亚马逊云科技

User Guide

Amazon Systems Manager Automation runbook reference

Amazon Systems Manager Automation runbook reference: User Guide

Table of Contents

Systems Manager Automation runbook reference

To help you get started quickly, Amazon Systems Manager provides predefined runbooks. These runbooks are maintained by Amazon Web Services, Amazon Web Services Support, and Amazon Config. The runbook reference describes each of the predefined runbooks provided by Systems Manager, Amazon Web Services Support, and Amazon Config.

Important

If you run an automation workflow that invokes other services by using an Amazon Identity and Access Management (IAM) service role, be aware that the service role must be configured with permission to invoke those services. This requirement applies to all Amazon Automation runbooks (`AWS-*` runbooks) such as the `AWS-ConfigureS3BucketLogging`, `AWS-CreateDynamoDBBackup`, and `AWS-RestartEC2Instance` runbooks, to name a few. This requirement also applies to any custom Automation runbooks you create that invoke other Amazon services by using actions that call other services. For example, if you use the `aws:executeAwsApi`, `aws:createStack`, or `aws:copyImage` actions, then you must configure the service role with permission to invoke those services. You can enable permissions to other Amazon services by adding an IAM inline policy to the role. For more information, see Add an Automation inline policy to invoke other Amazon services.

This reference includes topics that describe each of the Systems Manager runbooks that are owned by Amazon, Amazon Web Services Support, and Amazon Config. Runbooks are organized by the relevant Amazon Web Services service. Each page provides an explanation of the required and optional parameters that you can specify when using the runbook. Each page also lists the steps in the runbook and the output of the automation, if any.

This reference does *not* include a separate page for runbooks that require approval such as the `AWS-CreateManagedLinuxInstanceWithApproval` or `AWS-StopEC2InstanceWithApproval` runbook. Any runbook name that includes `WithApproval`, means the runbook includes the `aws:approve` action. This action temporarily pauses an automation until designated principals either approve or reject the action. After the required number of approvals is reached, the automation resumes.

For information about running automations, see Running a simple automation. For information about running automations on multiple targets, see Running automations that use targets and rate controls.

Topics

- Amazon Glue
- Amazon FSx
- GuardDuty
- IAM
- Amazon Kinesis Data Streams
- Amazon KMS
- Lambda
- Amazon Managed Workflows for Apache Airflow
- Neptune
- Amazon RDS
- Amazon Redshift
- Amazon S3
- Amazon SES
- SageMaker AI
- Secrets Manager
- Security Hub
- Amazon Shield
- Amazon SNS
- Amazon SQS
- Step Functions
- Systems Manager
- Third-party
- Amazon VPC
- Amazon WAF
- Amazon WorkSpaces
- X-Ray

View runbook content

You can view the content for runbooks in the Systems Manager console.

To view runbook content

1. Open the Amazon Systems Manager console at https://console.amazonaws.cn/systems-manager/.
2. In the navigation pane, choose **Documents**.

   -or-

   If the Amazon Systems Manager home page opens first, choose the menu icon (☰) to open the navigation pane, and then choose **Documents** in the navigation pane.
3. In the **Categories** section, choose **Automation documents**.
4. Choose a runbook, and then choose **View details**.
5. Choose the **Content** tab.

API Gateway

Amazon Systems Manager Automation provides predefined runbooks for Amazon API Gateway. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-DeleteAPIGatewayStage
- AWSConfigRemediation-EnableAPIGatewayTracing
- AWSConfigRemediation-UpdateAPIGatewayMethodCaching

AWSConfigRemediation-DeleteAPIGatewayStage

Description

The `AWSConfigRemediation-DeleteAPIGatewayStage` runbook deletes an Amazon API Gateway (API Gateway) stage. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- StageArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the API Gateway stage you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `apigateway:GET`
- `apigateway:DELETE`

Document Steps

- `aws:executeScript` - Deletes the API Gateway stage specified in the `StageArn` parameter.

AWSConfigRemediation-EnableAPIGatewayTracing

Description

The AWSConfigRemediation-EnableAPIGatewayTracing runbook enables tracing on an Amazon API Gateway (API Gateway) stage. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- StageArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the API Gateway stage you want to enable tracing on.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:GetAutomationExecution
- ssm:StartAutomationExecution
- config:GetResourceConfigHistory
- apigateway:GET
- apigateway:PATCH

Document Steps

- aws:executeScript - Enables tracing on the API Gateway stage specified in the StageArn parameter.

AWSConfigRemediation-UpdateAPIGatewayMethodCaching

Description

The AWSConfigRemediation-UpdateAPIGatewayMethodCaching runbook updates the cache method setting for an Amazon API Gateway stage resource.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- CachingAuthorizedMethods

  Type: StringList

  Description: (Required) The methods authorized to have caching enabled. The list must be some combination of `DELETE` , `GET` , `HEAD` , `OPTIONS` , `PATCH` , `POST` , and `PUT` . Caching is enabled for selected methods and disabled for non-selected methods. Caching is enabled for all methods if `ANY` is selected and is disabled for all methods if `NONE` is selected.
- StageArn

  Type: String

  Description: (Required) The API Gateway stage ARN for the `REST` API.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `apigateway:PATCH`
- `apigateway:GET`

Document Steps

- `aws:executeScript` - Accepts the stage resource ID as input, updates the cache method setting for an API Gateway stage using the `UpdateStage` API action, and verifies the update.

Amazon Batch

Amazon Systems Manager Automation provides predefined runbooks for Amazon Batch. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-TroubleshootAWSBatchJob

AWSSupport-TroubleshootAWSBatchJob

Description

The AWSSupport-TroubleshootAWSBatchJob runbook helps you to troubleshoot issues that prevent an Amazon Batch job from progressing from RUNNABLE to STARTING status.

How does it work?

This runbook performs the following checks:

- If the compute environment is in an INVALID or DISABLED state.
- If the compute environment's Max vCPU parameter is large enough to accommodate the job volume in the job queue.
- If the jobs require more vCPUs or memory resources than what the compute environment's instance types can provide.
- If the jobs should run on GPU-based instances but the compute environment is not configured to use GPU-based instances.
- If the Auto Scaling group for the compute environment failed to launch instances.
- If the launched instances can join the underlying Amazon Elastic Container Service (Amazon ECS) cluster; if not, it runs the AWSSupport-TroubleshootECSContainerInstance runbook.
- If any permissions issue is blocking specific actions that are required to run the job.

Important

- This runbook must be initiated in the same Amazon Region as your job that is stuck in RUNNABLE status.
- This runbook can be initiated for Amazon Batch jobs scheduled on Amazon ECS, Amazon Fargate or Amazon Elastic Compute Cloud (Amazon EC2) instances. If the automation is initiated for an Amazon Batch job on Amazon Elastic Kubernetes Service (Amazon EKS), the initiation stops.
- If instances are available to run the job but fail to register the Amazon ECS cluster, this runbook initiates the AWSSupport-TroubleshootECSContainerInstance automation runbook to try determine why. For more information, reference the AWSSupport-TroubleshootECSContainerInstance runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- JobId

  Type: String

  Description: (Required) The ID of the Amazon Batch Job that is stuck in RUNNABLE status.

  Allowed Pattern: `^[a-f0-9]{8}(-[a-f0-9]{4}){3}-[a-f0-9]{12}(:[0-9]+)?(#[0-9]+)?$`

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `autoscaling:DescribeAutoScalingGroups`
- `autoscaling:DescribeScalingActivities`
- `batch:DescribeComputeEnvironments`

- batch:DescribeJobs
- batch:DescribeJobQueues
- batch:ListJobs
- cloudtrail:LookupEvents
- ec2:DescribeIamInstanceProfileAssociations
- ec2:DescribeInstanceAttribute
- ec2:DescribeInstances
- ec2:DescribeInstanceTypeOfferings
- ec2:DescribeInstanceTypes
- ec2:DescribeNetworkAcls
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeSpotFleetInstances
- ec2:DescribeSpotFleetRequests
- ec2:DescribeSpotFleetRequestHistory
- ec2:DescribeSubnets
- ec2:DescribeVpcEndpoints
- ec2:DescribeVpcs
- ecs:DescribeClusters
- ecs:DescribeContainerInstances
- ecs:ListContainerInstances
- iam:GetInstanceProfile
- iam:GetRole
- iam:ListRoles
- iam:PassRole
- iam:SimulateCustomPolicy
- iam:SimulatePrincipalPolicy
- ssm:DescribeAutomationExecutions
- ssm:DescribeAutomationStepExecutions
- ssm:GetAutomationExecution

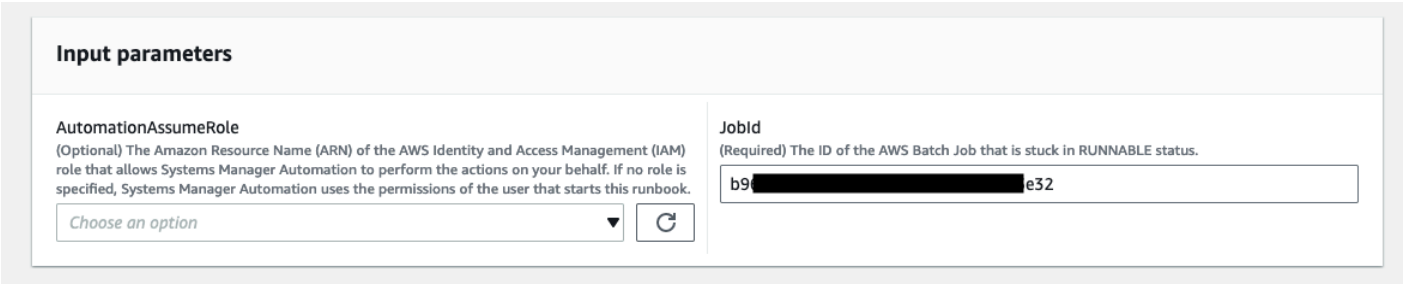
- `ssm:StartAutomationExecution`
- `sts:GetCallerIdentity`

Instructions

1. Navigate to the AWSSupport-TroubleshootAWSBatchJob in the Amazon Systems Manager Console.
2. Select **Execute Automation**
3. For input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **JobId (Required):**

     The ID of the Amazon Batch Job that is stuck in the RUNNABLE status.



4. Select **Execute**.
5. Notice that the automation initiates.
6. The document performs the following steps:
   - **PreflightPermissionChecks:**

     Performs preflight IAM permission checks against the initiating user/role. If there are any missing permissions, this step provides the API Actions missing in the global output section.
   - **ProceedOnlyIfUserHasPermission:**

     Branches based on if you have permissions to all required actions for the runbook.
   - **AWSBatchJobEvaluation:**

Performs checks against the Amazon Batch Job verifying it exists and is in the RUNNABLE status.

- **ProceedOnlyIfBatchJobExistsAndIsinRunnableState:**

  Branches based on if the jobs exists and is in the RUNNABLE status.

- **BatchComputeEnvironmentEvaluation:**

  Performs checks against the Amazon Batch Compute Environment.

- **ProceedOnlyIfComputeEnvironmentChecksAreOK:**

  Branches based on if compute environment checks succeeded.

- **UnderlyingInfraEvaluation:**

  Performs checks against the underlying Auto Scaling Group or Spot Fleet Request.

- **ProceedOnlyIfInstancesNotJoiningEcsCluster:**

  Branches based on if there are instances not joining the Amazon ECS cluster.

- **EcsAutomationRunner:**

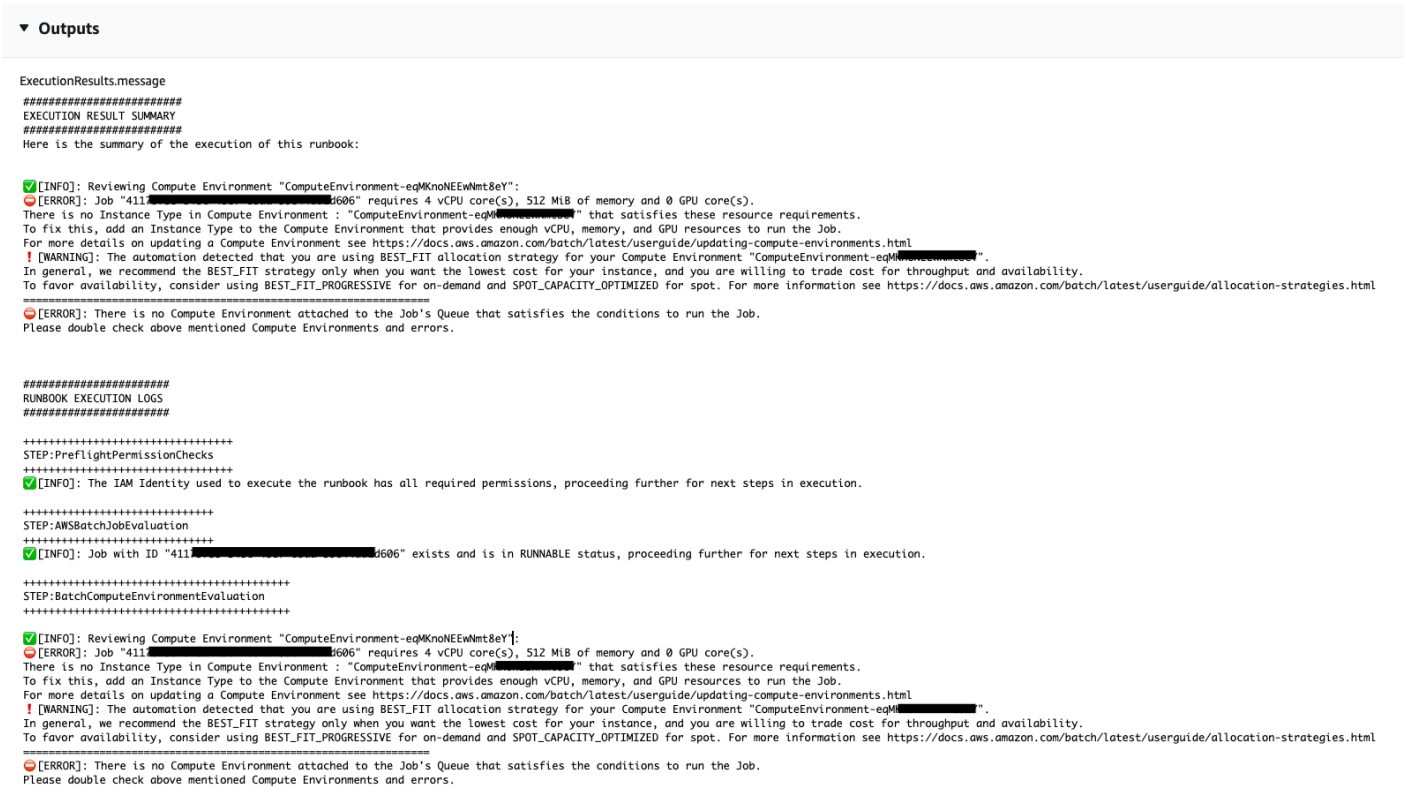
  Runs the Amazon ECS automation for the instances not joining the cluster.

- **ExecutionResults:**

  Generates output based on previous steps.

7. After completing, the URI for the assessment report HTML file is provided:

   S3 Console link and Amazon S3 URI for the Report on successful execution of the runbook



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon CloudFormation

Amazon Systems Manager Automation provides predefined runbooks for Amazon CloudFormation. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-DeleteCloudFormationStack
- AWS-EnableCloudFormationSNSNotification

- AWS-RunCfnLint
- AWSSupport-TroubleshootCFNCustomResource
- AWS-UpdateCloudFormationStack

AWS-DeleteCloudFormationStack

Description

Delete an Amazon CloudFormation stack.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- StackNameOrId

  Type: String

  Description: (Required) Name or Unique ID of the CloudFormation stack to be deleted

AWS-EnableCloudFormationSNSNotification

Description

The AWS-EnableCloudFormationSNSNotification runbook enables Amazon Simple Notification Service (Amazon SNS) notifications for the Amazon CloudFormation (Amazon CloudFormation) stack you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- StackArn

  Type: String

  Description: (Required) The ARN or name of the Amazon CloudFormation stack you want to enable Amazon SNS notifications for.

- NotificationArn

  Type: String

Description: (Required) The ARN of the Amazon SNS topic you want to associate with the Amazon CloudFormation stack.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- ssm:GetAutomationExecution
- ssm:StartAutomationExecution
- cloudformation:DescribeStacks
- cloudformation:UpdateStack
- kms:Decrypt
- kms:GenerateDataKey
- sns:Publish
- sqs:GetQueueAttributes

Document Steps

- CheckCfnSnsLimits (`aws:executeScript`) - Verifies the maximum number of Amazon SNS topics haven't already been associated with the Amazon CloudFormation stack you specify.
- EnableCfnSnsNotification (`aws:executeAwsApi`) - Enables Amazon SNS notifications for the Amazon CloudFormation stack.
- VerificationCfnSnsNotification (`aws:executeScript`) - Verifies that Amazon SNS notifications have been enabled for the Amazon CloudFormation stack.

Outputs

CheckCfnSnsLimits.NotificationArnList - A list of ARNs that receive Amazon SNS notifications for the Amazon CloudFormation stack.

VerificationCfnSnsNotification.VerifySnsTopicsResponse - Response from the API operation confirming Amazon SNS notifications have been enabled for the Amazon CloudFormation stack.

AWS-RunCfnLint

Description

This runbook uses an Amazon CloudFormation Linter (`cfn-python-lint`) to validate YAML and JSON templates against the Amazon CloudFormation resource specification. The `AWS-RunCfnLint` runbook performs additional checks, such as ensuring that valid values have been entered for resource properties. If validation is not successful, the `RunCfnLintAgainstTemplate` step fails and the linter tool's output is provided in an error message. This runbook is using cfn-lint v0.24.4.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ConfigureRuleFlag

  Type: String

  Description: (Optional) Configuration options for a rule to pass to the `--configure-rule` parameter.

Example: E2001:strict=false,E3012:strict=false.

- FormatFlag

  Type: String

  Description: (Optional) Value to pass to the `--format` parameter to specify the output format.

  Valid values: Default | quiet | parseable | json

  Default: Default
- IgnoreChecksFlag

  Type: String

  Description: (Optional) IDs of rules to pass to the --ignore-checks parameter. These rules are not checked.

  Example: E1001,E1003,W7001
- IncludeChecksFlag

  Type: String

  Description: (Optional) IDs of rules to pass to the `--include-checks` parameter. These rules are checked.

  Example: E1001,E1003,W7001
- InfoFlag

  Type: String

  Description: (Optional) Option for the `--info` parameter. Include the option to enable additional logging information about the template processing.

  Default: false
- TemplateFileName

  Type: String

  Description: The name, or key, of the template file in the S3 bucket.
- TemplateS3BucketName

Type: String

Description: The name of the S3 bucket containing the packer template.

- RegionsFlag

  Type: String

  Description: (Optional) Values to pass to the for `--regions` parameter to test the template against specified Amazon Web Services Regions.

  Example: us-east-1,us-west-1

Document Steps

RunCfnLintAgainstTemplate – Runs the `cfn-python-lint` tool against the specified Amazon CloudFormation template.

Outputs

RunCfnLintAgainstTemplate.output – The stdout from the `cfn-python-lint` tool.

AWSSupport-TroubleshootCFNCustomResource

Description

The `AWSSupport-TroubleshootCFNCustomResource` runbook helps diagnose why an Amazon CloudFormation stack failed in creating, updating, or deleting a custom resource. The runbook checks the service token used for the custom resource and the error message that was returned. After reviewing the details for the custom resource, the runbook output provides an explanation of the stack behavior and troubleshooting steps for the custom resource.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- StackName

  Type: String

  Description: (Required) The name of the Amazon CloudFormation stack where the custom resource failed.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudformation:DescribeStacks`
- `cloudformation:DescribeStackEvents`
- `cloudformation:ListStackResources`
- `ec2:DescribeRouteTables`
- `ec2:DescribeNatGateways`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeVpcs`
- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeSubnets`
- `logs:FilterLogEvents`

Document Steps

- `validateCloudFormationStack` - Verifies that the Amazon CloudFormation stack exists in the same Amazon Web Services account and Amazon Web Services Region.
- `checkCustomResource` - Analyzes the Amazon CloudFormation stack, checks the failed custom resource, and outputs information about how to troubleshoot the failed custom resource.

AWS-UpdateCloudFormationStack

Description

Update an Amazon CloudFormation stack by using an Amazon CloudFormation template stored in an Amazon S3 bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- LambdaAssumeRole

Type: String

Description: (Required) The ARN of the role assumed by Lambda

- StackNameOrId

  Type: String

  Description: (Required) Name or Unique ID of the Amazon CloudFormation stack to be updated

- TemplateUrl

  Type: String

  Description: (Required) S3 bucket location that contains the updated CloudFormation template (e.g. `https://s3.amazonaws.com.cn/amzn-s3-demo-bucket2/updated.template`)

CloudFront

Amazon Systems Manager Automation provides predefined runbooks for Amazon CloudFront. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-EnableCloudFrontDefaultRootObject
- AWSConfigRemediation-EnableCloudFrontAccessLogs
- AWSConfigRemediation-EnableCloudFrontOriginAccessIdentity
- AWSConfigRemediation-EnableCloudFrontOriginFailover
- AWSConfigRemediation-EnableCloudFrontViewerPolicyHTTPS

`AWSConfigRemediation-EnableCloudFrontDefaultRootObject`

Description

The `AWSConfigRemediation-EnableCloudFrontDefaultRootObject` runbook configures the default root object for the Amazon CloudFront (CloudFront) distribution that you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- CloudFrontDistributionId

  Type: String

  Description: (Required) The ID of the CloudFront distribution that you want to configure the default root object for.
- DefaultRootObject

  Type: String

  Description: (Required) The object that you want CloudFront to return when a viewer request points to your root URL.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`

- `cloudfront:GetDistributionConfig`
- `cloudfront:UpdateDistribution`

Document Steps

- `aws:executeScript` - Configures the default root object for the CloudFront distribution that you specify in the `CloudFrontDistributionId` parameter.

AWSConfigRemediation-EnableCloudFrontAccessLogs

Description

The `AWSConfigRemediation-EnableCloudFrontAccessLogs` runbook enables access logging for the Amazon CloudFront (CloudFront) distribution you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- BucketName

Type: String

Description: (Required) The name of the Amazon Simple Storage Service (Amazon S3) bucket you want to store access logs in. Buckets in the af-south-1, ap-east-1, eu-south-1, and me-south-1 Amazon Web Services Region are not supported.

- CloudFrontId

  Type: String

  Description: (Required) The ID of the CloudFront distribution you want to enable access logging on.

- IncludeCookies

  Type: Boolean

  Valid values: true | false

  Description: (Required) Set this parameter to `true` , if you want cookies to be included in the access logs.

- Prefix

  Type: String

  Description: (Optional) An optional string that you want CloudFront to prefix to the access log `filenames` for your distribution, for example, `myprefix/`.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `cloudfront:GetDistribution`
- `cloudfront:GetDistributionConfig`
- `cloudfront:UpdateDistribution`
- `s3:GetBucketLocation`

- s3:GetBucketAcl
- s3:PutBucketAcl

> **Note**
>
> The s3:GetBucketLocation API can only be used for S3 buckets in same account. You cannot use it for cross-account S3 buckets.

Document Steps

- aws:executeScript - Enables access logging for the CloudFront distribution you specify in the CloudFrontDistributionId parameter.

AWSConfigRemediation-EnableCloudFrontOriginAccessIdentity

Description

The AWSConfigRemediation-EnableCloudFrontOriginAccessIdentity runbook enables origin access identity for the Amazon CloudFront (CloudFront) distribution you specify. This automation assigns the same CloudFront Origin Access Identity for all Origins of the Amazon Simple Storage Service (Amazon S3) Origin type without origin access identity for the CloudFront distribution you specify. This automation does not grant read permission to the origin access identity for CloudFront to access objects in your Amazon S3 bucket. You must update your Amazon S3 bucket permissions to allow access.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- CloudFrontDistributionId

  Type: String

  Description: (Required) The ID of the CloudFront distribution you want to enable origin failover on.
- OriginAccessIdentityId

  Type: String

  Description: (Required) The ID of the CloudFront origin access identity to associate with the origin.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `cloudfront:GetDistributionConfig`
- `cloudfront:UpdateDistribution`

Document Steps

- `aws:executeScript` - Enables origin access identity for the CloudFront distribution you specify in the `CloudFrontDistributionId` parameter, and verifies the origin access identity was assigned.

AWSConfigRemediation-EnableCloudFrontOriginFailover

Description

The AWSConfigRemediation-EnableCloudFrontOriginFailover runbook enables origin failover for the Amazon CloudFront (CloudFront) distribution you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- CloudFrontDistributionId

  Type: String

  Description: (Required) The ID of the CloudFront distribution you want to enable origin failover on.
- OriginGroupId

  Type: String

  Description: (Required) The ID of the origin group.

- PrimaryOriginId

  Type: String

  Description: (Required) The ID of the primary origin in the origin group.

- SecondaryOriginId

  Type: String

  Description: (Required) The ID of the secondary origin in the origin group.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `cloudfront:GetDistributionConfig`
- `cloudfront:UpdateDistribution`

Document Steps

- `aws:executeScript` - Enables origin failover for the CloudFront distribution you specify in the `CloudFrontDistributionId` parameter, and verifies that failover has been enabled.

AWSConfigRemediation-EnableCloudFrontViewerPolicyHTTPS

Description

The `AWSConfigRemediation-EnableCloudFrontViewerPolicyHTTPS` runbook enables the viewer protocol policy for the Amazon CloudFront (CloudFront) distribution you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- CloudFrontDistributionId

  Type: String

  Description: (Required) The ID of the CloudFront distribution you want to enable the viewer protocol policy on.
- ViewerProtocolPolicy

  Type: String

  Valid values: https-only, redirect-to-https

  Description: (Required) The protocol that viewers can use to access the files in the origin.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `cloudfront:GetDistributionConfig`
- `cloudfront:UpdateDistribution`

- `cloudfront:GetDistribution`

Document Steps

- `aws:executeScript` - Enables the viewer protocol policy for the CloudFront distribution you specify in the `CloudFrontDistributionId` parameter, and verifies the policy was assigned.

CloudTrail

Amazon Systems Manager Automation provides predefined runbooks for Amazon CloudTrail. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

`AWSConfigRemediation-CreateCloudTrailMultiRegionTrail`

Description

The `AWSConfigRemediation-CreateCloudTrailMultiRegionTrail` runbook creates an Amazon CloudTrail (CloudTrail) trail that delivers log files from multiple Amazon Web Services Regions to the Amazon Simple Storage Service (Amazon S3) bucket of your choice.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- BucketName

  Type: String

  Description: (Required) The name of the Amazon S3 bucket you want to upload logs to.
- KeyPrefix

  Type: String

  Description: (Optional) The Amazon S3 key prefix that comes after the name of the bucket you designated for log file delivery.
- TrailName

  Type: String

  Description: (Required) The name of the CloudTrail trail to be created.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`

- cloudtrail:CreateTrail
- cloudtrail:StartLogging
- cloudtrail:GetTrail
- s3:PutObject
- s3:GetBucketAcl
- s3:PutBucketLogging
- s3:ListBucket

Document Steps

- aws:executeAwsApi - Accepts the trail name and the Amazon S3 bucket name as input and creates a CloudTrail trail.
- aws:executeAwsApi - Enables logging on the created trail and starts log delivery to the Amazon S3 bucket you specified.
- aws:assertAwsResourceProperty - Verifies that the CloudTrail trail has been created.

AWS-EnableCloudTrail

Description

Create an Amazon CloudTrail trail and configure logging to an S3 bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- S3BucketName

  Type: String

  Description: (Required) Name of the S3 bucket designated for publishing log files.

  > **Note**
  >
  > The S3 bucket must exist and the bucket policy must grant CloudTrail permission to write to it. For information, see Amazon S3 Bucket Policy for CloudTrail .

- TrailName

  Type: String

  Description: (Required) The name of the new trail.

AWS-EnableCloudTrailCloudWatchLogs

Description

This runbook updates the configuration of one or more Amazon CloudTrail trails to send events to an Amazon CloudWatch Logs log group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- CloudWatchLogsLogGroupArn

  Type: String

  Description: (Required) The ARN of the CloudWatch Logs log group where the CloudTrail logs will be delivered.
- CloudWatchLogsRoleArn

  Type: String

  Description: (Required) The ARN of the IAM role CloudWatch Logs Logs assumes to write to the specified log group.
- TrailNames

  Type: StringList

  Description: (Required) A comma separated list of the names of the CloudTrail trails whose events you want to send to CloudWatch Logs.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudtrail:UpdateTrail`
- `iam:PassRole`

Document Steps

- `aws:executeScript` - Updates the specified CloudTrail trails to deliver events to the specified CloudWatch Logs log group.

AWSConfigRemediation-EnableCloudTrailEncryptionWithKMS

Description

The `AWSConfigRemediation-EnableCloudTrailEncryptionWithKMS` runbook encrypts an Amazon CloudTrail (CloudTrail) trail using the Amazon Key Management Service (Amazon KMS) customer managed key you specify. This runbook should only be used as a baseline to ensure that your CloudTrail trails are encrypted according to minimum recommended security best practices. We recommend encrypting multiple trails with different KMS keys. CloudTrail digest files are not encrypted. If you have previously set the `EnableLogFileValidation` parameter to `true` for the trail, see the "Use server-side encryption with Amazon KMS managed keys" section of the CloudTrail Preventative Security Best Practices topic in the *Amazon CloudTrail User Guide* for more information.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- KMSKeyId

  Type: String

  Description: (Required) The ARN, key ID, or the key alias of the of the customer managed key you want to use to encrypt the trail you specify in the `TrailName` parameter.
- TrailName

  Type: String

  Description: (Required) The ARN or name of the trail you want to update to be encrypted.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `cloudtrail:GetTrail`
- `cloudtrail:UpdateTrail`

Document Steps

- `aws:executeAwsApi` - Enables encryption on the trail you specify in the `TrailName` parameter.
- `aws:executeAwsApi` - Gathers the ARN for the customer managed key you specify in the `KMSKeyId` parameter.
- `aws:assertAwsResourceProperty` - Verifies that encryption has been enabled on the CloudTrail trail.

AWS-EnableCloudTrailKmsEncryption

Description

This runbook updates the configuration of one or more Amazon CloudTrail trails to use Amazon Key Management Service (Amazon KMS) encryption.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- KMSKeyId

  Type: String

  Description: (Required) The key ID of the of the customer managed key you want to use to encrypt the trail you specify in the `TrailName` parameter. The value can be an alias name prefixed by "alias/", a fully specified ARN to an alias, or a fully specified ARN to a key.
- TrailNames

  Type: StringList

  Description: (Required) A comma separated list of the trails you want to update to be encrypted.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- cloudtrail:UpdateTrail
- kms:DescribeKey
- kms:ListKeys

Document Steps

- aws:executeScript - Enables Amazon KMS encryption on the trails you specify in the TrailName parameter.

AWSConfigRemediation-EnableCloudTrailLogFileValidation

Description

The AWSConfigRemediation-EnableCloudTrailLogFileValidation runbook enables log file validation for your Amazon CloudTrail trail.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- TrailName

  Type: String

  Description: (Required) The name or Amazon Resource Name (ARN) of the trail you want to enable log validation for.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `cloudtrail:GetTrail`
- `cloudtrail:UpdateTrail`

Document Steps

- `aws:executeAwsApi` - Enables log validation for the Amazon CloudTrail trail you specify in the `TrailName` parameter.
- `aws:assertAwsResourceProperty` - Verifies log validation is enabled for your trail.

AWS-EnableCloudTrailLogFileValidation

Description

The `AWS-EnableCloudTrailLogFileValidation` runbook enables log file validation for the Amazon CloudTrail trails you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- TrailNames

  Type: StringList

  Description: (Required) A comma separated list of the names of the CloudTrail trails you want to enable log validation for.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudtrail:GetTrail`
- `cloudtrail:UpdateTrail`

Document Steps

- `aws:executeScript` - Enables log validation for the Amazon CloudTrail trails you specify in the `TrailNames` parameter.

AWS-QueryCloudTrailLogs

Description

The `AWS-QueryCloudTrailLogs` runbook creates an Amazon Athena table from the Amazon Simple Storage Service (Amazon S3) bucket of your choice containing Amazon CloudTrail (CloudTrail) logs. After creating the table, the automation runs SQL queries you specify and then deletes the table.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Query

  Type: String

  Description: (Required) The SQL query you want to run.
- SourceBucketPath

  Type: String

  Description: (Required) The name of the Amazon S3 bucket containing the CloudTrail log files you want to query.
- TableName

Type: String

Description: (Optional) The name of the Athena table created by the automation.

Default: cloudtrail_logs

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `athena:GetQueryResults`
- `athena:GetQueryExecution`
- `athena:StartQueryExecution`
- `glue:CreateTable`
- `glue:DeleteTable`
- `glue:GetDatabase`
- `glue:GetPartitions`
- `glue:GetTable`
- `s3:AbortMultipartUpload`
- `s3:CreateBucket`
- `s3:GetBucketLocation`
- `s3:GetObject`
- `s3:ListBucket`
- `s3:ListBucketMultipartUploads`
- `s3:ListMultipartUploadParts`
- `s3:PutObject`

Document Steps

- `aws:executeAwsApi` - Creates an Athena table.
- `aws:executeAwsApi` - Runs the query string you specify in the `Query` parameter.
- `aws:executeScript` - Polls and waits for the query to complete.

- `aws:executeAwsApi` - Gets the results of the query.
- `aws:executeAwsApi` - Deletes the table created by the automation.

CloudWatch

Amazon Systems Manager Automation provides predefined runbooks for Amazon CloudWatch. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-ConfigureCloudWatchOnEC2Instance
- AWS-EnableCWAlarm

`AWS-ConfigureCloudWatchOnEC2Instance`

Description

Enable or disable Amazon CloudWatch detailed monitoring on managed instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance on which you want to enable CloudWatch monitoring.

- properties

  Type: String

  Description: (Optional) This parameter is not supported. It is listed here for backwards compatibility.

- status

  Valid values: Enabled | Disabled

  Description: (Optional) Specifies whether to enable or disable CloudWatch.

  Default: Enabled

Document Steps

configureCloudWatch - Configures CloudWatch on the Amazon EC2 instance with the given status.

Outputs

This automation has no output.

AWS-EnableCWAlarm

Description

The `AWS-EnableCWAlarm` runbook creates Amazon CloudWatch (CloudWatch) alarms for Amazon resources in your Amazon Web Services account that do not already have one. CloudWatch alarms are created for the following Amazon resources:

- Amazon Elastic Compute Cloud (Amazon EC2) instances
- Amazon Elastic Block Store (Amazon EBS) volumes

- Amazon Simple Storage Service (Amazon S3) buckets
- Amazon Relational Database Service (Amazon RDS) clusters

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ComparisonOperator

  Type: String

  Valid values: GreaterThanOrEqualToThreshold | GreaterThanThreshold | GreaterThanUpperThreshold | LessThanLowerOrGreaterThanUpperThreshol | LessThanLowerThreshold | LessThanOrEqualToThreshold | LessThanThreshold

  Description: (Required) The arithmetic operation to use when comparing the specified statistic and threshold.

- MetricName

  Type: String

Description: (Required) The name for the metric associated with the alarm.

- Period

  Type: Integer

  Valid values: 10 | 30 | 60 | A multiple of 60

  Description: (Required) The period, in seconds, over which the statistic is applied.

- ResourceARNs

  Type: StringList

  Description: (Required) A comma separated list of ARNs of the resources to create a CloudWatch alarm for

- Statistic

  Type: String

  Valid values: Average | Maximum | Minimum | SampleCount | Sum

  Description: (Required) The statistic for the metric associated with the alarm.

- Threshold

  Type: Integer

  Description: (Required) The value to compare with the specified statistic.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudwatch:PutMetricAlarm`

Document Steps

- `aws:executeScript` - Creates a CloudWatch alarm according to the values specified in the runbook parameters for the resources you specify in the `ResourceARNs` parameter .

Outputs

EnableCWAlarm.FailedResources: A maplist of resource ARNs for which a CloudWatch alarm was not created and the reason for the failure.

EnableCWAlarm.SuccessfulResources: A list of resource ARNs for which a CloudWatch alarm was successfully created.

Amazon DocumentDB

Amazon Systems Manager Automation provides predefined runbooks for Amazon DocumentDB (with MongoDB compatibility). For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-EnableDocDbClusterBackupRetentionPeriod

AWS-EnableDocDbClusterBackupRetentionPeriod

Description

The `AWS-EnableDocDbClusterBackupRetentionPeriod` runbook enables a backup retention period for the Amazon DocumentDB cluster you specify. This feature sets the total number of days for which an automated backup is retained. To modify a cluster, the cluster must be in the available state with an engine type of `docdb`.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- DBClusterResourceId

  Type: String

  Description: (Required) The resource ID for the Amazon DocumentDB cluster you want to enable the backup retention period for.

- BackupRetentionPeriod

  Type: Integer

  Description: (Required) The number of days for which automated backups are retained. Must be a value from 7-35 days.

- PreferredBackupWindow

  Type: String

  Description: (Optional) A daily time range in Universal Time Coordinated (UTC) in the format hh24:mm-hh24:mm, for example 07:14-07:44. The value must be at least 30 minutes and can't conflict with the preferred maintenance window.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `docdb:DescribeDBClusters`
- `docdb:ModifyDBCluster`
- `rds:DescribeDBClusters`
- `rds:ModifyDBCluster`

Document Steps

- GetDocDbClusterIdentifier (`aws:executeAwsApi`) - Returns the Amazon DocumentDB cluster identifier using the provided resource ID.
- VerifyDocDbEngine (`aws:assertAwsResourceProperty`) - Verifies the Amazon DocumentDB engine type is `docdb` to prevent inadvertent changes to other Amazon RDS engine types.
- VerifyDocDbStatus (`aws:waitAwsResourceProperty`) - Verifies the Amazon DocumentDB cluster status is `available`.
- ModifyDocDbRetentionPeriod (`aws:executeAwsApi`) - Sets the retention period using the provided values for the specified Amazon DocumentDB cluster.
- VerifyDocDbBackupsEnabled (`aws:executeScript`) - Verifies the retention period for the Amazon DocumentDB cluster and the preferred back up window, if specified, were successfully set.

Outputs

ModifyDocDbRetentionPeriod.ModifyDbClusterResponse - Response from the `ModifyDBCluster` API operation.

VerifyDocDbBackupsEnabled.VerifyDbClusterBackupsEnabledResponse - Output from the `VerifyDocDbBackupsEnabled` step confirming successful modification of the Amazon DocumentDB cluster.

CodeBuild

Amazon Systems Manager Automation provides predefined runbooks for Amazon CodeBuild. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-ConfigureCodeBuildProjectWithKMSCMK
- AWSConfigRemediation-DeleteAccessKeysFromCodeBuildProject

`AWSConfigRemediation-ConfigureCodeBuildProjectWithKMSCMK`

Description

The AWSConfigRemediation-ConfigureCodeBuildProjectWithKMSCMK runbook encrypts an Amazon CodeBuild (CodeBuild) project's build artifacts using the Amazon Key Management Service (Amazon KMS) customer managed key you specify. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- KMSKeyId

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon KMS customer managed key you want to use to encrypt the CodeBuild project you specify in the ProjectId parameter.

- ProjectId

  Type: String

  Description: (Required) The ID of the CodeBuild project whose build artifacts you want to encrypt.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- codebuild:BatchGetProjects
- codebuild:UpdateProject
- config:GetResourceConfigHistory

Document Steps

- aws:executeAwsApi - Gathers the CodeBuild project name from the project ID.
- aws:executeAwsApi - Enables encryption on the CodeBuild project you specify in the ProjectId parameter.
- aws:assertAwsResourceProperty - Verifies that encryption has been enabled on the CodeBuild project.

Outputs

UpdateLambdaConfig.UpdateFunctionConfigurationResponse - Response from the UpdateFunctionConfiguration API call.

AWSConfigRemediation-DeleteAccessKeysFromCodeBuildProject

Description

The AWSConfigRemediation-DeleteAccessKeysFromCodeBuildProject runbook deletes the AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY environment variables from the Amazon CodeBuild (CodeBuild) project you specify. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ResourceId

  Type: String

  Description: (Required) The ID of the CodeBuild project whose access key environment variables you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `codebuild:BatchGetProjects`
- `codebuild:UpdateProject`

Document Steps

- `aws:executeScript` - Deletes the access key environment variables for the CodeBuild project specified in the `ResourceId` parameter.

Amazon CodeDeploy

Amazon Systems Manager Automation provides predefined runbooks for Amazon CodeDeploy. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-TroubleshootCodeDeploy

`AWSSupport-TroubleshootCodeDeploy`

Description

The `AWSSupport-TroubleshootCodeDeploy` runbook helps diagnose why an Amazon CodeDeploy deployment failed on an Amazon Elastic Compute Cloud (Amazon EC2) instance. The runbook outputs steps to help you resolve the issue or troubleshoot further. Best practices for CodeDeploy are also provided to help you avoid similar issues in the future.

This runbook can help you to resolve the following issues:

- The CodeDeploy agent is not installed or not running on the Amazon EC2 instance
- The Amazon EC2 instance does not have an Amazon Identity and Access Management (IAM) instance profile attached
- The IAM instance profile attached to the Amazon EC2 instance does not have the required Amazon Simple Storage Service (Amazon S3) permissions
- A revision stored in Amazon S3 is missing, or the Amazon S3 bucket used is in an Amazon Web Services Region that is different than the Amazon EC2 instance
- Application specification (AppSpec) file issues
- "File already exists at location" errors
- Failed CodeDeploy managed lifecycle event hooks
- Failed customer managed lifecycle event hooks
- Scale-in events during the deployment

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DeploymentId

  Type: String

  Description: (Required) The ID of the deployment which failed.
- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance where the deployment failed.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `codedeploy:GetDeployment`
- `codedeploy:GetDeploymentTarget`

- `ec2:DescribeInstances`

Document Steps

- `aws:executeAwsApi` - Verifies the values provided for the `DeploymentId` and `InstanceId` parameters.
- `aws:executeScript` - Collects information from the Amazon EC2 instance such as the state of the instance and IAM instance profile details.
- `aws:executeScript` - Reviews the specified deployment, and returns an analysis regarding why the deployment failed.

Amazon Config

Amazon Systems Manager Automation provides predefined runbooks for Amazon Config. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-SetupConfig

`AWSSupport-SetupConfig`

Description

The `AWSSupport-SetupConfig` runbook creates an Amazon Identity and Access Management (IAM) service-linked role, a configuration recorder powered by Amazon Config, and a delivery channel with an Amazon Simple Storage Service (Amazon S3) bucket where Amazon Config sends configuration snapshots and configuration history files. If you specify values for the `AggregatorAccountId` and `AggregatorAccountRegion` parameters, the runbook also creates authorizations for data aggregation to collect Amazon Config configuration and compliance data from multiple Amazon Web Services accounts and multiple Amazon Web Services Regions. To learn more about aggregating data from multiple accounts and Regions, see Multi-Account Multi-Region Data Aggregation in the *Amazon Config Developer Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- AggregatorAccountId

  Type: String

  Description: (Optional) The ID of the Amazon Web Services account where an aggregator will be added to aggregate Amazon Config configuration and compliance data from multiple accounts and Amazon Web Services Regions. This account is also used by the aggregator to authorize the source accounts.
- AggregatorAccountRegion

  Type: String

  Description: (Optional) The Region where an aggregator will be added to aggregate Amazon Config configuration and compliance data from multiple accounts and Regions.
- IncludeGlobalResourcesRegion

  Type: String

  Default: us-east-1

  Description: (Required) To avoid recording global resource data in each Region, specify one Region to record global resource data from.

- Partition

  Type: String

  Default: `aws`

  Description: (Required) The partition you want to collect Amazon Config configuration and compliance data from.

- S3BucketName

  Type: String

  Default: `aws-config-delivery-channel`

  Description: (Optional) The name you want to apply to the Amazon S3 bucket created for the delivery channel. The account ID is appended to the end of the name.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:DescribeConfigurationRecorders`
- `config:DescribeDeliveryChannels`
- `config:PutAggregationAuthorization`
- `config:PutConfigurationRecorder`
- `config:PutDeliveryChannel`
- `config:StartConfigurationRecorder`
- `iam:CreateServiceLinkedRole`
- `iam:PassRole`
- `s3:CreateBucket`
- `s3:ListAllMyBuckets`
- `s3:PutBucketPolicy`

Document Steps

- `aws:executeScript` - Creates a service-linked IAM role for Amazon Config if one does not already exist.
- `aws:executeScript` - Creates a configuration recorder if one does not already exist.
- `aws:executeScript` - Creates an Amazon S3 bucket to be used by the delivery channel if one does not already exist.
- `aws:executeScript` - Creates a delivery channel using the resources created by the runbook.
- `aws:executeAwsApi` - Starts the configuration recorder.
- `aws:executeScript` - If you specified values for the `AggregatorAccountId` and `AggregatorAccountRegion` parameters, authorizations for multi-account and multi-Region data aggregation are configured.

Amazon Connect

Amazon Systems Manager Automation provides predefined runbooks for Amazon Connect. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

`AWSSupport-AssociatePhoneNumbersToConnectContactFlows`

Description

The `AWSSupport-AssociatePhoneNumbersToConnectContactFlows` helps you associate phone numbers to contact flows in your Amazon Connect instance. By providing the mappings of phone numbers and contact flows in an input comma-separated values (CSV) file, the runbook associates as many phone numbers to contact flows as possible within 14.5 minutes. The runbook produces a CSV file of all phone number and contact flow pairs that it couldn't associate within the time limit so that you can input them in the next run.

How does it work?

The runbook AWSSupport-AssociatePhoneNumbersToConnectContactFlows helps you associate phone numbers to contact flows in your Amazon Connect instance using a CSV file of mapping data that is stored in an Amazon Simple Storage Service (Amazon S3) bucket. The input CSV file should align to the following format, with PhoneNumber values in E.164 format.

Example of the input CSV file

```
PhoneNumber,ContactFlowName
+1800555xxxx,ContactFlowA
+1800555yyyy,ContactFlowB
+1800555zzzz,ContactFlowC
```

The automation runbook also creates the following files in the destination location specified in the DestinationFileBucket and DestinationFilePath.

- **automation:EXECUTION_ID/ResourceIdList.csv**: A temporary file that contains the PhoneNumberId and ContactFlowId pairs that are required for the AssociatePhoneNumberContactFlow API.
- **automation:EXECUTION_ID/ErrorResourceList.csv**: A file that contains the phone number and contact flow pairs that could not be processed due to an error, such as ResourceNotFoundException in the format of PhoneNumber,ContactFlowName,ErrorMessage.
- **automation:EXECUTION_ID/NonProcessedResourceList.csv**: A file that contains the phone number and contact flow pairs that weren't processed. The runbook tries to process as many phone numbers and contact flows as possible within 14.5 min (15 min of Amazon Lambda function timeout - 30 seconds of buffer). If there are some phone numbers / contact flows that could not be processed due to the time constraint, the runbook includes them in a CSV file to use as an input for the next runbook execution.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketAcl",
                "s3:GetObject",
                "s3:GetObjectAttributes",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
            "arn:aws:s3:::YOUR-BUCKET/*",
            "arn:aws:s3:::YOUR-BUCKET"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DescribeStacks",
                "cloudformation:DeleteStack",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:GetRole",
                "iam:PutRolePolicy",
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:TagResource",
                "connect:AssociatePhoneNumberContactFlow",
                "logs:CreateLogGroup",
                "logs:TagResource",
```

```
                    "logs:PutRetentionPolicy",
                    "logs:DeleteLogGroup",
                    "s3:GetAccountPublicAccessBlock"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Action": [
                    "connect:DescribeInstance",
                    "connect:ListPhoneNumbers",
                    "connect:ListContactFlows",
                    "ds:DescribeDirectories"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Condition": {
                    "StringLikeIfExists": {
                        "iam:PassedToService": [
                            "ssm.amazonaws.com",
                            "lambda.amazonaws.com"
                        ]
                    }
                },
                "Action": [
                    "iam:PassRole"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-AssociatePhoneNumbersToConnectContactFlows in Systems Manager under Documents.
2. Select Execute automation.

3. For the input parameters, enter the following:

- **AutomationAssumeRole (Optional)**

  The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

- **ConnectInstanceId (Required)**

  The ID of your Amazon Connect instance.

- **SourceFileBucket (Required)**

  The Amazon S3 bucket that stores the CSV file that contains the phone number and contact flow pairs.

- **SourceFilePath (Required)**

  The Amazon S3 object key of the CSV file that contains the phone number and contact flow pairs. For example, `path/to/input.csv`.

- **DestinationFileBucket (Required)**

  The Amazon S3 bucket into which the automation will place an intermediate file and result report.

- **DestinationFilePath (Optional)**

  The Amazon S3 object path in `DestinationFileBucket` under which an intermediate file and result report should be stored. For example, if you specify `path/to/files/`, files are stored under `s3://[DestinationFileBucket]/path/to/files/[automation:EXECUTION_ID]/`.

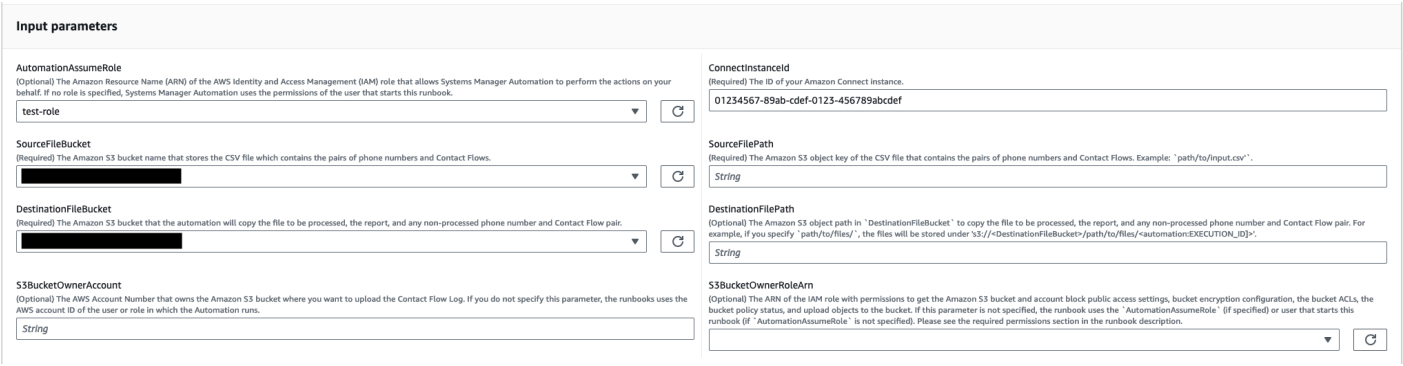
- **S3BucketOwnerAccount (Optional)**

  The Amazon Account Number that owns the Amazon S3 bucket where you want to upload the Contact Flow Log. If you do not specify this parameter, the runbooks uses the Amazon account ID of the user or role in which the Automation runs.

- **S3BucketOwnerRoleArn (Optional)**

  The ARN of the IAM role with permissions to get the Amazon S3 bucket and account block public access settings, bucket encryption configuration, the bucket ACLs, the bucket

policy status, and upload objects to the bucket. If this parameter is not specified, the runbook uses the `AutomationAssumeRole` (if specified) or user that starts this runbook (if `AutomationAssumeRole` is not specified). Please see the required permissions section in the runbook description.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

- **CheckConnectInstanceExistance**

  Checks if the Amazon Connect instance provided in `ConnectInstanceId` exists.

- **CheckS3BucketPublicStatus**

  Checks if the Amazon S3 buckets specified in the `SourceFileBucket` and `DestinationFileBucket` allow anonymous or public read or write access permissions.

- **CheckSourceFileExistenceAndSize**

  Checks if the source CSV file specified in the `SourceFilePath` exists and if the file size exceeds the of 25 MiB limit.

- **GenerateResourceIdMap**

  Downloads the source CSV file specified in the `SourceFilePath` and identify `PhoneNumberId` and `ContactFlowId` for each resource. After it's done, it uploads a CSV file that contains `PhoneNumber`, `PhoneNumberId`, `ContactFlowName`, and `ContactFlowId` to the destination Amazon S3 bucket specified in `DestinationFileBucket`. If `PhoneNumberId` cannot be identified for a certain number, the filed will be empty in the CSV file.

- **AssociatePhoneNumbersToContactFlows**

Creates an Amazon Lambda function in your account using an Amazon CloudFormation stack. The Amazon Lambda function associates each number to a contact flow listed in the source CSV file specified in `SourceFileBucket` and `SourceFilePath` and the Amazon CloudFormation stack invokes the function. The Amazon Lambda function map as many phone numbers to contact flows as possible before it times out (15 minutes). The list of phone numbers and contact flows that could not be processed due to error is uploaded in `[automation:EXECUTION_ID]/ErrorResourceList.csv`. The ones that could not be processed due to an excess of the maximum number of phone numbers that can be processed in a single execution are uploaded in `[automation:EXECUTION_ID]/NonProcessedResourceList.csv`. If this step fails, it goes to the `DescribeCloudFormationErrorFromStackEvents` step to show why it failed from Amazon CloudFormation stack events.

- **WaitForPhoneNumberContactFlowAssociationCompletion**

  Waits until the Amazon Lambda function that maps phone numbers to contact flows is created and the Amazon CloudFormation stack completes its invocation.

- **GenerateReport**

  Generates the report that contains the number of phone numbers mapped to contact flows, the ones that could not be processed due to error, and the ones that could not be processed due to an excess of the maximum number of phone numbers that can be processed in single execution. The report also shows the location (Amazon S3 URI and Amazon S3 console URL) for `[automation:EXECUTION_ID]/ErrorResourceList.csv` or `[automation:EXECUTION_ID]/NonProcessedResourceList.csv`, if applicable.

- **`DeleteCloudFormationStack`**

  Deletes the Amazon CloudFormation stack, including the Lambda function for mapping.

- **`DescribeCloudFormationErrorFromStackEvent`**

  Describes errors from the Amazon CloudFormation stack of the `AssociatePhoneNumbersToContactFlows` step.

7. After completed, review the Outputs section for the detailed results of the execution:

- **GenerateReport.OutputPayload**

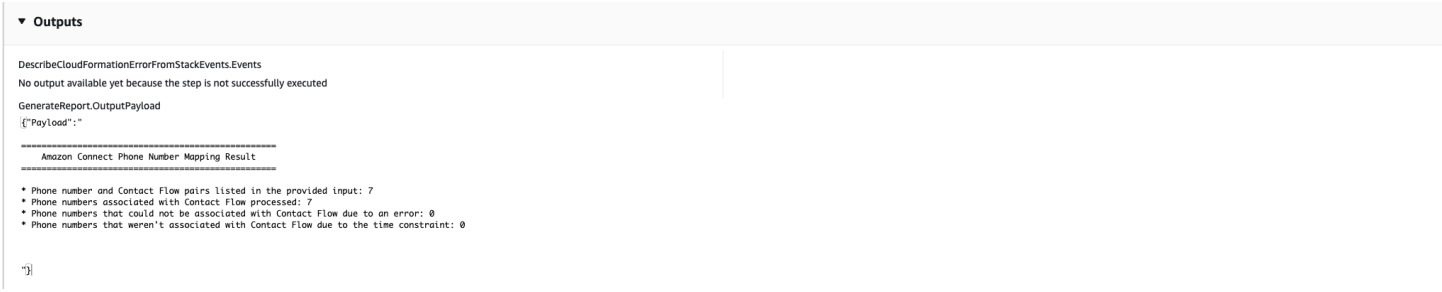
  Output of phone number and contact flow associations. This report contains following information:

- The number of phone number and contact flow pairs listed in the input CSV file
- The number of phone numbers associated with contact flows as specified in the input CSV file
- The number of phone numbers that could not be associated with contact flows due to error
- The number of phone numbers that weren't associated with contact flows due to time constraint
- The location (Amazon S3 URI and Amazon S3 console URL) of the CSV file that contains the phone number and contact flow pairs that could not be associated due to error
- The location (Amazon S3 URI and Amazon S3 Console URL) of the CSV file that contains the phone number and contact flow pairs that weren't associated due to time constraint

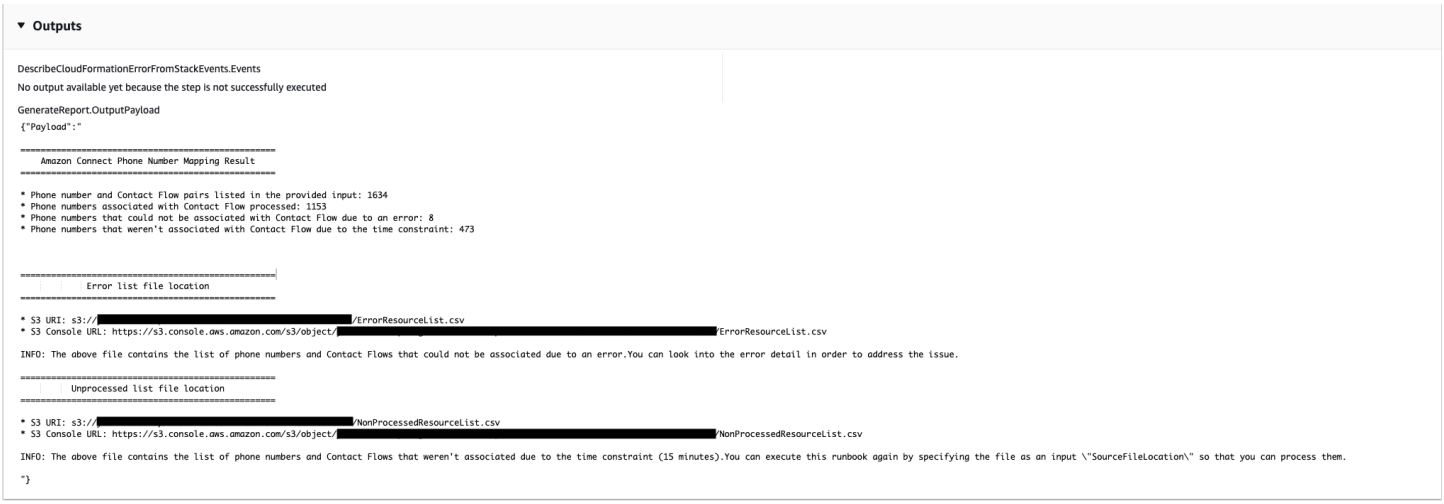
- **DescribeCloudFormationErrorFromStackEvents.Events**

  Output that shows Amazon CloudFormation stack events if the `AssociatePhoneNumbersToContactFlows` step fails.

Output of execution with a small number of phone numbers and contact flows



Output of execution with a large number of phone numbers and contact flows and phone numbers that weren't associated due to error or time constraint



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-CollectAmazonConnectContactFlowLog

Description

The AWSSupport-CollectAmazonConnectContactFlowLog automation runbook is used to collect the Amazon Connect contact flow logs for a specific contact ID. By providing your Amazon Connect instance ID and contact ID, the runbook searches contact flow logs for the contact from the Amazon CloudWatch log group and uploads them to the Amazon Simple Storage Service (Amazon S3) bucket that is specified in the request parameter. The runbook generates output that provides Amazon S3 console URL and Amazon CLI command for you to download the logs.

How does it work?

The AWSSupport-CollectAmazonConnectContactFlowLog automation runbook helps to collect the Amazon Connect contact flow logs for a specific contact ID stored in the configured CloudWatch log group and uploads them to a specified Amazon S3 bucket. To help with the security of the logs gathered from your Amazon Connect contact flow, the automation evaluates the Amazon S3 bucket configuration to determine if the bucket grants

public `read` or `write` access permissions and is owned by the Amazon account specified in the `S3BucketOwnerAccountId` parameter. If your Amazon S3 bucket uses server-side encryption with Amazon Key Management Service keys (SSE-KMS), make sure that the user or Amazon Identity and Access Management (IAM) role that is running this automation has the `kms:GenerateDataKey` permissions on the Amazon KMS key. For more information about the logs generated by your Amazon Connect instance, see Flow logs stored in an Amazon CloudWatch log group.

Important

The CloudWatch Logs Insights queries incur charges based on the amount of data that is queried. Free tier customers are charged only for usage that exceeds service quotas. For more information, see Amazon CloudWatch Pricing.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetBucketPublicAccessBlock",
```

```
                    "s3:GetBucketPolicyStatus",
                    "s3:GetBucketAcl",
                    "s3:GetObject",
                    "s3:GetObjectAttributes",
                    "s3:PutObject",
                    "s3:PutObjectAcl"
                ],
                "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-bucket"
                ],
                "Effect": "Allow"
            },
            {
                "Action": [
                    "connect:DescribeInstance",
                    "connect:DescribeContact",
                    "ds:DescribeDirectories"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Action": [
                    "logs:StartQuery",
                    "logs:GetQueryResults"

                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-CollectAmazonConnectContactFlowLog in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:

- **AutomationAssumeRole (Optional):**

  The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

- **ConnectInstanceId (Required):**

  The ID of your Amazon Connect instance.

- **ContactId (Required):**

  The ID of the contact that you want to collect contact flow log for.

- **S3BucketName (Required):**

  The Amazon S3 bucket name in your account where you want to upload the contact flow log. Make sure that bucket policy does not grant unnecessary read/write permissions to parties that do not need access to the collected logs.

- **S3ObjectPrefix (Optional):**

  The Amazon S3 object path in the Amazon S3 bucket for an uploaded contact flow log. For example, if you specify `CollectedLogs`, the log will be uploaded as `s3://your-s3-bucket/CollectedLogs/ContactFlowLog_[ContactId][AWSAccountId].gz`. If you do not specify this parameter, the Systems Manager Automation execution ID is used, for example: `s3://your-s3-bucket/[automation:EXECUTION_ID]/ContactFlowLog[ContactId]_[AWSAccountId].gz`. Note: if you specify a value for `S3ObjectPrefix` and run this automation using the same [ContactId], the contact flow log will be overwritten.

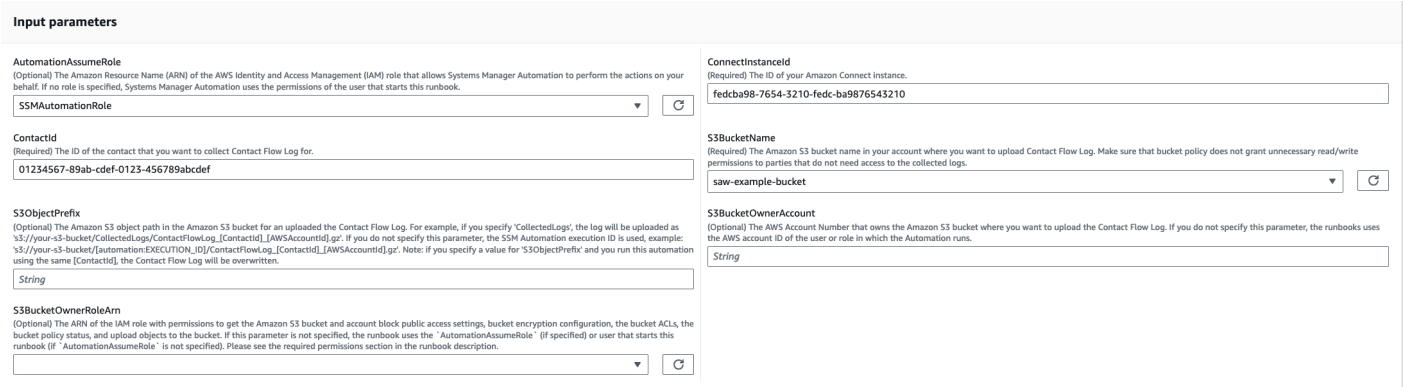
- **S3BucketOwnerAccount (Optional):**

  The Amazon account number that owns the Amazon S3 bucket where you want to upload the contact flow log. If you do not specify this parameter, the runbook uses the Amazon account ID of the user or role in which the automation runs.

- **S3BucketOwnerRoleArn (Optional):**

  The ARN of the IAM role with permissions to get the Amazon S3 bucket and account block public access settings, bucket encryption configuration, bucket ACLs, bucket policy status, and upload objects to the bucket. If this parameter is not specified, the runbook

uses the AutomationAssumeRole (if specified) or user that starts this runbook (if AutomationAssumeRole is not specified). See the required permissions section in the runbook description.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

- **CheckConnectInstanceExistance**

  Checks if the Amazon Connect instance provided in the ConnectInstanceId is ACTIVE.

- **CheckS3BucketPublicStatus**

  Checks if the Amazon S3 bucket specified in the S3BucketName allows anonymous or public read or write access permissions.

- **GenerateLogSearchTimeRange**

  Generates StartTime and EndTime for the StartQuery step based on the InitiationTimestamp and LastUpdateTimestamp returned by the DescribeContact API. StartTime will be an hour before InitiationTimestamp and EndTime will be an hour after LastUpdateTimestamp.

- **StartQuery**

  Starts a query log for the provided ContactId in the CloudWatch Logs log group associated with the Amazon Connect instance provided in ConnectInstanceId. Queries time out after 60 minutes of runtime. If your query times out, reduce the time range being searched. You can view the queries currently in progress as well as your recent query history in the CloudWatch console. For more information see View running queries or query history.

- **WaitForQueryCompletion**

Waits for the CloudWatch Logs query log for the provided `ContactId` to complete. Notice that the query times out after 60 minutes of runtime. If your query times out, reduce the time range being searched. You can view the queries currently in progress as well as your recent query history in the Amazon Connect console. For more information see View running queries or query history.

- **UploadContactFlowLog**

  Gets the query result and uploads the contact flow log to the Amazon S3 bucket specified in `S3BucketName`.

- **GenerateReport**

  Returns the Amazon S3 console URL where the contact flow log was uploaded and an example Amazon CLI command that you can use to download the log file.

7. After completed, review the Outputs section for the detailed results of the execution:

- **GenerateReport.OutputPayload**

  Output that tells you the runbook successfully retrieved contact flow logs for the specified contact. This report also contains Amazon S3 console URL and an example Amazon CLI command so that you can download the log file.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation

- Support Automation Workflows landing page

Amazon Directory Service

Amazon Systems Manager Automation provides predefined runbooks for Amazon Directory Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-CreateDSManagementInstance
- AWSSupport-TroubleshootADConnectorConnectivity
- AWSSupport-TroubleshootDirectoryTrust

`AWS-CreateDSManagementInstance`

Description

The `AWS-CreateDSManagementInstance` runbook creates an Amazon Elastic Compute Cloud (Amazon EC2) Windows instance that you can use to manage your Amazon Directory Service directory. The management instance can't be used to manage AD Connector directories.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AmiID

  Type: String

  Default: `{{ ssm:/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-Base }}`

  Description: (Required) The ID of the Amazon Machine Image (AMI) you want to use to launch the management instance.

- DirectoryId

  Type: String

  Description: (Required) The ID of the Amazon Directory Service directory you want to manage. The instance is joined to the directory you specify.

- IamInstanceProfileName

  Type: String

  Description: (Required) The name you specify is applied to the IAM instance profile that is created by the automation and attached to the management instance.

- InstanceType

  Type: String

  Default: t3.medium

  Allowed values:

  - t2.nano
  - t2.micro
  - t2.small
  - t2.medium

- t2.large
- t2.xlarge
- t2.2xlarge
- t3.nano
- t3.micro
- t3.small
- t3.medium
- t3.large
- t3.xlarge
- t3.2xlarge

Description: (Required) The type of instance you want to launch.

- KeyPairName

  Type: String

  Description: (Optional) The key pair to use when creating the instance. If you do not specify a value, no key pair is associated with the instance.

- RemoteAccessCidr

  Type: String

  Description: (Required) The CIDR block you want to allow RDP traffic (port 3389) from. The CIDR block you specify is applied to an inbound rule that's added to the security group created by the automation.

- SecurityGroupName

  Type: String

  Description: (Required) The name you specify is applied to the security group that is created by the automation and associated with the management instance.

- Tags

  Type: MapList

  Description: (Optional) A key-value pair you want to apply to the resources created by the automation.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ds:DescribeDirectories
- ec2:AuthorizeSecurityGroupIngress
- ec2:CreateSecurityGroup
- ec2:CreateTags
- ec2:DeleteSecurityGroup
- ec2:DescribeInstances
- ec2:DescribeInstanceStatus
- ec2:DescribeKeyPairs
- ec2:DescribeSecurityGroups
- ec2:DescribeVpcs
- ec2:RunInstances
- ec2:TerminateInstances
- iam:AddRoleToInstanceProfile
- iam:AttachRolePolicy
- iam:CreateInstanceProfile
- iam:CreateRole
- iam:DeleteInstanceProfile
- iam:DeleteRole
- iam:DetachRolePolicy
- iam:GetInstanceProfile
- iam:GetRole
- iam:ListAttachedRolePolicies
- iam:ListInstanceProfiles
- iam:ListInstanceProfilesForRole
- iam:PassRole
- iam:RemoveRoleFromInstanceProfile

- iam:TagInstanceProfile
- iam:TagRole
- ssm:CreateDocument
- ssm:DeleteDocument
- ssm:DescribeInstanceInformation
- ssm:GetAutomationExecution
- ssm:GetParameters
- ssm:ListCommandInvocations
- ssm:ListCommands
- ssm:ListDocuments
- ssm:SendCommand
- ssm:StartAutomationExecution

Document Steps

- aws:executeAwsApi - Gathers details about the directory you specify in the DirectoryId parameter.
- aws:executeAwsApi - Gets the CIDR block of the virtual private cloud (VPC) where the directory was launched.
- aws:executeAwsApi - Creates a security group using the value you specify in the SecurityGroupName parameter.
- aws:executeAwsApi - Creates an inbound rule for the newly created security group that allows RDP traffic from the CIDR you specify in the RemoteAccessCidr parameter.
- aws:executeAwsApi - Creates an IAM role and instance profile using the value you specify in the IamInstanceProfileName parameter.
- aws:executeAwsApi - Launches an Amazon EC2 instance based on the values you specify in the runbook parameters.
- aws:executeAwsApi - Creates an Amazon Systems Manager document to join the newly launched instance to your directory.
- aws:runCommand - Joins the new instance to your directory.
- aws:runCommand - Installs remote server administration tools on the new instance.

AWSSupport-TroubleshootADConnectorConnectivity

Description

The AWSSupport-TroubleshootADConnectorConnectivity runbook verifies the following prerequisites for an AD Connector:

- Checks if the required traffic is allowed by the security group and network access control list (ACL) rules associated with your AD Connector.
- Checks if the Amazon Systems Manager, Amazon Security Token Service, and Amazon CloudWatch interface VPC endpoints exist in the same virtual private cloud (VPC) as the AD Connector.

When the prerequisite checks complete successfully, the runbook launches two Amazon Elastic Compute Cloud (Amazon EC2) Linux t2.micro instances in the same subnets as your AD Connector. Network connectivity tests are then performed using the netcat and nslookup utilities.

Run this Automation (console)

> **Important**
>
> Using this runbook might incur extra charges to your Amazon Web Services account for the Amazon EC2 instances, Amazon Elastic Block Store volumes and Amazon Machine Image (AMI) created during the automation. For more information, see Amazon Elastic Compute Cloud Pricing and Amazon Elastic Block Store Pricing.
> If the aws:deletestack step fails, go to the Amazon CloudFormation console to manually delete the stack. The stack name created by this runbook begins with AWSSupport-TroubleshootADConnectorConnectivity. For information about deleting Amazon CloudFormation stacks, see Deleting a stack in the *Amazon CloudFormation User Guide*.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DirectoryId

  Type: String

  Description: (Required) The ID of the AD Connector directory you want to troubleshoot connectivity to.
- Ec2InstanceProfile

  Type: String

  Maximum characters: 128

  Description: (Required) The name of the instance profile you want to assign to the instances that are launched to perform connectivity tests. The instance profile you specify must have the `AmazonSSMManagedInstanceCore` policy or equivalent permissions attached.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`
- `ec2:DescribeImages`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`

- ec2:DescribeNetworkAcls
- ec2:DescribeVpcEndpoints
- ec2:CreateTags
- ec2:RunInstances
- ec2:StopInstances
- ec2:TerminateInstances
- cloudformation:CreateStack
- cloudformation:DescribeStacks
- cloudformation:ListStackResources
- cloudformation:DeleteStack
- ds:DescribeDirectories
- ssm:SendCommand
- ssm:ListCommands
- ssm:ListCommandInvocations
- ssm:GetParameters
- ssm:DescribeInstanceInformation
- iam:PassRole

Document Steps

- aws:assertAwsResourceProperty - Confirms the directory specified in the DirectoryId parameter is an AD Connector.
- aws:executeAwsApi - Gathers information about the AD Connector.
- aws:executeAwsApi - Gathers information about the security groups that are associated with the AD Connector.
- aws:executeAwsApi - Gathers information about the network ACL rules that are associated with the subnets for the AD Connector.
- aws:executeScript - Evalutes the AD Connector security group rules to verify that the required outbound traffic is allowed.
- aws:executeScript - Evalutes the AD Connector network ACL rules to verify that the required outbound and inbound network traffic is allowed.

- `aws:executeScript` - Checks if the Amazon Systems Manager, Amazon Security Token Service and Amazon CloudWatch interface endpoints exist in the same VPC as the AD Connector.
- `aws:executeScript` - Compiles the outputs of the checks performed in the previous steps.
- `aws:branch` - Branches the automation depending on the output of previous steps. The automation stops here if the required outbound and inbound rules are missing for the security groups and network ACLs.
- `aws:createStack` - Creates an Amazon CloudFormation stack to launch Amazon EC2 instances to perform connectivity tests.
- `aws:executeAwsApi` - Gathers the IDs of newly launched Amazon EC2 instances.
- `aws:waitForAwsResourceProperty` - Waits for the first newly launched Amazon EC2 instance to report as managed by Amazon Systems Manager.
- `aws:waitForAwsResourceProperty` - Waits for the second newly launched Amazon EC2 instance to report as managed by Amazon Systems Manager.
- `aws:runCommand` - Performs network connectivity tests to the on-premises DNS server IP addresses from the first Amazon EC2 instance.
- `aws:runCommand` - Performs network connectivity tests to the on-premises DNS server IP addresses from the second Amazon EC2 instance.
- `aws:changeInstanceState` - Stops the Amazon EC2 instances used for the connectivity tests.
- `aws:deleteStack` - Deletes the Amazon CloudFormation stack.
- `aws:executeScript` - Outputs instructions about how to manually delete the Amazon CloudFormation stack if the automation fails to delete the stack.

AWSSupport-TroubleshootDirectoryTrust

Description

The `AWSSupport-TroubleshootDirectoryTrust` runbook diagnoses trust creation issues between an Amazon Managed Microsoft AD and a Microsoft Active Directory. The automation ensures the directory type supports trusts, and then checks the associated security group rules, network access control lists (network ACLs), and route tables for potential connectivity issues.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DirectoryId

  Type: String

  Allowed pattern: ^d-[a-z0-9]{10}$

  Description: (Required) The ID of the Amazon Managed Microsoft AD to troubleshoot.
- RemoteDomainCidrs

  Type: StringList

  Allowed pattern: ^(([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.){3}([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])(\/(3[0-2]|[1-2][0-9]|[1-9]))$

  Description: (Required) The CIDR(s) of the remote domain you are attempting to establish a trust relationship with. You can add multiple CIDRs using comma-separated values. For example, 172.31.48.0/20, 192.168.1.10/32.
- RemoteDomainName

  Type: String

  Description: (Required) The fully qualified domain name of the remote domain you are establishing a trust relationship with.

- RequiredTrafficACL

  Type: String

  Description: (Required) The default port requirements for Amazon Managed Microsoft AD. In most cases, you should not modify the default value.

  Default: {"inbound":{"tcp":[[53,53],[88,88],[135,135],[389,389],[445,445],[464,464],[636,636],[1024,65535]],"udp":[[53,53],[88,88],[123.123],[138,138],[389,389],[445,445],[464,464]],"icmp":[[-1,-1]]},"outbound":{"-1":[[0,65535]]}}

- RequiredTrafficSG

  Type: String

  Description: (Required) The default port requirements for Amazon Managed Microsoft AD. In most cases, you should not modify the default value.

  Default: {"inbound":{"tcp":[[53,53],[88,88],[135,135],[389,389],[445,445],[464,464],[636,636],[1024,65535]],"udp":[[53,53],[88,88],[123.123],[138,138],[389,389],[445,445],[464,464]],"icmp":[[-1,-1]]},"outbound":{"-1":[[0,65535]]}}

- TrustId

  Type: String

  Description: (Optional) The ID of the trust relationship to troubleshoot.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ds:DescribeConditionalForwarders`
- `ds:DescribeDirectories`
- `ds:DescribeTrusts`
- `ds:ListIpRoutes`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`

Document Steps

- `aws:assertAwsResourceProperty` - Confirms the directory type is Amazon Managed Microsoft AD.
- `aws:executeAwsApi` - Gets information about the Amazon Managed Microsoft AD.
- `aws:branch` - Branches automation if a value is provided for the `TrustId` input parameter.
- `aws:executeAwsApi` - Gets information about the trust relationship.
- `aws:executeAwsApi` - Gets the conditional forwarder DNS IP addresses for the `RemoteDomainName` .
- `aws:executeAwsApi` - Gets information about IP routes that have been added to the Amazon Managed Microsoft AD.
- `aws:executeAwsApi` - Gets the CIDRs of the Amazon Managed Microsoft AD subnets.
- `aws:executeAwsApi` - Gets information about the security groups associated with the Amazon Managed Microsoft AD.
- `aws:executeAwsApi` - Gets information about the network ACLs associated with the Amazon Managed Microsoft AD.
- `aws:executeScript` - Confirms the `RemoteDomainCidrs` are valid values. Confirms that the Amazon Managed Microsoft AD has conditional forwarders for the `RemoteDomainCidrs` , and that the requisite IP routes have been added to the Amazon Managed Microsoft AD if the `RemoteDomainCidrs` are non-RFC 1918 IP addresses.
- `aws:executeScript` - Evaluates security group rules.
- `aws:executeScript` - Evaluates network ACLs.

Outputs

evalDirectorySecurityGroup.output - Results from evaluating whether the security group rules associated with the Amazon Managed Microsoft AD allow the requisite traffic for trust creation.

evalAclEntries.output - Results from evaluating whether the network ACLs associated with the Amazon Managed Microsoft AD allow the requisite traffic for trust creation.

evaluateRemoteDomainCidr.output - Results from evaluating whether the `RemoteDomainCidrs` are valid values. Confirms that the Amazon Managed Microsoft AD has conditional forwarders for the `RemoteDomainCidrs` , and that the requisite IP routes have been added to the Amazon Managed Microsoft AD if the `RemoteDomainCidrs` are non-RFC 1918 IP addresses.

Amazon AppSync

Amazon Systems Manager Automation provides predefined runbooks for Amazon AppSync. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWS-EnableAppSyncGraphQLApiLogging

Description

The `AWS-EnableAppSyncGraphQLApiLogging` runbook enables field-level logging and request-level logging for the Amazon AppSync GraphQL API you specify. The runbook will apply changes to the specified GraphQL API even if logging has already been enabled.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ApiId

  Type: String

  Description: (Required) The ID of the API you want to enable logging for.

- FieldLogLevel

  Type: String

  Valid Values: ERROR | ALL

  Description: (Required) The field logging level.

- CloudWatchLogsRoleArn

  Type: String

  Description: (Required) The ARN of the service role that Amazon AppSync assumes to publish to Amazon CloudWatch Logs.

- ExcludeVerboseContent

  Type: Boolean

  Default: False

  Description: (Optional) Set to `True` to exclude information such as headers, context, and evaluated mapping templates, regardless of logging level.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `appsync:GetGraphqlApi`
- `appsync:UpdateGraphqlApi`
- `iam:PassRole`

Document Steps

- aws:executeAwsApi - Gathers the authentication type and configuration information relevant for the primary authentication type.
- aws:branch - Branches based on the authentication type.
- aws:executeAwsApi - Updates the logging configuration for the Amazon AppSync GraphQL API based on the values specified for the runbook's input parameters.

Outputs

- `EnableApiLoggingWithApiKeyOrAwsIamAuthorization.UpdateGraphQLApiResponse`: Response from the `UpdateGraphqlApi` call.
- `EnableApiLoggingWithLambdaAuthorization.UpdateGraphQLApiResponse`: Response from the `UpdateGraphqlApi` call.
- `EnableApiLoggingWithCognitoAuth.UpdateGraphQLApiResponse`: Response from the `UpdateGraphqlApi` call.
- `EnableApiLoggingWithOpenIdAuthorization.UpdateGraphQLApiResponse`: Response from the `UpdateGraphqlApi` call.

Amazon Athena

Amazon Systems Manager Automation provides predefined runbooks for Amazon Athena. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-EnableAthenaWorkGroupEncryptionAtRest

`AWS-EnableAthenaWorkGroupEncryptionAtRest`

Description

The `AWS-EnableAthenaWorkGroupEncryptionAtRest` runbook enables encryption at rest for the Amazon Athena workgroup you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- WorkGroup

  Type: String

  Description: (Required) The workgroup that you want to enable encryption at rest for.
- EncryptionOption

  Type: String

  Valid Values: SSE_S3 | SSE_KMS | CSE_KMS

  Description: (Required) Specifies which encryption option is used. You can choose server-side encryption with Amazon S3 managed keys (SSE_S3), server-side encryption with Amazon KMS managed keys (SSE_KMS), or client-side encryption with Amazon KMS managed keys (CSE_KMS).
- KmsKeyId

  Type: String

  Description: (Optional) If you're using a Amazon KMS encryption option, specify the key ARN, key ID, or the key alias of the key you want to use.

- EnableMinimumEncryptionConfiguration

  Type: Boolean

  Default: True

  Description: (Optional) Enforces a minimal level of encryption for the workgroup for query and calculation results that are written to Amazon S3. When enabled, workgroup users can set encryption only to the minimum level set by the administrator or higher when they submit queries. This setting does not apply to Spark-enabled workgroups.
- EnforceWorkGroupConfiguration

  Type: Boolean

  Default: True

  Description: (Optional) If set to `True`, the settings for the workgroup override client-side settings. If set to `False`, client-side settings are used.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `athena:GetWorkGroup`
- `athena:UpdateWorkGroup`

Document Steps

- aws:branch - Branches based on the encryption option specified in the `EncryptionOption` parameter.
- aws:executeAwsApi - This step updates the Athena Work Group with the specified encryption setting.
- aws:executeAwsApi - Updates the Athena Work Group with the specified encryption setting.
- aws:assertAwsResourceProperty - Verifies that encryption for the workgroup has been enabled.

DynamoDB

Amazon Systems Manager Automation provides predefined runbooks for Amazon DynamoDB. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWS-ChangeDDBRWCapacityMode

Description

The `AWS-ChangeDDBRWCapacityMode` runbook changes the read/write capacity mode for one or more Amazon DynamoDB (DynamoDB) tables to either on-demand mode, or provisioned mode.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- CapacityMode

  Type: String

  Valid values: PROVISIONED | PAY_PER_REQUEST

  Description: (Required) The desired read/write capacity mode. When switching from on-demand(pay-per-request) to provisioned capacity, initial provisioned capacity values must be set. The initial provisioned capacity values are estimated based on the consumed read and write capacity of your table and global secondary indexes over the past 30 minutes.

- ReadCapacityUnits

  Type: Integer

  Default: 0

  Description: (Optional) The maximum number of strongly consistent reads consumed per second before DynamoDB returns a throttling exception.

- TableNames

  Type: String

  Description: (Required) Comma separated list of DynamoDB table names to change the read/write capacity mode for..

- WriteCapacityUnits

  Type: Integer

  Default: 0

  Description: (Optional) The maximum number of writes consumed per second before DynamoDB returns a throttling exception.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `dynamodb:DescribeTable`
- `dynamodb:UpdateTable`

Document Steps

- `aws:executeScript` - Changes the read/write capacity mode for the DynamoDB tables specified in the `TableNames` parameter.

Outputs

ChangeDDBRWCapacityMode.SuccessesTables - List of DynamoDB table names where the capacity mode was successfully changed

ChangeDDBRWCapacityMode.FailedTables - Maplist of DynamoDB table names where changing the capacity mode failed and the reason for the failure.

AWS-CreateDynamoDBBackup

Description

Create a backup of an Amazon DynamoDB table.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BackupName

  Type: String

  Description: (Required) Name of the backup to create.
- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.
- TableName

  Type: String

  Description: (Required) Name of the DynamoDB table.

AWS-DeleteDynamoDbBackup

Description

Delete the backup of an Amazon DynamoDB table.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BackupArn

  Type: String

  Description: (Required) ARN of the DynamoDB table backup to delete.

AWSConfigRemediation-DeleteDynamoDbTable

Description

The `AWSConfigRemediation-DeleteDynamoDbTable` runbook deletes the Amazon DynamoDB (DynamoDB) table you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- TableName

  Type: String

  Description: (Required) The name of the DynamoDB table you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `dynamodb:DeleteTable`
- `dynamodb:DescribeTable`

Document Steps

- `aws:executeScript` - Deletes the DynamoDB table specified in the `TableName` parameter.
- `aws:executeScript` - Verifies the DynamoDB table has been deleted.

AWS-DeleteDynamoDbTableBackups

Description

Delete DynamoDB table backups based on retention days or count.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.
- RetentionCount

  Type: String

  Default: 10

  Description: (Optional) The number of backups to retain for the table. If more than the specified number of backup exist, the oldest backups beyond that number are deleted. Either RetentionCount or RetentionDays can be used, not both.
- RetentionDays

  Type: String

  Description: (Optional) The number of days to retain backups for the table. Backups older than the specified number of days are deleted. Either RetentionCount or RetentionDays can be used, not both.

- TableName

  Type: String

  Description: (Required) Name of the DynamoDB table.

AWSConfigRemediation-EnableEncryptionOnDynamoDbTable

Description

The `AWSConfigRemediation-EnableEncryptionOnDynamoDbTable` runbook encrypts an Amazon DynamoDB (DynamoDB) table using the Amazon Key Management Service (Amazon KMS) customer managed key you specify for the `KMSKeyId` parameter.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- KMSKeyId

  Type: String

Description: (Required) The ARN of the customer managed key you want to use to encrypt the DynamoDB table you specify in the `TableName` parameter.

- TableName

  Type: String

  Description: (Required) The name of the DynamoDB table you want to encrypt.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `dynamodb:DescribeTable`
- `dynamodb:UpdateTable`

Document Steps

- `aws:executeAwsApi` - Encrypts the DynamoDB table you specify in the `TableName` parameter.
- `aws:waitForAwsResourceProperty` - Verifies the `Enabled` property for the DynamoDB table's `SSESpecification` is set to `true` .
- `aws:assertAwsResourceProperty` - Verifies the DynamoDB table is encrypted with the customer managed key specified in the `KMSKeyId` parameter.

AWSConfigRemediation-EnablePITRForDynamoDbTable

Description

The `AWSConfigRemediation-EnablePITRForDynamoDbTable` runbook enables point-in-time recovery (PITR) on the Amazon DynamoDB table you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- TableName

  Type: String

  Description: (Required) The name of the DynamoDB table to enable point-in-time recovery on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `dynamodb:DescribeContinuousBackups`
- `dynamodb:UpdateContinuousBackups`

Document Steps

- `aws:executeAwsApi` - Enables point-in-time recovery on the DynamoDB table you specify in the `TableName` parameter.

- `aws:assertAwsResourceProperty` - Confirms point-in-time recovery is enabled on the DynamoDB table.

AWS-EnableDynamoDbAutoscaling

Description

The `AWS-EnableDynamoDbAutoscaling` runbook enables Application Auto Scaling for the provisioned capacity Amazon DynamoDB table you specify. Application Auto Scaling dynamically adjusts provisioned throughput capacity in response to traffic patterns. For more information, see Managing throughput capacity automatically with DynamoDB auto scaling in the *Amazon DynamoDB Developer Guide*.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- TableName

  Type: String

  Description: (Required) The name of the DynamoDB table you want to enable Application Auto Scaling on.

- MinReadCapacity

  Type: Integer

  Description: (Required) The minimum number of provisioned throughput read capacity units for the DynamoDB table.

- MaxReadCapacity

  Type: Integer

  Description: (Required) The maximum number of provisioned throughput read capacity units for the DynamoDB table.

- TargetReadCapacityUtilization

  Type: Integer

  Description: (Required) The desired target read capacity utilization. Target utilization is the percentage of consumed provisioned throughput at a point in time. You can set the auto scaling target utilization values between 20 and 90 percent.

- ReadScaleOutCooldown

  Type: Integer

  Description: (Required) The amount of time in seconds to wait for a previous read capacity scale-out activity to take effect.

- ReadScaleInCooldown

  Type: Integer

  Description: (Required) The amount of time in seconds after a read capacity scale-in activity completes before another scale-in activity can start.

- MinWriteCapacity

  Type: Integer

  Description: (Required) The minimum number of provisioned throughput write units for the DynamoDB table.

- MaxWriteCapacity

  Type: Integer

Description: (Required) The maximum number of provisioned throughput write units for the DynamoDB table.

- TargetWriteCapacityUtilization

  Type: Integer

  Description: (Required) The desired target write capacity utilization. Target utilization is the percentage of consumed provisioned throughput at a point in time. You can set the auto scaling target utilization values between 20 and 90 percent.

- WriteScaleOutCooldown

  Type: Integer

  Description: (Required) The amount of time in seconds to wait for a previous write capacity scale-out activity to take effect.

- WriteScaleInCooldown

  Type: Integer

  Description: (Required) The amount of time in seconds after a write capacity scale-in activity completes before another scale-in activity can start.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `application-autoscaling:DescribeScalableTargets`
- `application-autoscaling:DescribeScalingPolicies`
- `application-autoscaling:PutScalingPolicy`
- `application-autoscaling:RegisterScalableTarget`

- RegisterAppAutoscalingTargetWrite (`aws:executeAwsApi`) - Configures Application Auto Scaling on the DynamoDB table you specify.

- RegisterAppAutoscalingTargetWriteDelay (`aws:sleep`) - Sleeps to avoid API throttling.
- PutScalingPolicyWrite (`aws:executeAwsApi`) - Configures the target write capacity utilization for the DynamoDB table.
- PutScalingPolicyWriteDelay (`aws:sleep`) - Sleeps to avoid API throttling.
- RegisterAppAutoscalingTargetRead (`aws:executeAwsApi`) - Configures minimum and maximum read capacity units for the DynamoDB table.
- RegisterAppAutoscalingTargetReadDelay (`aws:sleep`) - Sleeps to avoid API throttling.
- PutScalingPolicyRead (`aws:executeAwsApi`) - Configures the target read capacity utilization for the DynamoDB table.
- VerifyDynamoDbAutoscalingEnabled (`aws:executeScript`) - Verifies Application Auto Scaling is enabled for the DynamoDB table according to the values you specify.

Outputs

- RegisterAppAutoscalingTargetWrite.Response
- PutScalingPolicyWrite.Response
- RegisterAppAutoscalingTargetRead.Response
- PutScalingPolicyRead.Response
- VerifyDynamoDbAutoscalingEnabled.DynamoDbAutoscalingEnabledResponse

AWS-RestoreDynamoDBTable

Description

The `AWS-RestoreDynamoDBTable` runbook restores the Amazon DynamoDB table that you specify using point-in-time recovery (PITR).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EnablePointInTimeRecoverAsNeeded

  Type: Boolean

  Default: true

  Description: (Optional) Determines whether the automation turns on point-in-time recovery as needed to restore the table.

- GlobalSecondaryIndexOverride

  Type: String

  Description: (Optional) The new global secondary indexes to replace the existing secondary indexes for the new table.

- LocalSecondaryIndexOverride

  Type: String

  Description: (Optional) The new local secondary indexes to replace the existing secondary indexes for the new table.

- RestoreDateTime

  Type: String

  Description: (Required) The point-in-time recovery that you want to restore your table to during the last 35 days. Specify the date and time using the following format: `DD/MM/YYYY HH:MM:SS`

- SourceTableArn

  Type: String

  Description: (Required) The ARN of the table that you want to restore.

- SseSpecificationOverride

  Type: String

  Description: (Optional) The server-side encryption settings to use for the new table.

- TargetTableName

  Type: String

  Description: (Required) The name of the table to restore.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `dynamodb:BatchWriteItem`
- `dynamodb:DeleteItem`
- `dynamodb:DescribeTable`
- `dynamodb:GetItem`
- `dynamodb:PutItem`
- `dynamodb:Query`
- `dynamodb:RestoreTableToPointInTime`
- `dynamodb:Scan`
- `dynamodb:UpdateItem`

Document Steps

- `aws:executeScript` - Restores the DynamoDB table that you specify in the `TargetTableName` parameter using point-in-time recovery.

Amazon EBS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Elastic Block Store. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

`AWSSupport-AnalyzeEBSResourceUsage`

Description

The `AWSSupport-AnalyzeEBSResourceUsage` automation runbook is used to analyze resource usage on Amazon Elastic Block Store (Amazon EBS). It analyzes volume usage and identifies abandoned volumes, images, and snapshots in a given Amazon Region.

How does it work?

The runbook performs the following four tasks:

1. Verifies that an Amazon Simple Storage Service (Amazon S3) bucket exists, or creates a new Amazon S3 bucket.
2. Gathers all the Amazon EBS volumes in the available state.
3. Gathers all Amazon EBS snapshots for which source volume has been deleted.
4. Gathers all Amazon Machine Images (AMIs) which are not in use by any non-terminated Amazon Elastic Compute Cloud (Amazon EC2) instances.

The runbook generates CSV reports and stores them in a user-provided Amazon S3 bucket. The provided bucket should be secured following Amazon security best practices as outlined in the end. If the user provided Amazon S3 bucket does not exist in the account, the runbook creates a new Amazon S3 bucket with the name format `<User-provided-name>-awssupport-YYYY-MM-DD`, encrypted with a custom Amazon Key Management Service (Amazon KMS) key, with object versioning enabled, blocked public access, and require requests to use SSL/TLS.

If you want to specify your own Amazon S3 bucket, please make sure it is configured following these best practices:

- Block public access to the bucket (set `IsPublic` to `False`).
- Turn on Amazon S3 access logging.
- Allow only SSL requests to your bucket.
- Turn on object versioning.
- Use an Amazon Key Management Service (Amazon KMS) key to encrypt your bucket.

> **Important**
>
> Using this runbook might incur extra charges against your account for the creation of Amazon S3 buckets and objects. See Amazon S3 Pricing for more details on the charges that might incur.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- S3BucketName

  Type: `AWS::S3::Bucket::Name`

  Description: (Required) The Amazon S3 bucket in your account to upload the report to. Ensure the bucket policy does not grant unnecessary read/write permissions to parties that do not need access to the collected logs. If the bucket specified does not exist in the account, then automation creates a new bucket in the Region where automation is initiated with the name format `<User-provided-name>-awssupport-YYYY-MM-DD`, encrypted with a custom Amazon KMS key.

  Allowed Pattern: `$|^(?!(^(([0-9]{1,3}[.]){3}[0-9]{1,3}$)))^((?!xn—)(?!.*-s3alias))[a-z0-9][-.a-z0-9]{1,61}[a-z0-9]$`

- CustomerManagedKmsKeyArn

  Type: String

  Description: (Optional) The custom Amazon KMS key Amazon Resource Name (ARN) for encrypting the new Amazon S3 bucket that will create if the bucket specified does not exist in the account. Automation fails if the bucket creation is attempted without specifying a custom Amazon KMS key ARN.

  Allowed Pattern: `(^$|^arn:aws:kms:[-a-z0-9]:[0-9]:key/[-a-z0-9]*$)`

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeImages`
- `ec2:DescribeInstances`
- `ec2:DescribeSnapshots`
- `ec2:DescribeVolumes`
- `kms:Decrypt`
- `kms:GenerateDataKey`
- `s3:CreateBucket`
- `s3:GetBucketAcl`
- `s3:GetBucketPolicyStatus`
- `s3:GetBucketPublicAccessBlock`
- `s3:ListBucket`
- `s3:ListAllMyBuckets`
- `s3:PutObject`
- `s3:PutBucketLogging`
- `s3:PutBucketPolicy`
- `s3:PutBucketPublicAccessBlock`
- `s3:PutBucketTagging`
- `s3:PutBucketVersioning`
- `s3:PutEncryptionConfiguration`
- `ssm:DescribeAutomationExecutions`

Example policy with minimum required IAM Permissions to run this runbook:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "Read_Only_Permissions",
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeImages",
```

```
                    "ec2:DescribeInstances",
                    "ec2:DescribeSnapshots",
                    "ec2:DescribeVolumes",
                    "ssm:DescribeAutomationExecutions"
                ],
                "Resource": ""
            }, {
                "Sid": "KMS_Generate_Permissions",
                "Effect": "Allow",
                "Action": ["kms:GenerateDataKey", "kms:Decrypt"],
                "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }, {
                "Sid": "S3_Read_Only_Permissions",
                "Effect": "Allow",
                "Action": [
                    "s3:GetBucketAcl",
                    "s3:GetBucketPolicyStatus",
                    "s3:GetBucketPublicAccessBlock",
                    "s3:ListBucket"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket1",
                    "arn:aws:s3:::amzn-s3-demo-bucket2/"
                ]
            }, {
                "Sid": "S3_Create_Permissions",
                "Effect": "Allow",
                "Action": [
                    "s3:CreateBucket",
                    "s3:PutObject",
                    "s3:PutBucketLogging",
                    "s3:PutBucketPolicy",
                    "s3:PutBucketPublicAccessBlock",
                    "s3:PutBucketTagging",
                    "s3:PutBucketVersioning",
                    "s3:PutEncryptionConfiguration"
                ],
                "Resource": "*"
            }]
        }
```

Instructions

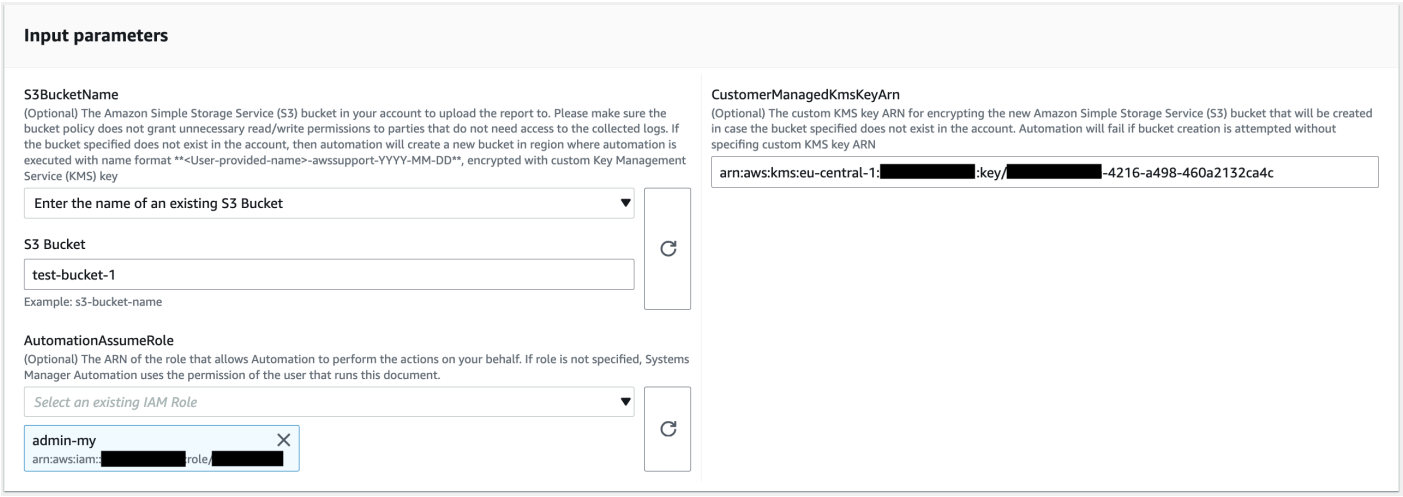
Follow these steps to configure the automation:

1. Navigate to the AWSSupport-AnalyzeEBSResourceUsage in the Amazon Systems Manager console.
2. For the input parameters enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **S3BucketName (Required):**

     The Amazon S3 bucket in your account to upload the report to.
   - **CustomerManagedKmsKeyArn (Optional):**

     The custom Amazon KMS key Amazon Resource Name (ARN) for encrypting the new Amazon S3 bucket that will create if the bucket specified does not exist in the account.



3. Select **Execute.**
4. The automation initiates.
5. The automation runbook performs the following steps:
   - **checkConcurrency:**

Ensures there is only one initiation of this runbook in the Region. If the runbook finds another execution in progress, it returns an error and ends.

- **verifyOrCreateS3bucket:**

  Verifies if the Amazon S3 bucket exists. If not, it creates a new Amazon S3 bucket in the Region where automation is initiated with the name format `<User-provided-name>-awssupport-YYYY-MM-DD`, encrypted with a custom Amazon KMS key.

- **gatherAmiDetails:**

  Searches for AMIs, which are not in use by any Amazon EC2 instances, generates the report with the name format `<region>-images.csv`, and uploads it to the Amazon S3 bucket.

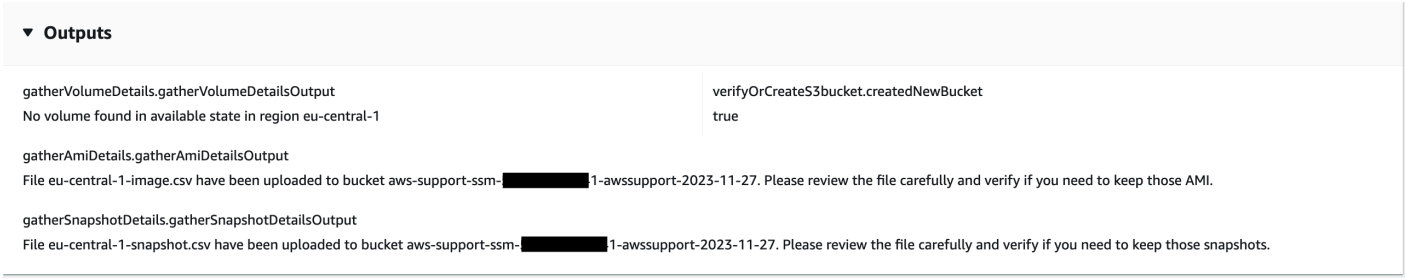
- **gatherVolumeDetails:**

  Verifies Amazon EBS volumes in the available state, generates the report with the name format `<region>-volume.csv`, and uploads it in an Amazon S3 bucket.

- **gatherSnapshotDetails:**

  Looks for the Amazon EBS snapshots of the Amazon EBS volumes that are deleted already, generates the report with the name format `<region>-snapshot.csv`, and uploads it to Amazon S3 bucket.

6. After completed, review the Outputs section for the detailed results of the execution.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWS-ArchiveEBSSnapshots

Description

The AWS-ArchiveEBSSnapshots runbook helps you archive snapshots for Amazon Elastic Block Store (Amazon EBS) volumes by specifying the tag you've applied to your snapshots. Alternatively, you can provide the ID of a volume if your snapshots are not tagged.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Description

  Type: String

  Description: (Optional) A description for the Amazon EBS snapshot.
- DryRun

  Type: String

  Valid values: Yes | No

Description: (Required) Checks whether you have the required permissions for the action, without actually making the request, and provides an error response.

- RetentionCount

  Type: String

  Description: (Optional) The number of snapshots you want to archive. Don't specify a value for this parameter if you specify a value for `RetentionDays`.

- RetentionDays

  Type: String

  Description: (Optional) The number of previous days of snapshots you want to archive. Don't specify a value for this parameter if you specify a value for `RetentionCount`.

- SnapshotWithTag

  Type: String

  Valid values: Yes | No

  Description: (Required) Specifies whether the snapshots you want to archive are tagged.

- TagKey

  Type: String

  Description: (Optional) The key of the tag assigned to the snapshots you want to archive.

- TagValue

  Type: String

  Description: (Optional) The value of the tag assigned to the snapshots you want to archive.

- VolumeId

  Type: String

  Description: (Optional) The ID of the volume whose snapshots you want to archive. Use this parameter if your snapshots are not tagged.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:ArchiveSnapshots`
- `ec2:DescribeSnapshots`

Document Steps

`aws:executeScript` - Archives snapshots using the tag you specify using the `TagKey` and `TagValue` parameters, or the `VolumeId` parameter.

AWS-AttachEBSVolume

Description

Attach an Amazon Elastic Block Store (Amazon EBS) volume to an Amazon Elastic Compute Cloud (Amazon EC2) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Device

  Type: String

  Description: (Required) The device name (for example, /dev/sdh or xvdh).

- InstanceId

  Type: String

  Description: (Required) The ID of the instance where you want to attach the volume.

- VolumeId

  Type: String

  Description: (Required) The ID of the Amazon EBS volume. The volume and instance must be in the same Availability Zone.

AWSSupport-CalculateEBSPerformanceMetrics

Description

The `AWSSupport-CalculateEBSPerformanceMetrics` runbook helps diagnose Amazon EBS performance issues by calculating and publishing performance metrics to a CloudWatch dashboard. The dashboard displays the estimated average IOPS and throughput for a target Amazon EBS volume or all the volumes attached to the target Amazon Elastic Compute Cloud (Amazon EC2) instance. For Amazon EC2 instances, it also shows the instance's average IOPS and throughput. The runbook outputs the link to the newly created CloudWatch dashboard that displays the relevant calculated CloudWatch metrics. The CloudWatch dashboard is created in your account with the name: `AWSSupport-<ResourceId>-EBS-Performance-<automation:EXECUTION_ID>`.

How does it work?

The runbook performs the following steps:

- Ensures that the specified timestamps are valid.
- Validates if the Resource ID (Amazon EBS Volume or Amazon EC2 Instance) is valid.

- When you provide an Amazon EC2 as a ResourceID, it creates a CloudWatch dashboard with Actual Total IOPS/Throughput for that Amazon EC2 instance and Estimated Average IOPS/Throughput graph for all Amazon EBS volumes attached to an Amazon EC2 instance.
- When you provide an Amazon EBS Volume as a ResourceID, it creates a CloudWatch dashboard with Estimated Average IOPS/Throughput graph for that volume.
- After the CloudWatch dashboard is generated, if Estimated Average IOPS or Estimated Average Throughput is more than Maximum IOPS or Maximum Throughput, respectively, then microbursting is possible for the volume or volumes attached to an Amazon EC2 instance.

Note

For burstable volumes (gp2, sc2, and st1), the maximum IOPS/throughput should be consider, until you have burst balance. After burst balance is completely utilized i.e. it becomes zero, consider baseline IOPS/throughput metrics.

Important

Creating the CloudWatch dashboard might result in your extra charges to your account. For more information, consult the Amazon CloudWatch Pricing guide.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeVolumes`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceTypes`
- `cloudwatch:PutDashboard`

Sample Policy

```
        {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "VisualEditor0",
                    "Effect": "Allow",
                    "Action": "cloudwatch:PutDashboard",
                    "Resource": "arn:aws:cloudwatch::Account-id:dashboard/*-EBS-
Performance-*"
                },
                {
                    "Sid": "VisualEditor1",
                    "Effect": "Allow",
                    "Action": [
                        "ec2:DescribeInstances",
                        "ec2:DescribeVolumes",
                        "ec2:DescribeInstanceTypes"
                    ],
                    "Resource": "*"
                }
            ]
        }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-CalculateEBSPerformanceMetrics in Systems Manager under Documents.

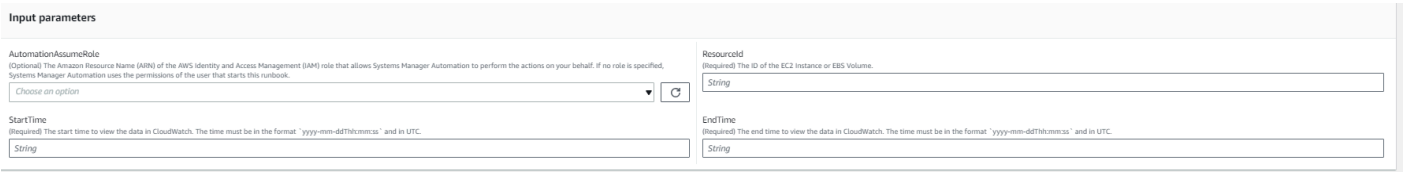
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **ResourceID (Required):**

     The ID of the Amazon EC2 instance or Amazon EBS volume.
   - **Start time (Required):**

     The start time to view the data in CloudWatch. The time must be in the format `yyyy-mm-ddThh:mm:ss` and in UTC.
   - **End time (Required):**

     The end time to view the data in CloudWatch. The time must be in the format `yyyy-mm-ddThh:mm:ss` and in UTC.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:
   - **CheckResourceIdAndTimeStamps:**

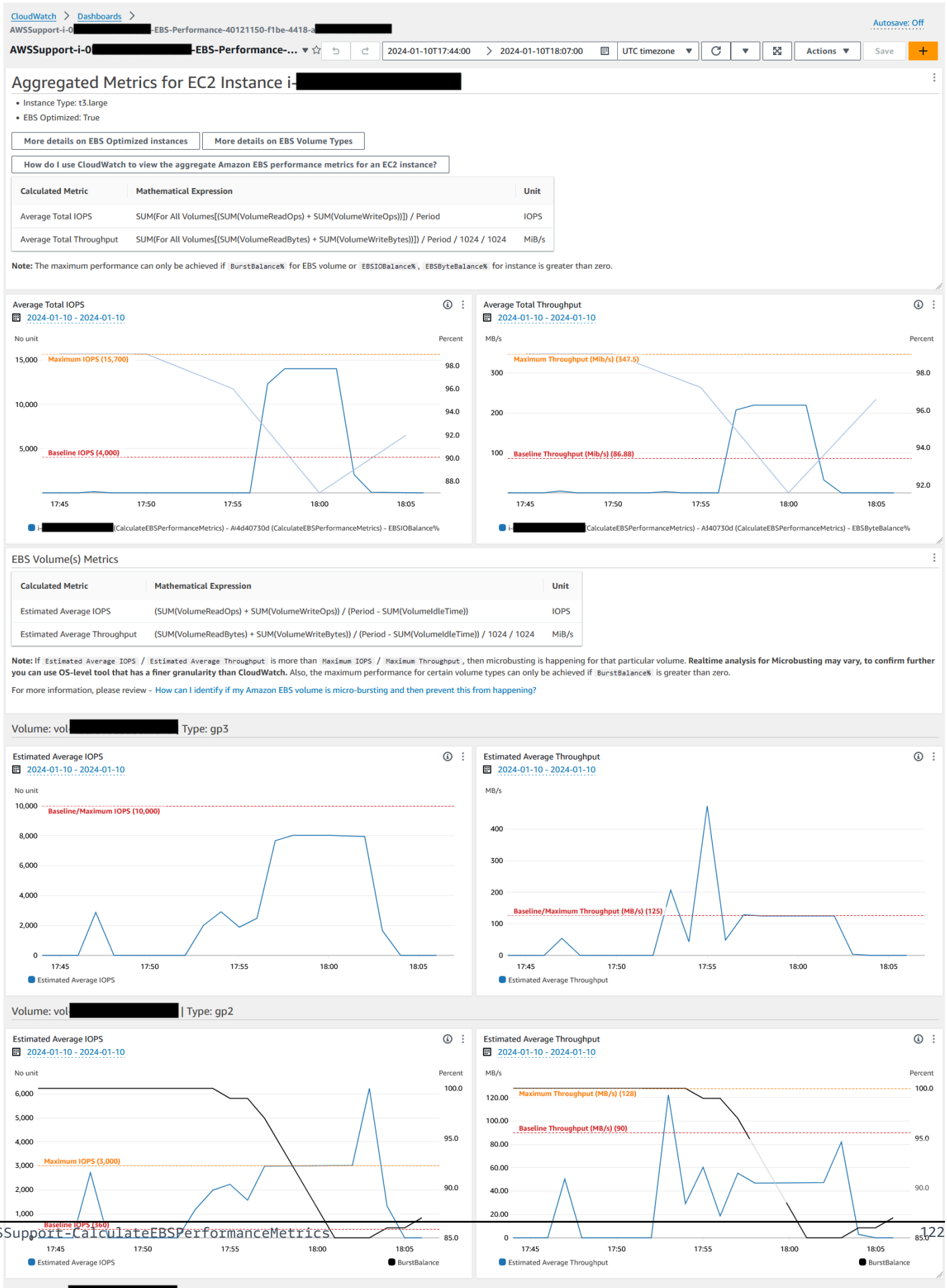
     Checks if end time is greater than start time by at least one minute and if the resource provided exists.
   - **CreateCloudWatchDashboard:**

     Calculates Amazon EBS performance and displays a graph based on your Resource ID. If you provide an Amazon EBS Volume ID for the parameter Resource ID, this runbook creates a CloudWatch dashboard with estimated average IOPS and estimated average throughput for the Amazon EBS volume. If you provide an Amazon EC2 Instance ID for the parameter

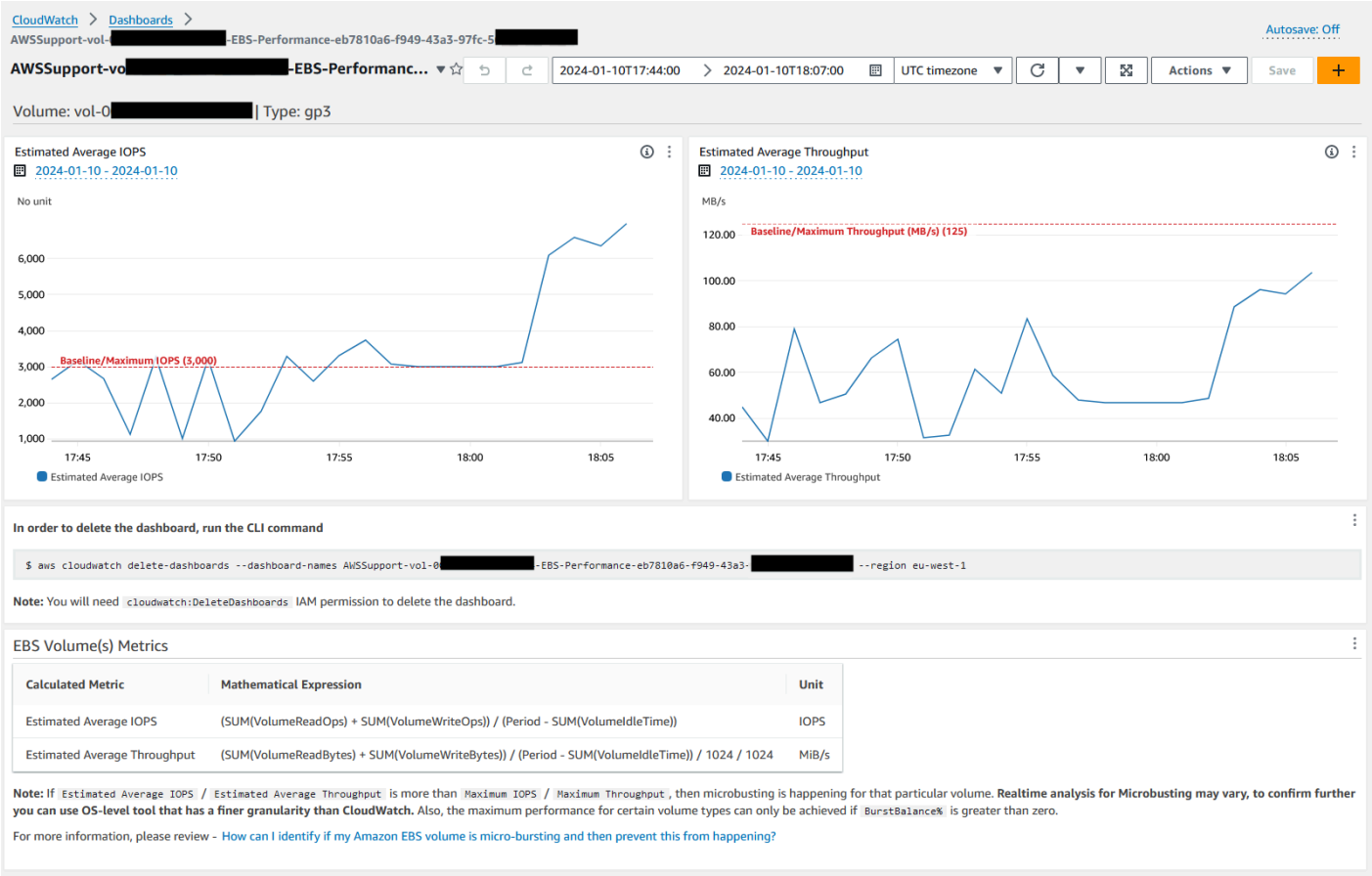
Resource ID, this runbook creates a CloudWatch dashboard with Average Total IOPS and Average Total Throughput for Amazon EC2 instance and with Estimated average IOPS and estimated average throughput for all Amazon EBS volumes attached to the Amazon EC2 instance.

7. After completed, review the Outputs section for the detailed results of the execution:

Example CloudWatch Dashboard For Resource ID as Amazon EC2 instance


CloudWatch > Dashboards >
AWSSupport-i-0[redacted]-EBS-Performance-40121150-f1be-4418-a[redacted]
Autosave: Off
AWSSupport-i-0[redacted]-EBS-Performance-...
2024-01-10T17:44:00 > 2024-01-10T18:07:00
UTC timezone
Actions
Save
Aggregated Metrics for EC2 Instance i-[redacted]
• Instance Type: t3.large
• EBS Optimized: True
More details on EBS Optimized instances
More details on EBS Volume Types
How do I use CloudWatch to view the aggregate Amazon EBS performance metrics for an EC2 instance?
Calculated Metric | Mathematical Expression | Unit
Average Total IOPS | SUM(For All Volumes[(SUM(VolumeReadOps) + SUM(VolumeWriteOps))]) / Period | IOPS
Average Total Throughput | SUM(For All Volumes[(SUM(VolumeReadBytes) + SUM(VolumeWriteBytes))]) / Period / 1024 / 1024 | MiB/s
Note: The maximum performance can only be achieved if BurstBalance% for EBS volume or EBSIOBalance%, EBSByteBalance% for instance is greater than zero.
Average Total IOPS
2024-01-10 - 2024-01-10
No unit
Percent
Maximum IOPS (15,700)
Baseline IOPS (4,000)
15,000
10,000
5,000
98.0
96.0
94.0
92.0
90.0
88.0
17:45
17:50
17:55
18:00
18:05
i-[redacted](CalculateEBSPerformanceMetrics) - AI4d40730d (CalculateEBSPerformanceMetrics) - EBSIOBalance%
Average Total Throughput
2024-01-10 - 2024-01-10
MB/s
Percent
Maximum Throughput (Mib/s) (347.5)
Baseline Throughput (Mib/s) (86.88)
300
200
100
98.0
96.0
94.0
92.0
17:45
17:50
17:55
18:00
18:05
i-[redacted]CalculateEBSPerformanceMetrics) - AI40730d (CalculateEBSPerformanceMetrics) - EBSByteBalance%
EBS Volume(s) Metrics
Calculated Metric | Mathematical Expression | Unit
Estimated Average IOPS | (SUM(VolumeReadOps) + SUM(VolumeWriteOps)) / (Period - SUM(VolumeIdleTime)) | IOPS
Estimated Average Throughput | (SUM(VolumeReadBytes) + SUM(VolumeWriteBytes)) / (Period - SUM(VolumeIdleTime)) / 1024 / 1024 | MiB/s
Note: If Estimated Average IOPS / Estimated Average Throughput is more than Maximum IOPS / Maximum Throughput, then microbusting is happening for that particular volume. Realtime analysis for Microbusting may vary, to confirm further you can use OS-level tool that has a finer granularity than CloudWatch. Also, the maximum performance for certain volume types can only be achieved if BurstBalance% is greater than zero.
For more information, please review - How can I identify if my Amazon EBS volume is micro-bursting and then prevent this from happening?
Volume: vol[redacted] Type: gp3
Estimated Average IOPS
2024-01-10 - 2024-01-10
No unit
Baseline/Maximum IOPS (10,000)
10,000
8,000
6,000
4,000
2,000
0
17:45
17:50
17:55
18:00
18:05
Estimated Average IOPS
Estimated Average Throughput
2024-01-10 - 2024-01-10
MB/s
Baseline/Maximum Throughput (MB/s) (125)
400
300
200
100
0
17:45
17:50
17:55
18:00
18:05
Estimated Average Throughput
Volume: vol[redacted] | Type: gp2
Estimated Average IOPS
2024-01-10 - 2024-01-10
No unit
Percent
Maximum IOPS (3,000)
Baseline IOPS (360)
6,000
5,000
4,000
3,000
2,000
1,000
0
100.0
95.0
90.0
85.0
17:45
17:50
17:55
18:00
18:05
Estimated Average IOPS
BurstBalance
Estimated Average Throughput
2024-01-10 - 2024-01-10
MB/s
Percent
Maximum Throughput (MB/s) (128)
Baseline Throughput (MB/s) (90)
120.00
100.00
80.00
60.00
40.00
20.00
0
100.0
95.0
90.0
85.0
17:45
17:50
17:55
18:00
18:05
Estimated Average Throughput
BurstBalance
Volume: vol[redacted] | Type: gp3
Estimated Average IOPS
Estimated Average Throughput

Example CloudWatch Dashboard For Resource ID as Amazon EBS volume id



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- How can I identify if my Amazon EBS volume is micro-bursting and then prevent this from happening?
- How do I use CloudWatch to view the aggregate Amazon EBS performance metrics for an EC2 instance?

AWS-CopySnapshot

Description

Copies a point-in-time snapshot of an Amazon Elastic Block Store (Amazon EBS) volume. You can copy the snapshot within the same Amazon Web Services Region or from one Region to another. Copies of encrypted Amazon EBS snapshots remain encrypted. Copies of unencrypted snapshots remain unencrypted. To copy an encrypted snapshot that was shared from another account, you must have permissions for the KMS key used to encrypt the snapshot. Snapshots created by copying another snapshot have an arbitrary volume ID that should not be used for any purpose.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Description

  Type: String

  Description: (Optional) A description for the Amazon EBS snapshot.
- SnapshotId

Type: String

Description: (Required) The ID of the Amazon EBS snapshot to copy.

- SourceRegion

  Type: String

  Description: (Required) The Region where the source snapshot currently exists.

Document Steps

copySnapshot - Copies a snapshot of an Amazon EBS volume.

Outputs

copySnapshot.SnapshotId - The ID of the new snapshot.

AWS-CreateSnapshot

Description

Create a snapshot of an Amazon EBS volume.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Description

  Type: String

  Description: (Optional) A description for the snapshot

- VolumeId

  Type: String

  Description: (Required) The ID of the volume.

AWS-DeleteSnapshot

Description

Delete a snapshot of an Amazon EBS volume.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- SnapshotId

  Type: String

  Description: (Required) The ID of the EBS snapshot.

AWSConfigRemediation-DeleteUnusedEBSVolume

Description

The `AWSConfigRemediation-DeleteUnusedEBSVolume` runbook deletes an unused Amazon Elastic Block Store (Amazon EBS) volume.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- CreateSnapshot

  Type: Boolean

  Description: (Optional) If set to `true` , the automation creates a snapshot of the Amazon EBS volume before it is deleted.

- VolumeId

  Type: String

  Description: (Required) The ID of the Amazon EBS volume that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:CreateSnapshot`
- `ec2:DeleteVolume`
- `ec2:DescribeSnapshots`
- `ec2:DescribeVolumes`

Document Steps

- `aws:executeScript` - Verifies the Amazon EBS volume you specify in the `VolumeId` parameter is not in use, and creates a snapshot depending on the value you choose for the `CreateSnapshot` parameter.
- `aws:branch` - Branches based on the value you chose for the `CreateSnapshot` parameter.
- `aws:waitForAwsResourceProperty` - Waits for the snapshot to complete.
- `aws:executeAwsApi` - Deletes the snapshot if the snapshot creation failed.
- `aws:executeAwsApi` - Deletes the Amazon EBS volume you specify in the `VolumeId` parameter.
- `aws:executeScript` - Verifies the Amazon EBS volume has been deleted.

AWS-DeregisterAMIs

Description

The AWS-DeregisterAMIs runbook helps you deregister Amazon Machine Images (AMIs) by specifying the tag that you've applied to your AMIs.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DryRun

  Type: String

  Valid values: Yes | No

  Description: (Required) Checks whether you have the required permissions for the action, without actually making the request, and provides an error response.
- RetainNumber

  Type: String

Description: (Optional) The number of AMIs that you want to retain. Don't specify a value for this parameter if you specify a value for Age.

- Age

  Type: String

  Description: (Optional) The number of previous days of AMIs that you want to retain. Don't specify a value for this parameter if you specify a value for RetainNumber.

- TagKey

  Type: String

  Description: (Required) The key of the tag assigned to the AMIs that you want to deregister.

- TagValue

  Type: String

  Description: (Required) The value of the tag assigned to the AMIs that you want to deregister.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ec2:DeregisterImage
- ec2:DescribeImages

Document Steps

- aws:executeAwsApi - Validates the values that you specify for the runbook input parameters.
- aws:executeAwsApi - Deregisters AMIs using the tag that you specify using the TagKey and TagValue parameters.

AWS-DetachEBSVolume

Description

Detach an Amazon EBS volume from an Amazon Elastic Compute Cloud (Amazon EC2) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role assumed by Lambda
- VolumeId

  Type: String

  Description: (Required) The ID of the EBS volume. The volume and instance must be within the same Availability Zone

AWSConfigRemediation-EnableEbsEncryptionByDefault

Description

The AWSConfigRemediation-EnableEbsEncryptionByDefault runbook enables encryption on all new Amazon Elastic Block Store (Amazon EBS) volumes in the Amazon Web Services account and Amazon Web Services Region where you run the automation. Volumes that were created before you run the automation are not encrypted.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ec2:EnableEbsEncryptionByDefault
- ec2:GetEbsEncryptionByDefault
- ssm:StartAutomationExecution
- ssm:GetAutomationExecution

Document Steps

- `aws:executeAwsApi` - Enables the default Amazon EBS encryption setting in the current account and Region.
- `aws:assertAwsResourceProperty` - Verifies that the default Amazon EBS encryption setting has been enabled.

AWS-ExtendEbsVolume

Description

The `AWS-ExtendEbsVolume` runbook increases the size of an Amazon EBS volume and extends the file system. This automation supports the `xfs` and `ext4` file systems.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DriveLetter

  Type: String

  Description: (Optional) The letter of the drive whose file system you want to extend. This parameter is required for Windows instances.

- InstanceId

  Type: String

  Description: (Optional) The ID of the Amazon EC2 instance that the Amazon EBS volume you want to extend is attached to.

- KeepSnapshot

  Type: Boolean

  Default: true

  Description: (Optional) Determines whether to keep the snapshot created before increasing the size of your Amazon EBS volume.

- MountPoint

  Type: String

  Description: (Optional) The mount point of the drive whose file system you want to extend. This parameter is required for Linux instances.

- SizeGib

  Type: String

  Description: (Required) The size in GiB that you want to modify your Amazon EBS volume to.

- VolumeId

  Type: String

  Description: (Required) The ID of the EBS volume that you want to extend.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:CreateSnapshot`
- `ec2:CreateTags`
- `ec2:DeleteSnapshot`

- ec2:DescribeVolumes
- ec2:ModifyVolume
- ssm:DescribeInstanceInformation
- ssm:GetCommandInvocation
- ssm:SendCommand

Document Steps

- aws:executeScript - Increases the size of the volume to the value that you specify in the VolumeId parameter and extends the file system.

AWSSupport-ModifyEBSSnapshotPermission

Description

The AWSSupport-ModifyEBSSnapshotPermission runbook helps you to modify permissions for multiple Amazon Elastic Block Store (Amazon EBS) snapshots. Using this runbook, you can make snapshots Public or Private and share them with other Amazon Web Services accounts. Snapshots encrypted with a default KMS key can't be shared with other accounts using this runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AccountIds

  Type: StringList

  Default: none

  Description: (Optional) The IDs of the accounts you want to share snapshots with. This parameter is required if you enter `No` for the value of the `Private` parameter.

- AccountPermissionOperation

  Type: String

  Valid values: add | remove

  Default: none

  Description: (Optional) The type of operation to perform.

- Private

  Type: String

  Valid values: Yes | No

  Description: (Required) Enter `No` for the value if you want to share snapshots with specific accounts.

- SnapshotIds

  Type: StringList

  Description: (Required) The IDs of Amazon EBS snapshots whose permission you want to modify.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ec2:DescribeSnapshots
- ec2:ModifySnapshotAttribute

Document Steps

1. aws:executeScript - Verifies the IDs of the snapshots provided in the SnapshotIds parameter. After verifying the IDs, the script checks for encrypted snapshots and outputs a list if any are found.
2. aws:branch - Branches the automation based on the value you enter for the Private parameter.
3. aws:executeScript - Modifies permissions of the snapshots specified to share it with the accounts specified.
4. aws:executeScript - Modifies permissions of the snapshots to change them from Public to Private .

Outputs

ValidateSnapshots.EncryptedSnapshots

SharewithOtherAccounts.Result

MakePrivate.Result

MakePrivate.Commands

AWSConfigRemediation-ModifyEBSVolumeType

Description

The AWSConfigRemediation-ModifyEBSVolumeType runbook modifies the volume type of an Amazon Elastic Block Store (Amazon EBS) volume. After the volume type is modified, the volume enters an optimizing state. For information about monitoring the progress of volume modifications, see Monitor the progress of volume modifications in the *Amazon EC2 User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- EbsVolumeId

  Type: String

  Description: (Required) The ID of the Amazon EBS volume that you want to modify.
- EbsVolumeType

  Type: String

  Valid values: standard | io1 | io2 | gp2 | gp3 | sc1 | st1

  Description: The volume type you want to change the Amazon EBS volume to. For information about Amazon EBS volume types, see Amazon EBS volume types in the *Amazon EC2 User Guide* .

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeVolumes`

- `ec2:ModifyVolume`

Document Steps

- `aws:waitForAwsResourceProperty` - Verifies the state of the volume is `available` or `in-use` .
- `aws:executeAwsApi` - Modifies the Amazon EBS volume you specify in the `EbsVolumeId` parameter.
- `aws:waitForAwsResourceProperty` - Verifies the type of the volume has been changed to the value you specified in the `EbsVolumeType` parameter.

Amazon EC2

Amazon Systems Manager Automation provides predefined runbooks for Amazon Elastic Compute Cloud. Runbooks for Amazon Elastic Block Store are located in the Amazon EBS section of the runbook reference. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-UpdateWindowsAmi
- AWSConfigRemediation-EnableAutoScalingGroupELBHealthCheck
- AWSConfigRemediation-EnforceEC2InstanceIMDSv2
- AWSEC2-CloneInstanceAndUpgradeSQLServer
- AWSEC2-CloneInstanceAndUpgradeWindows
- AWSEC2-ConfigureSTIG
- AWSEC2-PatchLoadBalancerInstance
- AWSEC2-SQLServerDBRestore
- AWSSupport-ActivateWindowsWithAmazonLicense
- AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2
- AWSPremiumSupport-ChangeInstanceTypeIntelToAMD
- AWSSupport-CheckXenToNitroMigrationRequirements
- AWSSupport-ConfigureEC2Metadata
- AWSSupport-ContainEC2Instance
- AWSSupport-CopyEC2Instance
- AWSSupport-EnableWindowsEC2SerialConsole
- AWSSupport-ExecuteEC2Rescue
- AWSSupport-ListEC2Resources
- AWSSupport-ManageRDPSettings
- AWSSupport-ManageWindowsService
- AWSSupport-MigrateEC2ClassicToVPC
- AWSSupport-MigrateXenToNitroLinux
- AWSSupport-ResetAccess
- AWSSupport-ResetLinuxUserPassword
- AWSPremiumSupport-ResizeNitroInstance
- AWSSupport-RestoreEC2InstanceFromSnapshot
- AWSSupport-SendLogBundleToS3Bucket
- AWSSupport-StartEC2RescueWorkflow
- AWSPremiumSupport-TroubleshootEC2DiskUsage
- AWSSupport-TroubleshootEC2InstanceConnect

- AWSSupport-TroubleshootLinuxMGNDRSAgentLogs
- AWSSupport-TroubleshootRDP
- AWSSupport-TroubleshootSSH
- AWSSupport-TroubleshootSUSERegistration
- AWSSupport-TroubleshootWindowsPerformance
- AWSSupport-TroubleshootWindowsUpdate
- AWSSupport-UpgradeWindowsAWSDrivers

AWS-ASGEnterStandby

Description

Change the standby state of an Amazon Elastic Compute Cloud (Amazon EC2) instance in an Auto Scaling group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

Type: String

Description: (Required) ID of an Amazon EC2 instance for which you want to change the standby state within an Auto Scaling group.

- LambdaRoleArn

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.

AWS-ASGExitStandby

Description

Change the standby state of an Amazon Elastic Compute Cloud (Amazon EC2) instance in an Auto Scaling group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) ID of an EC2 instance for which you want to change the standby state within an Auto Scaling group.

- LambdaRoleArn

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.

AWS-CreateImage

Description

Create a new Amazon Machine Image (AMI) from an Amazon Elastic Compute Cloud (Amazon EC2) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) The ID of the EC2 instance.

- NoReboot

  Type: Boolean

  Description: (Optional) Do not reboot the instance before creating the image.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateImage",
                "ec2:DescribeImages"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

AWS-DeleteImage

Description

Delete an Amazon Machine Image (AMI) and all associated snapshots.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ImageId

  Type: String

  Description: (Required) The ID of the AMI.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "ec2:DeleteSnapshot",
            "Resource": "arn:aws:ec2:{region}::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeImages",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeregisterImage",
            "Resource": "*"
        }
    ]
}
```

AWS-PatchAsgInstance

Description

Patch Amazon Elastic Compute Cloud (Amazon EC2) instances in an Auto Scaling group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) ID of the instance to patch. Don't specify an instance ID that is configured to run during a maintenance window.

- LambdaRoleArn

  Type: String

  Description: (Optional) The ARN of the role that allows the Lambda created by Automation to perform the actions on your behalf. If not specified, a transient role will be created to run the Lambda function.

- WaitForInstance

  Type: String

  Default: PT2M

  Description: (Optional) Duration that the Automation should sleep to allow the instance to come back into service.

- WaitForReboot

  Type: String

  Default: PT5M

  Description: (Optional) Duration that the Automation should sleep to allow a patched instance to reboot.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`

- ssm:GetCommandInvocation
- ssm:GetParameter
- ssm:SendCommand
- cloudformation:CreateStack
- cloudformation:DeleteStack
- cloudformation:DescribeStacks
- ec2:CreateTags
- ec2:DescribeInstances
- ec2:RunInstances
- iam:AttachRolePolicy
- iam:CreateRole
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:DetachRolePolicy
- iam:GetRole
- iam:PassRole
- iam:PutRolePolicy
- lambda:CreateFunction
- lambda:DeleteFunction
- lambda:GetFunction
- lambda:InvokeFunction

AWS-PatchInstanceWithRollback

Description

Brings an EC2 instance into compliance with the applicable patch baseline. Rolls back root volume on failure.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) EC2 InstanceId to which we apply the patch-baseline.
- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.
- ReportS3Bucket

  Type: String

  Description: (Optional) Amazon S3 Bucket destination for the Compliance Report generated during process.

Document Steps

Step number	Step name	Automation action
1	createDocumentStack	`aws:createStack`
2	IdentifyRootVolume	`aws:invokeLambdaFunction`
3	PrePatchSnapshot	`aws:executeAutomation`
4	installMissingUpdates	`aws:runCommand`
5	SleepThruInstallation	`aws:invokeLambdaFunction`
6	CheckCompliance	`aws:invokeLambdaFunction`
7	SaveComplianceReportToS3	`aws:invokeLambdaFunction`
8	ReportSuccessOrFailure	`aws:invokeLambdaFunction`
9	RestoreFromSnapshot	`aws:invokeLambdaFunction`
10	DeleteSnapshot	`aws:invokeLambdaFunction`
11	deleteCloudFormationTemplate	`aws:deleteStack`

Outputs

IdentifyRootVolume.Payload

PrePatchSnapshot.Output

SaveComplianceReportToS3.Payload

RestoreFromSnapshot.Payload

CheckCompliance.Payload

AWS-QuarantineEC2Instance

Description

With the AWS-QuarantineEC2Instance runbook, you can assign a security group to an Amazon Elastic Compute Cloud (Amazon EC2) instance that doesn't allow any inbound or outbound traffic.

> **Important**
>
> Changes to the RDP settings should be carefully reviewed before running this runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) The ID of the managed instance to manage the RDP settings of.

- IsolationSecurityGroup

  Type: String

  Description: (Required) The name of the security group that you want to assign to the instance to prevent inbound or outbound traffic.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `autoscaling:DescribeAutoScalingInstances`
- `autoscaling:DetachInstances`
- `ec2:CreateSecurityGroup`
- `ec2:CreateSnapshot`
- `ec2:DescribeInstances`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSnapshots`
- `ec2:ModifyInstanceAttribute`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:RevokeSecurityGroupIngress`

Document Steps

- `aws:executeAwsApi` - Gathers details about the instance.
- `aws:executeScript` - Verifies the instance isn't part of an Auto Scaling group.
- `aws:executeAwsApi` - Creates a snapshot of the root volume attached to the instance.
- `aws:waitForAwsResourceProperty` - Waits for the snapshot state to be `completed`.

- `aws:executeAwsApi` - Assigns the security group specified in the `IsolationSecurityGroup` parameter to your instance.

Outputs

`GetEC2InstanceResources.RevokedSecurityGroupsIds`

`GetEC2InstanceResources.RevokedSecurityGroupsNames`

`createSnapshot.SnapId`

AWS-ResizeInstance

Description

Change the instance type of an Amazon Elastic Compute Cloud (Amazon EC2) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) The ID of the instance.
- InstanceType

  Type: String

  Description: (Required) The instance type.
- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role assumed by Lambda.

AWS-RestartEC2Instance

Description

Restart one or more Amazon Elastic Compute Cloud (Amazon EC2) instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: StringList

  Description: (Required) The IDs of the Amazon EC2 instances to restart.

AWS-SetupJupyter

Description

The `AWS-SetupJupyter` runbook helps you set up Jupyter Notebook on an Amazon Elastic Compute Cloud (Amazon EC2) instance. You can either specify an existing instance, or provide an Amazon Machine Image (AMI) ID for the automation to launch and set up a new instance. Before you begin, you must create a `SecureString` parameter in Parameter Store to use as the password for Jupyter Notebook. Parameter Store is a tool in Amazon Systems Manager. For information about creating parameters, see Creating parameters in the *Amazon Systems Manager User Guide*.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AmiId

  Type: String

  Description: (Optional) The ID of the AMI that you want to use to launch a new instance and set up Jupyter Notebook.

- InstanceId

  Type: String

  Description: (Required) The ID of the instance that you want to set up Jupyter Notebook on.

- InstanceType

  Type: String

  Default: t3.medium

  Description: (Optional) If you're launching a new instance to set up Jupyter Notebook, specify the instance type that you want to use.

- JupyterPasswordSSMKey

  Type: String

  Description: (Required) The name of the `SecureString` parameter in Parameter Store that you want to use as the password for Jupyter Notebook.

- KeyPairName

  Type: String

  Description: (Optional) The key pair that you want to associate with the newly launched instance.

- RemoteAccessCidr

  Type: String

  Default: 0.0.0.0/0

Description: (Optional) The CIDR range that you want to allow SSH traffic from.

- RoleName

  Type: String

  Default: SSMManagedInstanceProfileRole

  Description: (Optional) The name of the instance profile for the newly launched instance.

- StackName

  Type: String

  Default: CreateManagedInstanceStack{{automation:EXECUTION_ID}}

  Description: (Optional) The Amazon CloudFormation stack name that you want the automation to use.

- SubnetId

  Type: String

  Default: Default

  Description: (Optional) The subnet that you want to launch the new instance to use.

- VpcId

  Type: String

  Default: Default

  Description: (Optional) The ID of the virtual private cloud (VPC) that you want to launch the new instance in to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:GetCommandInvocation`

- ssm:GetParameter
- ssm:SendCommand
- ssm:StartAutomationExecution
- cloudformation:CreateStack
- cloudformation:DeleteStack
- cloudformation:DescribeStacks
- ec2:DescribeInstances
- ec2:DescribeKeyPairs
- ec2:RunInstances
- iam:AttachRolePolicy
- iam:CreateRole
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:DetachRolePolicy
- iam:GetRole
- iam:PassRole
- iam:PutRolePolicy
- lambda:CreateFunction
- lambda:DeleteFunction
- lambda:GetFunction
- lambda:InvokeFunction

Document Steps

- aws:executeScript - Sets up Jupyter Notebook on the instance you specify, or on a newly launched instance, using the values that you specify for the runbook input parameters.

AWS-StartEC2Instance

Description

Start one or more Amazon Elastic Compute Cloud (Amazon EC2) instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: StringList

  Description: (Required) EC2 instances to start.

AWS-StopEC2Instance

Description

Stops one or more Amazon Elastic Compute Cloud (Amazon EC2) instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: StringList

  Description: (Required) EC2 instances to stop.

AWS-TerminateEC2Instance

Description

Terminate one or more Amazon Elastic Compute Cloud (Amazon EC2) instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: StringList

  Description: (Required) IDs of one or more EC2 instances to terminate.

AWS-UpdateLinuxAmi

Description

Update an Amazon Machine Image (AMI) with Linux distribution packages and Amazon software.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ExcludePackages

  Type: String

  Default: none

  Description: (Optional) Names of packages to hold back from updates, under all conditions. By default ("none"), no package is excluded.

- IamInstanceProfileName

  Type: String

  Default: ManagedInstanceProfile

  Description: (Required) The instance profile that enables Systems Manager to manage the instance.

- IncludePackages

  Type: String

  Default: all

  Description: (Optional) Only update these named packages. By default ("all"), all available updates are applied.

- InstanceType

  Type: String

  Default: t2.micro

  Description: (Optional) Type of instance to launch as the workspace host. Instance types vary by Region.

- MetadataOptions

  Type: StringMap

Default: {"HttpEndpoint": "enabled", "HttpTokens": "optional"}

Description: (Optional) The metadata options for the instance. For more information, see InstanceMetadataOptionsRequest.

- PostUpdateScript

  Type: String

  Default: none

  Description: (Optional) URL of a script to run after package updates are applied. Default ("none") is to not run a script.

- PreUpdateScript

  Type: String

  Default: none

  Description: (Optional) URL of a script to run before updates are applied. Default ("none") is to not run a script.

- SecurityGroupIds

  Type: String

  Description: (Required) A comma separated list of the IDs of the security groups you want to apply to the AMI.

- SourceAmiId

  Type: String

  Description: (Required) The source Amazon Machine Image ID.

- SubnetId

  Type: String

  Description: (Optional) The ID of the subnet you want to launch the instance into. If you have deleted your default VPC, this parameter is required.

- TargetAmiName

Type: String

Default: UpdateLinuxAmi_from_{{SourceAmiId}}_on_{{global:DATE_TIME}}

Description: (Optional) The name of the new AMI that will be created. Default is a system-generated string including the source AMI id, and the creation time and date.

AWS-UpdateWindowsAmi

Description

Update a Microsoft Windows Amazon Machine Image (AMI). By default, this runbook installs all Windows updates, Amazon software, and Amazon drivers. It then runs Sysprep to create a new AMI. Supports Windows Server 2008 R2 or later.

> **Important**
>
> If your instances connect to Amazon Systems Manager using VPC endpoints, this runbook will fail unless used in the us-east-1 Region. Instances must have TLS 1.2 enabled to use this runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Categories

  Type: String

  Description: (Optional) Specify one or more update categories. You can filter categories using comma-separated values. Options: Application, Connectors, CriticalUpdates, DefinitionUpdates, DeveloperKits, Drivers, FeaturePacks, Guidance, Microsoft, SecurityUpdates, ServicePacks, Tools, UpdateRollups, Updates. Valid formats include a single entry, for example: CriticalUpdates. Or you can specify a comma separated list: CriticalUpdates,SecurityUpdates. NOTE: There cannot be any spaces around the commas.

- ExcludeKbs

  Type: String

  Description: (Optional) Specify one or more Microsoft Knowledge Base (KB) article IDs to exclude. You can exclude multiple IDs using comma-separated values. Valid formats: KB9876543 or 9876543.

- IamInstanceProfileName

  Type: String

  Default: ManagedInstanceProfile

  Description: (Required) The name of the role that enables Systems Manager to manage the instance.

- IncludeKbs

  Type: String

  Description: (Optional) Specify one or more Microsoft Knowledge Base (KB) article IDs to include. You can install multiple IDs using comma-separated values. Valid formats: KB9876543 or 9876543.

- InstanceType

  Type: String

  Default: t2.medium

  Description: (Optional) Type of instance to launch as the workspace host. Instance types vary by region. Default is t2.medium.

- MetadataOptions

  Type: StringMap

  Default: {"HttpEndpoint": "enabled", "HttpTokens": "optional"}

  Description: (Optional) The metadata options for the instance. For more information, see InstanceMetadataOptionsRequest.

- PostUpdateScript

  Type: String

  Description: (Optional) A script provided as a string. It will run after installing OS updates.

- PreUpdateScript

  Type: String

  Description: (Optional) A script provided as a string. It will run prior to installing OS updates.

- PublishedDateAfter

  Type: String

  Description: (Optional) Specify the date that the updates should be published after. For example, if 01/01/2017 is specified, any updates that were found during the Windows Update search that have been published on or after 01/01/2017 will be returned.

- PublishedDateBefore

  Type: String

  Description: (Optional) Specify the date that the updates should be published before. For example, if 01/01/2017 is specified, any updates that were found during the Windows Update search that have been published on or before 01/01/2017 will be returned.

- PublishedDaysOld

  Type: String

  Description: (Optional) Specify the amount of days old the updates must be from the published date. For example, if 10 is specified, any updates that were found during the Windows Update search that have been published 10 or more days ago will be returned.

- SecurityGroupIds

  Type: String

  Description: (Required) A comma separated list of the IDs of the security groups you want to apply to the AMI.

- SeverityLevels

  Type: String

  Description: (Optional) Specify one or more MSRC severity levels associated with an update. You can filter severity levels using comma-separated values. By default patches for all security levels are selected. If value supplied, the update list is filtered by those values. Options: Critical, Important, Low, Moderate or Unspecified. Valid formats include a single entry, for example: Critical. Or, you can specify a comma separated list: Critical,Important,Low.

- SourceAmiId

  Type: String

  Description: (Required) The source AMI ID.

- SubnetId

  Type: String

  Description: (Optional) The ID of the subnet you want to launch the instance into. If you have deleted your default VPC, this parameter is required.

- TargetAmiName

  Type: String

  Default: UpdateWindowsAmi_from_{{SourceAmiId}}_on_{{global:DATE_TIME}}

Description: (Optional) The name of the new AMI that will be created. Default is a system-generated string including the source AMI id, and the creation time and date.

AWSConfigRemediation-EnableAutoScalingGroupELBHealthCheck

Description

The `AWSConfigRemediation-EnableAutoScalingGroupELBHealthCheck` runbook enables health checks for the Amazon EC2 Auto Scaling (Auto Scaling) group you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- AutoScalingGroupARN

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the auto scaling group that you want to enable health checks on.

- HealthCheckGracePeriod

  Type: Integer

  Default: 300

  Description: (Optional) The amount of time, in seconds, that Auto Scaling waits before checking the health status of an Amazon Elastic Compute Cloud (Amazon EC2) instance that has come into service.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeAutoScalingGroups`
- `ec2:UpdateAutoScalingGroup`

Document Steps

- `aws:executeScript` - Enables health checks on the Auto Scaling group you specify in the `AutoScalingGroupARN` parameter.

AWSConfigRemediation-EnforceEC2InstanceIMDSv2

Description

The `AWSConfigRemediation-EnforceEC2InstanceIMDSv2` runbook requires the Amazon Elastic Compute Cloud (Amazon EC2) instance you specify to use Instance Metadata Service Version 2 (IMDSv2).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance you want to require to use IMDSv2.
- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- HttpPutResponseHopLimit

  Type: Integer

  Description: (Optional) The Hop response limit from the IMDS service back to the requester. Set to 2 or greater for EC2 instances hosting containers. Set to 0 to not change (Default).

  Allowed pattern: `^([1-5]?\d|6[0-4])$`

  Default: 0

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeInstances`

- ec2:ModifyInstanceMetadataOptions

Document Steps

- aws:executeScript - Sets the HttpTokens option to required on the Amazon EC2 instance you specify in the InstanceId parameter.
- aws:assertAwsResourceProperty - Verifies IMDSv2 is required on the Amazon EC2 instance.

AWSEC2-CloneInstanceAndUpgradeSQLServer

Description

Create an AMI from an EC2 instance for Windows Server running SQL Server 2008 or later, and then upgrade the AMI to a later version of SQL Server. Only English versions of SQL Server are supported.

The following upgrade paths are supported:

- SQL Server 2008 to SQL Server 2017, 2016, or 2014
- SQL Server 2008 R2 to SQL Server 2017, 2016, or 2014
- SQL Server 2012 to SQL Server 2019, 2017, 2016, or 2014
- SQL Server 2014 to SQL Server 2019, 2017, or 2016
- SQL Server 2016 to SQL Server 2019 or 2017

If you are using an earlier version of Windows Server that is incompatible with SQL Server 2019, the automation document must upgrade your Windows Server version to 2016.

The upgrade is a multi-step process that can take 2 hours to complete. The automation creates the AMI from the instance, and then launches a temporary instance from the new AMI in the specified SubnetID. The security groups associated with your original instance are applied to the temporary instance. The automation then performs an in-place upgrade to the TargetSQLVersion on the temporary instance. After the upgrade, the automation creates a new AMI from the temporary instance and then terminates the temporary instance.

You can test application functionality by launching the new AMI in your VPC. After you finish testing, and before you perform another upgrade, schedule application downtime before completely switching over to the upgraded instance.

Note

If you want to modify the computer name of the EC2 instance launched from the new AMI , see Rename a Computer that Hosts a Stand-Alone Instance of SQL Server.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

Prerequisites

- TLS version 1.2.
- Only English versions of SQL Server are supported.
- The EC2 instance must use a version of Windows Server that is Windows Server 2008 R2 (or later) and SQL Server 2008 (or later).
- Verify that SSM Agent is installed on your instance. For more information, see Installing and configuring SSM Agent on EC2 instances for Windows Server.
- Configure the instance to use an Amazon Identity and Access Management (IAM) instance profile role. For more information, see Create an IAM instance profile for Systems Manager.
- Verify that the instance has 20 GB of free disk space in the instance boot disk.
- For instances that use a Bring Your Own License (BYOL) SQL Server version, the following additional prerequisites apply:
  - Provide an EBS snapshot ID that includes the target SQL Server installation media. To do this:

    1. Verify that the EC2 instance is running Windows Server 2008 R2 or later.

2. Create a 6 GB EBS volume in the same Availability Zone where the instance is running. Attach the volume to the instance. Mount it, for example, as drive D.
3. Right-click the ISO and mount it to an instance as, for example, drive E.
4. Copy the content of the ISO from drive E:\ to drive D:\
5. Create an EBS snapshot of the 6 GB volume created in step 2.

Limitations

- The upgrade can be performed on only a SQL Server using Windows authentication.
- Verify that no security patch updates are pending on the instances. Open **Control Panel**, then choose **Check for updates**.
- SQL Server deployments in HA and mirroring mode are not supported.

Parameters

- IamInstanceProfile

  Type: String

  Description: (Required) The IAM instance profile.
- InstanceId

  Type: String

  Description: (Required) The instance running Windows Server 2008 R2 (or later) and SQL Server 2008 (or later).
- KeepPreUpgradeImageBackUp

  Type: String

  Description: (Optional) If set to `true`, the automation doesn't delete the AMI created from the instance before the upgrade. If set to `true`, then you must delete the AMI. By default, the AMI is deleted.
- SubnetId

  Type: String

Description: (Required) Provide a subnet for the upgrade process. Verify that the subnet has outbound connectivity to Amazon services, Amazon S3, and Microsoft (to download patches).

- SQLServerSnapshotId

  Type: String

  Description: (Conditional) Snapshot ID for target SQL Server installation media. This parameter is required for instances that use a BYOL SQL Server version. This parameter is optional for SQL Server license-included instances (instances launched using an Amazon provided Amazon Machine Image for Windows Server with Microsoft SQL Server).

- RebootInstanceBeforeTakingImage

  Type: String

  Description: (Optional) If set to `true`, the automation reboots the instance before creating a pre-upgrade AMI. By default, the automation doesn't reboot before upgrade.

- TargetSQLVersion

  Type: String

  Description: (Optional) Select the target SQL Server version.

  Possible targets:

  - SQL Server 2019
  - SQL Server 2017
  - SQL Server 2016
  - SQL Server 2014

  Default target: SQL Server 2016

Outputs

AMIId: The ID of the AMI created from the instance that was upgraded to a later version of SQL Server.

AWSEC2-CloneInstanceAndUpgradeWindows

Description

Create an Amazon Machine Image (AMI) from a Windows Server 2008 R2, 2012 R2, 2016, or 2019 instance, and then upgrade the AMI to Windows Server 2016, 2019, or 2022. The supported upgrade paths are as follows.

- Windows Server 2008 R2 to Windows Server 2016.
- Windows Server 2012 R2 to Windows Server 2016.
- Windows Server 2012 R2 to Windows Server 2019.
- Windows Server 2012 R2 to Windows Server 2022.
- Windows Server 2016 to Windows Server 2019.
- Windows Server 2016 to Windows Server 2022.
- Windows Server 2019 to Windows Server 2022.

The upgrade operation is a multi-step process that can take 2 hours to complete. We recommend performing an operating system upgrade on instances with at least 2 vCPUs and 4GB of RAM. The automation creates an AMI from the instance and then launches a temporary instance from the newly created AMI in the `SubnetId` that you specify. The security groups associated with your original instance are applied to the temporary instance. The automation then performs an in-place upgrade to the `TargetWindowsVersion` on the temporary instance. To upgrade your Windows Server 2008 R2 instance to Windows Server 2016, 2019, or 2022, an in-place upgrade is performed twice because directly upgrading Windows Server 2008 R2 to Windows Server 2016, 2019, or 2022 is not supported. The automation also updates or installs the Amazon drivers required by the temporary instance. After the upgrade, the automation creates a new AMI from the temporary instance and then terminates the temporary instance.

You can test application functionality by launching a test instance from the upgraded AMI in your Amazon Virtual Private Cloud (Amazon VPC). After you finish testing, and before you perform another upgrade, schedule application downtime before completely switching over to the upgraded AMI.

Run this Automation (console)

Document Type

Automation

Owner

Amazon

Platforms

Windows Server 2008 R2, 2012 R2, 2016, or 2019 Standard and Datacenter editions

Prerequisites

- TLS version 1.2.
- Verify that SSM Agent is installed on your instance. For more information, see Installing and configuring SSM Agent on EC2 instances for Windows Server.
- Windows PowerShell 3.0 or later must be installed on your instance.
- For instances that are joined to a Microsoft Active Directory domain, we recommend specifying a `SubnetId` that does not have connectivity to your domain controllers to help avoid hostname conflicts.
- The instance subnet must have outbound connectivity to the internet, which provides access to Amazon Web Services services such as Amazon S3 and access to download patches from Microsoft. This requirement is met if either the subnet is a public subnet and the instance has a public IP address, or if the subnet is a private subnet with a route that sends internet traffic to a public NAT device.
- This Automation works only with Windows Server 2008 R2, 2012 R2, 2016, and 2019 instances.
- Configure the Windows Server instance with an Amazon Identity and Access Management (IAM) instance profile that provides the requisite permissions for Systems Manager. For more information, see Create an IAM instance profile for Systems Manager.
- Verify that the instance has 20 GB of free disk space in the boot disk.
- If the instance does not use an Amazon-provided Windows license, then specify an Amazon EBS snapshot ID that includes Windows Server 2012 R2 installation media. To do this:
  - Verify that the EC2 instance is running Windows Server 2012 or later.
  - Create a 6 GB EBS volume in the same Availability Zone where the instance is running. Attach the volume to the instance. Mount it, for example, as drive D.
  - Right-click the ISO and mount it to an instance as, for example, drive E.
  - Copy the content of the ISO from drive E:\ to drive D:\
  - Create an EBS snapshot of the 6 GB volume created in step 2 above.

Limitations

This Automation doesn't support upgrading Windows domain controllers, clusters, or Windows desktop operating systems. This Automation also doesn't support EC2 instances for Windows Server with the following roles installed.

- Remote Desktop Session Host (RDSH)
- Remote Desktop Connection Broker (RDCB)
- Remote Desktop Virtualization Host (RDVH)
- Remote Desktop Web Access (RDWA)

Parameters

- AlternativeKeyPairName

  Type: String

  Description: (Optional) The name of an alternative key pair to use during the upgrade process. This is useful in situations where the key pair assigned to the original instance is unavailable. If the original instance was not assigned a key pair, you must specify a value for this parameter.
- BYOLWindowsMediaSnapshotId

  Type: String

  Description: (Optional) The ID of the Amazon EBS snapshot to copy that includes Windows Server 2012R2 installation media. Required only if you are upgrading a BYOL instance.
- IamInstanceProfile

  Type: String

  Description: (Required) The name of the IAM instance profile that enables Systems Manager to manage the instance.
- InstanceId

  Type: String

  Description: (Required) The EC2 instance running Windows Server 2008 R2, 2012 R2, 2016, or 2019.
- KeepPreUpgradeImageBackUp

Type: String

Description: (Optional) If set True, the Automation doesn't delete the AMI created from the EC2 instance before the upgrade. If set to True, then you must delete the AMI. By default, the AMI is deleted.

- SubnetId

  Type: String

  Description: (Required) This is the subnet for the upgrade process and where your source EC2 instance resides. Verify that the subnet has outbound connectivity to Amazon services, Amazon S3, and Microsoft (to download patches).

- TargetWindowsVersion

  Type: String

  Description: (Required) Select the target Windows version.

  Default: 2022

- RebootInstanceBeforeTakingImage

  Type: String

  Description: (Optional) If set True, the Automation reboots the instance before creating a pre-upgrade AMI. By default, the Automation doesn't reboot before upgrade.

AWSEC2-ConfigureSTIG

Security Technical Implementation Guides (STIGs) are the configuration hardening standards created by the Defense Information Systems Agency (DISA) to secure information systems and software. To make your systems compliant with STIG standards, you must install, configure, and test a variety of security settings.

Amazon EC2 provides a Systems Manager command document, AWSEC2-ConfigureSTIG, which you can use to apply STIG settings to an instance. This document helps you to quickly build compliant images for STIG standards. The STIG Systems Manager document scans for misconfigurations and runs a remediation script. It also installs InstallRoot from the Department of Defense (DoD) on Windows AMIs to install and update the DoD certificates and to remove unnecessary certificates

to maintain STIG compliance. There are no additional charges for using the STIG Systems Manager document.

> **Note**
>
> AWSEC2-ConfigureSTIG is a command document. To search for AWSEC2-ConfigureSTIG in the Amazon Systems Manager console, select the category **Command documents**.

This page lists all STIGs that Amazon EC2 supports that the STIG hardening components apply to your EC2 instance.

You can choose which STIG compliance category to apply.

Compliance levels

- **High (Category I)**

  The most severe risk. Includes any vulnerability that can result in loss of confidentiality, availability, or integrity.
- **Medium (Category II)**

  Includes any vulnerability that can result in loss of confidentiality, availability, or integrity but the risk can be mitigated.
- **Low (Category III)**

  Includes any vulnerability that degrades measures to protect against loss of confidentiality, availability, or integrity.

Topics

STIG hardening component downloads

Amazon groups STIG hardening components together into operating system related bundles for each release. Bundles are archive files that are appropriate for the target operating system where they download and run. Linux component bundles are stored as TAR files (.tgz file extension). Windows component bundles are stored as ZIP files (.zip file extension).

Amazon stores the component bundles in the Image Builder S3 `STIG` bucket in each Amazon Web Services Region. Use SSL/TLS to communicate with Amazon resources. We require TLS 1.2 and recommend TLS 1.3.

> **Important**
>
> With few exceptions, the STIG hardening components that the Systems Manager document downloads do not install third-party packages. If third-party packages are already installed on the instance, and if there are related STIGs that Amazon EC2 supports for that package, those STIGs are applied.

Patterns and examples for component storage paths and bundle file names are as follows:

Component storage path

`s3://aws-windows-downloads-<region>/STIG/<bundle file name>`

Component path variables

region

Amazon Web Services Region (Each Region has its own components bucket.)

bundle file name

The format is *<os bundle name>*_*<YYYY>*_Q*<quarter>*[_*<release>*].*<file extension>*. *Note that the name has underscores between the nodes, not periods.*

os bundle name

The standard name prefix for the operating system bundle is either `LinuxAWSConfigureSTIG` or `AWSConfigureSTIG`. To maintain backwards compatibility, the download for Windows doesn't include a platform prefix.

YYYY

The four digit year of the release.

quarter

Identifies the quarter of the year: 1, 2, 3, or 4.

release

Incremental number that starts at one, and increments by one for each new release. The release is not included for the first release in a quarter and is only added for subsequent releases.

file extension

Compressed file format `tgz` (Linux) or `zip` (Windows).

Example bundle file names

- `LinuxAWSConfigureSTIG_2023_Q1_2.tgz`
- `AWSConfigureSTIG_2022_Q4.zip`

Windows STIG settings

Amazon EC2 Windows STIG AMIs and hardening components are designed for standalone servers and apply Local Group Policy. STIG-compliant components install InstallRoot from the Department of Defense (DoD) on Windows AMIs to download, install and update the DoD certificates. They also remove unnecessary certificates to maintain STIG compliance. Currently, Amazon EC2 supports STIG baselines for the following versions of Windows Server: 2012 R2, 2016, 2019, and 2022.

This section lists current STIG settings that Amazon EC2 supports for your Windows infrastructure, followed by a version history log.

You can apply low, medium, or high STIG settings.

Windows STIG Low (Category III)

The following list contains STIG settings that Amazon EC2 supports to your infrastructure. If a supported setting isn't applicable for your infrastructure, Amazon EC2 skips that setting, and moves on. For example, some STIG hardening settings might not apply to standalone servers.

Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

- **Windows Server 2022 STIG Version 2 Release 2**

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363, and V-254481

- **Windows Server 2019 STIG Version 3 Release 2**

  V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871, and V-205923

- **Windows Server 2016 STIG Version 2 Release 9**

  V-224916, V-224917, V-224918, V-224919, V-224931, V-224942, and V-225060

- **Windows Server 2012 R2 MS STIG Version 3 Release 5**

  V-225537, V-225536, V-225526, V-225525, V-225514, V-225511, V-225490, V-225489, V-225488, V-225487, V-225485, V-225484, V-225483, V-225482, V-225481, V-225480, V-225479, V-225476, V-225473, V-225468, V-225462, V-225460, V-225459, V-225412, V-225394, V-225392, V-225376, V-225363, V-225362, V-225360, V-225359, V-225358, V-225357, V-225355, V-225343, V-225342, V-225336, V-225335, V-225334, V-225333, V-225332, V-225331, V-225330, V-225328, V-225327, V-225324, V-225319, V-225318, and V-225250

- **Microsoft .NET Framework 4.0 STIG Version 2 Release 2**

  No STIG settings apply to the Microsoft .NET Framework for Category III vulnerabilities.

- **Windows Firewall STIG Version 2 Release 2**

  V-241994, V-241995, V-241996, V-241999, V-242000, V-242001, V-242006, V-242007, and V-242008

- **Internet Explorer 11 STIG Version 2 Release 5**

  V-46477, V-46629, and V-97527

- **Microsoft Edge STIG Version 2 Release 2 (Windows Server 2022 only)**

  V-235727, V-235731, V-235751, V-235752, and V-235765

Windows STIG Medium (Category II)

The following list contains STIG settings that Amazon EC2 supports to your infrastructure. If a supported setting isn't applicable for your infrastructure, Amazon EC2 skips that setting, and moves on. For example, some STIG hardening settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

> **Note**
>
> The Windows STIG Medium category includes all of the listed STIG hardening settings that apply to Windows STIG low (Category III), in addition to the STIG hardening settings that Amazon EC2 supports for Category II vulnerabilities.

- **Windows Server 2022 STIG Version 2 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-254247, V-254265, V-254269, V-254270, V-254271, V-254272, V-254273, V-254274, V-254276, V-254277, V-254278, V-254285, V-254286, V-254287, V-254288, V-254289, V-254290, V-254291, V-254292, V-254300, V-254301, V-254302, V-254303, V-254304, V-254305, V-254306, V-254307, V-254308, V-254309, V-254310, V-254311, V-254312, V-254313, V-254314, V-254315, V-254316, V-254317, V-254318, V-254319, V-254320, V-254321, V-254322, V-254323, V-254324, V-254325, V-254326, V-254327, V-254328, V-254329, V-254330, V-254331, V-254332, V-254333, V-254334, V-254339, V-254341, V-254342, V-254344, V-254345, V-254346, V-254347, V-254348, V-254349, V-254350, V-254355, V-254356, V-254356, V-254358, V-254359, V-254360, V-254361, V-254362, V-254364, V-254365, V-254366, V-254367, V-254368, V-254369, V-254370, V-254371, V-254372, V-254373, V-254375, V-254376, V-254377, V-254379, V-254380, V-254382, V-254383, V-254384, V-254431, V-254432, V-254433, V-254434, V-254435, V-254436, V-254438, V-254439, V-254442, V-254443, V-254444, V-254445, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254468, V-254470, V-254471, V-254472, V-254473, V-254476, V-254477, V-254478, V-254479, V-254480, V-254482, V-254483,

V-254484, V-254485, V-254486, V-254487, V-254488, V-254489, V-254490, V-254493, V-254494, V-254495, V-254497, V-254499, V-254501, V-254502, V-254503, V-254504, V-254505, V-254507, V-254508, V-254509, V-254510, V-254511, and V-254512

- **Windows Server 2019 STIG Version 3 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-205625, V-205626, V-205627, V-205629, V-205630, V-205633, V-205634, V-205635, V-205636, V-205637, V-205638, V-205639, V-205643, V-205644, V-205648, V-205649, V-205650, V-205651, V-205652, V-205655, V-205656, V-205659, V-205660, V-205662, V-205671, V-205672, V-205673, V-205675, V-205676, V-205678, V-205679, V-205680, V-205681, V-205682, V-205683, V-205684, V-205685, V-205686, V-205687, V-205688, V-205689, V-205690, V-205692, V-205693, V-205694, V-205697, V-205698, V-205708, V-205709, V-205712, V-205714, V-205716, V-205717, V-205718, V-205719, V-205720, V-205722, V-205729, V-205730, V-205733, V-205747, V-205751, V-205752, V-205754, V-205756, V-205758, V-205759, V-205760, V-205761, V-205762, V-205764, V-205765, V-205766, V-205767, V-205768, V-205769, V-205770, V-205771, V-205772, V-205773, V-205774, V-205775, V-205776, V-205777, V-205778, V-205779, V-205780, V-205781, V-205782, V-205783, V-205784, V-205795, V-205796, V-205797, V-205798, V-205801, V-205808, V-205809, V-205810, V-205811, V-205812, V-205813, V-205814, V-205815, V-205816, V-205817, V-205821, V-205822, V-205823, V-205824, V-205825, V-205826, V-205827, V-205828, V-205830, V-205832, V-205833, V-205834, V-205835, V-205836, V-205837, V-205838, V-205839, V-205840, V-205841, V-205842, V-205861, V-205863, V-205865, V-205866, V-205867, V-205868, V-205869, V-205872, V-205873, V-205874, V-205911, V-205912, V-205915, V-205916, V-205917, V-205918, V-205920, V-205921, V-205922, V-205924, V-205925, V-236001, and V-257503

- **Windows Server 2016 STIG Version 2 Release 9**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-224850, V-224852, V-224853, V-224854, V-224855, V-224856, V-224857, V-224858, V-224859, V-224866, V-224867, V-224868, V-224869, V-224870, V-224871, V-224872, V-224873, V-224881, V-224882, V-224883, V-224884, V-224885, V-224886, V-224887, V-224888, V-224889, V-224890, V-224891, V-224892, V-224893, V-224894, V-224895, V-224896, V-224897, V-224898, V-224899, V-224900, V-224901, V-224902, V-224903,

V-224904, V-224905, V-224906, V-224907, V-224908, V-224909, V-224910, V-224911, V-224912, V-224913, V-224914, V-224915, V-224920, V-224922, V-224924, V-224925, V-224926, V-224927, V-224928, V-224929, V-224930, V-224935, V-224936, V-224937, V-224938, V-224939, V-224940, V-224941, V-224943, V-224944, V-224945, V-224946, V-224947, V-224948, V-224949, V-224951, V-224952, V-224953, V-224955, V-224956, V-224957, V-224959, V-224960, V-224962, V-224963, V-225010, V-225013, V-225014, V-225015, V-225016, V-225017, V-225018, V-225019, V-225021, V-225022, V-225023, V-225024, V-225028, V-225029, V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040, V-225041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055, V-225056, V-225057, V-225058, V-225059, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066, V-225067, V-225068, V-225069, V-225072, V-225073, V-225074, V-225076, V-225078, V-225080, V-225081, V-225082, V-225083, V-225084, V-225086, V-225087, V-225088, V-225089, V-225092, V-225093, V-236000, and V-257502

- **Windows Server 2012 R2 MS STIG Version 3 Release 5**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-225574, V-225573, V-225572, V-225571, V-225570, V-225569, V-225568, V-225567, V-225566, V-225565, V-225564, V-225563, V-225562, V-225561, V-225560, V-225559, V-225558, V-225557, V-225555, V-225554, V-225553, V-225551, V-225550, V-225549, V-225548, V-225546, V-225545, V-225544, V-225543, V-225542, V-225541, V-225540, V-225539, V-225538, V-225535, V-225534, V-225533, V-225532, V-225531, V-225530, V-225529, V-225528, V-225527, V-225524, V-225523, V-225522, V-225521, V-225520, V-225519, V-225518, V-225517, V-225516, V-225515, V-225513, V-225510, V-225509, V-225508, V-225506, V-225504, V-225503, V-225502, V-225501, V-225500, V-225494, V-225486, V-225478, V-225477, V-225475, V-225474, V-225472, V-225471, V-225470, V-225469, V-225464, V-225463, V-225461, V-225458, V-225457, V-225456, V-225455, V-225454, V-225453, V-225452, V-225448, V-225443, V-225442, V-225441, V-225415, V-225414, V-225413, V-225411, V-225410, V-225409, V-225408, V-225407, V-225406, V-225405, V-225404, V-225402, V-225401, V-225400, V-225398, V-225397, V-225395, V-225393, V-225391, V-225389, V-225386, V-225385, V-225384, V-225383, V-225382, V-225381, V-225380, V-225379, V-225378, V-225377, V-225375, V-225374, V-225373, V-225372, V-225371, V-225370, V-225369, V-225368, V-225367, V-225356, V-225353, V-225352, V-225351, V-225350, V-225349, V-225348, V-225347, V-225346, V-225345, V-225344, V-225341, V-225340, V-225339, V-225338, V-225337, V-225329, V-225326,

V-225325, V-225317, V-225316, V-225315, V-225314, V-225305, V-225304, V-225303, V-225302, V-225301, V-225300, V-225299, V-225298, V-225297, V-225296, V-225295, V-225294, V-225293, V-225292, V-225291, V-225290, V-225289, V-225288, V-225287, V-225286, V-225285, V-225284, V-225283, V-225282, V-225281, V-225280, V-225279, V-225278, V-225277, V-225276, V-225275, V-225273, V-225272, V-225271, V-225270, V-225269, V-225268, V-225267, V-225266, V-225265, V-225264, V-225263, V-225261, V-225260, V-225259, and V-225239

- **Microsoft .NET Framework STIG 4.0 Version 2 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-225238

- **Windows Firewall STIG Version 2 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-241989, V-241990, V-241991, V-241993, V-241993, V-241998, V-241998, V-242003, and V-242003

- **Internet Explorer 11 STIG Version 2 Release 5**

  Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

  V-46473, V-46475, V-46481, V-46483, V-46501, V-46507, V-46509, V-46511, V-46513, V-46515, V-46517, V-46521, V-46523, V-46525, V-46543, V-46545, V-46547, V-46549, V-46553, V-46555, V-46573, V-46575, V-46577, V-46579, V-46581, V-46583, V-46587, V-46589, V-46591, V-46593, V-46597, V-46599, V-46601, V-46603, V-46605, V-46607, V-46609, V-46615, V-46617, V-46619, V-46621, V-46625, V-46633, V-46635, V-46637, V-46639, V-46641, V-46643, V-46645, V-46647, V-46649, V-46653, V-46663, V-46665, V-46669, V-46681, V-46685, V-46689, V-46691, V-46693, V-46695, V-46701, V-46705, V-46709, V-46711, V-46713, V-46715, V-46717, V-46719, V-46721, V-46723, V-46725, V-46727, V-46729, V-46731, V-46733, V-46779, V-46781, V-46787, V-46789, V-46791, V-46797, V-46799, V-46801, V-46807, V-46811, V-46815, V-46819, V-46829, V-46841, V-46847, V-46849, V-46853, V-46857, V-46859, V-46861, V-46865, V-46869, V-46879, V-46883, V-46885, V-46889, V-46893, V-46895, V-46897, V-46903, V-46907, V-46921, V-46927, V-46939, V-46975, V-46981, V-46987, V-46995, V-46997, V-46999, V-47003,

V-47005, V-47009, V-64711, V-64713, V-64715, V-64717, V-64719, V-64721, V-64723, V-64725, V-64729, V-72757, V-72759, V-72761, V-72763, V-75169, and V-75171

- **Microsoft Edge STIG Version 2 Release 2 (Windows Server 2022 only)**

  V-235720, V-235721, V-235723, V-235724, V-235725, V-235726, V-235728, V-235729, V-235730, V-235732, V-235733, V-235734, V-235735, V-235736, V-235737, V-235738, V-235739, V-235740, V-235741, V-235742, V-235743, V-235744, V-235745, V-235746, V-235747, V-235748, V-235749, V-235750, V-235754, V-235756, V-235760, V-235761, V-235763, V-235764, V-235766, V-235767, V-235768, V-235769, V-235770, V-235771, V-235772, V-235773, V-235774, and V-246736

- **Defender STIG Version 2 Release 4**

  V-213427, V-213429, V-213430, V-213431, V-213432, V-213433, V-213434, V-213435, V-213436, V-213437, V-213438, V-213439, V-213440, V-213441, V-213442, V-213443, V-213444, V-213445, V-213446, V-213447, V-213448, V-213449, V-213450, V-213451, V-213455, V-213464, V-213465, and V-213466

Windows STIG High (Category I)

The following list contains STIG settings that Amazon EC2 supports to your infrastructure. If a supported setting isn't applicable for your infrastructure, Amazon EC2 skips that setting, and moves on. For example, some STIG hardening settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list of Windows STIGs, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

> **Note**
>
> The Windows STIG High category includes all of the listed STIG hardening settings that apply for Windows STIG Medium and Low categories, in addition to the STIG hardening settings that Amazon EC2 supports for Category I vulnerabilities.

- **Windows Server 2022 STIG Version 2 Release 2**

V-254293, V-254352, V-254353, V-254354, V-254374, V-254378, V-254381, V-254446, V-254465, V-254466, V-254467, V-254469, V-254474, V-254475, and V-254500

- **Windows Server 2019 STIG Version 3 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

  V-205653, V-205654, V-205711, V-205713, V-205724, V-205725, V-205757, V-205802, V-205804, V-205805, V-205806, V-205849, V-205908, V-205913, V-205914, and V-205919

- **Windows Server 2016 STIG Version 2 Release 9**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

  V-224874, V-224932, V-224933, V-224934, V-224954, V-224958, V-224961, V-225025, V-225044, V-225045, V-225046, V-225048, V-225053, V-225054, and V-225079

- **Windows Server 2012 R2 MS STIG Version 3 Release 5**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

  V-225556, V-225552, V-225547, V-225507, V-225505, V-225498, V-225497, V-225496, V-225493, V-225492, V-225491, V-225449, V-225444, V-225399, V-225396, V-225390, V-225366, V-225365, V-225364, V-225354, and V-225274

- **Microsoft .NET Framework STIG 4.0 Version 2 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities for the Microsoft .NET Framework. No additional STIG settings apply for Category I vulnerabilities.

- **Windows Firewall STIG Version 2 Release 2**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

  V-241992, V-241997, and V-242002

- **Internet Explorer 11 STIG Version 2 Release 5**

Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities for Internet Explorer 11. No additional STIG settings apply for Category I vulnerabilities.

- **Microsoft Edge STIG Version 2 Release 2 (Windows Server 2022 only)**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

  V-235758 and V-235759

- **Defender STIG Version 2 Release 4**

  Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

  V-213426, V-213426, V-213452, V-213452, V-213452, V-213453, V-213453, and V-213453

Windows STIG version history

This section logs Windows component version history for the quarterly STIG updates. To see the changes and published versions for a quarter, choose the title to expand the information.

2024 Q4 changes - 12/10/2024:

Updated STIG versions and applied STIGS for the 2024 Q4 release as follows:

STIG-Build-Windows-Low version 2024.4.0

- Windows Server 2022 STIG Version 2 Release 2
- Windows Server 2019 STIG Version 3 Release 2
- Windows Server 2016 STIG Version 2 Release 9
- Windows Server 2012 R2 MS STIG Version 3 Release 5
- Microsoft .NET Framework 4.0 STIG Version 2 Release 2
- Windows Firewall STIG Version 2 Release 2
- Internet Explorer 11 STIG Version 2 Release 5
- Microsoft Edge STIG Version 2 Release 2 (Windows Server 2022 only)

STIG-Build-Windows-Medium version 2024.4.0

- Windows Server 2022 STIG Version 2 Release 2
- Windows Server 2019 STIG Version 3 Release 2
- Windows Server 2016 STIG Version 2 Release 9
- Windows Server 2012 R2 MS STIG Version 3 Release 5
- Microsoft .NET Framework 4.0 STIG Version 2 Release 2
- Windows Firewall STIG Version 2 Release 2
- Internet Explorer 11 STIG Version 2 Release 5
- Microsoft Edge STIG Version 2 Release 2 (Windows Server 2022 only)
- Defender STIG Version 2 Release 4

STIG-Build-Windows-High version 2024.4.0

- Windows Server 2022 STIG Version 2 Release 2
- Windows Server 2019 STIG Version 3 Release 2
- Windows Server 2016 STIG Version 2 Release 9
- Windows Server 2012 R2 MS STIG Version 3 Release 5
- Microsoft .NET Framework 4.0 STIG Version 2 Release 2
- Windows Firewall STIG Version 2 Release 2
- Internet Explorer 11 STIG Version 2 Release 5
- Microsoft Edge STIG Version 2 Release 2 (Windows Server 2022 only)
- Defender STIG Version 2 Release 4

2024 Q3 changes - 10/04/2024 (no changes):

There were no changes for Windows component STIGS for the 2024 third quarter release.

2024 Q2 changes - 05/10/2024 (no changes):

There were no changes for Windows component STIGS for the 2024 second quarter release.

2024 Q1 changes - 02/23/2024 (no changes):

There were no changes for Windows component STIGS for the 2024 first quarter release.

2023 Q4 changes - 12/07/2023 (no changes):

There were no changes for Windows component STIGS for the 2023 fourth quarter release.

2023 Q3 changes - 10/04/2023 (no changes):

There were no changes for Windows component STIGS for the 2023 third quarter release.

2023 Q2 changes - 05/03/2023 (no changes):

There were no changes for Windows component STIGS for the 2023 second quarter release.

2023 Q1 changes - 03/27/2023 (no changes):

There were no changes for Windows component STIGS for the 2023 first quarter release.

2022 Q4 changes - 02/01/2023:

Updated STIG versions and applied STIGS for the 2022 Q4 release as follows:

STIG-Build-Windows-Low version 2022.4.0

- Windows Server 2022 STIG Version 1 Release 1
- Windows Server 2019 STIG Version 2 Release 5
- Windows Server 2016 STIG Version 2 Release 5
- Windows Server 2012 R2 MS STIG Version 3 Release 5
- Microsoft .NET Framework 4.0 STIG Version 2 Release 2
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 2 Release 3
- Microsoft Edge STIG Version 1 Release 6 (Windows Server 2022 only)

STIG-Build-Windows-Medium version 2022.4.0

- Windows Server 2022 STIG Version 1 Release 1
- Windows Server 2019 STIG Version 2 Release 5
- Windows Server 2016 STIG Version 2 Release 5
- Windows Server 2012 R2 MS STIG Version 3 Release 5

- Microsoft .NET Framework 4.0 STIG Version 2 Release 2
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 2 Release 3
- Microsoft Edge STIG Version 1 Release 6 (Windows Server 2022 only)
- Defender STIG Version 2 Release 4 (Windows Server 2022 only)

STIG-Build-Windows-High version 2022.4.0

- Windows Server 2022 STIG Version 1 Release 1
- Windows Server 2019 STIG Version 2 Release 5
- Windows Server 2016 STIG Version 2 Release 5
- Windows Server 2012 R2 MS STIG Version 3 Release 5
- Microsoft .NET Framework 4.0 STIG Version 2 Release 2
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 2 Release 3
- Microsoft Edge STIG Version 1 Release 6 (Windows Server 2022 only)
- Defender STIG Version 2 Release 4 (Windows Server 2022 only)

2022 Q3 changes - 09/30/2022 (no changes):

There were no changes for Windows component STIGS for the 2022 third quarter release.

2022 Q2 changes - 08/02/2022:

Updated STIG versions and applied STIGS for the 2022 Q2 release.

STIG-Build-Windows-Low version 1.5.0

- Windows Server 2019 STIG Version 2 Release 4
- Windows Server 2016 STIG Version 2 Release 4
- Windows Server 2012 R2 MS STIG Version 3 Release 3
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 2 Release 1

- Internet Explorer 11 STIG Version 1 Release 19

STIG-Build-Windows-Medium version 1.5.0

- Windows Server 2019 STIG Version 2 Release 4
- Windows Server 2016 STIG Version 2 Release 4
- Windows Server 2012 R2 MS STIG Version 3 Release 3
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 1 Release 19

STIG-Build-Windows-High version 1.5.0

- Windows Server 2019 STIG Version 2 Release 4
- Windows Server 2016 STIG Version 2 Release 4
- Windows Server 2012 R2 MS STIG Version 3 Release 3
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 1 Release 19

2022 Q1 changes - 08/02/2022 (no changes):

There were no changes for Windows component STIGS for the 2022 first quarter release.

2021 Q4 changes - 12/20/2021:

Updated STIG versions and applied STIGS for the 2021 fourth quarter release.

STIG-Build-Windows-Low version 1.5.0

- Windows Server 2019 STIG Version 2 Release 3
- Windows Server 2016 STIG Version 2 Release 3
- Windows Server 2012 R2 MS STIG Version 3 Release 3
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1

- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 1 Release 19

STIG-Build-Windows-Medium version 1.5.0

- Windows Server 2019 STIG Version 2 Release 3
- Windows Server 2016 STIG Version 2 Release 3
- Windows Server 2012 R2 MS STIG Version 3 Release 3
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 1 Release 19

STIG-Build-Windows-High version 1.5.0

- Windows Server 2019 STIG Version 2 Release 3
- Windows Server 2016 STIG Version 2 Release 3
- Windows Server 2012 R2 MS STIG Version 3 Release 3
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 2 Release 1
- Internet Explorer 11 STIG Version 1 Release 19

2021 Q3 changes - 09/30/2021:

Updated STIG versions and applied STIGS for the 2021 third quarter release.

STIG-Build-Windows-Low version 1.4.0

- Windows Server 2019 STIG Version 2 Release 2
- Windows Server 2016 STIG Version 2 Release 2
- Windows Server 2012 R2 MS STIG Version 3 Release 2
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 1 Release 7
- Internet Explorer 11 STIG Version 1 Release 19

STIG-Build-Windows-Medium version 1.4.0

- Windows Server 2019 STIG Version 2 Release 2
- Windows Server 2016 STIG Version 2 Release 2
- Windows Server 2012 R2 MS STIG Version 3 Release 2
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 1 Release 7
- Internet Explorer 11 STIG Version 1 Release 19

STIG-Build-Windows-High version 1.4.0

- Windows Server 2019 STIG Version 2 Release 2
- Windows Server 2016 STIG Version 2 Release 2
- Windows Server 2012 R2 MS STIG Version 3 Release 2
- Microsoft .NET Framework 4.0 STIG Version 2 Release 1
- Windows Firewall STIG Version 1 Release 7
- Internet Explorer 11 STIG Version 1 Release 19

Linux STIG settings

This section contains information about the Linux STIG hardening settings that Amazon EC2 supports, followed by a version history log. If the Linux distribution doesn't have STIG hardening settings of its own, Amazon EC2 uses RHEL settings. Supported STIG hardening settings apply to Amazon EC2 Linux AMIs and components based on the Linux distribution, as follows:

- Red Hat Enterprise Linux (RHEL) 7 STIG settings
  - RHEL 7
  - CentOS 7
  - Amazon Linux 2 (AL2)
- RHEL 8 STIG settings
  - RHEL 8
  - CentOS 8
  - Amazon Linux 2023 (AL 2023)
- RHEL 9 STIG settings

- RHEL 9
- CentOS Stream 9

Linux STIG Low (Category III)

The following list contains STIG settings that Amazon EC2 supports to your infrastructure. If a supported setting isn't applicable for your infrastructure, Amazon EC2 skips that setting, and moves on. For example, some STIG hardening settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

RHEL 7 STIG Version 3 Release 14

- **RHEL 7/CentOS 7/AL2**

  V-204452, V-204576, and V-204605

RHEL 8 STIG Version 2 Release 1

- **RHEL 8/CentOS 8/AL 2023**

  V-230241, V-230269, V-230270, V-230281, V-230285, V-230346, V-230381, V-230395, V-230468, V-230469, V-230485, V-230486, V-230491, V-230494, V-230495, V-230496, V-230497, V-230498, V-230499, and V-244527

RHEL 9 STIG Version 2 Release 2

- **RHEL 9/CentOS Stream 9**

  V-257782, V-257795, V-257796, V-257824, V-257880, V-257946, V-257947, V-258037, V-258067, V-258069, V-258076, V-258138, and V-258173

Ubuntu 18.04 STIG Version 2 Release 15

V-219163, V-219164, V-219165, V-219172, V-219173, V-219174, V-219175, V-219178, V-219180, V-219210, V-219301, V-219327, V-219332, and V-219333

Ubuntu 20.04 STIG Version 2 Release 1

V-238202, V-238221, V-238222, V-238223, V-238224, V-238226, V-238234, V-238235, V-238237, V-238308, V-238323, V-238357, V-238362, and V-238373

Ubuntu 22.04 STIG Version 2 Release 2

V-260472, V-260476, V-260479, V-260480, V-260481, V-260520, V-260521, V-260549, V-260550, V-260551, V-260552, V-260581, and V-260596

Linux STIG Medium (Category II)

The following list contains STIG settings that Amazon EC2 supports to your infrastructure. If a supported setting isn't applicable for your infrastructure, Amazon EC2 skips that setting, and moves on. For example, some STIG hardening settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

> **Note**
>
> The Linux STIG Medium category includes all of the listed STIG hardening settings that apply for Linux STIG Low (Category III), in addition to the STIG hardening settings that Amazon EC2 supports for Category II vulnerabilities.

RHEL 7 STIG Version 3 Release 15

Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

- **RHEL 7/CentOS 7/AL2**

  V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204416, V-204417, V-204418, V-204422, V-204423, V-204426, V-204427, V-204431, V-204434, V-204435, V-204437, V-204449, V-204450, V-204451, V-204457, V-204466, V-204490, V-204491, V-204503, V-204507, V-204508, V-204510, V-204511, V-204512, V-204514, V-204515, V-204516, V-204517, V-204521,

V-204524, V-204531, V-204536, V-204537, V-204538, V-204539, V-204540, V-204541, V-204542, V-204543, V-204544, V-204545, V-204546, V-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204578, V-204579, V-204584, V-204585, V-204587, V-204588, V-204589, V-204590, V-204591, V-204592, V-204593, V-204595, V-204596, V-204597, V-204598, V-204599, V-204600, V-204601, V-204602, V-204609, V-204610, V-204611, V-204612, V-204613, V-204614, V-204615, V-204616, V-204617, V-204619, V-204622, V-204625, V-204630, V-204631, V-204633, V-233307, V-237634, V-237635, V-251703, V-255925, V-255927, V-255928, and V-25697

RHEL 8 STIG Version 2 Release 1

Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

- **RHEL 8/CentOS 8/AL 2023**

V-230228, V-230231, V-230233, V-230236, V-230237, V-230238, V-230239, V-230240, V-230244, V-230245, V-230246, V-230247, V-230248, V-230249, V-230250, V-230255, V-230257, V-230258, V-230259, V-230262, V-230266, V-230267, V-230268, V-230273, V-230275, V-230277, V-230278, V-230280, V-230282, V-230286, V-230287, V-230288, V-230290, V-230291, V-230296, V-230298, V-230310, V-230311, V-230312, V-230313, V-230314, V-230315, V-230324, V-230330, V-230332, V-230333, V-230335, V-230337, V-230339, V-230341, V-230343, V-230345, V-230348, V-230353, V-230356, V-230357, V-230358, V-230359, V-230360, V-230361, V-230362, V-230363, V-230365, V-230366, V-230368, V-230369, V-230370, V-230373, V-230375, V-230376, V-230377, V-230378, V-230382, V-230383, V-230386, V-230387, V-230390, V-230392, V-230393, V-230394, V-230396, V-230397, V-230398, V-230399, V-230400, V-230401, V-230402, V-230403, V-230404, V-230405, V-230406, V-230407, V-230408, V-230409, V-230410, V-230411, V-230412, V-230413, V-230418, V-230419, V-230421, V-230422, V-230423, V-230424, V-230425, V-230426, V-230427, V-230428, V-230429, V-230430, V-230431, V-230432, V-230433, V-230434, V-230435, V-230436, V-230437, V-230438, V-230439, V-230444, V-230446, V-230447, V-230448, V-230449, V-230455, V-230456, V-230462, V-230463, V-230464, V-230465, V-230466, V-230467, V-230471, V-230472, V-230473, V-230474, V-230478, V-230480, V-230483, V-230488, V-230489, V-230502, V-230503, V-230507, V-230526, V-230527, V-230532, V-230535, V-230536, V-230537, V-230538, V-230539,

V-230540, V-230541, V-230542, V-230543, V-230544, V-230545, V-230546, V-230547, V-230548, V-230549, V-230550, V-230555, V-230556, V-230559, V-230560, V-230561, V-237640, V-237642, V-237643, V-244523, V-244524, V-244525, V-244526, V-244528, V-244533, V-244542, V-244550, V-244551, V-244552, V-244553, V-244554, V-250315, V-250316, V-250317, V-251711, V-251713, V-251714, V-251715, V-251716, V-251717, V-251718, V-256974, and V-257258

RHEL 9 STIG Version 2 Release 2

Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

- **RHEL 9/CentOS Stream 9**

  V-257780, V-257781, V-257783, V-257786, V-257788, V-257790, V-257791, V-257792, V-257793, V-257794, V-257797, V-257798, V-257799, V-257800, V-257801, V-257802, V-257803, V-257804, V-257805, V-257806, V-257807, V-257808, V-257809, V-257810, V-257811, V-257812, V-257813, V-257814, V-257815, V-257816, V-257817, V-257818, V-257825, V-257827, V-257828, V-257829, V-257830, V-257831, V-257832, V-257833, V-257834, V-257836, V-257838, V-257839, V-257840, V-257841, V-257842, V-257849, V-257882, V-257883, V-257884, V-257885, V-257886, V-257887, V-257888, V-257889, V-257890, V-257891, V-257892, V-257893, V-257894, V-257895, V-257896, V-257897, V-257898, V-257899, V-257900, V-257901, V-257902, V-257903, V-257904, V-257905, V-257906, V-257907, V-257908, V-257909, V-257910, V-257911, V-257912, V-257913, V-257914, V-257915, V-257916, V-257917, V-257918, V-257919, V-257920, V-257921, V-257922, V-257923, V-257924, V-257925, V-257926, V-257927, V-257928, V-257929, V-257930, V-257933, V-257934, V-257935, V-257936, V-257939, V-257940, V-257942, V-257943, V-257944, V-257948, V-257949, V-257951, V-257952, V-257953, V-257954, V-257957, V-257958, V-257959, V-257960, V-257961, V-257962, V-257963, V-257964, V-257965, V-257966, V-257967, V-257968, V-257969, V-257970, V-257971, V-257972, V-257973, V-257974, V-257975, V-257976, V-257977, V-257978, V-257979, V-257980, V-257981, V-257982, V-257983, V-257985, V-257987, V-257988, V-257989, V-257991, V-257992, V-257993, V-257994, V-257995, V-257996, V-257997, V-257998, V-257999, V-258000, V-258001, V-258002, V-258003, V-258004, V-258005, V-258006, V-258007, V-258008, V-258009, V-258010, V-258011, V-258028, V-258034, V-258035, V-258036, V-258038, V-258040, V-258041, V-258043, V-258046, V-258049, V-258052, V-258054, V-258055, V-258056, V-258057, V-258060, V-258063, V-258064, V-258065, V-258066,

V-258068, V-258070, V-258071, V-258072, V-258073, V-258074, V-258075, V-258077, V-258079, V-258080, V-258081, V-258082, V-258083, V-258084, V-258085, V-258088, V-258089, V-258091, V-258092, V-258093, V-258095, V-258097, V-258098, V-258099, V-258100, V-258101, V-258102, V-258103, V-258104, V-258105, V-258107, V-258108, V-258109, V-258110, V-258111, V-258112, V-258113, V-258114, V-258115, V-258116, V-258117, V-258118, V-258119, V-258120, V-258122, V-258123, V-258124, V-258125, V-258126, V-258128, V-258129, V-258130, V-258133, V-258134, V-258137, V-258140, V-258141, V-258142, V-258144, V-258145, V-258146, V-258147, V-258148, V-258150, V-258151, V-258152, V-258153, V-258154, V-258156, V-258157, V-258158, V-258159, V-258160, V-258161, V-258162, V-258163, V-258164, V-258165, V-258166, V-258167, V-258168, V-258169, V-258170, V-258171, V-258172, V-258175, V-258176, V-258177, V-258178, V-258179, V-258180, V-258181, V-258182, V-258183, V-258184, V-258185, V-258186, V-258187, V-258188, V-258189, V-258190, V-258191, V-258192, V-258193, V-258194, V-258195, V-258196, V-258197, V-258198, V-258199, V-258200, V-258201, V-258202, V-258203, V-258204, V-258205, V-258206, V-258207, V-258208, V-258209, V-258210, V-258211, V-258212, V-258213, V-258214, V-258215, V-258216, V-258217, V-258218, V-258219, V-258220, V-258221, V-258222, V-258223, V-258224, V-258225, V-258226, V-258227, V-258228, V-258229, V-258232, V-258233, V-258234, V-258237, V-258239, and V-258240

Ubuntu 18.04 STIG Version 2 Release 15

V-219149, V-219155, V-219156, V-219160, V-219166, V-219168, V-219176, V-219181, V-219184, V-219186, V-219188, V-219189, V-219190, V-219191, V-219192, V-219193, V-219194, V-219195, V-219196, V-219197, V-219198, V-219199, V-219200, V-219201, V-219202, V-219203, V-219204, V-219205, V-219206, V-219207, V-219208, V-219209, V-219213, V-219214, V-219215, V-219216, V-219217, V-219218, V-219219, V-219220, V-219221, V-219222, V-219223, V-219224, V-219225, V-219226, V-219227, V-219228, V-219229, V-219230, V-219231, V-219232, V-219233, V-219234, V-219235, V-219236, V-219238, V-219239, V-219240, V-219241, V-219242, V-219243, V-219244, V-219250, V-219254, V-219257, V-219263, V-219264, V-219265, V-219266, V-219267, V-219268, V-219269, V-219270, V-219271, V-219272, V-219273, V-219274, V-219275, V-219276, V-219277, V-219279, V-219281, V-219287, V-219291, V-219297, V-219298, V-219299, V-219300, V-219303, V-219304, V-219306, V-219309, V-219310, V-219311, V-219312, V-219315, V-219318, V-219319, V-219323, V-219326, V-219328, V-219330, V-219331, V-219335, V-219336, V-219337, V-219338, V-219339, V-219342, V-219344, V-233779, V-233780, and V-255906

Ubuntu 20.04 STIG Version 2 Release 1

V-238200, V-238205, V-238207, V-238209, V-238210, V-238211, V-238212, V-238213, V-238216, V-238220, V-238225, V-238227, V-238228, V-238230, V-238231, V-238232, V-238236, V-238238, V-238239, V-238240, V-238241, V-238242, V-238244, V-238245, V-238246, V-238247, V-238248, V-238249, V-238250, V-238251, V-238252, V-238253, V-238254, V-238255, V-238256, V-238257, V-238258, V-238264, V-238268, V-238271, V-238277, V-238278, V-238279, V-238280, V-238281, V-238282, V-238283, V-238284, V-238285, V-238286, V-238287, V-238288, V-238289, V-238290, V-238291, V-238292, V-238293, V-238294, V-238295, V-238297, V-238299, V-238300, V-238301, V-238302, V-238303, V-238304, V-238309, V-238310, V-238315, V-238316, V-238317, V-238318, V-238319, V-238320, V-238324, V-238325, V-238329, V-238330, V-238333, V-238334, V-238337, V-238338, V-238339, V-238340, V-238341, V-238342, V-238343, V-238344, V-238345, V-238346, V-238347, V-238348, V-238349, V-238350, V-238351, V-238352, V-238353, V-238355, V-238356, V-238359, V-238360, V-238369, V-238370, V-238371, V-238376, V-238377, V-238378, V-251505, and V-255912

Ubuntu 22.04 STIG Version 2 Release 2

Includes all STIG hardening settings that Amazon EC2 supports for Category III (Low) vulnerabilities, plus:

V-260471, V-260473, V-260474, V-260475, V-260477, V-260478, V-260485, V-260486, V-260487, V-260488, V-260489, V-260490, V-260491, V-260492, V-260493, V-260494, V-260495, V-260496, V-260497, V-260498, V-260499, V-260500, V-260505, V-260506, V-260507, V-260508, V-260509, V-260510, V-260511, V-260512, V-260513, V-260514, V-260522, V-260527, V-260528, V-260530, V-260531, V-260532, V-260533, V-260534, V-260535, V-260537, V-260538, V-260540, V-260542, V-260543, V-260545, V-260546, V-260547, V-260553, V-260554, V-260555, V-260556, V-260557, V-260560, V-260561, V-260562, V-260563, V-260564, V-260565, V-260566, V-260567, V-260569, V-260572, V-260573, V-260574, V-260576, V-260582, V-260584, V-260585, V-260586, V-260588, V-260589, V-260590, V-260591, V-260594, V-260597, V-260598, V-260599, V-260600, V-260601, V-260602, V-260603, V-260604, V-260605, V-260606, V-260607, V-260608, V-260609, V-260610, V-260611, V-260612, V-260613, V-260614, V-260615, V-260616, V-260617, V-260618, V-260619, V-260620, V-260621, V-260622, V-260623, V-260624, V-260625, V-260626, V-260627, V-260628, V-260629, V-260630, V-260631, V-260632, V-260633, V-260634, V-260635, V-260636, V-260637, V-260638, V-260639, V-260640,

V-260641, V-260642, V-260643, V-260644, V-260645, V-260646, V-260647, V-260648, and V-260649

Linux STIG High (Category I)

The following list contains STIG settings that Amazon EC2 supports to your infrastructure. If a supported setting isn't applicable for your infrastructure, Amazon EC2 skips that setting, and moves on. For example, some STIG hardening settings might not apply to standalone servers. Organization-specific policies can also affect which settings apply, such as a requirement for administrators to review document settings.

For a complete list, see the STIGs Document Library. For information about how to view the complete list, see STIG Viewing Tools.

> **Note**
>
> The Linux STIG High category includes all of the listed STIG hardening settings that apply for Linux STIG Medium and Low categories, in addition to the STIG hardening settings that Amazon EC2 supports for Category I vulnerabilities.

RHEL 7 STIG Version 3 Release 15

Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

- **RHEL 7/CentOS 7/AL2**

  V-204424, V-204425, V-204442, V-204443, V-204447, V-204448, V-204455, V-204497, V-204502, V-204594, V-204620, and V-204621

RHEL 8 STIG Version 2 Release 1

Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

- **RHEL 8/CentOS 8/AL 2023**

  V-230223, V-230264, V-230265, V-230487, V-230492, V-230529, V-230531, V-230533, V-230558, and V-244540

RHEL 9 STIG Version 2 Release 2

Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

- **RHEL 9/CentOS Stream 9**

  V-257784, V-257785, V-257820, V-257821, V-257826, V-257835, V-257955, V-257956, V-257984, V-257986, V-258059, V-258078, V-258094, and V-258235

Ubuntu 18.04 STIG Version 2 Release 15

V-219157, V-219158, V-219177, V-219212, V-219308, V-219314, V-219316, V-251507, and V-264388

Ubuntu 20.04 STIG Version 2 Release 1

V-238201, V-238218, V-238219, V-238326, V-238327, V-238380, and V-251504

Ubuntu 22.04 STIG Version 2 Release 2

Includes all STIG hardening settings that Amazon EC2 supports for Categories II and III (Medium and Low) vulnerabilities, plus:

V-260469, V-260482, V-260483, V-260523, V-260524, V-260526, V-260529, V-260539, V-260570, V-260571, and V-260579

Linux STIG version history

This section logs Linux component version history for the quarterly STIG updates. To see the changes and published versions for a quarter, choose the title to expand the information.

2024 Q4 changes - 12/10/2024:

Updated the following STIG versions, applied STIGS for the 2024 fourth quarter release, and added information about two new input parameters for the Linux components:

STIG-Build-Linux-Low version 2024.4.x

- RHEL 7 STIG Version 3 Release 15
- RHEL 8 STIG Version 2 Release 1

- RHEL 9 STIG Version 2 Release 2
- Ubuntu 18.04 STIG Version 2 Release 15
- Ubuntu 20.04 STIG Version 2 Release 1
- Ubuntu 22.04 STIG Version 2 Release 2

STIG-Build-Linux-Medium version 2024.4.x

- RHEL 7 STIG Version 3 Release 15
- RHEL 8 STIG Version 2 Release 1
- RHEL 9 STIG Version 2 Release 2
- Ubuntu 18.04 STIG Version 2 Release 15
- Ubuntu 20.04 STIG Version 2 Release 1
- Ubuntu 22.04 STIG Version 2 Release 2

STIG-Build-Linux-High version 2024.4.x

- RHEL 7 STIG Version 3 Release 15
- RHEL 8 STIG Version 2 Release 1
- RHEL 9 STIG Version 2 Release 2
- Ubuntu 18.04 STIG Version 2 Release 15
- Ubuntu 20.04 STIG Version 2 Release 1
- Ubuntu 22.04 STIG Version 2 Release 2

2024 Q3 changes - 10/04/2024 (no changes):

There were no changes for Linux component STIGS for the 2024 third quarter release.

2024 Q2 changes - 05/10/2024:

Updated STIG versions and applied STIGS for the 2024 second quarter release. Also added support for RHEL 9, CentOS Stream 9, and Ubuntu 22.04, as follows:

STIG-Build-Linux-Low version 2024.2.x

- RHEL 7 STIG Version 3 Release 14

- RHEL 8 STIG Version 1 Release 14
- RHEL 9 STIG Version 1 Release 3
- Ubuntu 18.04 STIG Version 2 Release 14
- Ubuntu 20.04 STIG Version 1 Release 12
- Ubuntu 22.04 STIG Version 1 Release 1

STIG-Build-Linux-Medium version 2024.2.x

- RHEL 7 STIG Version 3 Release 14
- RHEL 8 STIG Version 1 Release 14
- RHEL 9 STIG Version 1 Release 3
- Ubuntu 18.04 STIG Version 2 Release 14
- Ubuntu 20.04 STIG Version 1 Release 12
- Ubuntu 22.04 STIG Version 1 Release 1

STIG-Build-Linux-High version 2024.2.x

- RHEL 7 STIG Version 3 Release 14
- RHEL 8 STIG Version 1 Release 14
- RHEL 9 STIG Version 1 Release 3
- Ubuntu 18.04 STIG Version 2 Release 14
- Ubuntu 20.04 STIG Version 1 Release 12
- Ubuntu 22.04 STIG Version 1 Release 1

2024 Q1 changes - 02/06/2024:

Updated STIG versions and applied STIGS for the 2024 first quarter release as follows:

STIG-Build-Linux-Low version 2024.1.x

- RHEL 7 STIG Version 3 Release 14
- RHEL 8 STIG Version 1 Release 13
- Ubuntu 18.04 STIG Version 2 Release 13
- Ubuntu 20.04 STIG Version 1 Release 11

STIG-Build-Linux-Medium version 2024.1.x

- RHEL 7 STIG Version 3 Release 14
- RHEL 8 STIG Version 1 Release 13
- Ubuntu 18.04 STIG Version 2 Release 13
- Ubuntu 20.04 STIG Version 1 Release 11

STIG-Build-Linux-High version 2024.1.x

- RHEL 7 STIG Version 3 Release 14
- RHEL 8 STIG Version 1 Release 13
- Ubuntu 18.04 STIG Version 2 Release 13
- Ubuntu 20.04 STIG Version 1 Release 11

2023 Q4 changes - 12/07/2023:

Updated STIG versions and applied STIGS for the 2023 fourth quarter release as follows:

STIG-Build-Linux-Low version 2023.4.x

- RHEL 7 STIG Version 3 Release 13
- RHEL 8 STIG Version 1 Release 12
- Ubuntu 18.04 STIG Version 2 Release 12
- Ubuntu 20.04 STIG Version 1 Release 10

STIG-Build-Linux-Medium version 2023.4.x

- RHEL 7 STIG Version 3 Release 13
- RHEL 8 STIG Version 1 Release 12
- Ubuntu 18.04 STIG Version 2 Release 12
- Ubuntu 20.04 STIG Version 1 Release 10

STIG-Build-Linux-High version 2023.4.x

- RHEL 7 STIG Version 3 Release 13

- RHEL 8 STIG Version 1 Release 12
- Ubuntu 18.04 STIG Version 2 Release 12
- Ubuntu 20.04 STIG Version 1 Release 10

2023 Q3 changes - 10/04/2023:

Updated STIG versions and applied STIGS for the 2023 third quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 12
- RHEL 8 STIG Version 1 Release 11
- Ubuntu 18.04 STIG Version 2 Release 11
- Ubuntu 20.04 STIG Version 1 Release 9

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 12
- RHEL 8 STIG Version 1 Release 11
- Ubuntu 18.04 STIG Version 2 Release 11
- Ubuntu 20.04 STIG Version 1 Release 9

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 12
- RHEL 8 STIG Version 1 Release 11
- Ubuntu 18.04 STIG Version 2 Release 11
- Ubuntu 20.04 STIG Version 1 Release 9

2023 Q2 changes - 05/03/2023:

Updated STIG versions and applied STIGS for the 2023 second quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 11

- RHEL 8 STIG Version 1 Release 10
- Ubuntu 18.04 STIG Version 2 Release 11
- Ubuntu 20.04 STIG Version 1 Release 8

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 11
- RHEL 8 STIG Version 1 Release 10
- Ubuntu 18.04 STIG Version 2 Release 11
- Ubuntu 20.04 STIG Version 1 Release 8

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 11
- RHEL 8 STIG Version 1 Release 10
- Ubuntu 18.04 STIG Version 2 Release 11
- Ubuntu 20.04 STIG Version 1 Release 8

2023 Q1 changes - 03/27/2023:

Updated STIG versions and applied STIGS for the 2023 first quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 10
- RHEL 8 STIG Version 1 Release 9
- Ubuntu 18.04 STIG Version 2 Release 10
- Ubuntu 20.04 STIG Version 1 Release 7

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 10
- RHEL 8 STIG Version 1 Release 9
- Ubuntu 18.04 STIG Version 2 Release 10
- Ubuntu 20.04 STIG Version 1 Release 7

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 10
- RHEL 8 STIG Version 1 Release 9
- Ubuntu 18.04 STIG Version 2 Release 10
- Ubuntu 20.04 STIG Version 1 Release 7

2022 Q4 changes - 02/01/2023:

Updated STIG versions and applied STIGS for the 2022 fourth quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 9
- RHEL 8 STIG Version 1 Release 8
- Ubuntu 18.04 STIG Version 2 Release 9
- Ubuntu 20.04 STIG Version 1 Release 6

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 9
- RHEL 8 STIG Version 1 Release 8
- Ubuntu 18.04 STIG Version 2 Release 9
- Ubuntu 20.04 STIG Version 1 Release 6

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 9
- RHEL 8 STIG Version 1 Release 8
- Ubuntu 18.04 STIG Version 2 Release 9
- Ubuntu 20.04 STIG Version 1 Release 6

2022 Q3 changes - 09/30/2022 (no changes):

There were no changes for Linux component STIGS for the 2022 third quarter release.

2022 Q2 changes - 08/02/2022:

Introduced Ubuntu support, updated STIG versions and applied STIGS for the 2022 second quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 7
- RHEL 8 STIG Version 1 Release 6
- Ubuntu 18.04 STIG Version 2 Release 6 (new)
- Ubuntu 20.04 STIG Version 1 Release 4 (new)

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 7
- RHEL 8 STIG Version 1 Release 6
- Ubuntu 18.04 STIG Version 2 Release 6 (new)
- Ubuntu 20.04 STIG Version 1 Release 4 (new)

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 7
- RHEL 8 STIG Version 1 Release 6
- Ubuntu 18.04 STIG Version 2 Release 6 (new)
- Ubuntu 20.04 STIG Version 1 Release 4 (new)

2022 Q1 changes - 04/26/2022:

Refactored to include better support for containers. Combined the previous AL2 script with RHEL 7. Updated STIG versions and applied STIGS for the 2022 first quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 6
- RHEL 8 STIG Version 1 Release 5

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 6
- RHEL 8 STIG Version 1 Release 5

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 6
- RHEL 8 STIG Version 1 Release 5

2021 Q4 changes - 12/20/2021:

Updated STIG versions, and applied STIGS for the 2021 fourth quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 5
- RHEL 8 STIG Version 1 Release 4

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 5
- RHEL 8 STIG Version 1 Release 4

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 5
- RHEL 8 STIG Version 1 Release 4

2021 Q3 changes - 09/30/2021:

Updated STIG versions, and applied STIGS for the 2021 third quarter release as follows:

Linux STIG Low (Category III)

- RHEL 7 STIG Version 3 Release 4
- RHEL 8 STIG Version 1 Release 3

Linux STIG Medium (Category II)

- RHEL 7 STIG Version 3 Release 4
- RHEL 8 STIG Version 1 Release 3

Linux STIG High (Category I)

- RHEL 7 STIG Version 3 Release 4
- RHEL 8 STIG Version 1 Release 3

AWSEC2-PatchLoadBalancerInstance

Description

Upgrade and patch minor version of an Amazon EC2 instance (Windows or Linux) attached to any load balancer (classic, ALB, or NLB). The default connection draining time is applied before the instance is patched. You can override the wait time by entering your custom draining time in minutes (1-59) for the **ConnectionDrainTime** parameter.

The automation workflow is as follows:

1. The load balancer or target group to which the instance is attached is determined, and the instance is verified as healthy.
2. The instance is removed from the load balancer or target group.
3. The automation waits for the period of time specified for the connection draining time.
4. The AWS-RunPatchBaseline automation is called to patch the instance.
5. The instance is reattached to the load balancer or target group.

Run this Automation (console)

Document Type

Automation

Owner

Amazon

Prerequisites

- Verify that SSM Agent is installed on your instance. For more information, see Working with SSM Agent on EC2 instances for Windows Server.

Parameters

- **InstanceId**

  Type: String

  Description: (Required) ID of the instance to patch that is associated with a load balancer (classic, ALB, or NLB).
- **ConnectionDrainTime**

  Type: String

  Description: (Optional) The connection draining time of the load balancer, in minutes (1-59).
- **S3BucketLog**

  Type: String

  Description: (Optional) The name of the Amazon S3 bucket to use to store the command output responses. You can specify a bucket that you own or a bucket that is shared with you. If you provide this parameter, you must also provide **runCommandAssumeRole**.
- **runCommandAssumeRole**

  Type: String

  Description: (Optional) The ARN of the IAM role to use to run the command on the instance. The role must have a trust relationship with the `ssm.amazonaws.com` service principal, it must have the **AmazonSSMManagedInstanceCore** policy attached, and it must have write permissions for the Amazon S3 bucket specified for **S3BucketLog**.

AWSEC2-SQLServerDBRestore

Description

The `AWSEC2-SQLServerDBRestore` runbook restores Microsoft SQL Server database backups stored in Amazon S3 to SQL Server 2017 running on an Amazon Elastic Compute Cloud (EC2) Linux instance. You may provide your own EC2 instance running SQL Server 2017 Linux. If an

EC2 instance is not provided, the automation launches and configures a new Ubuntu 16.04 EC2 instance with SQL Server 2017. The automation supports restoring full, differential, and transactional log backups. This automation accepts multiple database backup files and automatically restores the most recent valid backup of each database in the files provided.

To automate both backup and restore of an on-premises SQL Server database to an EC2 instance running SQL Server 2017 Linux, you can use the Amazon-signed PowerShell script `MigrateSQLServerToEC2Linux`.

> **Important**
>
> This runbook resets the SQL Server server administrator (SA) user password every time the automation runs. After the automation is complete, you must set your own SA user password again before you connect to the SQL Server instance.

Run this Automation (console)

Document Type

Automation

Owner

Amazon

Platforms

Linux

Prerequisites

To run this automation, you must meet the following prerequisites:

- The IAM user or role that runs this automation must have an inline policy attached with the permissions outlined in Required IAM permissions.
- If you provide your own EC2 instance:
  - The EC2 instance that you provide must be a Linux instance running Microsoft SQL Server 2017.

- The EC2 instance that you provide must be configured with an Amazon Identity and Access Management (IAM) instance profile that has the `AmazonSSMManagedInstanceCore` managed policy attached. For more information, see Create an IAM instance profile for Systems Manager.
- The SSM Agent must be installed on your EC2 instance. For more information, see Installing and configuring SSM Agent on EC2 instances for Linux.
- The EC2 instance must have enough free disk space to download and restore the SQL Server backups.

Limitations

This automation does not support restoring to SQL Server running on EC2 instances for Windows Server. This automation only restores database backups that are compatible with SQL Server Linux 2017. For more information, see Editions and Supported Features of SQL Server 2017 on Linux.

Parameters

This automation has the following parameters:

- **DatabaseNames**

  Type: String

  Description: (Optional) Comma-separated list of the names of databases to restore.
- **DataDirectorySize**

  Type: String

  Description: (Optional) Desired volume size (GiB) of the SQL Server Data directory for the new EC2 instance.

  Default value: 100
- **KeyPair**

  Type: String

  Description: (Optional) Key pair to use when creating the new EC2 instance.
- **IamInstanceProfileName**

Type: String

Description: (Optional) The IAM instance profile to attach to the new EC2 instance. The IAM instance profile must have the `AmazonSSMManagedInstanceCore` managed policy attached.

- **InstanceId**

  Type: String

  Description: (Optional) The instance running SQL Server 2017 on Linux. If no InstanceId is provided, the automation launches a new EC2 instance using the InstanceType and SQLServerEdition provided.

- **InstanceType**

  Type: String

  Description: (Optional) The instance type of the EC2 instance to be launched.

- **IsS3PresignedUrl**

  Type: String

  Description: (Optional) If S3Input is a pre-signed S3 URL, indicate `yes`.

  Default value: no

  Valid values: yes | no

- **LogDirectorySize**

  Type: String

  Description: (Optional) Desired volume size (GiB) of the SQL Server Log directory for the new EC2 instance.

  Default value: 100

- **S3Input**

  Type: String

  Description: (Required) S3 bucket name, comma-separated list of S3 object keys, or comma-separated list of pre-signed S3 URLs containing the SQL backup files to be restored.

- **SQLServerEdition**

  Type: String

  Description: (Optional) The edition of SQL Server 2017 to be installed on the newly created EC2 instance.

  Valid values: Standard | Enterprise | Web | Express
- **SubnetId**

  Type: String

  Description: (Optional) The subnet in which to launch the new EC2 instance. The subnet must have outbound connectivity to Amazon services. If a value for SubnetId is not provided, the automation uses the default subnet.
- **TempDbDirectorySize**

  Type: String

  Description: (Optional) Desired volume size (GiB) of the SQL Server TempDB directory for the new EC2 instance.

  Default value: 100

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to successfully use the runbook.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags",
                "ec2:DescribeImages",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:RebootInstances",
                "ec2:RunInstances",
```

```
                "ssm:DescribeInstanceInformation",
                "ssm:GetAutomationExecution",
                "ssm:ListCommandInvocations",
                "ssm:ListCommands",
                "ssm:SendCommand",
                "ssm:StartAutomationExecution"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::ACCOUNTID:role/ROLENAME"
        }
    ]
}
```

Document Steps

To use this automation, follow the steps that apply to your instance type:

For new EC2 instances:

1. `aws:executeAwsApi` - Retrieve the AMI ID for SQL Server 2017 on Ubuntu 16.04.
2. `aws:runInstances` - Launch a new EC2 instance for Linux.
3. `aws:waitForAwsResourceProperty` - Wait for the newly created EC2 instance to be ready.
4. `aws:executeAwsApi` - Reboot the instance if the instance is not ready.
5. `aws:assertAwsResourceProperty` - Verify that SSM Agent is installed.
6. `aws:runCommand` - Run the SQL Server restore script in PowerShell.

For existing EC2 instances:

1. `aws:waitForAwsResourceProperty` - Verify that the EC2 instance is ready.
2. `aws:executeAwsApi` - Reboot the instance if the instance is not ready.
3. `aws:assertAwsResourceProperty` - Verify that SSM Agent is installed.
4. `aws:runCommand` - Run the SQL Server restore script in PowerShell.

Outputs

getInstance.InstanceId

restoreToNewInstance.Output

restoreToExistingInstance.Output

AWSSupport-ActivateWindowsWithAmazonLicense

Description

The `AWSSupport-ActivateWindowsWithAmazonLicense` runbook activates an Amazon Elastic Compute Cloud (Amazon EC2) instance for Windows Server with a license provided by Amazon. The automation verifies and configures required key management service operating system settings and attempts activation. This includes operating system routes to Amazon's key management servers and key management service operating system settings. Setting the `AllowOffline` parameter to `true` allows the automation to successfully target instances that are not managed by Amazon Systems Manager, but requires a stop and start of the instance.

> **Note**
>
> This runbook cannot be used on Bring Your Own License (BYOL) model Windows Server instances. For information about using your own license, see Microsoft Licensing on Amazon.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AllowOffline

Type: String

Valid values: true | false

Default: false

Description: (Optional) Set it to `true` if you allow an offline Windows activation remediation in case the online troubleshooting fails, or if the provided instance is not a managed instance.

> **Important**
>
> The offline method requires that the provided EC2 instance be stopped and then started. Data stored in instance store volumes will be lost. The public IP address will change if you are not using an Elastic IP.

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ForceActivation

  Type: String

  Valid values: true | false

  Default: false

  Description: (Optional) Set it to `true` if you want to proceed even if Windows is already activated.

- InstanceId

  Type: String

  Description: (Required) ID of your managed EC2 instance for Windows Server.

- SubnetId

Type: String

Default: CreateNewVPC

Description: (Optional) Offline only - The subnet ID for the EC2Rescue instance used to perform the offline troubleshooting. Use `SelectedInstanceSubnet` to use the same subnet as your instance, or use `CreateNewVPC` to create a new VPC. IMPORTANT: The subnet must be in the same Availability Zone as InstanceId, and it must allow access to the SSM endpoints.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

We recommend that the EC2 instance receiving the command has an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. You must have at least **ssm:StartAutomationExecution** and **ssm:SendCommand** to run the automation and send the command to the instance, plus **ssm:GetAutomationExecution** to be able to read the automation output. For the offline remediation, see the permissions needed by `AWSSupport-StartEC2RescueWorkflow`.

Document Steps

1. `aws:assertAwsResourceProperty` - Check the provided instance's platform is Windows.
2. `aws:assertAwsResourceProperty` - Confirm the provided instance is a managed instance:
   a. (Online activation fix) If the input instance is a managed instance, then run `aws:runCommand` to run the PowerShell script to attempt to fix Windows activation.
   b. (Offline activation fix) If the input instance is not a managed instance:
      i. `aws:assertAwsResourceProperty` - Verifies the `AllowOffline` flag is set to `true`. If so, the offline fix starts; otherwise the automation ends.
      ii. `aws:executeAutomation` - Invoke `AWSSupport-StartEC2RescueWorkflow` with the Windows activation offline fix script. The script uses either EC2Config or EC2Launch, depending on the OS version.
      iii. `aws:executeAwsApi` - Read the result from `AWSSupport-StartEC2RescueWorkflow`.

Outputs

activateWindows.Output

getActivateWindowsOfflineResult.Output

AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2

Description

The `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2` runbook analyzes connectivity from an Amazon Elastic Compute Cloud (Amazon EC2) instance or elastic network interface to an Amazon Web Services service endpoint. IPv6 is not supported. The runbook uses the value that you specify for the `ServiceEndpoint` parameter to analyze connectivity to an endpoint. If an Amazon PrivateLink endpoint can't be found in your VPC, the runbook uses a public IP address for the service in the current Amazon Web Services Region. This automation uses Reachability Analyzer from Amazon Virtual Private Cloud. For more information, see What is Reachability Analyzer?, in *Reachability Analyzer*.

This automation checks the following:

- Checks whether your virtual private cloud (VPC) is configured to use the Amazon provided DNS server.
- Checks whether an Amazon PrivateLink endpoint exists in the VPC for the Amazon Web Services service that you specify. If an endpoint is found, the automation verifies that the `privateDns` attribute is turned on.
- Checks if the Amazon PrivateLink endpoint is using the default endpoint policy.

Considerations

- You are charged per analysis run between a source and destination. For more information, see Amazon VPC Pricing.
- During the automation, a network insights path and network insights analysis are created. If the automation completes successfully, the runbook deletes these resources . If the cleanup step fails, the network insights path is not deleted by the runbook and you will need to delete it manually. If you don't delete the network insights path manually, it continues to count towards the quota for your Amazon Web Services account. For more information about quotas for Reachability Analyzer, see Quotas for Reachability Analyzer in *Reachability Analyzer*.
- Operating system-level configurations such as the use of a proxy, local DNS resolver, or hosts file can affect connectivity even if the Reachability Analyzer returns PASS.

- Review the evaluation of all checks performed by the Reachability Analyzer. If any of the checks return with a status of `FAIL`, that might affect connectivity even if the overall reachability check returns a status of PASS.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Source

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance or the network interface from which you want to analyze reachability.
- ServiceEndpoint

  Type: String

  Description: (Required) The hostname of the service endpoint that you want to analyze reachability to.
- RetainVpcReachabilityAnalysis

Type: String

Default: false

Description: (Optional) Determines whether the network insight path and related analysis created are retained. By default, the resources used for analyze reachability are deleted after successful analysis. If you choose to retain the analysis, the runbook does not delete the analysis and you can visualize it in the Amazon VPC console. A console link is available in the automation output.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:CreateNetworkInsightsPath`
- `ec2:DeleteNetworkInsightsAnalysis`
- `ec2:DeleteNetworkInsightsPath`
- `ec2:DescribeAvailabilityZones`
- `ec2:DescribeCustomerGateways`
- `ec2:DescribeDhcpOptions`
- `ec2:DescribeInstances`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNatGateways`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeNetworkInsightsAnalyses`
- `ec2:DescribeNetworkInsightsPaths`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribePrefixLists`
- `ec2:DescribeRegions`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`

- ec2:DescribeSubnets
- ec2:DescribeTransitGatewayAttachments
- ec2:DescribeTransitGatewayPeeringAttachments
- ec2:DescribeTransitGatewayConnects
- ec2:DescribeTransitGatewayRouteTables
- ec2:DescribeTransitGateways
- ec2:DescribeTransitGatewayVpcAttachments
- ec2:DescribeVpcAttribute
- ec2:DescribeVpcEndpoints
- ec2:DescribeVpcEndpointServiceConfigurations
- ec2:DescribeVpcPeeringConnections
- ec2:DescribeVpcs
- ec2:DescribeVpnConnections
- ec2:DescribeVpnGateways
- ec2:GetManagedPrefixListEntries
- ec2:GetTransitGatewayRouteTablePropagations
- ec2:SearchTransitGatewayRoutes
- ec2:StartNetworkInsightsAnalysis
- elasticloadbalancing:DescribeListeners
- elasticloadbalancing:DescribeLoadBalancerAttributes
- elasticloadbalancing:DescribeLoadBalancers
- elasticloadbalancing:DescribeRules
- elasticloadbalancing:DescribeTags
- elasticloadbalancing:DescribeTargetGroups
- elasticloadbalancing:DescribeTargetHealth
- tiros:CreateQuery
- tiros:GetQueryAnswer
- tiros:GetQueryExplanation

Document Steps

1. aws:executeScript: Validates the service endpoint by attempting to resolve the hostname.
2. aws:executeScript: Gathers details about the VPC and subnet.
3. aws:executeScript: Evaluates the DNS configuration of the VPC.
4. aws:executeScript: Evaluates the VPC endpoint checks.
5. aws:executeScript: Locates an internet gateway to connect to the public service endpoint.
6. aws:executeScript: Determines the destination to be used for reachability analysis.
7. aws:executeScript: Analyzes the reachability from source to the endpoint using Reachability Analyzer and cleans up the resources if the analysis is successful.
8. aws:executeScript: Generates a reachability evaluation report.
9. aws:executeScript: Generates the output in JSON.

Outputs

- generateReport.EvalReport - The results of the checks performed by the automation in text format.
- generateJsonOutput.Output - A minimal version of the results in JSON format.

AWSPremiumSupport-ChangeInstanceTypeIntelToAMD

Description

TheAWSPremiumSupport-ChangeInstanceTypeIntelToAMD runbook automates migrations from Intel powered Amazon Elastic Compute Cloud (Amazon EC2) instances to the equivalent AMD powered instance types. This runbook supports general purpose (M), burstable general purpose (T), compute optimized (C), and memory optimized (R) instances built on the Nitro system. This runbook can be used on instances that aren't managed by Systems Manager.

To reduce the potential risk of data loss and downtime, the runbook checks the instance's stop behavior, whether the instance is in an Amazon EC2 Auto Scaling group, the health of the instance, and that the equivalent AMD powered instance type is available in the same Availability Zone. By default, this runbook will not change the instance type if instance store volumes are attached, or if the instance is part of an Amazon CloudFormation stack. If you want to change this behavior, specify yes for either of the AllowInstanceStoreInstances and AllowCloudFormationInstances parameters.

Important

Access to `AWSPremiumSupport-*` runbooks requires either an Enterprise or Business Support Subscription. For more information, see Compare Amazon Web Services Support Plans.

Considerations

- We recommend backing up your instance before using this runbook.
- Changing the instance type requires the runbook to stop your instance. When an instance is stopped, any data stored in the RAM or the instance store volumes is lost, and the automatic public IPv4 address is released. For more information, see Stop and start your instance.
- If you don't specify a value for the `TargetInstanceType` parameter, the runbook attempts to identify the equivalent AMD instance in terms of virtual CPUs and memory within the same instance family. The runbook ends if it is not able to identify an equivalent AMD instance type.
- By using the `DryRun` option, you can capture the equivalent AMD instance type, and validate requirements without actually changing the instance type.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Acknowledge

  Type: String

  Description: (Required) Enter `yes` to acknowledge that your target instance will be stopped if it's running.

- InstanceId

  Type: String

  Description: (Required) The ID of Amazon EC2 instance whose type you want to change.

- TargetInstanceType

  Type: String

  Default: automatic

  Description: (Optional) The AMD instance type you want to change your instance to. The default `automatic` value uses the equivalent instance type in terms of virtual CPUs and memory. For example, an m5.large would be changed to m5a.large.

- AllowInstanceStoreInstances

  Type: String

  Valid values: no | yes

  Default: no

  Description: (Optional) If you specify `yes`, the runbook runs on instances that have instance store volumes attached.

- AllowCloudFormationInstances

  Type: String

  Valid values: no | yes

Default: no

Description: (Optional) If set to `yes`, the runbook runs on instances that are part of a Amazon CloudFormation stack.

- AllowCrossGeneration

  Type: String

  Valid values: no | yes

  Default: no

  Description: (Optional) If set to `yes`, the runbook attempts to find the newest equivalent AMD instance type within the same instance family.

- DryRun

  Type: String

  Valid values: no | yes

  Default: no

  Description: (Optional) If set to `yes`, the runbook returns the equivalent AMD instance type and validates migration requirements without making changes to the instance type.

- SleepWait

  Type: String

  Default: PT3S

  Description: (Optional) The time the runbook should wait before starting a new automation. The value you provide for this parameter must match the ISO 8601 standard. For more information about creating ISO 8601 strings, see Formatting date and time strings for Systems Manager.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DescribeAutomationExecutions`

- ssm:GetAutomationExecution
- ssm:StartAutomationExecution
- ec2:GetInstanceTypesFromInstanceRequirements
- ec2:DescribeInstanceAttribute
- ec2:DescribeInstances
- ec2:DescribeInstanceStatus
- ec2:DescribeInstanceTypeOfferings
- ec2:DescribeInstanceTypes
- ec2:DescribeTags
- ec2:ModifyInstanceAttribute
- ec2:StartInstances
- ec2:StopInstances

Document Steps

1. aws:assertAwsResourceProperty: Confirms the status of the target Amazon EC2 instance is running, pending, stopped, or stopping. Otherwise, the automation ends.
2. aws:executeAwsApi: Gathers properties from the target Amazon EC2 instance.
3. aws:branch: Branches the automation based on the state of the Amazon EC2 instance.
   a. If stopped or stopping, the automation runs aws:waitForAwsResourceProperty until the Amazon EC2 instance is fully stopped.
   b. If running or pending, the automation runs aws:waitForAwsResourceProperty until the Amazon EC2 instance passes status checks.
4. aws:assertAwsResourceProperty: Confirms the Amazon EC2 instance is not part of an Auto Scaling group by checking if the aws:autoscaling:groupName tag is applied.
5. aws:executeAwsApi: Gathers the current instance type properties to find the equivalent AMD instance type.
6. aws:assertAwsResourceProperty: Confirms a Amazon Web Services Marketplace product code is not associated with the Amazon EC2 instance. Some products are not available on all instance types.
7. aws:branch: Branches the automation depending on whether you want the automation to check if the Amazon EC2 instance is part of a Amazon CloudFormation stack

   a. If the `aws:cloudformation:stack-name` tag is applied to the instance, the automation runs `aws:assertAwsResourceProperty` to confirm the instance is not part of a Amazon CloudFormation stack.

8. `aws:branch`: Branches the automation based on whether the instance root volume type is Amazon Elastic Block Store (Amazon EBS).
9. `aws:assertAwsResourceProperty`: Confirms the instance shutdown behavior is `stop` and not `terminate`.
10. `aws:executeScript`: Confirms there is only one automation of this runbook targeting the current instance. If another automation is already in progress targeting the same instance, it returns an error and ends.
11. `aws:executeAwsApi`: Returns a list of the AMD instance types with same amount of memory and vCPUs.
12. `aws:executeScript`: Checks if the current instance type is supported and returns its equivalent AMD instance type. If there is no equivalent, the automation ends.
13. `aws:executeScript`: Confirms the AMD instance type is available in the same Availability Zone, and verifies the provided IAM permissions.
14. `aws:branch`: Branches the automation based on whether the `DryRun` parameter value is `yes`.
15. `aws:branch`: Checks if the original and the target instance type are the same. If they're the same, the automation ends.
16. `aws:executeAwsApi`: Gets the current instance state.
17. `aws:changeInstanceState`: Stops the Amazon EC2 instance.
18. `aws:changeInstanceState`: Forces the instance to stop if it's stuck in stopping state.
19. `aws:executeAwsApi`: Changes the instance type to the target AMD instance type.
20. `aws:sleep`: Waits 3 seconds after changing the instance type for eventual consistency.
21. `aws:branch`: Branches the automation based on the previous instance state. If it was `running`, the instance is started.
    a. `aws:changeInstanceState`: Starts the Amazon EC2 instance if it was running before changing the instance type.
    b. `aws:waitForAwsResourceProperty`: Waits for the Amazon EC2 instance to pass status checks. If the instance doesn't pass status checks, the instance is changed back to its original instance type.
        i. `aws:changeInstanceState`: Stops the Amazon EC2 instance before changing it to its original instance type.

ii. `aws:changeInstanceState`: Forces the Amazon EC2 instance to stop before changing it to its original instance type in case it gets stuck in a stopping state.

iii. `aws:executeAwsApi`: Changes Amazon EC2 instance to its original type.

iv. `aws:sleep`: Waits 3 seconds after changing instance type for eventual consistency.

v. `aws:changeInstanceState`: Starts the Amazon EC2 instance if it was running before changing the instance type.

vi. `aws:waitForAwsResourceProperty`: Waits for the Amazon EC2 instance to pass status checks.

22`aws:sleep`: Waits before ending the runbook.

AWSSupport-CheckXenToNitroMigrationRequirements

Description

The `AWSSupport-CheckXenToNitroMigrationRequirements` runbook verifies that an Amazon Elastic Compute Cloud (Amazon EC2) instance meets the prerequistes to successfully change the instance type from a Xen type instance to Nitro-based instance type. This automation checks the following:

- The root device is an Amazon Elastic Block Store (Amazon EBS) volume.
- The `enaSupport` attribute is enabled.
- The ENA module is installed on the instance.
- The NVMe module is installed on the instance. If yes, the module is installed and a script verifies that the module is loaded in the `initramfs` image.
- Analyzes `/etc/fstab` and looks for block devices being mounted using device names.
- Determines whether the operating system (OS) uses predictable network interface names by default.

This runbook supports the following operating systems:

- Red Hat Enterprise Linux
- CentOS
- Amazon Linux 2
- Amazon Linux

- Debian Server
- Ubuntu Server
- SUSE Linux Enterprise Server 15 SP2
- SUSE Linux Enterprise Server 12 SP5

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Default: false

  Description: (Required) The ID of the Amazon EC2 instance which you want to check prerequisites for before migrating to a Nitro-based instance type.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:DescribeInstanceInformation`
- `ssm:DescribeInstanceProperties`
- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:GetDocument`
- `ssm:ListCommands`
- `ssm:ListCommandInvocations`
- `ssm:ListDocuments`
- `ssm:StartAutomationExecution`
- `ssm:SendCommand`
- `iam:ListRoles`
- `ec2:DescribeInstances`
- `ec2:DescribeInstancesTypes`

Document Steps

- `aws:executeAwsApi` - Gathers details about the instance.
- `aws:executeAwsApi` - Gathers information about the hypervisor for the instance.
- `aws:branch` - Branches based on whether the target instance is already running a Nitro-based instance type.
- `aws:branch` - Checks whether the instance's OS is supported by Nitro-based instances.
- `aws:assertAwsResourceProperty` - Verifies the instance you specified is managed by Systems Manager, and that the status is `Online`.
- `aws:branch` - Branches based on whether the root device of the instance is an Amazon EBS volume.
- `aws:branch` - Branches based on whether the ENA attribute is enabled for the instance.

- `aws:runCommand` - Checks for ENA drivers on the instance.
- `aws:runCommand` - Checks for NVMe drivers on the instance.
- `aws:runCommand` - Checks the `fstab` file for unrecognized formats.
- `aws:runCommand` - Checks for predictable interface name configuration on the instance.
- `aws:executeScript` - Generates output based on previous steps.

Outputs

finalOutput.output - The results of the checks performed by the automation.

AWSSupport-ConfigureEC2Metadata

Description

This runbook helps you configure instance metadata service (IMDS) options for Amazon Elastic Compute Cloud (Amazon EC2) instances. Using this runbook, you can configure the following:

- Enforce the use of IMDSv2 for instance metadata.
- Configure the `HttpPutResponseHopLimit` value.
- Allow or deny instance metadata access.

For more information about instance metadata, see Configuring the Instance Metadata Service in the *Amazon EC2 User Guide*.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EnforceIMDSv2

  Type: String

  Valid values: required | optional

  Default: optional

  Description: (Optional) Enforce IMDSv2. If you choose `required`, the Amazon EC2 instance will only use IMDSv2. If you choose `optional`, you can choose between IMDSv1 and IMDSv2 for metadata access.

  > **Important**
  >
  > If you enforce IMDSv2, applications that use IMDSv1 might not function correctly. Before enforcing IMDSv2, make sure your applications that use IMDS are upgraded to a version that support IMDSv2. For information about Instance Metadata Service Version 2 (IMDSv2), see Configuring the Instance Metadata Service in the *Amazon EC2 User Guide*.

- HttpPutResponseHopLimit

  Type: Integer

  Valid values: 0-64

  Default: 0

  Description: (Optional) The desired HTTP PUT response hop limit value (1-64) for instance metadata requests. This value controls the number of hops that the PUT response can traverse. To prevent the response from traveling outside of the instance, specify 1 for the parameter value.

- InstanceId

Type: String

Description: (Required) The ID of the Amazon EC2 instance whose metadata settings you want to configure.

- MetadataAccess

  Type: String

  Valid values: enabled | disabled

  Default: enabled

  Description: (Optional) Allow or deny instance metadata access in the Amazon EC2 instance. If you specify `disabled`, all other parameters will be ignored and the metadata access will be denied for the instance.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`
- `ec2:ModifyInstanceMetadataOptions`
- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`

Document Steps

1. branchOnMetadataAccess - Branches automation based on the value of `MetadataAccess` parameter.
2. disableMetadataAccess - Calls the ModifyInstanceMetadataOptions API action to disable metadata endpoint access.
3. branchOnHttpPutResponseHopLimit - Branches automation based on the value of `HttpPutResponseHopLimit` parameter.
4. maintainHopLimitAndConfigureImdsVersion - If `HttpPutResponseHopLimit` is 0, maintains current hop limit and changes other metadata options.

5. waitBeforeAssertingIMDSv2State - Waits 30 seconds before asserting IMDSv2 status.
6. setHopLimitAndConfigureImdsVersion - If `HttpPutResponseHopLimit` is greater than 0, configures the metadata options using the given input parameters.
7. waitBeforeAssertingHopLimit - Waits 30 seconds before asserting metadata options.
8. assertHopLimit - Asserts the `HttpPutResponseHopLimit` property is set to the value you specified.
9. branchVerificationOnIMDSv2Option - Branches verification based on the value of `EnforceIMDSv2` parameter.
10. assertIMDSv2IsOptional - Asserts `HttpTokens` value set to `optional`.
11. assertIMDSv2IsEnforced - Asserts `HttpTokens` value set to `required`.
12. waitBeforeAssertingMetadataState - Waits 30 seconds before asserting the metadata state is disabled.
13. assertMetadataIsDisabled - Asserts metadata is `disabled`.
14. describeMetadataOptions - Gets the metadata options after the changes you've specified have been applied.

Outputs

describeMetadataOptions.State

describeMetadataOptions.MetadataAccess

describeMetadataOptions.IMDSv2

describeMetadataOptions.HttpPutResponseHopLimit

AWSSupport-ContainEC2Instance

Description

The `AWSSupport-ContainEC2Instance` runbook provides an automated solution for the procedure outlined in the article How do I isolate the Amazon EC2 Instance when faced with a potentially compromised or suspicious? The automation branches depending on the values you specify.

How does it work?

This Automation runbook `AWSSupport-ContainEC2Instance` performs network containment of an Amazon EC2 Instance through a series of coordinated steps. When executed in `Contain` mode, it first validates the input parameters and checks if the instance is not terminated. It then backs up the current security group configuration to an Amazon S3 bucket for later restoration. The runbook creates two security groups: a temporary "all access" security group and a final "containment" security group. It gradually transitions the instance's network interfaces from their original security groups to the all-access security group, and finally to the containment security group. If specified, it creates both unencrypted and encrypted AMI backups of the instance. For instances in an Auto Scaling group, it handles the necessary Auto Scaling group modifications and brings the instance to standby state. When executed in `Release` mode, it restores the instance to its original network configuration using the backed-up settings from Amazon S3. The runbook supports a `DryRun` parameter to preview actions without making actual changes, and includes comprehensive error handling and reporting mechanisms throughout the containment and release workflows.

Important

- This runbook performs various operations that require elevated privileges, such as modifying security groups, creating AMIs, and interacting with Auto Scaling groups. These actions could potentially lead to privilege escalation or impact other workloads in your account. You should review the permissions granted to the role specified by the `AutomationAssumeRole` parameter and ensure they are appropriate for the intended use case. You can refer to the following Amazon documentation for more information on IAM permissions: Amazon Identity and Access Management (IAM) Permissions Amazon Systems Manager Automation Permissions.
- This runbook performs mutative actions that could potentially cause unavailability or disruption to your workloads. Specifically, it modifies the security groups associated with the target Amazon EC2 Instance, which could impact network connectivity. Additionally, if the instance is part of an Auto Scaling group, the runbook may modify the group's configuration, potentially affecting its scaling behavior.
- During the containment process, this runbook creates additional resources, such as security groups and AMIs. While these resources are tagged for identification, you should be aware of their creation and ensure proper cleanup or management after the containment process is complete.
- If the `Action` parameter is set to `Release`, this runbook attempts to restore the Amazon EC2 Instance's configuration to its original state. However, there is a risk that the restoration process may fail, leaving the instance in an inconsistent state. The runbook

provides instructions for manual restoration in case of such failures, but you should be prepared to handle potential issues during the restoration process.

It is recommended to review the runbook thoroughly, understand its potential impacts, and test it in a non-production environment before executing it in your production environment.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- autoscaling:CreateOrUpdateTags
- autoscaling:DeleteTags
- autoscaling:DescribeAutoScalingGroups
- autoscaling:DescribeAutoScalingInstances
- autoscaling:DescribeTags
- autoscaling:EnterStandby
- autoscaling:ExitStandby
- autoscaling:UpdateAutoScalingGroup
- ec2:AuthorizeSecurityGroupEgress
- ec2:AuthorizeSecurityGroupIngress

- ec2:CopyImage
- ec2:CreateImage
- ec2:CreateSecurityGroup
- ec2:CreateSnapshot
- ec2:CreateTags
- ec2:DeleteSecurityGroup
- ec2:DeleteTags
- ec2:DescribeImages
- ec2:DescribeInstances
- ec2:DescribeSecurityGroups
- ec2:DescribeSnapshots
- ec2:DescribeTags
- ec2:ModifyNetworkInterfaceAttribute
- ec2:RevokeSecurityGroupEgress
- kms:CreateGrant
- kms:DescribeKey
- kms:GenerateDataKeyWithoutPlaintext
- kms:ReEncryptFrom
- kms:ReEncryptTo
- s3:CreateBucket
- s3:DeleteObjectTagging
- s3:GetAccountPublicAccessBlock
- s3:GetBucketAcl
- s3:GetBucketLocation
- s3:GetBucketOwnershipControls
- s3:GetBucketPolicy
- s3:GetBucketPolicyStatus
- s3:GetBucketPublicAccessBlock
- s3:GetObject
- s3:ListBucket

- s3:PutAccountPublicAccessBlock
- s3:PutBucketPolicy
- s3:PutBucketVersioning
- s3:PutObject
- s3:PutObjectTagging

Example Policy:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "ReadOperations",
   "Effect": "Allow",
   "Action": [
    "autoscaling:DescribeAutoScalingGroups",
    "autoscaling:DescribeAutoScalingInstances",
    "autoscaling:DescribeTags",
    "ec2:DescribeImages",
    "ec2:DescribeInstances",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSnapshots",
    "ec2:DescribeTags",
    "kms:DescribeKey",
    "s3:GetAccountPublicAccessBlock",
    "s3:GetBucketAcl",
    "s3:GetBucketLocation",
    "s3:GetBucketOwnershipControls",
    "s3:GetBucketPolicy",
    "s3:GetBucketPolicyStatus",
    "s3:GetBucketPublicAccessBlock",
    "s3:GetObject",
    "s3:ListBucket"
   ],
   "Resource": "*"
  },
  {
   "Sid": "WriteOperations",
   "Effect": "Allow",
   "Action": [
```

```
      "autoscaling:CreateOrUpdateTags",
      "autoscaling:DeleteTags",
      "autoscaling:EnterStandby",
      "autoscaling:ExitStandby",
      "autoscaling:UpdateAutoScalingGroup",
      "ec2:AuthorizeSecurityGroupEgress",
      "ec2:AuthorizeSecurityGroupIngress",
      "ec2:CopyImage",
      "ec2:CreateImage",
      "ec2:CreateSecurityGroup",
      "ec2:CreateSnapshot",
      "ec2:CreateTags",
      "ec2:DeleteSecurityGroup",
      "ec2:DeleteTags",
      "ec2:ModifyNetworkInterfaceAttribute",
      "ec2:RevokeSecurityGroupEgress",
      "kms:CreateGrant",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:ReEncryptFrom",
      "kms:ReEncryptTo",
      "s3:CreateBucket",
      "s3:DeleteObjectTagging",
      "s3:PutAccountPublicAccessBlock",
      "s3:PutBucketPolicy",
      "s3:PutBucketVersioning",
      "s3:PutObject",
      "s3:PutObjectTagging"
    ],
    "Resource": "*"
   }
  ]
}
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-ContainEC2Instance in Systems Manager under Documents.
2. Select **Execute automation.**
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

  - Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
  - Type: `AWS::IAM::Role::Arn`
- **Action (Required):**
  - Description: (Required) Select `Contain` to isolate the Amazon EC2 instance or `Restore` to try to restore the Amazon EC2 instance configuration original configuration from a previous backup.
  - Type: String
  - Allowed Pattern: `Contain|Restore`
- **DryRun (Optional):**
  - Description: (Optional) When set to `true`, the automation will not execute any of the commands, instead it will report on what it would have attempted to do, detailing out each step. Default value: `true`.
  - Type: Boolean
  - Allowed Values: `true|false`
- **CreateAMIBackup (Optional):**
  - Description: (Optional) When set to `true`, an AMI of the Amazon EC2 Instance will be created before performing the containment actions.
  - Type: Boolean
  - Allowed Values: `true|false`
- **KmsKey (Optional):**
  - Description: (Optional) The ID of the Amazon KMS key that will be used to create an encrypted AMI of target Amazon EC2 instance. Default is set to `alias/aws/ebs`.
  - Type: String
  - Allowed Pattern: `^(((arn:(aws|aws-cn|aws-us-gov):kms:([a-z]{2}|[a-z]{2}-gov)-[a-z]+-[0-9]{1}:[0-9]{12}:key/)?([a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}|mrk-[a-f0-9]{32}))|(arn:(aws|aws-cn|aws-us-gov):kms:([a-z]{2}|[a-z]{2}-gov)-[a-z]+-[0-9]{1}:[0-9]{12}:)?alias/.{1,})$`
- **BackupS3BucketName (Conditional):**

- Description: (Conditional) Amazon Amazon S3 bucket to upload the configuration when `Action` is `Contain` or to restore the configuration when `Action` is `Release`. **Note:** If the provided bucket doesn't exist in the account, the automation will create a Amazon S3 bucket on your behalf.
- Type: `AWS::S3::Bucket::Name`

- **TagIdentifier (Optional):**
  - Description: (Optional) A tag in the format `Key=BatchId,Value=78925` that will be added to the Amazon resources created or modified by this runbook during the containment workflow. This tag can be used to identify and manage resources associated during containment process. During the restore workflow, the tag specified by this parameter will be removed from the resources. **Note:** Tag keys and values are case-sensitive.
  - Type: String
  - Allowed Pattern: `^$|^[Kk][Ee][Yy]=[\\+\\-\\=\\.\\_\\:\\/@a-zA-Z0-9]{1,128},[Vv][Aa][Ll][Uu][Ee]=[\\+\\-\\=\\.\\_\\:\\/@a-zA-Z0-9]{0,128}$`
- **BackupS3BucketAccess (Conditional):**
  - Description: (Conditional) The ARN of the IAM users or roles that will be allowed access to the backup Amazon S3 bucket after running the containment actions. This parameter is required when `Action` is `Contain`. The `AutomationAssumeRole`, or in its absence the user under whose context the automation is running is automatically added to the list.
  - Type: String
  - Allowed Pattern: `^$|^arn:(aws|aws-cn|aws-us-gov|aws-iso(-[a-z])?):iam::[0-9]{12}:(role|user)\\/[\\w+\\/=,.@-]+$`
- **IngressTrafficRules (Optional):**
  - Description: (Optional) A comma separated map of security group ingress rules with Cidr, IpProtocol, FromPort and ToPort in the format `[{"Cidr": "1.2.3.4/32", "IpProtocol": "tcp", "FromPort":"22", "ToPort":"22"}]` to be applied to the Amazon EC2 instance. If no rules are provided, a security group without any ingress rules will be attached to the Amazon EC2 instance, effectively isolating it from any incoming traffic.
  - Type: MapList

- Allowed Pattern: ^\\{\\}$|^\\{\"Cidr\":\"[\\x00-\\x7F+]{1,128}\",\"IpProtocol\":\"[\\x00-\\x7F+]{1,128}\",\"FromPort\":\"[\\x00-\\x7F+]{1,128}\",\"ToPort\":\"[\\x00-\\x7F+]{0,255}\"\\}

- **EgressTrafficRules (Optional):**
  - Description: (Optional) A comma separated map of security group egress rules with Cidr, IpProtocol, FromPort and ToPort in the format `[{"Cidr": "1.2.3.4/32", "IpProtocol": "tcp", "FromPort":"22", "ToPort":"22"}]` to be applied to the Amazon Amazon EC2 instance. If no rules are provided, a security group without any egress rules will be attached to the Amazon EC2 instance, effectively preventing all outgoing traffic.
  - Type: MapList
  - Allowed Pattern: ^\\{\\}$|^\\{\"Cidr\":\"[\\x00-\\x7F+]{1,128}\",\"IpProtocol\":\"[\\x00-\\x7F+]{1,128}\",\"FromPort\":\"[\\x00-\\x7F+]{1,128}\",\"ToPort\":\"[\\x00-\\x7F+]{0,255}\"\\}

- **BackupS3KeyName (Optional):**
  - Description: (Optional) If `Action` is set to `Restore`, this specifies the Amazon S3 key the automation will use to try to restore the target Amazon EC2 instance configuration. The Amazon S3 key typically follows this format: `{year}/{month}/{day}/{hour}/{minute}/{automation_execution_id}.json`. The key can be obtained from the output of a previous containment automation execution.
  - Type: String
  - Allowed Pattern: ^[a-zA-Z0-9\\.\\-_\\\\!*'()/]{0,1024}$

4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

- **ValidateRequiredInputs**

  Validates that all required inputs are provided.

- **AssertInstanceIsNotTerminated**

  Checks if the target Amazon EC2 Instance is not in terminated (deleted).

- **GetAutoScalingInstanceInfo**

Gets the Amazon EC2 instance lifecycle and group name if the target Amazon EC2 instance is part of an Auto Scaling group.

- **CheckBackupS3BucketName**

  Checks if the target Amazon S3 bucket potentially grants `read` or `write` public access to its objects. A new Amazon S3 bucket is created if the `BackupS3BucketName` bucket doesn't exist.

- **BranchOnActionAndMode**

  Branches the automation based on the input parameters `Action` and `DryRun`.

- **BranchOnAutoScalingGroupMembership**

  Branches the automation based on if the target Amazon EC2 Instance is part of Auto Scaling group and its lifecycle state.

- **DescribeAutoScalingGroups**

  Gets and stores the associated Amazon EC2 Auto Scaling group configuration.

- **ModifyAutoScalingGroup**

  Modifies the associated Amazon EC2 Auto Scaling group configuration for the containment actions, setting the Amazon EC2 instance to the `Standby` state and adjusting the Auto Scaling group `MinSize` capacity.

- **BackupInstanceSecurityGroups**

  Gets and stores the configuration of the target Amazon EC2 Instance security groups.

- **CreateAllAccessSecurityGroup**

  Creates a temporary security group allowing all ingress traffic that replaces the target Amazon EC2 Instance's security groups.

- **CreateContainmentSecurityGroup**

  Creates a restrictive containment security group with the specified ingress and egress rules, and replaces the temporary all-access group with it.

- **BranchOnCreateAMIBackup**

  Branches the automation based on the `CreateAMIBackup` input parameter.

- **AssertSourceInstanceRootVolumeIsEbs**

Checks if the target Amazon EC2 Instance root volume is Amazon EBS.

- **CreateImage**

  Creates an AMI of the target Amazon EC2 Instance.

- **RestoreInstanceConfiguration**

  Restores the target Amazon EC2 Instance configuration from the backup.

- **ReportContain**

  Outputs dry run details for the containment actions.

- **ReportRestore**

  Outputs dry run details for the restoring actions.

- **ReportRestoreFailure**

  Provides instructions to restore the target Amazon EC2 Instance original configuration during a restore workflow failure scenario.

- **ReportContainmentFailure**

  Provides instructions to restore the target Amazon EC2 Instance original configuration during a containment workflow failure scenario.

- **FinalOutput**

  Outputs the details of the containment actions.

7. After the execution completes, review the Outputs section for the detailed results of the execution:

- **FinalOutput.Output**

  Outputs the details of the containment actions performed by this runbook when `DryRun` is set to False.

- **RestoreInstanceConfiguration.Output**

  Outputs the restore actions performed by this runbook when `DryRun` is set to False.

- **ReportContain.Output**

  Outputs the details of the containment actions performed by this runbook when `DryRun` is set to True.

- **ReportRestore.Output**

  Outputs the details of the restore actions performed by this runbook when `DryRun` is set to True.

- **ReportContainmentFailure.Output**

  Provides instructions to restore the target Amazon EC2 Instance original configuration during a containment workflow failure scenario.

- **ReportRestoreFailure.Output**

  Provides instructions to restore the target Amazon EC2 Instance original configuration during a restore workflow failure scenario.

References

Systems Manager Automation

- Run this Automation (console)
- Running a simple automation
- Setting up Automation
- Support Automation Workflows

AWSSupport-CopyEC2Instance

Description

The `AWSSupport-CopyEC2Instance` runbook provides an automated solution for the procedure outlined in the Knowledge Center article How do I move my EC2 instance to another subnet, Availability Zone, or VPC? The automation branches depending on the values you specify for the `Region` and `SubnetId` parameters.

If you specify a value for the `SubnetId` parameter but not a value for the `Region` parameter, the automation creates an Amazon Machine Image (AMI) of the target instance and launches a new instance from the AMI in the subnet you specified.

If you specify a value for the `SubnetId` parameter and the `Region` parameter, the automation creates an AMI of the target instance, copies the AMI to the Amazon Web Services Region you specified, and launches a new instance from the AMI in the subnet you specified.

If you specify a value for the `Region` parameter but not a value for the `SubnetId` parameter, the automation creates an AMI of the target instance, copies the AMI to the Region you specified, and launches a new instance from the AMI in the default subnet of your virtual private cloud (VPC) in the destination Region.

If no value is specified for either the `Region` or `SubnetId` parameters, the automation creates an AMI of the target instance, and launches a new instance from the AMI in the default subnet of your VPC.

To copy an AMI to a different Region, you must provide a value for the `AutomationAssumeRole` parameter. If the automation times out during the `waitForAvailableDestinationAmi` step, the AMI might still be copying. If this is the case, you can wait for the copy to complete and launch the instance manually.

Before running this automation, note the following:

- AMIs are based on Amazon Elastic Block Store (Amazon EBS) snapshots. For large file systems without a previous snapshot, AMI creation can take several hours. To decrease the AMI creation time, create an Amazon EBS snapshot before you create the AMI.
- Creating an AMI doesn't create a snapshot for instance store volumes on the instance. For information about backing up instance store volumes to Amazon EBS, see How do I back up an instance store volume on my Amazon EC2 instance to Amazon EBS?
- The new Amazon EC2 instance has a different private IPv4 or public IPv6 IP address. You must update all references to the old IP addresses (for example, in DNS entries) with the new IP addresses that are assigned to the new instance. If you're using an Elastic IP address on your source instance, be sure to attach it to the new instance.
- Domain security identifier (SID) conflict issues can occur when the copy launches and tries to contact the domain. Before you capture the AMI, use Sysprep or remove the domain-joined instance from the domain to prevent conflict issues. For more information, see How can I use Sysprep to create and install custom reusable Windows AMIs?

Run this Automation (console)

Important

We do not recommend using this runbook to copy Microsoft Active Directory Domain Controller instances.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the instance that you want to copy.
- KeyPair

  Type: String

  Description: (Optional) The key pair you want to associate with the new copied instance. If you're copying the instance to a different Region, make sure the key pair exists in the specified Region.
- Region

  Type: String

  Description: (Optional) The Region you want to copy the instance to. If you specify a value for this parameter, but do not specify values for the `SubnetId` and `SecurityGroupIds` parameters, the automation attempts to launch the instance in the default VPC with the default security group. If EC2-Classic is enabled in the destination Region, the launch will fail.

- SubnetId

  Type: String

  Description: (Optional) The ID of the subnet you want to copy the instance to. If EC2-Classic is enabled in the destination Region, you must provide a value for this parameter.

- InstanceType

  Type: String

  Description: (Optional) The instance type the copied instance should be launched as. If you do not specify a value for this parameter, the source instance type is used. If the source instance type is not supported in the Region the instance is being copied to, the automation fails.

- SecurityGroupIds

  Type: String

  Description: (Optional) A comma-separated list of security group IDs you want to associate with the copied instance. If you do not specify a value for this parameter, and the instance is not being copied to a different Region, the security groups associated with the source instance are used. If you're copying the instance to a different Region, the default security group for the default VPC in the destination Region is used.

- KeepImageSourceRegion

  Type: Boolean

  Valid values: true | false

  Default: true

  Description: (Optional) If you specify `true` for this parameter, the automation does not delete the AMI of the source instance. If you specify `false` for this parameter, the automation deregisters the AMI and deletes the associated snapshots.

- KeepImageDestinationRegion

  Type: Boolean

  Valid values: true | false

  Default: true

Description: (Optional) If you specify `true` for this parameter, the automation does not delete the AMI that is copied to the Region you specified. If you specify `false` for this parameter, the automation deregisters the AMI and deletes the associated snapshots.

- NoRebootInstanceBeforeTakingImage

  Type: Boolean

  Valid values: true | false

  Default: false

  Description: (Optional) If you specify `true` for this parameter, the source instance will not be restarted before creating the AMI. When this option is used, file system integrity on the created image can't be guaranteed.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:CreateImage`
- `ec2:DeleteSnapshot`
- `ec2:DeregisterImage`
- `ec2:DescribeInstances`
- `ec2:DescribeImages`
- `ec2:RunInstances`

If you're copying the instance to a different Region, you will also need the following permissions.

- `ec2:CopyImage`

Document Steps

- describeOriginalInstanceDetails - Gathers details from the instance to be copied.
- assertRootVolumeIsEbs - Checks if the root volume device type is `ebs`, and if not, ends the automation.

- evalInputParameters - Evaluates the values provided for the input parameters.
- createLocalAmi - Creates an AMI of the source instance.
- tagLocalAmi - Tags the AMI created in the previous step.
- branchAssertRegionIsSame - Branches based on whether the instance is being copied within the same Region or to a different Region.
- branchAssertSameRegionWithKeyPair - Branches based on whether a value was provided for the `KeyPair` parameter for an instance that's being copied within the same Region.
- sameRegionLaunchInstanceWithKeyPair - Launches an Amazon EC2 instance from the AMI of the source instance in the same subnet or the subnet you specify using the key pair that you specified.
- sameRegionLaunchInstanceWithoutKeyPair - Launches an Amazon EC2 instance from the AMI of the source instance in the same subnet or the subnet you specify without a key pair.
- copyAmiToRegion - Copies the AMI to the destination Region.
- waitForAvailableDestinationAmi - Waits for the copied AMI state to become `available`.
- destinationRegionLaunchInstance - Launches an Amazon EC2 Instance using the copied AMI.
- branchAssertDestinationAmiToDelete - Branches based on the value you provided for the `KeepImageDestinationRegion` parameter.
- deregisterDestinationAmiAndDeleteSnapshots - Deregisters the copied AMI and deletes associated snapshots.
- branchAssertSourceAmiTodelete - Branches based on the value you provided for the `KeepImageSourceRegion` parameter.
- deregisterSourceAmiAndDeleteSnapshots - Deregisters the AMI created from the source instance and deletes associated snapshots.
- sleep - Sleeps the automation for 2 seconds. This is a terminal step.

Outputs

sameRegionLaunchInstanceWithKeyPair.InstanceIds

sameRegionLaunchInstanceWithoutKeyPair.InstanceIds

destinationRegionLaunchInstance.DestinationInstanceId

AWSSupport-EnableWindowsEC2SerialConsole

Description

The runbook `AWSSupport-EnableWindowsEC2SerialConsole` helps you enable Amazon EC2 Serial Console, Special Admin Console (SAC), and boot menu on your Amazon EC2 Windows instance. With Amazon Elastic Compute Cloud (Amazon EC2) Serial Console feature, you have access to your Amazon EC2 instance's serial port to troubleshoot boot, network configuration, and other issues. The runbook automates the steps required to enable the feature on instances in running state and managed by Amazon Systems Manager, as well as ones in stopped state or not managed by Amazon Systems Manager.

How does it work?

The `AWSSupport-EnableWindowsEC2SerialConsole` automation runbook helps to enable SAC and boot menu on Amazon EC2 instances running Microsoft Windows Server. For instances in running state and managed by Amazon Systems Manager, the runbook runs an Amazon Systems Manager Run Command PowerShell script to enable SAC and boot menu. For instances in stopped state or not managed by Amazon Systems Manager, the runbook uses the AWSSupport-StartEC2RescueWorkflow to create a temporary Amazon EC2 instance to perform the required changes offline.

For more information see Amazon EC2 Serial Console for Windows instances.

Important

- If you enable SAC on an instance, the Amazon EC2 services that rely on password retrieval will not work from the Amazon EC2 console. For more information, see Use SAC to troubleshoot your Windows instance.
- To configure access to the serial console, you must grant serial console access at the account level and then configure Amazon Identity and Access Management (IAM) policies to grant access to your users. You must also configure a password-based user on every instance so that your users can use the serial console for troubleshooting. For more information see Configure access to the Amazon EC2 Serial Console.
- To see if the serial console is enabled on your account see View account access status to the serial console.
- Serial console access is only supported on virtualized instances built on the Nitro System.

For more information, see the Amazon EC2 Serial Console Prerequisites.

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingInstances",
                "ec2:GetSerialConsoleAccessStatus",
                "ec2:Describe*",
                "ec2:createTags",
                "ec2:createImage",
                "ssm:DescribeAutomationExecutions",
                "ssm:DescribeInstanceInformation",
                "ssm:GetAutomationExecution",
                "ssm:ListCommandInvocations",
                "ssm:ListCommands"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
```

```
                        "ec2:DetachVolume",
                        "ec2:ModifyInstanceAttribute",
                        "ec2:RebootInstances",
                        "ec2:StartInstances",
                        "ec2:StopInstances",
                        "iam:GetInstanceProfile",
                        "ssm:GetParameters",
                        "ssm:SendCommand",
                        "ssm:StartAutomationExecution"
                    ],
                    "Resource": [
                        "arn:${Partition}:ec2:${Region}:${AccountId}:instance/
${InstanceId}",
                        "arn:${Partition}:ec2:${Region}:${AccountId}:volume/
${VolumeId}",
                        "arn:${Partition}:iam::${AccountId}:instance-profile/
${InstanceProfileName}",
                        "arn:${Partition}:ssm:${Region}::parameter/aws/service/*",
                        "arn:${Partition}:ssm:${Region}::automation-definition/
AWSSupport-StartEC2RescueWorkflow:*",
                        "arn:${Partition}:ssm:${Region}::document/AWS-
ConfigureAWSPackage",
                        "arn:${Partition}:ssm:${Region}::document/AWS-
RunPowerShellScript"
                    ]
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "cloudformation:CreateStack"
                    ],
                    "Resource": "*",
                    "Condition": {
                        "StringLike": {
                            "aws:RequestTag/Name": "AWSSupport-EC2Rescue: *"
                        },
                        "ForAllValues:StringEquals": {
                            "aws:TagKeys": [
                                "AWSSupport-EC2Rescue-AutomationExecution",
                                "Name"
                            ]
                        }
                    }
                },
```

```
                {
                    "Effect": "Allow",
                    "Action": [
                        "cloudformation:DeleteStack",
                        "cloudformation:DescribeStackEvents",
                        "cloudformation:DescribeStackResource",
                        "cloudformation:DescribeStacks",
                        "ec2:AttachVolume",
                        "ec2:DetachVolume",
                        "ec2:RebootInstances",
                        "ec2:StopInstances",
                        "ec2:TerminateInstances",
                        "ssm:SendCommand"
                    ],
                    "Resource": "*",
                    "Condition": {
                        "StringLike": {
                            "aws:ResourceTag/Name": "AWSSupport-EC2Rescue: *"
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "ec2:CreateLaunchTemplate",
                        "ec2:DeleteLaunchTemplate",
                        "ec2:RunInstances"
                    ],
                    "Resource": "*",
                    "Condition": {
                        "ForAnyValue:StringEquals": {
                            "aws:CalledVia": [
                                "cloudformation.amazonaws.com"
                            ]
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "iam:PassRole"
                    ],
                    "Resource": "*",
                    "Condition": {
```

```
                    "StringLikeIfExists": {
                        "iam:PassedToService": [
                            "ssm.amazonaws.com",
                            "ec2.amazonaws.com"
                        ]
                    }
                }
            }
        ]
    }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to the `AWSSupport-EnableWindowsEC2SerialConsole` in the Amazon Systems Manager console.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **InstanceId: (Required)**

     The ID of the Amazon EC2 instance that you want to enable Amazon EC2 serial console, (SAC), and boot menu.
   - **AutomationAssumeRole: (Optional)**

     The Amazon Resource Name (ARN) of the IAM role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **HelperInstanceType: (Conditional)**

     The type of Amazon EC2 instance that the runbook provisions to configure Amazon EC2 serial console for an offline instance.
   - **HelperInstanceProfileName: (Conditional)**

     The name of an existing IAM instance profile for the helper instance. If you are enabling SAC and boot menu on an instance that is in stopped state or not managed by Amazon Systems Manager, this is required. If an IAM instance profile is not specified, the automation creates one on your behalf.

- **SubnetId: (Conditional)**

  The subnet ID for a helper instance. By default, it uses the the same subnet where the provided instance resides.

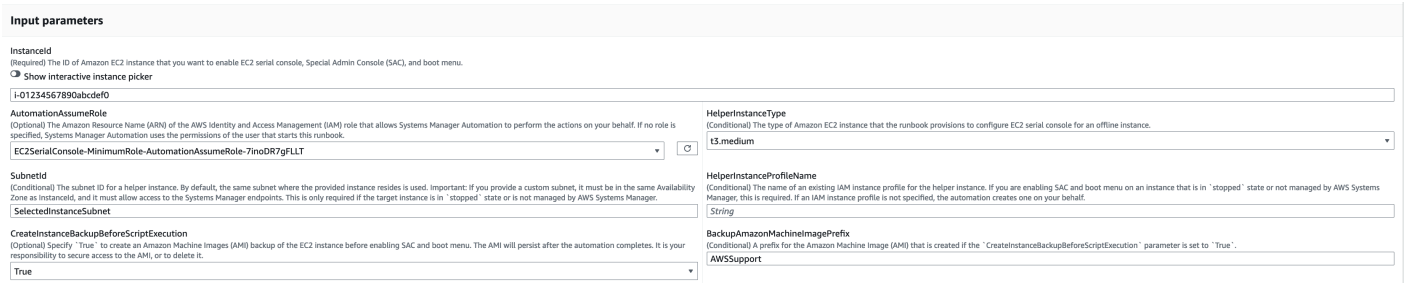
  > **Important**
  >
  > If you provide a custom subnet, it must be in the same Availability Zone as InstanceId, and it must allow access to the Systems Manager endpoints. This is only required if the target instance is in stopped state or is not managed by Amazon Systems Manager.

- **CreateInstanceBackupBeforeScriptExecution: (Optional)**

  Specify True to create an Amazon Machine Images (AMI) backup of the Amazon EC2 instance before enabling SAC and boot menu. The AMI will persist after the automation completes. It is your responsibility to secure access to the AMI or delete it.

- **BackupAmazonMachineImagePrefix: (Conditional)**

  A prefix for the Amazon Machine Image (AMI) that is created if the `CreateInstanceBackupBeforeScriptExecution` parameter is set to `True`.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

   - **CheckIfEc2SerialConsoleAccessEnabled:**

     Checks if Amazon EC2 Serial Console access is enabled at the account level. Note: Access to the serial console is not available by default. For more information see Configure access to the Amazon EC2 Serial Console.

   - **CheckIfEc2InstanceIsWindows:**

     Asserts if the target instance platform is Windows.

- **GetInstanceType:**

  Retrieves the instance type of the target instance.

- CheckIfInstanceTypeIsNitro:

  Checks if the instance type hypervisor is Nitro-based. Serial Console Access is only supported on virtualized instances built on the Nitro System.

- **CheckIfInstanceIsInAutoScalingGroup:**

  Checks if the Amazon EC2 instance is part of an Amazon EC2 Auto Scaling group by calling the `DescribeAutoScalingInstances` API. If the instance is part of an Amazon EC2 Auto Scaling group, it ensures that the Porting Assistant for .NET instance is in Standby lifecycle state.

- **WaitForEc2InstanceStateStablized:**

  Waits for the instance to be in running or stopped state.

- **GetEc2InstanceState:**

  Gets the current state of the instance.

- **BranchOnEc2InstanceState:**

  Branches based on the instance state retrieved in the previous step. If that instance state is running, it goes to the `CheckIfEc2InstanceIsManagedBySSM` step and if not, it goes to the `CheckIfHelperInstanceProfileIsProvided` step.

- **CheckIfEc2InstanceIsManagedBySSM:**

  Checks if the instance is managed by Amazon Systems Manager. If managed, the runbook enables SAC and boot menu using a PowerShell Run Command.

- **BranchOnPreEC2RescueBackup:**

  Branches based on the `CreateInstanceBackupBeforeScriptExecution` input parameter.

- **CreateAmazonMachineImageBackup:**

  Creates an AMI backup of the instance.

- **EnableSACAndBootMenu:**

  Enables SAC and boot menu by running a PowerShell Run Command script.

- **RebootInstance:**

  Reboots the Amazon EC2 instance to apply the configuration. This is the final step if the instance is online and is managed by Amazon Systems Manager.

- **CheckIfHelperInstanceProfileIsProvided:**

  Checks if the `HelperInstanceProfileName` specified exists before enabling SAC and boot menu offline using a temporary Amazon EC2 instance.

- **RunAutomationToInjectOfflineScriptForEnablingSACAndBootMenu:**

  Runs the `AWSSupport-StartEC2RescueWorkflow` to enable SAC and boot menu when the instance is in stopped state or not managed by Amazon Systems Manager.

- **GetExecutionDetails:**

  Retrieves Image ID of backup and offline script output.

7. After completed, review the Outputs section for the detailed results of the execution:

- **EnableSACAndBootMenu.Output:**

  Output of the command execution in the `EnableSACAndBootMenu` step.

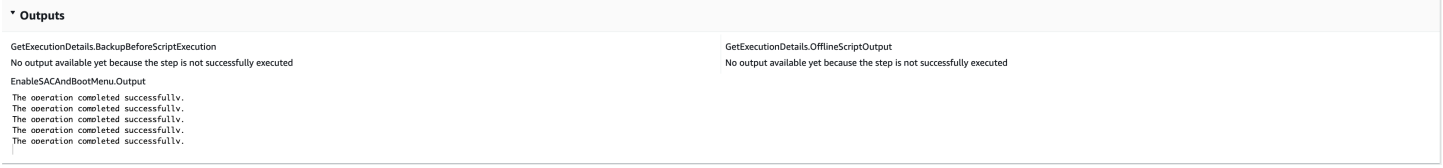
- **GetExecutionDetails.OfflineScriptOutput:**

  Output of the offline script executed in the `RunAutomationToInjectOfflineScriptForEnablingSACAndBootMenu` step.

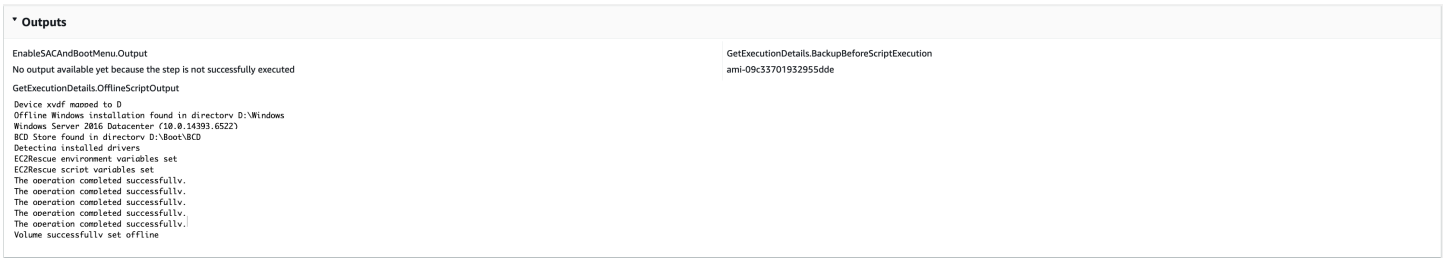
- **GetExecutionDetails.BackupBeforeScriptExecution:**

  Image ID of the AMI backup taken if `CreateInstanceBackupBeforeScriptExecution` input parameter is True.

Output of execution on an instance that is running and managed by Amazon Systems Manager



Output of execution on an instance that is stopped or not managed by Amazon Systems Manager



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-ExecuteEC2Rescue

Description

This runbook uses the EC2Rescue tool to troubleshoot and, where possible, repair common connectivity issues with the specified Amazon Elastic Compute Cloud (Amazon EC2) instance for Linux or Windows Server. Instances with encrypted root volumes are not supported.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EC2RescueInstanceType

  Type: String

  Valid values: t2.small | t2.medium | t2.large

  Default: t2.small

  Description: (Required) The EC2 instance type for the EC2Rescue instance. Recommended size: `t2.small`

- LogDestination

  Type: String

  Description: (Optional) Amazon S3 bucket name in your account where you want to upload the troubleshooting logs. Make sure the bucket policy does not grant unnecessary read/write permissions to parties that do not need access to the collected logs.

- SubnetId

  Type: String

  Default: CreateNewVPC

  Description: (Optional) The subnet ID for the EC2Rescue instance. By default, Amazon Systems Manager Automation creates a new VPC. Alternatively, use `SelectedInstanceSubnet` to use the same subnet as your instance, or specify a custom subnet ID.

  > **Important**
  >
  > The subnet must be in the same Availability Zone as `UnreachableInstanceId`, and it must allow access to the SSM endpoints.

- UnreachableInstanceId

  Type: String

Description: (Required) ID of your unreachable EC2 instance.

> **Important**
>
> Systems Manager Automation stops this instance, and creates an AMI before attempting any operations. Data stored in instance store volumes will be lost. The public IP address will change if you are not using an Elastic IP address.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

You must have at least `ssm:StartAutomationExecution` and `ssm:GetAutomationExecution` to be able to read the automation output. For more information about the required permissions, see `AWSSupport-StartEC2RescueWorkflow`.

Document Steps

1. `aws:assertAwsResourceProperty` - Asserts if the provided instance is Windows Server:
   a. (EC2Rescue for Windows Server) If the provided instance is a Windows Server instance:
      i. `aws:executeAutomation` - Invokes `AWSSupport-StartEC2RescueWorkflow` with the EC2Rescue for Windows Server offline script.
      ii. `aws:executeAwsApi` - Retrieves the backup AMI ID from the nested automation.
      iii. `aws:executeAwsApi` - Retrieves the EC2Rescue summary from the nested automation.
   b. (EC2Rescue for Linux) If the provided instance is a Linux instance:
      i. `aws:executeAutomation` - Invokes `AWSSupport-StartEC2RescueWorkflow` with the EC2Rescue for Linux offline scripts
      ii. `aws:executeAwsApi` - Retrieves the backup AMI ID from the nested automation.
      iii. `aws:executeAwsApi` - Retrieves the EC2Rescue summary from the nested automation.

Outputs

`getEC2RescueForWindowsResult.Output`

`getWindowsBackupAmi.ImageId`

`getEC2RescueForLinuxResult.Output`

`getLinuxBackupAmi.ImageId`

AWSSupport-ListEC2Resources

Description

The `AWSSupport-ListEC2Resources` runbook returns information about Amazon EC2 instances and related resources like Amazon Elastic Block Store (Amazon EBS) volumes, Elastic IP addresses, and Amazon EC2 Auto Scaling groups from the Amazon Web Services Regions you specify.
By default, the information is gathered from all Regions and is displayed in the output of the automation. Optionally, you can specify an Amazon Simple Storage Service (Amazon S3) bucket for the information to be uploaded to as a comma-separated values (.csv) file.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BucketName

  Type: String

Description: (Optional) The name of the S3 bucket where the information gathered is uploaded to.

- DisplayResourceDeletionDocumentation

  Type: String

  Default: true

  Description: (Optional) If set to `true`, the automation creates links in the output to documentation related to deleting your resources.

- RegionsToQuery

  Type: String

  Default: All

  Description: (Optional) The Regions you want to gather Amazon EC2 related information from.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `autoscaling:DescribeAutoScalingGroups`
- `ec2:DescribeAddresses`
- `ec2:DescribeImages`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRegions`
- `ec2:DescribeVolumes`
- `ec2:DescribeSnapshots`
- `elasticloadbalancing:DescribeLoadBalancers`

Additionally, to successfully upload the information gathered to the S3 bucket you specify, the `AutomationAssumeRole` requires the following actions:

- s3:GetBucketAcl
- s3:GetBucketPolicyStatus
- s3:PutObject

Document Steps

- aws:executeAwsApi - Gathers the Regions enabled for the account.
- aws:executeScript - Confirms the Regions enabled for the account support the Regions specified in the RegionsToQuery parameter.
- aws:branch - If no Regions are enabled for the account, the automation ends.
- aws:executeScript - Lists all EC2 instances for the account and Regions you specify.
- aws:executeScript - Lists all Amazon Machine Images (AMI) for the account and Regions you specify.
- aws:executeScript - Lists all EBS volumes for the account and Regions you specify.
- aws:executeScript - Lists all Elastic IP addresses for the account and Regions you specify.
- aws:executeScript - Lists all elastic network interfaces for the account and Regions you specify.
- aws:executeScript - Lists all Auto Scaling groups for the account and Regions you specify.
- aws:executeScript - Lists all load balancers for the account and Regions you specify.
- aws:executeScript - Uploads the information gathered to the S3 bucket specified if you provide a value for the Bucket parameter.

AWSSupport-ManageRDPSettings

Description

The AWSSupport-ManageRDPSettings runbook allows the user to manage common Remote Desktop Protocol (RDP) settings, such as the RDP port and Network Layer Authentication (NLA). By default, the runbook reads and outputs the values of the settings.

> **Important**
>
> Changes to the RDP settings should be carefully reviewed before running this runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the managed instance to manage the RDP settings of.
- NLASettingAction

  Type: String

  Valid values: Check | Enable | Disable

  Default: Check

  Description: (Required) An action to perform on the NLA setting: Check, Enable, Disable.
- RDPPort

  Type: String

  Default: 3389

Description: (Optional) Specify the new RDP port. Used only when the action is set to Modify. The port number must be between 1025-65535. Note: After the port is changed, the RDP service is restarted.

- RDPPortAction

  Type: String

  Valid values: Check | Modify

  Default: Check

  Description: (Required) An action to apply to the RDP port.

- RemoteConnections

  Type: String

  Valid values: Check | Enable | Disable

  Default: Check

  Description: (Required) An action to perform on the fDenyTSConnections setting.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

The EC2 instance receiving the command must have an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. The user must have at least **ssm:SendCommand** to send the command to the instance, plus **ssm:GetCommandInvocation** to be able to read the command output.

Document Steps

`aws:runCommand` - Run the PowerShell script to change or check the RDP settings on the target instance.

Outputs

manageRDPSettings.Output

AWSSupport-ManageWindowsService

Description

The AWSSupport-ManageWindowsService runbook enables you to stop, start, restart, pause, or disable any Windows service on the target instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the managed instance to manage the services of.
- ServiceAction

  Type: String

  Valid values: Check | Restart | Force-Restart | Start | Stop | Force-Stop | Pause

  Default: Check

Description: (Required) An action to apply to the Windows service. Note that `Force-Restart` and `Force-Stop` can be used to restart and to stop a service that has dependent services.

- StartupType

  Type: String

  Valid values: Check | Auto | Demand | Disabled | DelayedAutoStart

  Default: Check

  Description: (Required) A startup type to apply to the Windows service.
- WindowsServiceName

  Type: String

  Description: (Required) A valid Windows service name.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

It is recommended that the EC2 instance receiving the command has an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. The user must have at least **ssm:StartAutomationExecution** and **ssm:SendCommand** to run the automation and send the command to the instance, plus **ssm:GetAutomationExecution** to be able to read the automation output.

Document Steps

`aws:runCommand` - Run the PowerShell script to apply the desired configuration to the Windows service on the target instance.

Outputs

manageWindowsService.Output

AWSSupport-MigrateEC2ClassicToVPC

Description

The `AWSSupport-MigrateEC2ClassicToVPC` runbook migrates an Amazon Elastic Compute Cloud (Amazon EC2) instance from EC2-Classic to a virtual private cloud (VPC). This runbook supports migrating Amazon EC2 instances of the hardware virtual machine (HVM) virtualization type with Amazon Elastic Block Store (Amazon EBS) root volumes.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ApproverIAM

  Type: StringList

  Description: (Optional) The Amazon Resource Names (ARNs) of IAM users who can approve or deny the action. This parameter only applies if you specify the `CutOver` value for the `MigrationType` parameter.
- DestinationSecurityGroupId

  Type: StringList

  Description: (Optional) The ID of the security group that you want to associate with the Amazon EC2 instance that is launched in your VPC. If you don't specify a value for this parameter, the automation creates a security group in your VPC and copies the rules from the security group in

EC2-Classic. If the rules fail to copy to the new security group, the default security group of your VPC is associated with the Amazon EC2 instance.

- DestinationSubnetId

  Type: String

  Description: (Optional) The ID of the subnet that you want to migrate your Amazon EC2 instance to. If you do not specify a value for this parameter, the automation randomly chooses a subnet from your VPC.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance that you want to migrate.

- MigrationType

  Type: String

  Valid values: CutOver | Test

  Description: (Required) The type of migration that you want to perform.

  The `CutOver` option requires approval to stop your Amazon EC2 instance that's running in EC2-Classic. After this action is approved, the Amazon EC2 instance is stopped and the automation creates an Amazon Machine Image (AMI). When the AMI status is `available`, a new Amazon EC2 instance is launched from this AMI in the `DestinationSubnetId` you specify in your VPC. If your Amazon EC2 instance that's running in EC2-Classic has an Elastic IP address attached, the instance will be moved to the newly created Amazon EC2 instance in your VPC. If the Amazon EC2 instance launching in your VPC fails to create for any reason, it is terminated and approval is requested to start your Amazon EC2 instance in EC2-Classic.

  The `Test` option creates an AMI of your Amazon EC2 instance that's running in EC2-Classic without rebooting. Because the Amazon EC2 instance does not reboot, we can't guarantee the file system integrity of the created image. When the AMI status is `available`, a new Amazon EC2 instance is launched from this AMI in the `DestinationSubnetId` that you specify in your VPC. If your Amazon EC2 instance that's running in EC2-Classic has an Elastic IP address attached, the automation verifies that the `DestinationSubnetId` you specify is public. If the Amazon EC2 instance launching in your VPC fails to create for any reason, it is terminated and the automation ends.

- SNSNotificationARNforApproval

  Type: String

  Description: (Optional) The ARN of the Amazon Simple Notification Service (Amazon SNS) topic that you want to send approval requests to. This parameter only applies if you specify the `CutOver` value for the `MigrationType` parameter.
- TargetInstanceType

  Type: String

  Default: t2.2xlarge

  Description: (Optional) The type of Amazon EC2 instance that you want to launch in your VPC. Only Xen-based instance types, such as T2, M4, or C4, are supported.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetDocument`
- `ssm:ListDocumentVersions`
- `ssm:ListDocuments`
- `ssm:StartAutomationExecution`
- `sns:GetTopicAttributes`
- `sns:ListSubscriptions`
- `sns:ListTopics`
- `sns:Publish`
- `ec2:AssociateAddress`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateImage`
- `ec2:CreateSecurityGroup`
- `ec2:DeleteSecurityGroup`
- `ec2:MoveAddressToVpc`

- ec2:RunInstances
- ec2:StopInstances
- ec2:CreateTags
- ec2:DescribeAddresses
- ec2:DescribeInstanceAttribute
- ec2:DescribeInstances
- ec2:DescribeInstanceStatus
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroupReferences
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeTags
- ec2:DescribeVpcs
- ec2:DescribeInstanceTypes
- ec2:DescribeImages

Document Steps

- aws:executeAwsApi - Gathers details about the Amazon EC2 instance that you specify in the InstanceId parameter.
- aws:assertAwsResourceProperty - Confirms the instance type that you specify in the TargetInstanceType parameter is Xen-based.
- aws:assertAwsResourceProperty - Confirms the Amazon EC2 instance that you specify in the InstanceId parameter is of the HVM virtualization type.
- aws:assertAwsResourceProperty - Confirms the Amazon EC2 instance that you specify in the InstanceId parameter has an Amazon EBS root volume.
- aws:executeScript - Creates a security group as needed depending on the value that you specify for the DestinationSecurityGroupId parameter.
- aws:branch - Branches based on the value that you specify in the DestinationSubnetId parameter.
- aws:executeAwsApi - Identifies the default VPC in the Amazon Web Services Region where you run this automation.

- aws:executeAwsApi - Randomly chooses the ID of a subnet located in the default VPC.
- aws:createImage - Creates an AMI without rebooting the Amazon EC2 instance.
- aws:branch - Branches based on the value that you specify for the MigrationType parameter.
- aws:branch - Branches based on the value that you specify for the DestinationSubnetId parameter.
- aws:runInstances - Launches a new instance from the AMI created without rebooting the Amazon EC2 instance in EC2-Classic.
- aws:changeInstanceState - Terminates the newly launched Amazon EC2 instance if the previous step fails for any reason.
- aws:runInstances - Launches a new instance from the AMI created without rebooting the Amazon EC2 instance in EC2-Classic in the DestinationSubnetId if provided.
- aws:changeInstanceState - Terminates the newly launched Amazon EC2 instance if the previous step fails for any reason.
- aws:assertAwsResourceProperty - Confirms the stop behavior for the Amazon EC2 instance running in EC2-Classic.
- aws:approve - Waits for approval to stop the Amazon EC2 instance.
- aws:changeInstanceState - Stops the Amazon EC2 instance running in EC2-Classic.
- aws:changeInstanceState - Force stops the Amazon EC2 instance running in EC2-Classic if needed.
- aws:createImage - Creates an AMI of the Amazon EC2 instance after it has stopped.
- aws:branch - Branches based on the value specified for the DestinationSubnetId parameter.
- aws:runInstances - Launches a new instance from the AMI created of the stopped Amazon EC2 instance in EC2-Classic.
- aws:approve - Waits for approval to terminate the newly launched instance and starts the Amazon EC2 instance in EC2-Classic if the previous step fails for any reason.
- aws:changeInstanceState - Terminates the newly launched Amazon EC2 instance.
- aws:runInstances - Launches a new instance from the AMI created of the stopped Amazon EC2 instance in EC2-Classic from the DestinationSubnetId parameter.
- aws:approve - Waits for approval to terminate the newly launched instance and starts the Amazon EC2 instance in EC2-Classic if the previous step fails for any reason.
- aws:changeInstanceState - Terminates the newly launched Amazon EC2 instance.

- aws:changeInstanceState - Starts the Amazon EC2 instance that was stopped in EC2-Classic.
- aws:branch - Branches based on whether the Amazon EC2 instance has a public IP address.
- aws:executeAwsApi - Verifies whether the public IP address is an Elastic IP address.
- aws:branch - Branches based on the value that you specify in the MigrationType parameter.
- aws:executeAwsApi - Moves the Elastic IP address to your VPC.
- aws:executeAwsApi - Gathers the allocation ID of the Elastic IP address that was moved to your VPC.
- aws:branch - Branches based on which subnet the Amazon EC2 instance running in your VPC was launched.
- aws:executeAwsApi - Attaches the Elastic IP address to the newly launched instance in your VPC.
- aws:executeScript - Confirms the subnet your newly launched Amazon EC2 instance running in your VPC is public.

Outputs

getInstanceProperties.virtualizationType - The virtualization type of the Amazon EC2 instance running in EC2-Classic.

getInstanceProperties.rootDeviceType - The root device type of the Amazon EC2 instance running in EC2-Classic.

createAMIWithoutReboot.ImageId - The ID of the AMI created without rebooting the Amazon EC2 instance running in EC2-Classic.

getDefaultVPC.VpcId - The ID of the default VPC where the new Amazon EC2 instance is launched if a value for the DestinationSubnetId parameter is not provided.

getSubnetIdinDefaultVPC.subnetIdFromDefaultVpc - The ID of the subnet in the default VPC where the new Amazon EC2 instance is launched if a value for the DestinationSubnetId parameter is not provided.

launchTestInstanceDefaultVPC.InstanceIds - The ID of the newly launched Amazon EC2 instance in your default VPC during the Test migration type.

launchTestInstanceProvidedSubnet.InstanceIds - The ID of the newly launched Amazon EC2 instance in the DestinationSubnetId that you specified during the Test migration type.

`createAMIAfterStoppingInstance.ImageId` - The ID of the AMI created after stopping the Amazon EC2 instance running in EC2-Classic.

`launchCutOverInstanceProvidedSubnet.InstanceIds` - The ID of the newly launched Amazon EC2 instance in the `DestinationSubnetId` that you specified during the `CutOver` migration type.

`launchCutOverInstanceDefaultVPC.InstanceIds` - The ID of the newly launched Amazon EC2 instance in your default VPC during the `CutOver` migration type.

`verifySubnetIsPublicTestDefaultVPC.IsSubnetPublic` - Whether the subnet chosen by the automation in your default VPC is public.

`verifySubnetIsPublicTestProvidedSubnet.IsSubnetPublic` - Whether the subnet you specified in the `DestinationSubnetId` is public.

AWSSupport-MigrateXenToNitroLinux

Description

The `AWSSupport-MigrateXenToNitroLinux` runbook clones, prepares, and migrates an Amazon Elastic Compute Cloud (Amazon EC2) Linux Xen instance to a Nitro instance type. This runbook provide two options for operation types:

- `Clone&Migrate` – This option's workflow consists of the **Preliminary Checks**, **Testing**, and **Clone&Migrate** phases. The workflow is run using the `AWSSupport-CloneXenEC2InstanceAndMigrateToNitro` runbook.
- `FullMigration` – This option runs the `Clone&Migrate` workflow and then performs the additional step of **Replace root Amazon EBS volumes**.

> **Important**
>
> Using this runbook incurs costs to your account for the running time of Amazon EC2 instances, creation of Amazon Elastic Block Store (Amazon EBS) volumes, and AMIs. For more details, see Amazon EC2 Pricing and Amazon EBS Pricing.

Preliminary checks

The automation performs the following preliminary checks before continuing with the migration. If any of the checks fail, the automation ends. This phase is only part of the `Clone&Migrate` workflow.

- Checks if the target instance is already a Nitro instance type.
- Checks if the Spot Instances purchasing option was used for the target instance.
- Checks if instance store volumes are attached to the target instance.
- Verifies the target instance operating system (OS) is Linux.
- Checks if the target instance is a part of an Amazon EC2 Auto Scaling group. If it is part of an Auto Scaling group, the automation verifies that the instance is in the `standby` state.
- Verifies that the instance is managed by Amazon Systems Manager.

Testing

The automation creates an Amazon Machine Image (AMI) from the target instance and launches a test instance from the newly created AMI. This phase is part of only the `Clone&Migrate` workflow.

If the test instance passes all status checks, the automation pauses and approval from the designated principals is requested through Amazon Simple Notification Service (Amazon SNS) notification. If approval is provided, the automation terminates the test instance, stops the target instance, and continues with the migration, while the newly created AMI is deregistered at the end of the `Clone&Migrate` workflow.

> **Note**
>
> Before providing approval, we recommend verifying that all applications running on the target instance have been closed gracefully.

Clone and Migrate

The automation creates another AMI from the target instance, and launches a new instance to change to a Nitro instance type. The automation completes the following prerequisites before continuing with the migration. If any of the checks fail, the automation ends. This phase is also only part of the `Clone&Migrate` workflow.

- Turns on the enhanced networking (ENA) attribute.

- Installs the latest version of ENA drivers if they're not already installed, or updates the ENA drivers version to the latest version. To ensure maximum network performance, updating to the latest ENA driver version is required if the Nitro instance type is the 6th generation.
- Verifies that the NVMe module is installed. If the module is installed, the automation verifies that the module is loaded in `initramfs`.
- Analyzes `/etc/fstab` and replaces entries with block device names (`/dev/sd*` or `/dev/xvd*`) with their respective UUIDs. Before modifying the configuration, the automation creates a backup of the file at the path `/etc/fstab*`.
- Turns off predictable interface naming by adding the `net.ifnames=0` option to the `GRUB_CMDLINE_LINUX` line in the `/etc/default/grub` file if it exists, or to the kernel in `/boot/grub/menu.lst`.
- Removes the `/etc/udev/rules.d/70-persistent-net.rules` file if it exists. Before removing the file, the automation creates a backup of the file at the path `/etc/udev/rules.d/`.

After verifying all requirements, the instance type is changed to the Nitro instance type that you specify. The automation waits for the newly created instance to pass all status checks after starting as a Nitro instance type. The automation then waits for approval from the designated principals to create an AMI of the successfully launched Nitro instance. If approval is denied, the automation ends, leaving the newly created instance running, and the target instance remains stopped.

Replace root Amazon EBS Volume

If you choose `FullMigration` as the `OperationType`, the automation migrates the target Amazon EC2 instance to the Nitro instance type that you specify. Automation requests approval from designated principals to replace the root Amazon EBS volume of the target Amazon EC2 instance with the cloned Amazon EC2 instance's root volume. After the migration is successful, the cloned Amazon EC2 instance is terminated. If the automation fails, the original Amazon EBS root volume is attached to the target Amazon EC2 instance. If the root Amazon EBS volume attached to the target Amazon EC2 instance has tags with the `aws:` prefix applied, the `FullMigration` operation isn't supported.

Before you begin

The target instance must have outbound internet access. This is to access repositories for drivers and dependencies like kernel-devel, gcc, patch, rpm-build, wget, dracut, make, linux-headers, and unzip. Package manager is used if needed.

An Amazon SNS topic is required to send notifications for approvals and updates. For more information about creating an Amazon SNS topic, see Creating an Amazon SNS topic in the *Amazon Simple Notification Service Developer Guide*.

This runbook supports the following operating systems:

- RHEL 7.x - 8.5
- Amazon Linux (2018.03), Amazon Linux 2
- Debian Server
- Ubuntu Server 18.04 LTS, 20.04 LTS, and 20.10 STR
- SUSE Linux Enterprise Server (SUSE12SP5, SUSE15SP2)

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Acknowledgement

  Type: String

Description: (Required) Read the complete details of the actions performed by this automation runbook, and enter **`Yes, I understand and acknowledge`** to proceed with using the runbook.

- ApproverIAM

  Type: String

  Description: (Required) The ARNs of IAM roles, users, or user names who can provide approvals to the automation. You can specify a maximum of 10 approvers.

- DeleteResourcesOnFailure

  Type: Boolean

  Description: (Optional) Determines whether the newly created instance and AMI for the migration are deleted if the automation fails.

  Valid values: True | False

  Default: True

- MinimumRequiredApprovals

  Type: String

  Description: (Optional) The minimum number of approvals required to continue running the automation when approvals are requested.

  Valid values: 1-10

  Default: 1

- NitroInstanceType

  Type: String

  Description: (Required) The Nitro instance type that you want to change the instance to. Supported instance types include M5, M6, C5, C6, R5, R6, and T3.

  Default: m5.xlarge

- OperationType

Type: String

Description: (Required) The operation that you want to perform. The `FullMigration` option performs the same tasks as `Clone&Migrate` and additionally replaces the root volume of your target instance. The root volume of the target instance is replaced with the root volume from the newly created instance following the migration process. The `FullMigration` operation does not support root volumes defined by Logical Volume Manager (LVM).

Valid values: Clone&Migrate | FullMigration

- SNSTopicArn

  Type: String

  Description: (Required) The ARN of the Amazon SNS topic for approval notification. The Amazon SNS topic is used to send required approval notifications during the automation.

- TargetInstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instances to migrate.

Clone&Migrate workflow

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DescribeAutomationExecutions`
- `ssm:StartAutomationExecution`
- `ssm:DescribeInstanceInformation`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:SendCommand`
- `ssm:GetAutomationExecution`
- `ssm:ListCommands`
- `ssm:ListCommandInvocations`
- `ec2:DescribeInstances`

- ec2:DescribeInstanceTypeOfferings
- ec2:DescribeInstanceTypes
- ec2:DescribeImages
- ec2:CreateImage
- ec2:RunInstances
- ec2:DescribeInstanceStatus
- ec2:DeregisterImage
- ec2:DeleteSnapshot
- ec2:TerminateInstances
- ec2:StartInstances
- ec2:DescribeKeyPairs
- ec2:StopInstances
- kms:CreateGrant*
- kms:ReEncrypt
- ec2:ModifyInstanceAttribute
- autoscaling:DescribeAutoScalingInstances
- iam:passRole
- iam:ListRoles

Document Steps

- startOfPreliminaryChecksBranch - Branches to the Preliminary checks workflow.
- getTargetInstanceProperties - Gathers details from the target instance.
- checkIfNitroInstanceTypeIsSupportedInAZ - Determines if the target Amazon EC2 instance type is supported in the same Availability Zone as the target instance.
- getXenInstanceTypeDetails - Gathers details about the source instance type.
- checkIfInstanceHypervisorIsNitroAlready - Checks if the target instance is already running as a Nitro instance type.
- checkIfTargetInstanceLifecycleIsSpot - Checks if the purchasing option of the target instance is Spot.
- checkIfOperatingSystemIsLinux - Checks if the target instance OS is Linux.

- `verifySSMConnectivityForTargetInstance` - Verifies that the target instance is managed by Systems Manager.
- `checkIfEphemeralVolumeAreSupported` - Checks if the current instance type of the target instance supports instance store volumes.
- `verifyIfTargetInstanceHasEphemeralVolumesAttached` - Checks if the target instance has instance store volumes attached.
- `checkIfRootVolumeIsEBS` - Checks if the target instance's root volume type is EBS.
- `checkIfTargetInstanceIsInASG` - Checks if the target instance is a part of an Auto Scaling group.
- `endOfPreliminaryChecksBranch` - End of the Preliminary checks branch.
- `startOfTestBranch` - Branches to the Testing workflow.
- `createTestImage` - Creates a test AMI of the target instance.
- `launchTestInstanceInSameSubnet` - Launches a test instance from the test AMI using the same configuration as target instance.
- `cleanupTestInstance` - Terminates the test instance.
- `endOfTestBranch` - End of the Testing branch.
- `checkIfTestingBranchSucceeded` - Checks the status of the Testing branch.
- `approvalToStopTargetInstance` - Waits for approval from the designated principals to stop the target instance.
- `stopTargetEC2Instance` - Stops the target instance.
- `forceStopTargetEC2Instance` - Force stops the target instance only if the previous step fails to stop the instance.
- `startOfCloneAndMigrateBranch` - Branches to the Clone&Migrate workflow.
- `createBackupImage` - Creates an AMI of the target instance to serve as a backup.
- `launchInstanceInSameSubnet` - Launches a new instance from the backup AMI using the same configuration as the source instance.
- `waitForClonedInstanceToPassStatusChecks` - Waits for the newly created instance to pass all status checks.
- `verifySSMConnectivityForClonedInstance` - Verifies that the newly created instance is managed by Systems Manager.
- `checkAndInstallENADrivers` - Checks if ENA drivers are installed on the newly created instance, and installs the drivers if needed.

- checkAndAddNVMEDrivers - Checks if NVMe drivers are installed on the newly created instance, and installs the drivers if needed.
- checkAndModifyFSTABEntries - Checks if device names are used in /etc/fstab and replaces them with UUIDs if needed.
- stopClonedInstance - Stops the newly created instance.
- forceStopClonedInstance - Force stops the newly created instance only if the previous step fails to stop the instance.
- checkENAAttributeForClonedInstance - Checks if the enhanced networking attribute is turned on for the newly created instance.
- setNitroInstanceTypeForClonedInstance - Changes the instance type for the newly created instance to the Nitro instance type that you specify.
- startClonedInstance - Starts the newly created instance whose instance type you changed.
- approvalForCreatingImageAfterDriversInstallation - If the instance successfully starts as a Nitro instance type, the automation waits for approval from the required principals. If approval is provided, an AMI is created to be used as a Golden AMI.
- createImageAfterDriversInstallation - Creates an AMI to be used as a Golden AMI.
- endOfCloneAndMigrateBranch - End of Clone&Migrate branch.
- cleanupTestImage - Deregisters the AMI created for testing.
- failureHandling - Checks if you chose to terminate resources on failure.
- onFailureTerminateClonedInstance - Terminates the newly created instance if the automation fails.
- onFailurecleanupTestImage - Deregisters the AMI created for testing.
- onFailureApprovalToStartTargetInstance - If the automation fails, waits for approval from the designated principals to start the target instance.
- onFailureStartTargetInstance - If the automation fails, starts the target instance.

FullMigration workflow

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:DescribeAutomationExecutions

- ssm:DescribeInstanceInformation
- ssm:DescribeAutomationStepExecutions
- ssm:SendCommand
- ssm:GetAutomationExecution
- ssm:ListCommands
- ssm:ListCommandInvocations
- ec2:DescribeInstances
- ec2:DescribeInstanceTypeOfferings
- ec2:DescribeInstanceTypes
- ec2:DescribeImages
- ec2:CreateImage
- ec2:RunInstances
- ec2:DescribeInstanceStatus
- ec2:DeregisterImage
- ec2:DeleteSnapshot
- ec2:TerminateInstances
- ec2:StartInstances
- ec2:DescribeKeyPairs
- ec2:StopInstances
- kms:CreateGrant*
- kms:ReEncrypt
- ec2:ModifyInstanceAttribute
- ec2:DetachVolume
- ec2:AttachVolume
- ec2:DescribeVolumes
- autoscaling:DescribeAutoScalingInstances
- iam:PassRole
- ec2:CreateTags
- cloudformation:DescribeStackResources

Document Steps

The `FullMigration` workflow runs the same steps as the `Clone&Migrate` workflow and additionally performs the following steps:

- `checkConcurrency` - Verifies that there is only one automation of this runbook targeting the Amazon EC2 instance that you specify. If the runbook finds another automation in progress targeting the same instance, the automation ends.
- `getTargetInstanceProperties` - Gathers details from the target instance.
- `checkRootVolumeTags` - Determines if the root volume of the target Amazon EC2 instance contains any Amazon reserved tags.
- `cloneTargetInstanceAndMigrateToNitro` - Starts a child automation using the `AWS-CloneXenInstanceToNitro` runbook.
- `branchOnTheOperationType` - Branches on the value that you specify for the `OperationType` parameter.
- `getClonedInstanceId` - Retrieves the ID of the newly launched instance from the child automation.
- `checkIfRootVolumeIsBasedOnLVM` - Determines if the root partition is managed by LVM.
- `branchOnTheRootVolumeLVMStatus` - If the minimum required approvals are received from the principals, the automation proceeds with the root volume replacement.
- `manualInstructionsInCaseOfLVM` - If the root volume is managed by LVM, the automation sends output containing instructions for how to manually replace the root volumes.
- `startOfReplaceRootEBSVolumeBranch` - Starts the Replace Root EBS Volume branch workflow.
- `checkIfTargetInstanceIsManagedByCFN` - Determines if the target instance is managed by an Amazon CloudFormation stack.
- `branchOnCFNStackStatus` - Branches based on the status of the CloudFormation stack.
- `approvalForRootVolumesReplacement(WithCFN)` - If the target instance was launched by CloudFormation, the automation waits for approval after the newly launched instance successfully starts as a Nitro instance type. When approvals are provided, the Amazon EBS volumes of the target instance are replaced with the root volumes from the newly launched instance.
- `approvalForRootVolumesReplacement` - Waits for approval after the newly launched instance successfully starts as a Nitro instance type. When approvals are provided, the Amazon

EBS volumes of the target instance are replaced with the root volumes from the newly launched instance.

- `assertIfTargetEC2InstanceIsStillStopped` - Verifies that the target instance is in a `stopped` state before replacing the root volume.
- `stopTargetInstanceForRootVolumeReplacement` - If the target instance is running, the automation stops the instance before replacing the root volume.
- `forceStopTargetInstanceForRootVolumeReplacement` - Force stops the target instance if the previous step fails.
- `stopClonedInstanceForRootVolumeReplacement` - Stops the newly created instance before replacing the Amazon EBS volumes.
- `forceStopClonedInstanceForRootVolumeReplacement` - Force stops the newly created instance if the previous step fails.
- `getBlockDeviceMappings` - Retrieves the block device mappings for both the target and newly created instances.
- `replaceRootEbsVolumes` - Replaces the root volume of the target instance with the root volume of the newly created instance.
- `EndOfReplaceRootEBSVolumeBranch` - End of Replace Root EBS Volume branch workflow.
- `checkENAAttributeForTargetInstance` - Checks if the enhanced networking (ENA) attribute is turned on for the target Amazon EC2 instance.
- `enableENAAttributeForTargetInstance` - Turns on the ENA attribute for the target Amazon EC2 instance if needed.
- `setNitroInstanceTypeForTargetInstance` - Changes the target instance to the Nitro instance type that you specify.
- `replicateRootVolumeTags` - Replicates the tags on the root Amazon EBS volume from the target Amazon EC2 instance.
- `startTargetInstance` - Starts the target Amazon EC2 instance after changing the instance type.
- `onFailureStopTargetEC2Instance` - Stops the target Amazon EC2 instance if it fails to start as a Nitro instance type.
- `onFailureForceStopTargetEC2Instance` - Force stops the target Amazon EC2 instance if the previous step fails.
- `OnFailureRevertOriginalInstanceType` - Reverts the target Amazon EC2 instance to the original instance type if the target instance fails to start as a Nitro instance type.

- `onFailureRollbackRootVolumeReplacement` - Reverts all the changes made by the `replaceRootEbsVolumes` step if needed.
- `onFailureApprovalToStartTargetInstance` - Waits for designated principal's approval to start the target Amazon EC2 instance after rolling back the previous changes.
- `onFailureStartTargetInstance` - Starts the target Amazon EC2 instance.
- `terminateClonedEC2Instance` - Terminates the cloned Amazon EC2 instance after replacing the root Amazon EBS volume.

AWSSupport-ResetAccess

Description

This runbook will use the EC2Rescue tool on the specified EC2 instance to re-enable password decryption using the EC2 Console (Windows) or to generate and add a new SSH key pair (Linux). If you lost your key pair, this automation will create a password-enabled AMI that you can use to launch a new EC2 instance with a key pair you own (Windows).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EC2RescueInstanceType

  Type: String

  Valid values: t2.small | t2.medium | t2.large

  Default: t2.small

  Description: (Required) The EC2 instance type for the EC2Rescue instance. Recommended size: t2.small.

- InstanceId

  Type: String

  Description: (Required) ID of the EC2 instance you want to reset access for.

  > **Important**
  >
  > Systems Manager Automation stops this instance, and creates an AMI before attempting any operations. Data stored in instance store volumes will be lost. The public IP address will change if you are not using an Elastic IP.

- SubnetId

  Type: String

  Default: CreateNewVPC

  Description: (Optional) The subnet ID for the EC2Rescue instance. By default, Systems Manager Automation creates a new VPC. Alternatively, Use SelectedInstanceSubnet to use the same subnet as your instance, or specify a custom subnet ID.

  > **Important**
  >
  > The subnet must be in the same Availability Zone as InstanceId, and it must allow access to the SSM endpoints.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

You must have at least **ssm:StartAutomationExecution**, **ssm:GetParameter** (to retrieve the SSH key parameter name) and **ssm:GetAutomationExecution** to be able to read the automation output. For more information about the required permissions, see [AWSSupport-StartEC2RescueWorkflow].

Document Steps

1. `aws:assertAwsResourceProperty` - Assert if the provided instance is Windows.
   a. (EC2Rescue for Windows) If the provided instance is Windows:
      i. `aws:executeAutomation` - Invoke `AWSSupport-StartEC2RescueWorkflow` with the EC2Rescue for Windows offline password reset script
      ii. `aws:executeAwsApi` - Retrieve the backup AMI ID from the nested automation
      iii. `aws:executeAwsApi` - Retrieve the password-enabled AMI ID from the nested automation
      iv. `aws:executeAwsApi` - Retrieve the EC2Rescue summary from the nested automation
   b. (EC2Rescue for Linux) If the provided instance is Linux:
      i. `aws:executeAutomation` - Invoke `AWSSupport-StartEC2RescueWorkflow` with the EC2Rescue for Linux offline SSH key injection script
      ii. `aws:executeAwsApi` - Retrieve the backup AMI ID from the nested automation
      iii. `aws:executeAwsApi` - Retrieve the SSM parameter name for the injected SSH key
      iv. `aws:executeAwsApi` - Retrieve the EC2Rescue summary from the nested automation

Outputs

getEC2RescueForWindowsResult.Output

getWindowsBackupAmi.ImageId

getWindowsPasswordEnabledAmi.ImageId

getEC2RescueForLinuxResult.Output

getLinuxBackupAmi.ImageId

getLinuxSSHKeyParameter.Name

AWSSupport-ResetLinuxUserPassword

Description

The AWSSupport-ResetLinuxUserPassword runbook helps you reset the password of a local operating system (OS) user. This runbook is especially helpful for users who need to access their Amazon Elastic Compute Cloud (Amazon EC2) instances using the serial console. The runbook creates a temporary Amazon EC2 instance in your Amazon Web Services account and an Amazon Identity and Access Management (IAM) role with permissions to retrieve an Amazon Secrets Manager secret value containing the password.

The runbook stops your target Amazon EC2 instance, detaches the root Amazon Elastic Block Store (Amazon EBS) volume, and attaches it to the temporary Amazon EC2 instance. Using Run Command, a script runs on the temporary instance to set the password of the OS user that you specify. Then, the root Amazon EBS volume is reattached to your target instance. The runbook also provides an option to create a snapshot of the root volume at the beginning of the automation.

Before you begin

Create an Secrets Manager secret with the value of the password that you want to assign to your OS user. The value must be in plaintext. For more information, see Create an Amazon Secrets Manager secret in the *Amazon Secrets Manager User Guide*.

Considerations

- We recommend backing up your instance before using this runbook. Consider setting the value of the CreateSnapshot parameter as **Yes**.
- Changing the local user password requires the runbook to stop your instance. When an instance is stopped, any data stored in memory or on instance store volumes is lost. Also, any automatically assigned public IPv4 addresses are released. For more information about what happens when you stop an instance, see Stop and start your instance in the *Amazon EC2 User Guide*.
- If the Amazon EBS volumes attached to your target Amazon EC2 instance are encrypted with a customer managed Amazon Key Management Service (Amazon KMS) key, make sure the Amazon KMS key is not deleted or disabled or your instance will fail to start.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 Linux instance that contains the OS user password that you want to reset.
- LinuxUserName

  Type: String

  Default: ec2-user

  Description: (Optional) The OS user account whose password you want to reset.
- SecretArn

  Type: String

  Description: (Required) The ARN of your Secrets Manager secret containing the new password.
- SecurityGroupId

Type: String

Description: (Optional) The ID of the security group to attach to the temporary Amazon EC2 instance. If you don't provide a value for this parameter, the default Amazon Virtual Private Cloud (Amazon VPC) security group is used.

- SubnetId

  Type: String

  Description: (Optional) The ID of the subnet that you want to launch the Amazon EC2 temporary instance in to. By default, the automation chooses the same subnet as your target instance. If you choose to provide a different subnet, it must be in the same Availability Zone as the target instance and have access to Systems Manager endpoints.

- CreateSnapshot

  Type: String

  Valid values: Yes | No

  Default: Yes

  Description: (Optional) Determines whether a snapshot of the root volume of your target Amazon EC2 instance is created before the automation runs.

- StopConsent

  Type: String

  Valid values: Yes | No

  Default: No

  Description: Enter **`Yes`** to acknowledge that your target Amazon EC2 instance will be stopped during this automation. When the Amazon EC2 instance is stopped, any data stored in memory or instance store volumes is lost, and the automatic public IPv4 address is released. For more information, see Stop and start your instance in the *Amazon EC2 User Guide*.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DescribeInstanceInformation`
- `ssm:ListTagsForResource`
- `ssm:SendCommand`
- `ec2:AttachVolume`
- `ec2:CreateSnapshot`
- `ec2:CreateSnapshots`
- `ec2:CreateVolume`
- `ec2:DescribeImages`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`
- `ec2:DescribeSnapshotAttribute`
- `ec2:DescribeSnapshots`
- `ec2:DescribeSnapshotTierStatus`
- `ec2:DescribeVolumes`
- `ec2:DescribeVolumeStatus`
- `ec2:DetachVolume`
- `ec2:RunInstances`
- `ec2:StartInstances`
- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `cloudformation:CreateStack`
- `cloudformation:DeleteStack`
- `cloudformation:DescribeStackResource`
- `cloudformation:DescribeStacks`
- `cloudformation:ListStacks`
- `logs:CreateLogDelivery`

- logs:CreateLogGroup
- logs:DeleteLogDelivery
- logs:DeleteLogGroup
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- logs:PutLogEvents

Document Steps

1. aws:branch – Branches based on whether you have provided consent to stopping the target Amazon EC2 instance.
2. aws:assertAwsResourceProperty – Ensures the Amazon EC2 instance status is in a running or stopped state. Otherwise, the automation ends.
3. aws:executeAwsApi – Gets the Amazon EC2 instance properties.
4. aws:executeAwsApi – Gets the root volume properties.
5. aws:branch – Branches the automation depending on whether a subnet ID for the temporary Amazon EC2 instance was provided.
6. aws:assertAwsResourceProperty – Ensures the subnet that you specify in SubnetId parameter is in the same Availability Zone as the target Amazon EC2 instance.
7. aws:assertAwsResourceProperty– Ensures the target Amazon EC2 instance root volume is an Amazon EBS volume.
8. aws:assertAwsResourceProperty – Ensures the Amazon EC2 instance architecture is arm64 or x86_64.
9. aws:assertAwsResourceProperty – Ensures the Amazon EC2 instance shutdown behavior is stop and not terminate.
10. aws:branch – Ensures the Amazon EC2 instance is not a Spot Instance. Otherwise, the automation ends.
11. aws:executeScript – Ensures the Amazon EC2 instance is not part of an auto scaling group. If the instance is part of an auto scaling group, the automation confirms the Amazon EC2 instance is in a Standby lifecycle state.
12. aws:createStack – Creates a temporary Amazon EC2 instance that is used to reset the password for the OS user that you specify.

13 `aws:waitForAwsResourceProperty` – Waits until the newly launched temporary Amazon EC2 instance is running.

14 `aws:executeAwsApi` – Gets the ID of the temporary Amazon EC2 instance.

15 `aws:waitForAwsResourceProperty` – Waits for the temporary Amazon EC2 instance to report as managed by Systems Manager.

16 `aws:changeInstanceState`– Stops the target Amazon EC2 instance.

17 `aws:changeInstanceState` – Forces the target Amazon EC2 instance to stop in case it gets stuck in a stopping state.

18 `aws:branch` – Branches the automation depending on whether a snapshot of the root volume of the target Amazon EC2 instance was requested.

19 `aws:executeAwsApi` – Creates a snapshot of the target Amazon EC2 instance root Amazon EBS volume.

20 `aws:waitForAwsResourceProperty` – Waits for the snapshot to be in a `completed` state.

21 `aws:executeAwsApi` – Detaches the Amazon EBS root volume from the target Amazon EC2 instance.

22 `aws:waitForAwsResourceProperty` – Waits for the Amazon EBS root volume to be detached from the target Amazon EC2 instance.

23 `aws:executeAwsApi` – Attaches the root Amazon EBS volume to the temporary Amazon EC2 instance.

24 `aws:waitForAwsResourceProperty` – Waits for the Amazon EBS root volume to be attached to the temporary Amazon EC2 instance.

25 `aws:runCommand` – Resets the target user password by running a shell script using Run Command on the temporary Amazon EC2 instance.

26 `aws:executeAwsApi` – Detaches the Amazon EBS root volume from the temporary Amazon EC2 instance.

27 `aws:waitForAwsResourceProperty` – Waits for the Amazon EBS root volume to be detached from the temporary Amazon EC2 instance.

28 `aws:executeAwsApi` – Detaches the Amazon EBS root volume from the temporary Amazon EC2 instance after an error.

29 `aws:waitForAwsResourceProperty` – Waits for the Amazon EBS root volume to be detached from the temporary Amazon EC2 instance after an error.

30 `aws:branch` – Branches the automation depending on whether a snapshot of the root volume was requested to determine the recovery path in case of an error.

31 `aws:executeAwsApi` – Reattaches the root Amazon EBS volume to the target Amazon EC2 instance.

32 `aws:waitForAwsResourceProperty` – Waits for the Amazon EBS root volume to be attached to the Amazon EC2 instance.

33 `aws:executeAwsApi` – Creates a new Amazon EBS volume from the target Amazon EC2 instance root volume snapshot.

34 `aws:waitForAwsResourceProperty` – Waits until the new Amazon EBS volume is in an `available` state.

35 `aws:executeAwsApi` – Attaches the new Amazon EBS volume to the target instance as the root volume.

36 `aws:waitForAwsResourceProperty` – Waits for the Amazon EBS volume to be in an `attached` state.

37 `aws:executeAwsApi` – Describes the Amazon CloudFormation stack events if the runbooks fails to create or update the Amazon CloudFormation stack.

38 `aws:branch` – Branches the automation depending on the previous Amazon EC2 instance state. If the state was `running`, the instance is started. If it was in a `stopped` state, the automation continues.

39 `aws:changeInstanceState` – Starts the Amazon EC2 instance if needed.

40 `aws:waitForAwsResourceProperty` – Waits until the Amazon CloudFormation stack is in a terminal status before deleting.

41 `aws:executeAwsApi` – Deletes the Amazon CloudFormation stack including the temporary Amazon EC2 instance.

AWSPremiumSupport-ResizeNitroInstance

Description

The `AWSPremiumSupport-ResizeNitroInstance` runbook provides an automated solution for resizing Amazon Elastic Compute Cloud (Amazon EC2) instances built on the Nitro System.

To reduce the potential risk of data loss and downtime, the runbook verifies the following:

- Instance stop behavior.
- If the instance is part of an Amazon EC2 Auto Scaling group, and in `standby` mode.
- Instance state and tenancy.

- The instance type you want to change to supports the number of network interfaces currently attached to your instance.
- The processor architecture and virtualization type for both the current and target instance type are the same.
- If the instance is running, that it's passing all status checks.
- The instance type you want to change to is available in the same Availability Zone.

If the Amazon EC2 does not pass status checks after changing the instance type, the runbook automatically rolls back to the previous instance type.

By default, this runbook will not change the instance type if it is running and instance store volumes are attached. The runbook will also not change the instance type if the instance is part of an Amazon CloudFormation stack. If you want to change either of these behaviors, specify `yes` for the `AllowInstanceStoreInstances` and `AllowCloudFormationInstances` parameters.

The runbook provides two different ways to specify the instance type you want to change to:

- For simple automations targeting a single instance, specify the instance type you want to change to using the `TargetInstanceTypeFromParameter` parameter.
- For running automations at scale to change the instance type of several instances, specify the instance type using the `TargetInstanceTypeFromTagValue` parameter. For information about running automations at scale, see Run automations at scale.

If you don't specify a value for either parameter, the automation fails.

> **Important**
>
> Access to `AWSPremiumSupport-*` runbooks requires either an Enterprise or Business Support Subscription. For more information, see Compare Amazon Web Services Support Plans.

Considerations

- We recommend backing up your instance before using this runbook.
- For information about compatibility for changing instance types, see Compatibility for changing the instance type.

- If the automation fails and rolls back to the original instance type, see Troubleshoot changing the instance type.
- Changing the instance type requires the runbook to stop your instance. When an instance is stopped, any data stored in memory or on instance store volumes is lost. Also, any automatically assigned public IPv4 addresses are released. For more information about what happens when you stop an instance, see Stop and start your instance.
- By using the `SkipInstancesWithTagKey` parameter, you can skip instances that have a specific Amazon EC2 tag key applied.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Acknowledge

  Type: String

  Description: (Required) Enter **`yes`** to acknowledge that your instance will be stopped if it's currently running.

- AllowInstanceStoreInstances

  Type: String

  Valid values: no | yes

  Default: no

  Description: (Optional) If you specify `yes`, you allow the runbook to run on instances that have instance store volumes attached.

- AllowCloudFormationInstances

  Type: String

  Valid values: no | yes

  Default: no

  Description: (Optional) If you specify `yes`, the runbook runs on instances that are part of an Amazon CloudFormation stack.

- DryRun

  Type: String

  Valid values: no | yes

  Default: no

  Description: (Optional) If you specify `yes`, the runbook validates resizing requirements without making changes to the instance type.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance whose type you want to change.

- SkipInstancesWithTagKey

  Type: String

  Description: (Optional) The automation skips a target instance if the tag key you specify is applied to the instance.

- SleepTime

  Type: String

  Default: 3

  Description: (Optional) The number of seconds this runbook should sleep after completion.
- TagInstance

  Type: String

  Description: (Optional) Tag the instances with the key and value of your choice using the following format: *Key=ChangingType,Value=True*. This option allows you to track instances that have been targeted by this runbook. Tag keys and values are case sensitive.
- TargetInstanceTypeFromParameter

  Type: String

  Description: (Optional) The instance type you want to change your instance to. Leave this parameter empty if you want to use the value of the tag key provided in the `TargetInstanceTypeFromTagValue` parameter.
- TargetInstanceTypeFromTagValue

  Type: String

  Description: (Optional) The tag key applied to your target instances whose value contains the instance type you want to change to. If you specify a value for the `TargetInstanceTypeFromParameter` parameter, it overrides any value you specify for this parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `autoscaling:DescribeAutoScalingInstances`
- `cloudformation:DescribeStackResources`
- `ssm:GetAutomationExecution`
- `ssm:DescribeAutomationExecutions`

- ec2:DescribeInstanceAttribute
- ec2:DescribeInstances
- ec2:DescribeInstanceStatus
- ec2:DescribeInstanceTypeOfferings
- ec2:DescribeInstanceTypes
- ec2:DescribeTags
- ec2:ModifyInstanceAttribute
- ec2:StartInstances
- ec2:StopInstances

Document Steps

1. aws:assertAwsResourceProperty: Ensures the Amazon EC2 instance is not tagged with the resource tag key specified in the SkipInstancesWithTagKey parameter. If the tag key is found applied to the instance, the step fails and the automation ends.
2. aws:assertAwsResourceProperty: Confirms the status of the target Amazon EC2 instance is running, pending, stopped, or stopping. Otherwise, the automation ends.
3. aws:executeAwsApi: Gathers properties from the Amazon EC2 instance.
4. aws:executeAwsApi: Gathers details about the current Amazon EC2 instance type.
5. aws:branch: Checks if the current instance type and the instance type specified in the TargetInstanceTypeFromParameter parameter are the same. If they are, the automation ends.
6. aws:assertAwsResourceProperty: Ensures the instance is running on the Nitro System.
7. aws:branch: Ensures the Amazon EC2 instance root volume type is an Amazon Elastic Block Store (Amazon EBS) volume.
8. aws:assertAwsResourceProperty: Confirms the instance shutdown behavior is stop and not terminate.
9. aws:branch: Ensures the Amazon EC2 instance is not a Spot instance.
10. aws:branch: Ensures the Amazon EC2 instance tenancy is default and not dedicated host, or dedicated instance.
11. aws:executeScript: Confirms there is only one automation of this runbook targeting the current instance ID. If another automation is already in progress targeting the same instance, the automation returns an error and ends.

12 `aws:branch`: Branches the automation based on the state of the Amazon EC2 instance.

   a. If `stopped` or `stopping`, the automation runs `aws:waitForAwsResourceProperty` until the Amazon EC2 instance is fully stopped.

   b. If `running` or `pending`, the automation runs `aws:waitForAwsResourceProperty` until the Amazon EC2 instance passes status checks.

13 `aws:assertAwsResourceProperty`: Confirms that the Amazon EC2 instance is not part of an Auto Scaling group by calling the `DescribeAutoScalingInstances` API operation. If the instance is part of an Auto Scaling group, ensures the Amazon EC2 instance is in `standby` mode.

14 `aws:branch`: Branches the automation depending on whether you want the automation to check if the Amazon EC2 instance is part of an Amazon CloudFormation stack:

   a. `aws:executeScript` Ensures the Amazon EC2 instance is not part of an Amazon CloudFormation stack by calling the `DescribeStackResources` API operation.

15 `aws:executeAwsApi`: Returns a list of instance types with the same processor architecture type, virtulization type, and that supports the number of network interfaces currently attached to the target instance.

16 `aws:executeAwsApi`: Gets the target instance type value from the tag key specified in the `TargetInstanceTypeFromTagValue` parameter.

17 `aws:executeScript`: Confirms that the current and target instances types are compatible. Ensures that the target instance type is available in the same subnet. Verifies the principal who started the runbook has permissions to change the instance type, and stop and start the instance if it was running.

18 `aws:branch`: Branches the automation based on whether the `DryRun` parameter value is set to `yes`. If `yes`, the automation ends.

19 `aws:branch`: Checks if the original and the target instance type are the same. If they're the same, the automation ends.

20 `aws:executeAwsApi`: Gets the current instance state.

21 `aws:changeInstanceState`: Stops the Amazon EC2 instance.

22 `aws:changeInstanceState`: Forces the instance to stop if it's stuck in the `stopping` state.

23 `aws:executeAwsApi`: Changes the instance type to the target instance type.

24 `aws:sleep`: Waits 3 seconds after changing the instance type for eventual consistency.

25 `aws:branch`: Branches the automation based on the previous instance state. If it was `running`, the instance is started.

a. `aws:changeInstanceState`: Starts the Amazon EC2 instance if it was running before changing the instance type.

b. `aws:waitForAwsResourceProperty`: Waits for the Amazon EC2 instance to pass status checks. If the instance doesn't pass status checks, the instance is changed back to its original instance type.

   i. `aws:changeInstanceState`: Stops the Amazon EC2 instance before changing it to its original instance type.

   ii. `aws:changeInstanceState`: Forces the Amazon EC2 instance to stop before changing it to its original instance type in case it gets stuck in a stopping state.

   iii. `aws:executeAwsApi`: Changes the Amazon EC2 instance to its original type.

   iv. `aws:sleep`: Waits 3 seconds after changing the instance type for eventual consistency.

   v. `aws:changeInstanceState`: Starts the Amazon EC2 instance if it was running before changing the instance type.

   vi. `aws:waitForAwsResourceProperty`: Waits for the Amazon EC2 instance to pass status checks.

26`aws:sleep`: Waits before ending the runbook.

AWSSupport-RestoreEC2InstanceFromSnapshot

Description

The `AWSSupport-RestoreEC2InstanceFromSnapshot` runbook helps you identify and restore an Amazon Elastic Compute Cloud (Amazon EC2) instance from a working Amazon Elastic Block Store (Amazon EBS) snapshot of the root volume.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- EndDate

  Type: String

  Description: (Optional) The last date you want the automation to look for a snapshot.
- InplaceSwap

  Type: Boolean

  Valid values: true | false

  Description: (Optional) If the value for this parameter is set to `true`, the newly created volume from the snapshot replaces the existing root volume attached to your instance.
- InstanceId

  Type: String

  Description: (Required) The ID of the instance you want to restore from a snapshot.
- LookForInstanceStatusCheck

  Type: Boolean

  Valid values: true | false

  Default: true

  Description: (Optional) If the value for this parameter is set to `true`, the automation checks whether instance status checks fail on the test instances launched from the snapshots.
- SkipSnapshotsBy

Type: String

Description: (Optional) The interval at which snapshots are skipped when searching for snapshots to restore your instance. For example, if there are 100 snapshots available, and you specify a value of 2 for this parameter, then every third snapshot is reviewed.

Default: 0

- SnapshotId

  Type: String

  Description: (Optional) The ID of a snapshot you want to restore the instance from.

- StartDate

  Type: String

  Description: (Optional) The first date you want the automation to look for a snapshot.

- TotalSnapshotsToLook

  Type: String

  Description: (Optional) The number of snapshots the automation reviews.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:DescribeInstanceInformation`
- `ec2:AttachVolume`
- `ec2:CreateImage`
- `ec2:CreateTags`
- `ec2:CreateVolume`
- `ec2:DeleteTags`
- `ec2:DeregisterImage`

- ec2:DescribeInstances
- ec2:DescribeInstanceStatus
- ec2:DescribeImages
- ec2:DescribeSnapshots
- ec2:DescribeVolumes
- ec2:DetachVolume
- ec2:RunInstances
- ec2:StartInstances
- ec2:StopInstances
- ec2:TerminateInstances
- cloudwatch:GetMetricData

Document Steps

1. aws:executeAwsApi - Gathers details about the target instance.
2. aws:assertAwsResourceProperty - Verifies the target instance exists.
3. aws:assertAwsResourceProperty - Verifies the root volume is an Amazon EBS volume.
4. aws:assertAwsResourceProperty - Verifies that another automation isn't already running that targets this instance.
5. aws:executeAwsApi - Tags the target instance.
6. aws:executeAwsApi - Creates an AMI of the instance.
7. aws:executeAwsApi - Gathers details about the AMI created in the previous step.
8. aws:waitForAwsResourceProperty - Waits for the AMI state to become available before proceeding.
9. aws:executeScript - Launches a new instance from the newly created AMI.
10. aws:assertAwsResourceProperty - Verifies the instance state is available.
11. aws:executeAwsApi - Gathers details about the newly launched instance.
12. aws:branch - Branches based on whether you provided a value for the SnapshotId parameter.
13. aws:executeScript - Returns a list of snapshots within the time period specified.
14. aws:executeAwsApi - Stops the instance.

15 `aws:waitForAwsResourceProperty` - Waits for the volume state to be `available`.

16 `aws:waitForAwsResourceProperty` - Waits for the instance state to be `stopped`.

17 `aws:executeAwsApi` - Detaches the root volume.

18 `aws:waitForAwsResourceProperty` - Waits for the root volume to be detached.

19 `aws:executeAwsApi` - Attaches the new root volume.

20 `aws:waitForAwsResourceProperty` - Waits for the new volume to be attached.

21 `aws:executeAwsApi` - Starts the instance.

22 `aws:waitForAwsResourceProperty` - Waits for the instance state to be `available`.

23 `aws:waitForAwsResourceProperty` - Waits for system and instance status checks to pass for the instance.

24 `aws:executeScript` - Runs a script to find a snapshot that can be used to successfully create a volume.

25 `aws:executeScript` - Runs a script to recover the instance using the newly created volume from the snapshot identified by the automation, or using the volume created from the snapshot you specified in the `SnapshotId` parameter.

26 `aws:executeScript` - Deletes resources created by the automation.

Outputs

launchCloneInstance.InstanceIds

ListSnapshotByDate.finalSnapshots

ListSnapshotByDate.remainingSnapshotToBeCheckedInSameDateRange

findWorkingSnapshot.workingSnapshot

InstanceRecovery.result

`AWSSupport-SendLogBundleToS3Bucket`

Description

The `AWSSupport-SendLogBundleToS3Bucket` runbook uploads a log bundle generated by the EC2Rescue tool from the target instance to the specified S3 bucket. The runbook installs the platform specific version of EC2Rescue based on the platform of the target instance. EC2Rescue is then used to collect all the available operating system (OS) logs.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the Windows or Linux managed instance you want to collect logs from.
- S3BucketName

  Type: String

  Description: (Required) S3 bucket to upload the logs to.
- S3Path

  Type: String

  Default: `AWSSupport-SendLogBundleToS3Bucket/`

  Description: (Optional) S3 path for the collected logs.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

It is recommended that the EC2 instance receiving the command has an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. The user must have at least **ssm:StartAutomationExecution** and **ssm:SendCommand** to run the automation and send the command to the instance, plus **ssm:GetAutomationExecution** to be able to read the automation output.

Document Steps

1. `aws:runCommand` - Install EC2Rescue via `AWS-ConfigureAWSPackage`.
2. `aws:runCommand` - Run the PowerShell script to collect Windows troubleshooting logs with EC2Rescue.
3. `aws:runCommand` - Run the bash script to collect Linux troubleshooting logs with EC2Rescue.

Outputs

collectAndUploadWindowsLogBundle.Output

collectAndUploadLinuxLogBundle.Output

AWSSupport-StartEC2RescueWorkflow

Description

The `AWSSupport-StartEC2RescueWorkflow` runbook runs the provided base64 encoded script (Bash or Powershell) on a helper instance created to rescue your instance. The root volume of your instance is attached and mounted to the helper instance, also known as the EC2Rescue instance. If your instance is Windows, provide a Powershell script. Otherwise, use Bash. The runbook sets some environment variables which you can use in your script. The environment variables contain information about the input you provided, as well as information about the offline root volume. The offline volume is already mounted and ready to use. For example, you can save a Desired State Configuration file to an offline Windows root volume, or chroot to an offline Linux root volume and perform an offline remediation.

Run this Automation (console)

Important

Amazon EC2 instances created from Marketplace Amazon Machine Images (AMIs) are not supported by this automation.

Additional Information

To base64 encode a script, you can use either Powershell or Bash. Powershell:

```
[System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes([System.IO.File]::ReadA
```

Bash:

```
base64 PATH_TO_FILE
```

Here is a list of environment variables you can use in your offline scripts, depending on the target OS

Windows:

Variable	Description	Example value
$env:EC2RESCUE_ACCOUNT_ID	{{ global:ACCOUNT_ID }}	123456789012
$env:EC2RESCUE_DATE	{{ global:DATE }}	2018-09-07
$env:EC2RESCUE_DATE_TIME	{{ global:DATE_TIME }}	2018-09-07_18.09.59
$env:EC2RESCUE_EC2RW_DIR	EC2Rescue for Windows installation path	C:\Program Files\Amazon\EC2Rescue
$env:EC2RESCUE_EC2RW_DIR	EC2Rescue for Windows installation path	C:\Program Files\Amazon\EC2Rescue
$env:EC2RESCUE_EXECUTION_ID	{{ automation:EXECUTION_ID }}	7ef8008e-219b-4aca-8bb5-65e2e898e20b

Variable	Description	Example value
$env:EC2RESCUE_OFFLINE_CURRENT_CONTROL_SET	Offline Windows Current Control Set path	`HKLM:\AWSTempSystem\ControlSet001`
$env:EC2RESCUE_OFFLINE_DRIVE	Offline Windows drive letter	D:\
$env:EC2RESCUE_OFFLINE_EBS_DEVICE	Offline root volume EBS device	xvdf
$env:EC2RESCUE_OFFLINE_KERNEL_VER	Offline Windows Kernel version	6.1.7601.24214
$env:EC2RESCUE_OFFLINE_OS_ARCHITECTURE	Offline Windows architecture	AMD64
$env:EC2RESCUE_OFFLINE_OS_CAPTION	Offline Windows caption	Windows Server 2008 R2 Datacenter
$env:EC2RESCUE_OFFLINE_OS_TYPE	Offline Windows OS type	Server
$env:EC2RESCUE_OFFLINE_PROGRAM_FILES_DIR	Offline Windows Program files directory path	D:\Program Files
$env:EC2RESCUE_OFFLINE_PROGRAM_FILES_X86_DIR	Offline Windows Program files x86 directory path	D:\Program Files (x86)
$env:EC2RESCUE_OFFLINE_REGISTRY_DIR	Offline Windows registry directory path	D:\Windows\System32\config
$env:EC2RESCUE_OFFLINE_SYSTEM_ROOT	Offline Windows system root directory path	D:\Windows
$env:EC2RESCUE_REGION	{{ global:REGION }}	us-west-1
$env:EC2RESCUE_S3_BUCKET	{{ S3BucketName }}	amzn-s3-demo-bucket

Variable	Description	Example value
$env:EC2RESCUE_S3_PREFIX	{{ S3Prefix }}	myprefix/
$env:EC2RESCUE_SOURCE_INSTANCE	{{ InstanceId }}	i-abcdefgh123456789
$script:EC2RESCUE_OFFLINE_WINDOWS_INSTALL	Offline Windows Installation metadata	Customer Powershell Object

Linux:

Variable	Description	Example value
EC2RESCUE_ACCOUNT_ID	{{ global:ACCOUNT_ID }}	123456789012
EC2RESCUE_DATE	{{ global:DATE }}	2018-09-07
EC2RESCUE_DATE_TIME	{{ global:DATE_TIME }}	2018-09-07_18.09.59
EC2RESCUE_EC2RL_DIR	EC2Rescue for Linux installation path	/usr/local/ec2rl-1.1.3
EC2RESCUE_EXECUTION_ID	{{ automation:EXECUTION_ID }}	7ef8008e-219b-4aca-8bb5-65e2e898e20b
EC2RESCUE_OFFLINE_DEVICE	Offline device name	/dev/xvdf1
EC2RESCUE_OFFLINE_EBS_DEVICE	Offline root volume EBS device	/dev/sdf
EC2RESCUE_OFFLINE_SYSTEM_ROOT	Offline root volume mount point	/mnt/mount
EC2RESCUE_PYTHON	Python version	python2.7
EC2RESCUE_REGION	{{ global:REGION }}	us-west-1
EC2RESCUE_S3_BUCKET	{{ S3BucketName }}	amzn-s3-demo-bucket

Variable	Description	Example value
EC2RESCUE_S3_PREFIX	{{ S3Prefix }}	myprefix/
EC2RESCUE_SOURCE_INSTANCE	{{ InstanceId }}	i-abcdefgh123456789

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AMIPrefix

  Type: String

  Default: `AWSSupport-EC2Rescue`

  Description: (Optional) A prefix for the backup AMI name.
- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- CreatePostEC2RescueBackup

  Type: String

  Valid values: true | false

Default: false

Description: (Optional) Set it to `true` to create an AMI of InstanceId after running the script, before starting it. The AMI will persist after the automation completes. It is your responsibility to secure access to the AMI, or to delete it.

- CreatePreEC2RescueBackup

  Type: String

  Valid values: true | false

  Default: false

  Description: (Optional) Set it to `true` to create an AMI of InstanceId before running the script. The AMI will persist after the automation completes. It is your responsibility to secure access to the AMI, or to delete it.

- EC2RescueInstanceType

  Type: String

  Valid values: t2.small | t2.medium | t2.large | t3.small | t3.medium | t3.large | i3.large

  Default: t3.medium

  Description: (Optional) The EC2 instance type for the EC2Rescue instance.

- InstanceId

  Type: String

  Description: (Required) ID of your EC2 instance. IMPORTANT: Amazon Systems Manager Automation stops this instance. Data stored in instance store volumes will be lost. The public IP address will change if you are not using an Elastic IP.

- OfflineScript

  Type: String

  Description: (Required) Base64 encoded script to run against the helper instance. Use Bash if your source instance is Linux, and PowerShell if it is Windows.

- S3BucketName

Type: String

Description: (Optional) S3 bucket name in your account where you want to upload the troubleshooting logs. Make sure the bucket policy does not grant unnecessary read/write permissions to parties that do not need access to the collected logs.

- S3Prefix

  Type: String

  Default: `AWSSupport-EC2Rescue`

  Description: (Optional) A prefix for the S3 logs.

- SubnetId

  Type: String

  Default: SelectedInstanceSubnet

  Description: (Optional) The subnet ID for the EC2Rescue instance. By default, the same subnet where the provided instance resides is used. IMPORTANT: If you provide a custom subnet, it must be in the same Availability Zone as InstanceId, and it must allow access to the SSM endpoints.

- UniqueId

  Type: String

  Default: {{ automation:EXECUTION_ID }}

  Description: (Optional) A unique identifier for the automation.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

It is recommended the user who runs the automation have the **AmazonSSMAutomationRole** IAM managed policy attached. In addition to that policy, the user must have:

```
{
                "Version": "2012-10-17",
                "Statement": [
```

```
                    {
                       "Action": [
                          "lambda:InvokeFunction",
                          "lambda:DeleteFunction",
                          "lambda:GetFunction"
                       ],
                       "Resource": "arn:aws:lambda:*:An-AWS-Account-
ID:function:AWSSupport-EC2Rescue-*",
                       "Effect": "Allow"
                    },
                    {
                       "Action": [
                          "s3:GetObject",
                          "s3:GetObjectVersion"
                       ],
                       "Resource": [
                          "arn:aws:s3:::awssupport-ssm.*/*.template",
                          "arn:aws:s3:::awssupport-ssm.*/*.zip"
                       ],
                       "Effect": "Allow"
                    },
                    {
                       "Action": [
                          "iam:CreateRole",
                          "iam:CreateInstanceProfile",
                          "iam:GetRole",
                          "iam:GetInstanceProfile",
                          "iam:PutRolePolicy",
                          "iam:DetachRolePolicy",
                          "iam:AttachRolePolicy",
                          "iam:PassRole",
                          "iam:AddRoleToInstanceProfile",
                          "iam:RemoveRoleFromInstanceProfile",
                          "iam:DeleteRole",
                          "iam:DeleteRolePolicy",
                          "iam:DeleteInstanceProfile"
                       ],
                       "Resource": [
                          "arn:aws:iam::An-AWS-Account-ID:role/AWSSupport-EC2Rescue-*",
                          "arn:aws:iam::An-AWS-Account-ID:instance-profile/AWSSupport-
EC2Rescue-*"
                       ],
                       "Effect": "Allow"
                    },
```

```
                {
                   "Action": [
                      "lambda:CreateFunction",
                      "ec2:CreateVpc",
                      "ec2:ModifyVpcAttribute",
                      "ec2:DeleteVpc",
                      "ec2:CreateInternetGateway",
                      "ec2:AttachInternetGateway",
                      "ec2:DetachInternetGateway",
                      "ec2:DeleteInternetGateway",
                      "ec2:CreateSubnet",
                      "ec2:DeleteSubnet",
                      "ec2:CreateRoute",
                      "ec2:DeleteRoute",
                      "ec2:CreateRouteTable",
                      "ec2:AssociateRouteTable",
                      "ec2:DisassociateRouteTable",
                      "ec2:DeleteRouteTable",
                      "ec2:CreateVpcEndpoint",
                      "ec2:DeleteVpcEndpoints",
                      "ec2:ModifyVpcEndpoint",
                      "ec2:Describe*"
                   ],
                   "Resource": "*",
                   "Effect": "Allow"
                }
             ]
          }
```

Document Steps

1. `aws:executeAwsApi` - Describe the provided instance
2. `aws:executeAwsApi` - Describe the provided instance's root volume
3. `aws:assertAwsResourceProperty` - Check the root volume device type is EBS
4. `aws:assertAwsResourceProperty` - Check the root volume is not encrypted
5. `aws:assertAwsResourceProperty` - Check the provide subnet ID
   a. (Use current instance subnet) - If *SubnetId = SelectedInstanceSubnet* then run `aws:createStack` to deploy the EC2Rescue CloudFormation stack
   b. (Create new VPC) - If *SubnetId = CreateNewVPC* then run `aws:createStack` to deploy the EC2Rescue CloudFormation stack

c. (Use custom subnet) - In all other cases:

`aws:assertAwsResourceProperty` - Check the provided subnet is in the same Availability Zone as the provided instance

`aws:createStack` - Deploy the EC2Rescue CloudFormation stack

6. `aws:invokeLambdaFunction` - Perform additional input validation
7. `aws:executeAwsApi` - Update the EC2Rescue CloudFormation stack to create the EC2Rescue helper instance
8. `aws:waitForAwsResourceProperty` - Wait for the EC2Rescue CloudFormation stack update to complete
9. `aws:executeAwsApi` - Describe the EC2Rescue CloudFormation stack output to obtain the EC2Rescue helper instance ID
10. `aws:waitForAwsResourceProperty` - Wait for the EC2Rescue helper instance to become a managed instance
11. `aws:changeInstanceState` - Stop the provided instance
12. `aws:changeInstanceState` - Stop the provided instance
13. `aws:changeInstanceState` - Force stop the provided instance
14. `aws:assertAwsResourceProperty` - Check the CreatePreEC2RescueBackup input value
    a. (Create pre-EC2Rescue backup) - If *CreatePreEC2RescueBackup = true*
    b. `aws:executeAwsApi` - Create an AMI backup of the provided instance
    c. `aws:createTags` - Tag the AMI backup
15. `aws:runCommand` - Install EC2Rescue on the EC2Rescue helper instance
16. `aws:executeAwsApi` - Detach the root volume from the provided instance
17. `aws:assertAwsResourceProperty` - Check the provided instance platform
    a. (Instance is Windows):

    `aws:executeAwsApi` - Attach the root volume to the EC2Rescue helper instance as *xvdf*

    `aws:sleep` - Sleep 10 seconds

    `aws:runCommand` - Run the provided offline script in Powershell

    b. (Instance is Linux):

aws:executeAwsApi - Attach the root volume to the EC2Rescue helper instance as */dev/sdf*

aws:sleep - Sleep 10 seconds

aws:runCommand - Run the provided offline script in Bash

18 aws:changeInstanceState - Stop the EC2Rescue helper instance

19 aws:changeInstanceState - Force stop the EC2Rescue helper instance

20 aws:executeAwsApi - Detach the root volume from the EC2Rescue helper instance

21 aws:executeAwsApi - Attach the root volume back to the provided instance

22 aws:assertAwsResourceProperty - Check the CreatePostEC2RescueBackup input value

a. (Create post-EC2Rescue backup) - If *CreatePostEC2RescueBackup = true*

b. aws:executeAwsApi - Create an AMI backup of the provided instance

c. aws:createTags - Tag the AMI backup

23 aws:executeAwsApi - Restore the initial delete on termination state for the root volume of the provided instance

24 aws:changeInstanceState - Restore the initial state of the provided instance (running/stopped)

25 aws:deleteStack - Delete the EC2Rescue CloudFormation stack

Outputs

runScriptForLinux.Output

runScriptForWindows.Output

preScriptBackup.ImageId

postScriptBackup.ImageId

AWSPremiumSupport-TroubleshootEC2DiskUsage

Description

The AWSPremiumSupport-TroubleshootEC2DiskUsage runbook helps you investigate and potentially remediate issues with Amazon Elastic Compute Cloud (Amazon EC2) instance root

and non-root disk usage. If possible, the runbook attempts to remediate issues by extending the volume and its file system. To perform these tasks, this runbook orchestrates the execution of several runbooks based on the operating system of the affected instance.

The first runbook, `AWSPremiumSupport-DiagnoseDiskUsageOnWindows` or `AWSPremiumSupport-DiagnoseDiskUsageOnLinux`, determines if disk issues can be mitigated by expanding the volume.

The second runbook, `AWSPremiumSupport-ExtendVolumesOnWindows` or `AWSPremiumSupport-ExtendVolumesOnLinux`, uses the output of the first runbook to run Python code that modifies the volume. After the volume has been modified, the runbook extends the partition and file system of the affected volumes.

Important

Access to `AWSPremiumSupport-*` runbooks requires an Enterprise or Business Support Subscription. For more information, see Compare Amazon Web Services Support Plans.

This document was built in collaboration with Amazon Managed Services (AMS). AMS helps you manage your Amazon infrastructure more efficiently and securely. AMS also provides operational flexibility, enhanced security and compliance, capacity optimization, and cost-savings identification. For more information, see Amazon Managed Services.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, Windows

Parameters

- InstanceId

  Type: String

  Allowed values: ^i-[a-z0-9]{8,17}$

  Description: (Required) ID of your Amazon EC2 instance.
- VolumeExpansionEnabled

  Type: Boolean

  Description: (Optional) Flag to control whether the document will extend the volumes and partitions affected.

  Default: true
- VolumeExpansionUsageTrigger

  Type: String

  Description: (Optional) Minimum usage of partition space required to trigger extension (in percentage).

  Allowed values: ^[0-9]{1,2}$

  Default: 85
- VolumeExpansionCapSize

  Type: String

  Description: (Optional) Maximum size that the Amazon Elastic Block Store (Amazon EBS) volume will be increased to (in GiB).

  Allowed values: ^[0-9]{1,4}$

  Default: 2048
- VolumeExpansionGibIncrease

  Type: String

  Description: (Optional) Increase in GiB of the volume. The biggest net increase between VolumeExpansionGibIncrease and VolumeExpansionPercentageIncrease will be used.

Allowed values: ^[0-9]{1,4}$

Default: 20

- VolumeExpansionPercentageIncrease

  Type: String

  Description: (Optional) Increase in percentage of the volume. The biggest net increase between VolumeExpansionGibIncrease and VolumeExpansionPercentageIncrease will be used.

  Allowed values: ^[0-9]{1,2}$

  Default: 20

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeVolumes`
- `ec2:DescribeVolumesModifications`
- `ec2:ModifyVolume`
- `ec2:DescribeInstances`
- `ec2:CreateImage`
- `ec2:DescribeImages`
- `ec2:DescribeTags`
- `ec2:CreateTags`
- `ec2:DeleteTags`

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ssm:DescribeAutomationStepExecutions
- ssm:DescribeAutomationExecutions
- ssm:SendCommand
- ssm:DescribeInstanceInformation
- ssm:ListCommands
- ssm:ListCommandInvocations

Document Steps

1. aws:assertAwsResourceProperty - Check if the instance is managed by Systems Manager
2. aws:executeAwsApi - Describes the instance to get the platform.
3. aws:branch - Branches automation based on the instance's platform.
   a. If the instance is Windows:
      i. aws:executeAutomation - Run the AWSPremiumSupport-DiagnoseDiskUsageOnWindows runbook in order to diagnose disk usage issues on the instance.
      ii. aws:executeAwsApi - Gets the output of the previous automation.
      iii. aws:branch - Branches based on the output of the diagnostics, and if there are volumes that can be expanded to mitigate the alert.
         A. There are no volumes that need to be expanded: End the automation.
         B. There are volumes that need to be expanded:
            I. aws:executeAwsApi - Create an Amazon Machine Image (AMI) of the instance.
            II. aws:waitForAwsResourceProperty - Waits for the AMI state to be available.
            III. aws:executeAutomation - Run the AWSPremiumSupport-ExtendVolumesOnWindows runbook in order to perform the volume modification as well as the required steps in the operating system (OS) to make the new space available.
   b. (Platform is not windows) If the input instance is not Windows:

i. `aws:executeAutomation` - Run the `AWSPremiumSupport-DiagnoseDiskUsageOnLinux` runbook in order to diagnose disk usage issues on the instance.

ii. `aws:executeAwsApi` - Gets the output of the previous automation.

iii. `aws:branch` - Branches based on the output of the diagnostics, and if there are volumes that can be expanded to mitigate the alert.

A. There are no volumes that need to be expanded: End the automation.

B. There are volumes that need to be expanded:

I. `aws:executeAwsApi` - Create an AMI of the instance.

II. `aws:waitForAwsResourceProperty` - Waits for AMI state to be `available`.

III. `aws:executeAutomation` - Run the `AWSPremiumSupport-ExtendVolumesOnLinux` runbook in order to perform the volume modification as well as the required steps in the OS to make the new space available.

Outputs

diagnoseDiskUsageAlertOnWindows.Output

extendVolumesOnWindows.Output

diagnoseDiskUsageAlertOnLinux.Output

extendVolumesOnLinux.Output

BackupAMILinux.ImageId

BackupAMIWindows.ImageId

AWSSupport-TroubleshootEC2InstanceConnect

Description

`AWSSupport-TroubleshootEC2InstanceConnect` automation helps analyze and detect errors preventing the connection to an Amazon Elastic Compute Cloud (Amazon EC2) instance using Amazon EC2 Instance Connect. It identifies issues caused by an unsupported Amazon Machine Image (AMI), missing OS-level package installation or configuration, missing Amazon Identity and Access Management (IAM) permissions, or network configuration issues.

How does it work?

The runbook takes the Amazon EC2 instance ID, username, connection mode, source IP CIDR, SSH port, and Amazon Resource Name (ARN) for the IAM role or user experiencing issues with Amazon EC2 Instance Connect. It then checks the prerequisites for connecting to an Amazon EC2 instance using Amazon EC2 Instance Connect:

- The instance is running and in a healthy state.
- The instance is located in an Amazon region supported by Amazon EC2 Instance Connect.
- AMI of the instance is supported by Amazon EC2 Instance Connect.
- The instance can reach the Instance Metadata Service (IMDSv2).
- Amazon EC2 Instance Connect package is properly installed and configured at the OS level.
- The network configuration (security groups, network ACL, and route table rules) allows connection to the instance through Amazon EC2 Instance Connect.
- The IAM role or user that's used to leverage Amazon EC2 Instance Connect has access to push keys to the Amazon EC2 instance.

Important

- To check the instance AMI, IMDSv2 reachability, and Amazon EC2 Instance Connect package installation, the instance must be SSM managed. Otherwise, it skips those steps. For more information, see Why is my Amazon EC2 instance not displaying as a managed node.
- The network check will only detect if security group and network ACL rules block traffic when SourceIpCIDR is provided as an input parameter. Otherwise, it will only display SSH-related rules.
- Connections using Amazon EC2 Instance Connect Endpoint are not validated in this runbook.
- For private connections, the automation does not check if the SSH client is installed on the source machine and if it can reach the instance's private IP address.

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeRouteTables`
- `ec2:DescribeInternetGateways`
- `iam:SimulatePrincipalPolicy`
- `ssm:DescribeInstanceInformation`
- `ssm:ListCommands`
- `ssm:ListCommandInvocations`
- `ssm:SendCommand`

Instructions

Follow these steps to configure the automation:

1. Navigate to the `AWSSupport-TroubleshootEC2InstanceConnect` in the Amazon Systems Manager console.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **InstanceId (Required):**

     The ID of the target Amazon EC2 instance that you could not connect to using Amazon EC2 Instance Connect.
   - **AutomationAssumeRole (Optional):**

The ARN of the IAM role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

- **Username (Required):**

  The username used to connect to the Amazon EC2 instance using Amazon EC2 Instance Connect. It is used to evaluate if IAM access is granted for this particular user.

- **EC2InstanceConnectRoleOrUser (Required):**

  The ARN of the IAM role or user that is leveraging Amazon EC2 Instance Connect to push keys to the instance.

- **SSHPort (Optional):**

  The SSH port configured on the Amazon EC2 instance. Default value is 22. The port number must be between 1-65535.

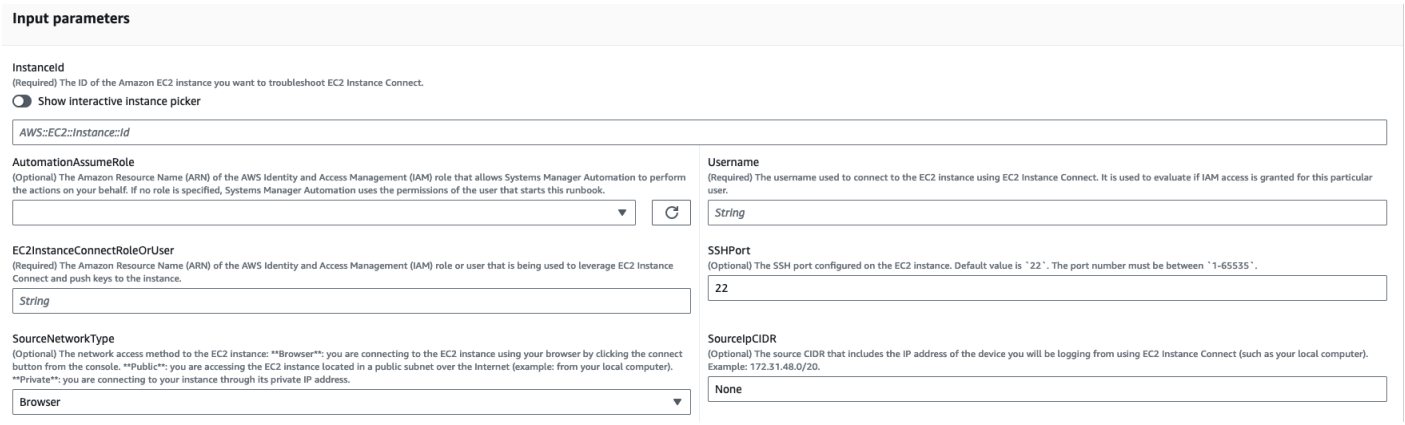
- **SourceNetworkType (Optional):**

  The network access method to the Amazon EC2 instance:

  - **Browser:** You connect from the Amazon Management Console.
  - **Public:** You connect to the instance located in a public subnet over the internet (for example, your local computer).
  - **Private:** You connect through the instance's private IP address.

- **SourceIpCIDR (Optional):**

  The source CIDR that includes the IP address of the device (such as your local computer) you will log from using Amazon EC2 Instance Connect. Example: 172.31.48.6/32. If no value is provided with public or private access mode, the runbook will not evaluate if the Amazon EC2 instance security group and network ACL rules allow SSH traffic. It will display SSH-related rules instead.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:
   - **AssertInitialState:**

     Ensures that the Amazon EC2 instance status is running. Otherwise, the automation ends.
   - **GetInstanceProperties:**

     Gets the current Amazon EC2 instance properties (PlatformDetails, PublicIpAddress, VpcId, SubnetId and MetadataHttpEndpoint).
   - **GatherInstanceInformationFromSSM:**

     Gets the Systems Manager instance's ping status and operating system details if the instance is SSM managed.
   - **CheckIfAWSRegionSupported:**

     Checks if the Amazon EC2 instance is located in an Amazon EC2 Instance Connect supported Amazon region.
   - **BranchOnIfAWSRegionSupported:**

     Continues the execution if the Amazon Region is supported by Amazon EC2 Instance Connect. Otherwise, it creates the output and exits the automation.
   - **CheckIfInstanceAMIIsSupported:**

     Checks if the AMI associated with the instance is supported by Amazon EC2 Instance Connect.
   - **BranchOnIfInstanceAMIIsSupported:**

If the instance AMI is supported, it performs the OS-level checks, like metadata reachability and Amazon EC2 Instance Connect package installation and configuration. Otherwise, it checks if HTTP metadata is enabled using Amazon API, then advances to the network check step.

- **CheckIMDSReachabilityFromOs:**

  Runs a Bash script on the target Amazon EC2 Linux instance to check if it is able to reach the IMDSv2.

- **CheckEICPackageInstallation:**

  Runs a Bash script on the target Amazon EC2 Linux instance to check if the Amazon EC2 Instance Connect package is properly installed and configured.

- **CheckSSHConfigFromOs:**

  Runs a Bash script on the target Amazon EC2 Linux instance to check if the configured SSH port matches the input parameter `SSHPort.`

- **CheckMetadataHTTPEndpointIsEnabled:**

  Checks if the instance metadata service HTTP endpoint is enabled.

- **CheckEICNetworkAccess:**

  Checks if the network configuration (security groups, network ACL, and route table rules) allows connection to the instance through Amazon EC2 Instance Connect.

- **CheckIAMRoleOrUserPermissions:**

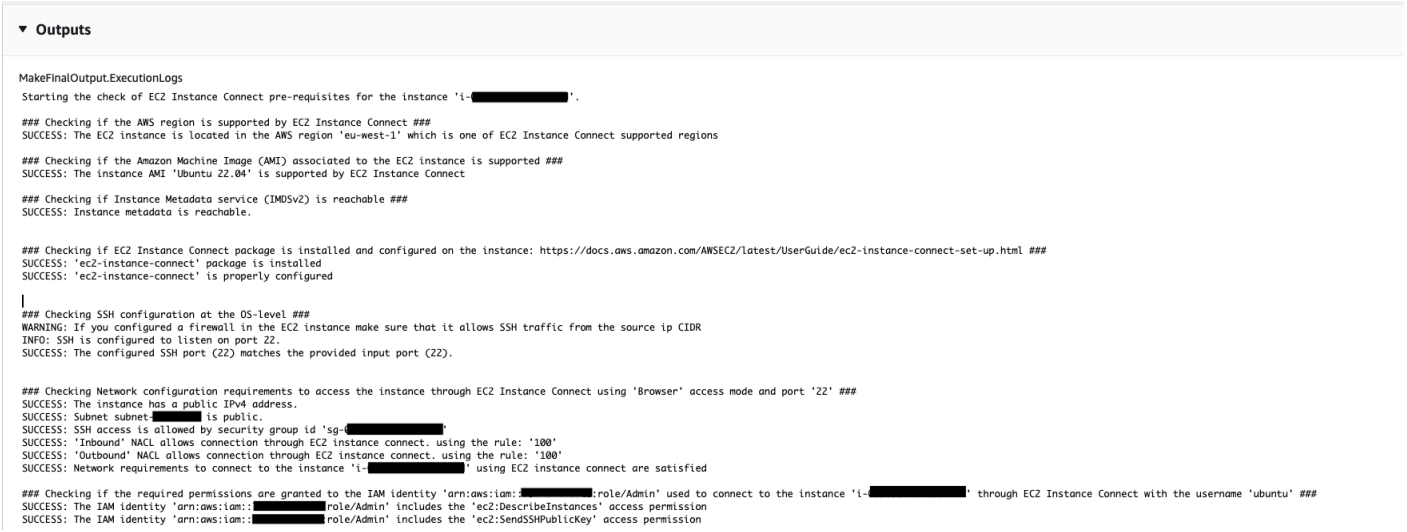
  Checks if the IAM role or user used to leverage Amazon EC2 Instance Connect has access to push keys to the Amazon EC2 instance using the provided username.

- **MakeFinalOutput:**

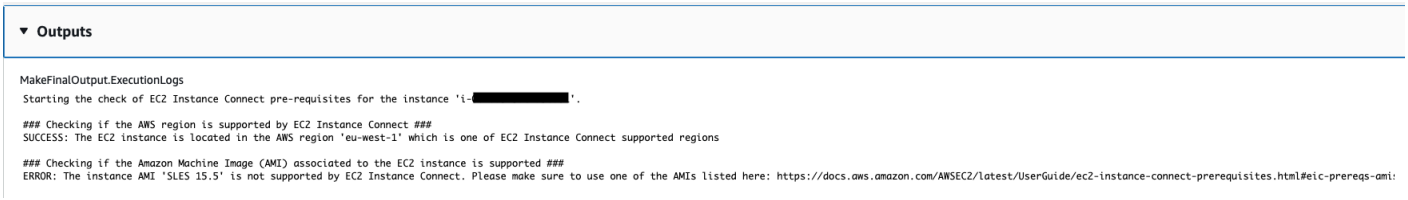
  Consolidates the output of all previous steps.

7. After completed, review the **Outputs** section for the detailed results of the execution:

   Execution where the target instance has all required prerequisites:



Execution where the AMI of the target instance is not supported:



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- How do I troubleshoot issues connecting to my Amazon EC2 instance using Amazon EC2 Instance Connect?

AWSSupport-TroubleshootLinuxMGNDRSAgentLogs

Description

`AWSSupport-TroubleshootLinuxMGNDRSAgentLogs` automation runbook is used to detect common errors when installing the Amazon Application Migration Service (Amazon MGN) and Amazon Elastic Disaster Recovery (Amazon DRS) replication agents in Linux servers to migrate source servers to the Amazon cloud.

How does it work?

The runbook `AWSSupport-TroubleshootLinuxMGNDRSAgentLogs` takes the Amazon Simple Storage Service (Amazon S3) path where the Amazon MGN or Amazon DRS installation log `aws_replication_agent_installer.log` is uploaded as parameter. Then, it performs the following tasks:

- **Validation:** Checks if the provided log file is valid and that it contains at least one agent installation.
- **Parsing:** Thoroughly parses the latest agent installation in the log file for known Amazon MGN or Amazon DRS errors.
- **Error Detection and resolution:** Based on the parsing, it detects and lists any errors or issues during the agent installation process. For each detected error, the runbook provides detailed steps to help resolve or mitigate the issue.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `s3:GetObject`
- `s3:ListBucket`

Instructions

Follow these steps to configure the automation:

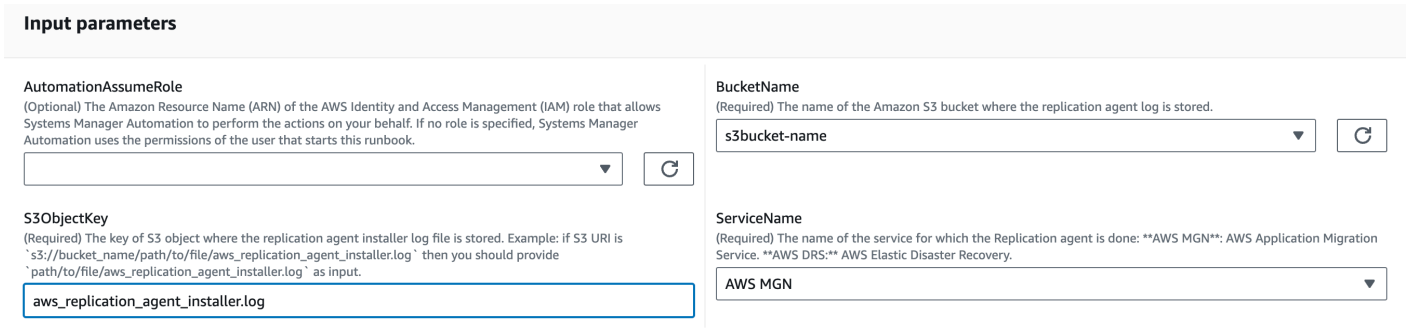
1. Navigate to `AWSSupport-TroubleshootLinuxMGNDRSAgentLogs` in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **BucketName (Required):**

     The name of the Amazon S3 bucket where the replication agent log is stored.
   - **S3ObjectKey (Required):**

     The key of the Amazon S3 object where the replication agent installer log file is stored. Example: If the Amazon S3 URI is `s3://bucket_name/path/to/file/aws_replication_agent_installer.log`, then you should input `path/to/file/aws_replication_agent_installer.log`.
   - **ServiceName (Required):**

     The name of the service for which the replication agent is installed. Allowed values: `Amazon MGN` or `Amazon DRS`

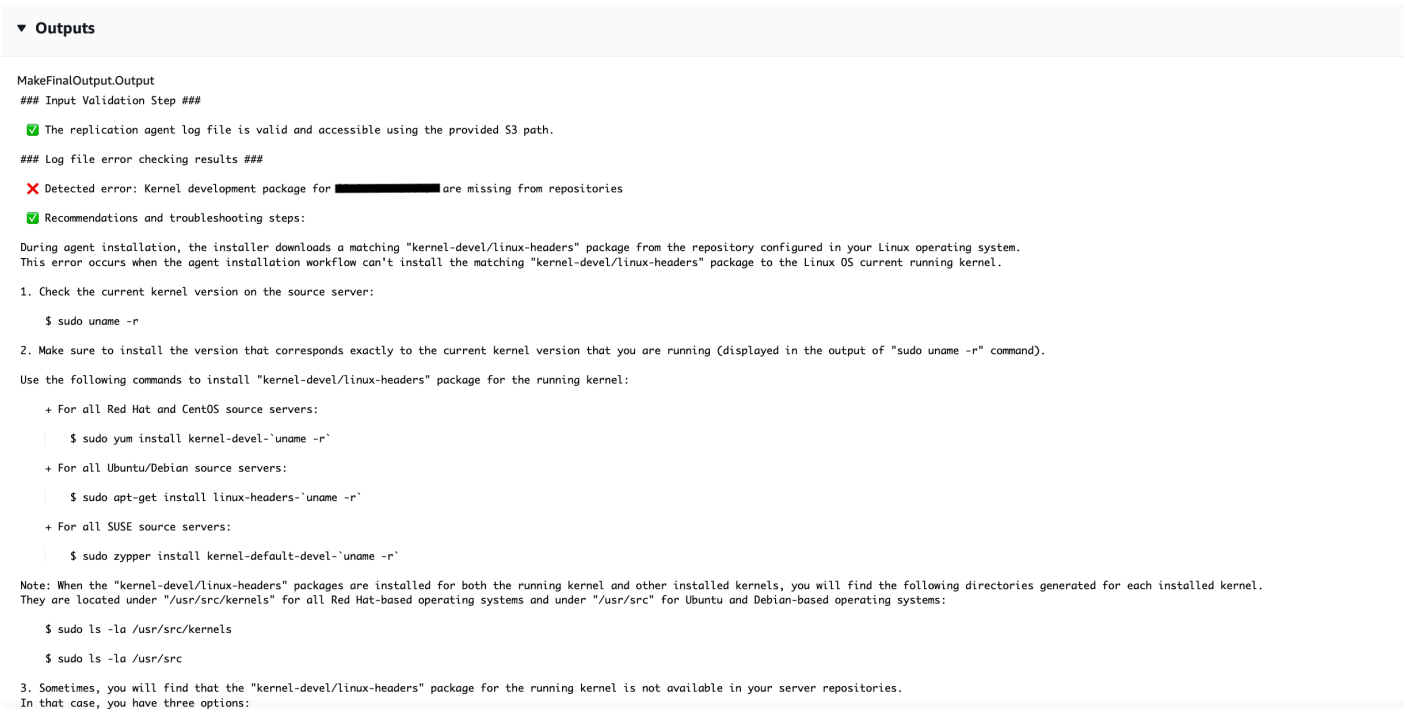


4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:
   - **`ValidateInput`**

     Ensures that the replication agent log file is valid and accessible using the provided Amazon S3 bucket name and path to the object, then returns the byte number of the latest agent installation.
   - **`CheckReplicationAgentLogErrors`**

     Reads the replication agent log file starting from the latest installation byte and search for known Amazon MGN or Amazon DRS errors.
   - **`MakeFinalOutput`**

     Creates the output from the previous checks including information about the errors found and troubleshooting recommendations.
7. After completed, review the Outputs section for the detailed results of the execution:



Outputs

MakeFinalOutput.Output

```
### Input Validation Step ###

 ✅ The replication agent log file is valid and accessible using the provided S3 path.

### Log file error checking results ###

 ❌ Detected error: Kernel development package for [redacted] are missing from repositories

 ✅ Recommendations and troubleshooting steps:

During agent installation, the installer downloads a matching "kernel-devel/linux-headers" package from the repository configured in your Linux operating system.
This error occurs when the agent installation workflow can't install the matching "kernel-devel/linux-headers" package to the Linux OS current running kernel.

1. Check the current kernel version on the source server:

    $ sudo uname -r

2. Make sure to install the version that corresponds exactly to the current kernel version that you are running (displayed in the output of "sudo uname -r" command).

Use the following commands to install "kernel-devel/linux-headers" package for the running kernel:

    + For all Red Hat and CentOS source servers:

        $ sudo yum install kernel-devel-`uname -r`

    + For all Ubuntu/Debian source servers:

        $ sudo apt-get install linux-headers-`uname -r`

    + For all SUSE source servers:

        $ sudo zypper install kernel-default-devel-`uname -r`

Note: When the "kernel-devel/linux-headers" packages are installed for both the running kernel and other installed kernels, you will find the following directories generated for each installed kernel.
They are located under "/usr/src/kernels" for all Red Hat-based operating systems and under "/usr/src" for Ubuntu and Debian-based operating systems:

    $ sudo ls -la /usr/src/kernels

    $ sudo ls -la /usr/src

3. Sometimes, you will find that the "kernel-devel/linux-headers" package for the running kernel is not available in your server repositories.
In that case, you have three options:
```

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-TroubleshootRDP

Description

The `AWSSupport-TroubleshootRDP` runbook allows the user to check or modify common settings on the target instance which may impact Remote Desktop Protocol (RDP) connections, such as the RDP port, Network Layer Authentication (NLA) and Windows Firewall profiles. Optionally, changes can be applied offline by stopping and starting the instance, if the user explicitly allows for offline remediation. By default, the runbook reads and outputs the values of the settings.

> **Important**
>
> Changes to the RDP settings, RDP service and Windows Firewall profiles should be carefully reviewed before using this runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- Action

  Type: String

  Valid values: CheckAll | FixAll | Custom

  Default: Custom

  Description: (Optional) [Custom] Use the values from Firewall, RDPServiceStartupType, RDPServiceAction, RDPPortAction, NLASettingAction and RemoteConnections to manage the settings. [CheckAll] Read the values of the settings without changing them. [FixAll] Restore RDP default settings, and disable the Windows Firewall.

- AllowOffline

  Type: String

  Valid values: true | false

  Default: false

Description: (Optional) Fix only - Set it to true if you allow an offline RDP remediation in case the online troubleshooting fails, or the provided instance is not a managed instance. Note: For the offline remediation, SSM Automation stops the instance, and creates an AMI before attempting any operations.

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Firewall

  Type: String

  Valid values: Check | Disable

  Default: Check

  Description: (Optional) Check or disable the Windows firewall (all profiles).

- InstanceId

  Type: String

  Description: (Required) The ID of the instance to troubleshoot the RDP settings of.

- NLASettingAction

  Type: String

  Valid values: Check | Disable

  Default: Check

  Description: (Optional) Check or disable Network Layer Authentication (NLA).

- RDPPortAction

  Type: String

Valid values: Check | Modify

Default: Check

Description: (Optional) Check the current port used for RDP connections, or modify the RDP port back to 3389 and restart the service.

- RDPServiceAction

  Type: String

  Valid values: Check | Start | Restart | Force-Restart

  Default: Check

  Description: (Optional) Check, start, restart, or force-restart the RDP service (TermService).

- RDPServiceStartupType

  Type: String

  Valid values: Check | Auto

  Default: Check

  Description: (Optional) Check or set the RDP service to automatically start when Windows boots.

- RemoteConnections

  Type: String

  Valid values: Check | Enable

  Default: Check

  Description: (Optional) An action to perform on the fDenyTSConnections setting: Check, Enable.

- S3BucketName

  Type: String

  Description: (Optional) Offline only - S3 bucket name in your account where you want to upload the troubleshooting logs. Make sure the bucket policy does not grant unnecessary read/write permissions to parties that do not need access to the collected logs.

- SubnetId

  Type: String

  Default: SelectedInstanceSubnet

  Description: (Optional) Offline only - The subnet ID for the EC2Rescue instance used to perform the offline troubleshooting. If no subnet ID is specified, Amazon Systems Manager Automation will create a new VPC. IMPORTANT: The subnet must be in the same Availability Zone as InstanceId, and it must allow access to the SSM endpoints.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

It is recommended that the EC2 instance receiving the command has an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. For the online remediation, the user must have at least **ssm:DescribeInstanceInformation**, **ssm:StartAutomationExecution** and **ssm:SendCommand** to run the automation and send the command to the instance, plus **ssm:GetAutomationExecution** to be able to read the automation output. For the offline remediation, the user must have at least **ssm:DescribeInstanceInformation**, **ssm:StartAutomationExecution**, **ec2:DescribeInstances**, plus **ssm:GetAutomationExecution** to be able to read the automation output. `AWSSupport-TroubleshootRDP` calls `AWSSupport-ExecuteEC2Rescue` to perform the offline remediation - please review the permissions for `AWSSupport-ExecuteEC2Rescue` to ensure you can run the automation successfully.

Document Steps

1. `aws:assertAwsResourceProperty` - Check if the instance is a Windows Server instance
2. `aws:assertAwsResourceProperty` - Check if the instance is a managed instance
3. (Online troubleshooting) If the instance is a managed instance, then:
   a. `aws:assertAwsResourceProperty` - Check the provided Action value
   b. (Online check) If the **Action = CheckAll**, then:

      `aws:runPowerShellScript` - Runs the PowerShell script to get the Windows Firewall profiles status.

`aws:executeAutomation` - Calls `AWSSupport-ManageWindowsService` to get the RDP service status.

`aws:executeAutomation` - Calls `AWSSupport-ManageRDPSettings` to get the RDP settings.

c. (Online fix) If the **Action = FixAll**, then:

`aws:runPowerShellScript` - Runs the PowerShell script to disable all Windows Firewall profiles.

`aws:executeAutomation` - Calls `AWSSupport-ManageWindowsService` to start the RDP service.

`aws:executeAutomation` - Calls `AWSSupport-ManageRDPSettings` to enable remote connections and disable NLA.

d. (Online management) If the **Action = Custom**, then:

`aws:runPowerShellScript` - Runs the PowerShell script to manage the Windows Firewall profiles.

`aws:executeAutomation` - Calls `AWSSupport-ManageWindowsService` to manage the RDP service.

`aws:executeAutomation` - Calls `AWSSupport-ManageRDPSettings` to manage the RDP settings.

4. (Offline remediation) If the instance is not a managed instance then:

a. `aws:assertAwsResourceProperty` - Assert **AllowOffline = true**

b. `aws:assertAwsResourceProperty` - Assert **Action = FixAll**

c. `aws:assertAwsResourceProperty` - Assert the value of SubnetId

(Use the provided instance's subnet) If SubnetId is SELECTED_INSTANCE_SUBNET

`aws:executeAwsApi` - Retrieve the current instance's subnet.

`aws:executeAutomation` - Run `AWSSupport-ExecuteEC2Rescue` with provided instance's subnet.

d. (Use the provided custom subnet) If SubnetId is not SELECTED_INSTANCE_SUBNET

`aws:executeAutomation` - Run `AWSSupport-ExecuteEC2Rescue` with provided SubnetId value.

Outputs

manageFirewallProfiles.Output

manageRDPServiceSettings.Output

manageRDPSettings.Output

checkFirewallProfiles.Output

checkRDPServiceSettings.Output

checkRDPSettings.Output

disableFirewallProfiles.Output

restoreDefaultRDPServiceSettings.Output

restoreDefaultRDPSettings.Output

troubleshootRDPOffline.Output

troubleshootRDPOfflineWithSubnetId.Output

`AWSSupport-TroubleshootSSH`

Description

The `AWSSupport-TroubleshootSSH` runbook installs the Amazon EC2Rescue tool for Linux, and then uses the EC2Rescue tool to check or attempt to fix common issues that prevent a remote connection to the Linux machine via SSH. Optionally, changes can be applied offline by stopping and starting the instance, if the user explicitly allows for offline remediation. By default, the runbook operates in read-only mode.

Run this Automation (console)

For information about working with the `AWSSupport-TroubleshootSSH` runbook, see this `AWSSupport-TroubleshootSSH` troubleshooting topic from Amazon Premium Support.

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- Action

  Type: String

  Valid values: CheckAll | FixAll

  Default: CheckAll

  Description: (Required) Specify whether to check for issues without fixing them or to check and automatically fix any discovered issues.
- AllowOffline

  Type: String

  Valid values: true | false

  Default: false

  Description: (Optional) Fix only - Set it to true if you allow an offline SSH remediation in case the online troubleshooting fails, or the provided instance is not a managed instance. Note: For the offline remediation, SSM Automation stops the instance, and creates an AMI before attempting any operations.
- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) ID of your EC2 instance for Linux.
- S3BucketName

  Type: String

  Description: (Optional) Offline only - S3 bucket name in your account where you want to upload the troubleshooting logs. Make sure the bucket policy does not grant unnecessary read/write permissions to parties that do not need access to the collected logs.
- SubnetId

  Type: String

  Default: SelectedInstanceSubnet

  Description: (Optional) Offline only - The subnet ID for the EC2Rescue instance used to perform the offline troubleshooting. If no subnet ID is specified, Amazon Systems Manager Automation will create a new VPC.

  > **Important**
  >
  > The subnet must be in the same Availability Zone as InstanceId, and it must allow access to the SSM endpoints.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

It is recommended that the EC2 instance receiving the command has an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. For the online remediation, the user must have at least **ssm:DescribeInstanceInformation**, **ssm:StartAutomationExecution** and **ssm:SendCommand** to run the automation and send the command to the instance, plus **ssm:GetAutomationExecution** to be able to read the automation output. For the offline remediation, the user must have at least **ssm:DescribeInstanceInformation**, **ssm:StartAutomationExecution**, **ec2:DescribeInstances**, plus **ssm:GetAutomationExecution** to

be able to read the automation output. AWSSupport-TroubleshootSSH calls AWSSupport-ExecuteEC2Rescue to perform the offline remediation - please review the permissions for AWSSupport-ExecuteEC2Rescue to ensure you can run the automation successfully.

Document Steps

1. aws:assertAwsResourceProperty - Check if the instance is a managed instance
   a. (Online remediation) If the instance is a managed instance, then:
      i. aws:configurePackage - Install EC2Rescue for Linux via AWS-ConfigureAWSPackage.
      ii. aws:runCommand - Run the bash script to run EC2Rescue for Linux.
   b. (Offline remediation) If the instance is not a managed instance then:
      i. aws:assertAwsResourceProperty - Assert **AllowOffline = true**
      ii. aws:assertAwsResourceProperty - Assert **Action = FixAll**
      iii. aws:assertAwsResourceProperty - Assert the value of SubnetId
      iv. (Use the provided instance's subnet) If SubnetId is SelectedInstanceSubnet us aws:executeAutomation to run AWSSupport-ExecuteEC2Rescue with provided instance's subnet.
      v. (Use the provided custom subnet) If SubnetId is not SelectedInstanceSubnet use aws:executeAutomation to run AWSSupport-ExecuteEC2Rescue with provided SubnetId value.

Outputs

troubleshootSSH.Output

troubleshootSSHOffline.Output

troubleshootSSHOfflineWithSubnetId.Output

AWSSupport-TroubleshootSUSERegistration

Description

The AWSSupport-TroubleshootSUSERegistration runbook helps you to identify why registering an Amazon Elastic Compute Cloud (Amazon EC2) SUSE Linux Enterprise Server instance with SUSE Update Infrastructure failed. The automation output provides steps to resolve, or helps

you troubleshoot, the issue. If the instance passes all checks during the automation, the instance is registered with SUSE Update Infrastructure.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance you want to troubleshoot.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:DescribeInstanceProperties`

- ssm:DescribeInstanceInformation
- ssm:ListCommandInvocations
- ssm:SendCommand
- ssm:ListCommands

Document Steps

- aws:assertAwsResourceProperty - Checks if the Amazon EC2 instance is managed by Amazon Systems Manager.
- aws:runCommand - Checks if the Amazon EC2 instance platform is SLES.
- aws:runCommand - Checks if the package cloud-regionsrv-client version is greater than or equal to the required version 9.0.10.
- aws:runCommand - Checks if the symbolic link for base product is broken, and fixes the link it it is broken.
- aws:runCommand - Checks if the hosts file (/etc/hosts) contains records for smt-ec2-suscloud.net. The automation removes any duplicate entries.
- aws:runCommand - Checks if the curl command is installed.
- aws:runCommand - Checks if the Amazon EC2 instance can access the Instance Metadata Service (IMDS) address 169.254.169.254.
- aws:runCommand - Checks if the Amazon EC2 instance has a billing code or Amazon Web Services Marketplace product code.
- aws:runCommand - Checks if the Amazon EC2 instance can reach at least 1 regional server over HTTPS.
- aws:runCommand - Checks if the Amazon EC2 instance can reach the Subscription Management Tool (SMT) servers over HTTP.
- aws:runCommand - Checks if the Amazon EC2 instance can reach the Subscription Management Tool (SMT) servers over HTTPS.
- aws:runCommand - Checks if the Amazon EC2 instance can reach the smt-ec2.susecloud.net address over HTTPS.
- aws:runCommand - Registers the Amazon EC2 instance with SUSE Update Infrastructure.
- aws:executeScript - Gathers and outputs the output of all the previous steps.

AWSSupport-TroubleshootWindowsPerformance

Description

The runbook AWSSupport-TroubleshootWindowsPerformance helps troubleshoot ongoing performance issues on Amazon Elastic Compute Cloud (Amazon EC2) Windows instance. The runbook captures logs from the target instance and analyzes CPU, memory, disk, and network performance metrics. Optionally, the automation can capture a process dump to help you determine the potential cause of performance degradation. The automation also captures the event and system logs by using the latest EC2Rescue tool, if you allow this runbook to install it.

How does it work?

The runbook performs the following steps:

- Checks the Amazon EC2 instance for prerequisites.
- Generates performance logs in the root disk of the Amazon EC2 Windows instance
- Stores captured logs in folder C:\ProgramData\Amazon\SSM\TroubleshootWindowsPerformance
- If an Amazon Simple Storage Service (Amazon S3) bucket is provided, and the automation assume role has the required permissions, the captured logs are uploaded to the Amazon S3 bucket.
- Installs the latest EC2Rescue tool to the Amazon EC2 Windows instance to capture events and system logs if you choose to install it, but it does not analyze the process dump and logs captured by EC2Rescue.

Important

- To execute this runbook, the Amazon EC2 Windows instance must be managed by Amazon Systems Manager. For more information, see Why is my Amazon EC2 instance not displaying as a managed node.
- To execute this runbook, the Amazon EC2 Windows instance must be running on versions Windows 8.1 / Windows Server 2012 R2 (6.3) or newer with PowerShell 4.0 or above. For more information, see Windows Operating System version.
- For the generation of performance logs, at least 10 GB of free space on the root device is required. If the root disk is larger than 100 GB, the free space must be greater than 10%

of the disk size. If you dump a process during execution, the free space must be greater than 10 GB plus the total memory size consumed by the process when the process consumes more than 10 GB memory.

- The logs generated on the root device are not deleted automatically.
- The runbook does not uninstall the `EC2Rescue` tool. For more information, see Use EC2Rescue for Windows Server.
- It is best practice to run this automation during a performance impact. You can also run it periodically using an Amazon Systems Manager State Manager association or by scheduling Amazon Systems Manager Maintenance Windows.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`
- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeInstanceInformation`
- `ssm:GetAutomationExecution`
- `ssm:ListCommands`
- `ssm:ListCommandInvocations`
- `ssm:SendCommand`

- s3:ListBucket
- s3:GetEncryptionConfiguration
- s3:GetBucketPublicAccessBlock
- s3:GetBucketPolicyStatus
- s3:PutObject
- s3:GetBucketAcl
- s3:GetAccountPublicAccessBlock

(Optional) The IAM role attached on the instance profile or IAM user configured on the instance requires the following actions to upload logs to the Amazon S3 bucket specified for parameter `LogUploadBucketName`:

- s3:PutObject
- s3:GetObject
- s3:ListBucket

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-TroubleshootWindowsPerformance in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **InstanceId (Required):**

     The ID of the target Amazon EC2 Windows instance where you want to run the automation. The instance must be managed by Systems Manager to execute the automation.
   - **CaptureProcessDump (Optional):**

The process dump type to capture. The automation can capture one process dump for the process that is potentially causing the performance impact in the beginning of the automation. The instance root volume requires at least 10 GB free space (greater than 10% of the disk size when the root volume size is bigger than 100 GB, and 10 GB plus the total memory size consumed by the process when the process consumes more than 10 GB memory).

- **LogCaptureDuration (Optional):**

  The number of minutes, between 1 and 15, that this automation will capture logs while the issue is present. Default is 5.

- **LogUploadBucketName (Optional):**

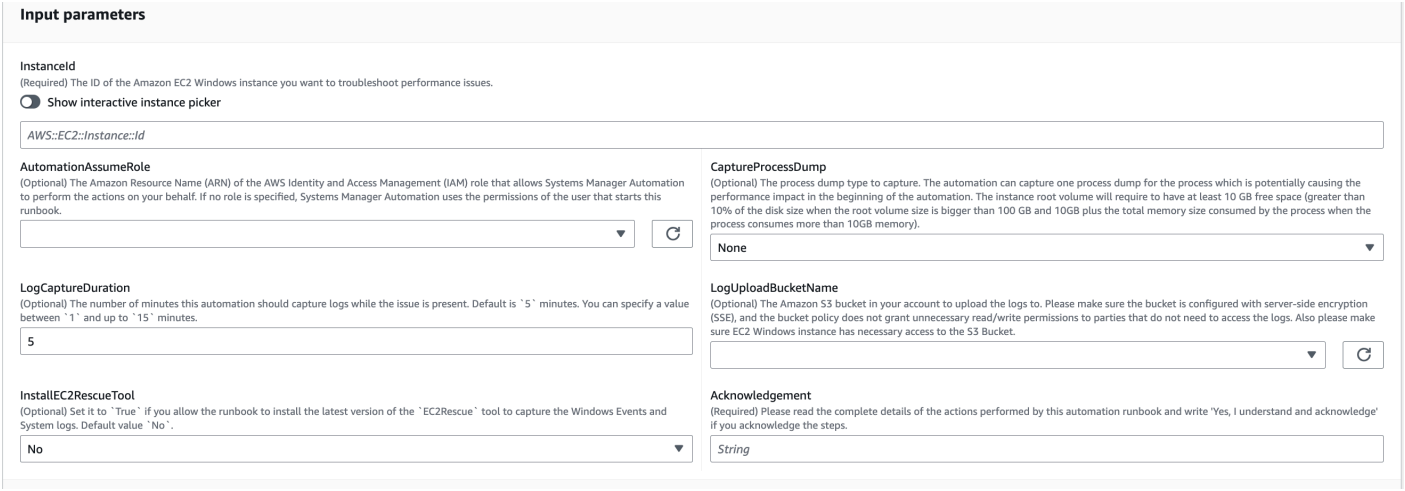
  The Amazon S3 bucket in your account where you want to upload the logs. The bucket must be configured with server-side encryption (SSE), and the bucket policy must not grant unnecessary read/write permissions to parties that do not need access to the captured logs. The Amazon EC2 Windows instance must have access to the Amazon S3 bucket.

- **InstallEC2RescueTool (Optional):**

  Set to `Yes` to allow the runbook to install the latest version of the `EC2Rescue` tool to capture the Windows Events and System logs. Default is `No`.

- **Acknowledgement (Required):**

  Read the complete details of the actions performed by this automation runbook and if you agree, type `Yes, I understand and acknowledge`.



4. Select Execute.

5. The automation initiates.
6. The document performs the following steps:

- **`CheckConcurrency:`**

  Ensures that there is only one execution of this runbook targeting the instance. If the runbook finds another execution targeting the same instance, it returns an error and ends.

- **`AssertInstanceIsWindows:`**

  Asserts that the Amazon EC2 instance is running on Windows Operating System. Otherwise, the automation ends.

- **`AssertInstanceIsManagedInstance:`**

  Asserts that the Amazon EC2 instance is managed by Amazon Systems Manager. Otherwise the automation ends.

- **`VerifyPrerequisites:`**

  Verifies the PowerShell version on the instance OS and ensures that the instance can be connected through Systems Manager to run PowerShell commands. This automation supports PowerShell 4.0 and above running on versions Windows 8.1 / Server 2012 R2 (6.3) or newer. If the version is older, the automation fails. When you choose to upload logs to Amazon S3 bucket, this automation Checks that the Amazon Tools for PowerShell module is available. If not, the automation ends.

- **`BranchOnProcessDump:`**

  Branches based on if you set it to capture the dump of processes that impacted performance.

- **`CaptureProcessDump:`**

  Checks if the instance has enough space to run this automation (when you choose Highest CPU / Memory).

- **`CapturePerformanceLogs:`**

  Checks the disk space again and runs the PowerShell script on the instance to create perfmon counters and start Performance Monitor and Windows Performance Recorder logging. The script stops after the defined `LogCaptureDuration` is met.

- **`SummarizePerformanceLogs:`**

Summarizes the XML report generated on the previous step, `CapturePerformanceLogs`, to find the responsible process consuming the most WorkingSet64 (Memory) and % Processor Time (CPU) shown as output on the automation. It generates similar information for usage of LogicalDisk, Network Interface, Memory, TCPv4, IPv4, and UDPv4 and saves it to `analysis_output.log` in the output folder.

- **`BranchOnInstallEC2Rescue:`**

  Branches if you set it to install the latest `EC2Rescue` tool in the Amazon EC2 instance.

- **`InstallEC2RescueTool:`**

  Installs the `EC2Rescue` tool in the instance OS to capture `EC2Rescue` logs using `AWS-ConfigureAWSPackage`.

- **`RunEC2RescueTool:`**

  Runs the `EC2Rescue` tool in the instance OS to capture all logs needed. `EC2Rescue` captures only the required logs to save space.

- **`BranchOnIfS3BucketProvided:`**

  Branches based on user input of `LogUploadBucketName` to see if there is a bucket name available to upload logs.

- **`GetS3BucketPublicStatus:`**

  Determines if an Amazon S3 bucket is provided, and if so, confirms that the Amazon S3 bucket is not public and is configured with SSE.

- **`UploadLogResult:`**

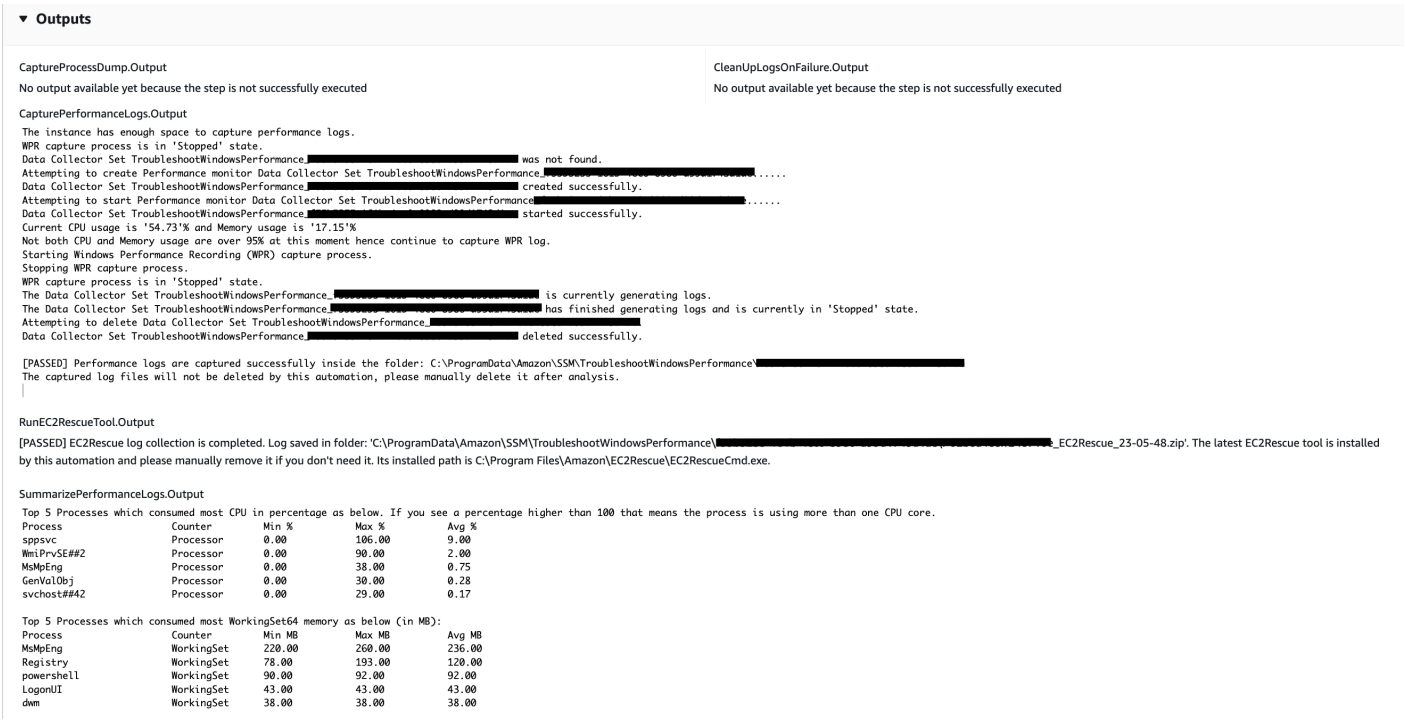
  Uploads the logs to the Amazon S3 bucket provided. If the PowerShell version is 5.0 or above, it compresses the logs to a ZIP archive and uploads them. It deletes the ZIP file after upload completes. If the PowerShell version is below 5.0, it uploads the files directly to a folder.

- **`CleanUpLogsOnFailure:`**

  Cleans all the logs generated by the `CapturePerformanceLogs` step when it fails. The `CleanUpLogsOnFailure` step may fail or timeout if SSM Agent isn't working correctly, or the Windows system is unresponsive.

7. After completed, review the Outputs section for the detailed results of the execution:

Execution where the target instance has all required prerequisites.



Outputs

CaptureProcessDump.Output

No output available yet because the step is not successfully executed

CleanUpLogsOnFailure.Output

No output available yet because the step is not successfully executed

CapturePerformanceLogs.Output

```
The instance has enough space to capture performance logs.
WPR capture process is in 'Stopped' state.
Data Collector Set TroubleshootWindowsPerformance_[redacted] was not found.
Attempting to create Performance monitor Data Collector Set TroubleshootWindowsPerformance_[redacted].....
Data Collector Set TroubleshootWindowsPerformance_[redacted] created successfully.
Attempting to start Performance monitor Data Collector Set TroubleshootWindowsPerformance[redacted].....
Data Collector Set TroubleshootWindowsPerformance_[redacted] started successfully.
Current CPU usage is '54.73'% and Memory usage is '17.15'%
Not both CPU and Memory usage are over 95% at this moment hence continue to capture WPR log.
Starting Windows Performance Recording (WPR) capture process.
Stopping WPR capture process.
WPR capture process is in 'Stopped' state.
The Data Collector Set TroubleshootWindowsPerformance_[redacted] is currently generating logs.
The Data Collector Set TroubleshootWindowsPerformance_[redacted] has finished generating logs and is currently in 'Stopped' state.
Attempting to delete Data Collector Set TroubleshootWindowsPerformance_[redacted]
Data Collector Set TroubleshootWindowsPerformance_[redacted] deleted successfully.

[PASSED] Performance logs are captured successfully inside the folder: C:\ProgramData\Amazon\SSM\TroubleshootWindowsPerformance\[redacted]
The captured log files will not be deleted by this automation, please manually delete it after analysis.
```

RunEC2RescueTool.Output

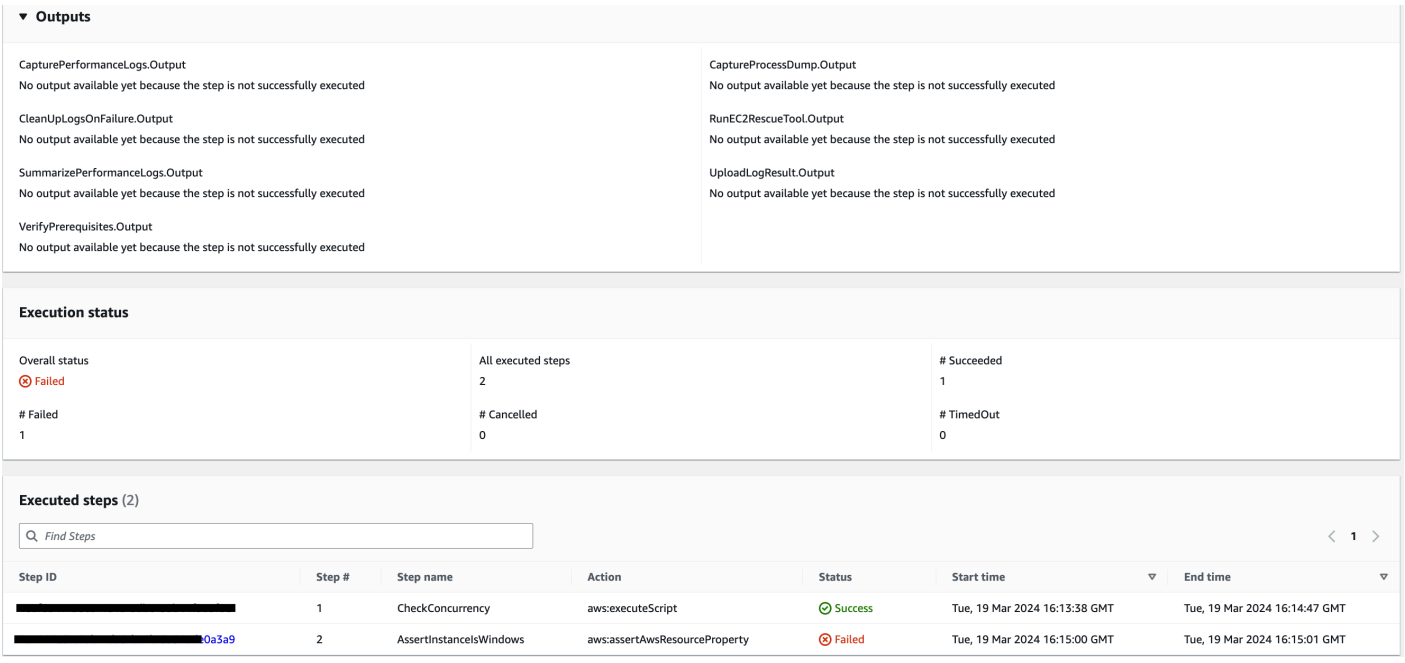
[PASSED] EC2Rescue log collection is completed. Log saved in folder: 'C:\ProgramData\Amazon\SSM\TroubleshootWindowsPerformance\[redacted]_EC2Rescue_23-05-48.zip'. The latest EC2Rescue tool is installed by this automation and please manually remove it if you don't need it. Its installed path is C:\Program Files\Amazon\EC2Rescue\EC2RescueCmd.exe.

SummarizePerformanceLogs.Output

```
Top 5 Processes which consumed most CPU in percentage as below. If you see a percentage higher than 100 that means the process is using more than one CPU core.
Process                 Counter         Min %           Max %           Avg %
sppsvc                  Processor       0.00            106.00          9.00
WmiPrvSE##2             Processor       0.00            90.00           2.00
MsMpEng                 Processor       0.00            38.00           0.75
GenValObj               Processor       0.00            30.00           0.28
svchost##42             Processor       0.00            29.00           0.17

Top 5 Processes which consumed most WorkingSet64 memory as below (in MB):
Process                 Counter         Min MB          Max MB          Avg MB
MsMpEng                 WorkingSet      220.00          260.00          236.00
Registry                WorkingSet      78.00           193.00          120.00
powershell              WorkingSet      90.00           92.00           92.00
LogonUI                 WorkingSet      43.00           43.00           43.00
dwm                     WorkingSet      38.00           38.00           38.00
```

Execution where the target instance is on Linux platform and the execution failed. You would select the step ID to see the failure details.



Outputs

CapturePerformanceLogs.Output

No output available yet because the step is not successfully executed

CaptureProcessDump.Output

No output available yet because the step is not successfully executed

CleanUpLogsOnFailure.Output

No output available yet because the step is not successfully executed

RunEC2RescueTool.Output

No output available yet because the step is not successfully executed

SummarizePerformanceLogs.Output

No output available yet because the step is not successfully executed

UploadLogResult.Output

No output available yet because the step is not successfully executed

VerifyPrerequisites.Output

No output available yet because the step is not successfully executed

Execution status

Overall status	All executed steps	# Succeeded
Failed	2	1
# Failed	**# Cancelled**	**# TimedOut**
1	0	0

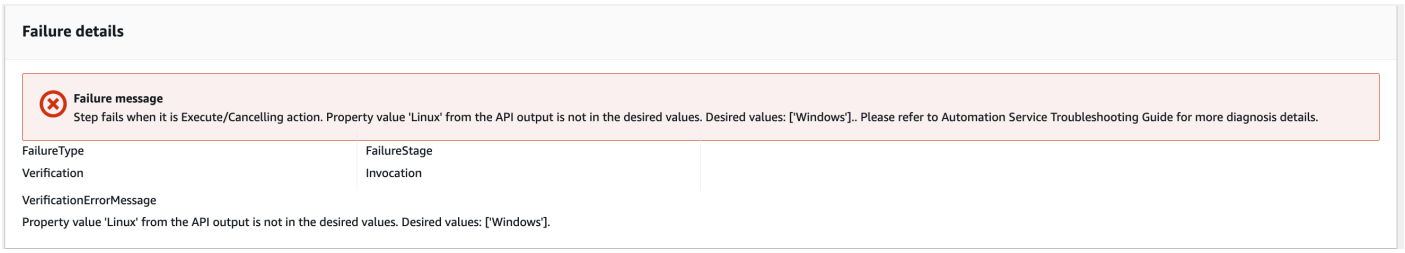
Executed steps (2)

Find Steps

1

Step ID	Step #	Step name	Action	Status	Start time	End time
[redacted]	1	CheckConcurrency	aws:executeScript	Success	Tue, 19 Mar 2024 16:13:38 GMT	Tue, 19 Mar 2024 16:14:47 GMT
[redacted]0a3a9	2	AssertInstanceIsWindows	aws:assertAwsResourceProperty	Failed	Tue, 19 Mar 2024 16:15:00 GMT	Tue, 19 Mar 2024 16:15:01 GMT

The failure details of step `AssertInstanceIsWindows`.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-TroubleshootWindowsUpdate

Description

The `AWSSupport-TroubleshootWindowsUpdate` runbook is used to identify issues that could fail the Windows updates for Amazon Elastic Compute Cloud (Amazon EC2) Windows instances.

How does it work?

The runbook performs the following steps:

- Checks if the target Amazon EC2 instance is managed by Amazon Systems Manager.
- Checks if the Amazon Systems Manager Agent (SSM Agent) and Windows Server versions are supported for Systems Manager patching operations.
- Checks the available disk space recommended for Windows updates and if a reboot is pending. A pending reboot normally indicates that updates are pending, and a reboot is required before performing additional updates.
- Configures the proxy settings at the operating system level, which can help troubleshoot connectivity issues.
- Performs an Amazon Simple Storage Service (Amazon S3) endpoint connectivity test and calls the `GetDeployablePatchSnapshotForInstance` API operation to retrieve the current snapshot for the patch baseline the managed node uses.

- If the connection fails, provides the option to run the `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2` runbook to analyze the instance's connectivity to Amazon S3 endpoints.
- Validates the Windows updates configuration and tests Windows Server Update Services (WSUS) (if applicable).

Important

- Active Directory domain controllers are not supported.
- Windows Server version 2008 R2 or previous versions are not supported.
- SSM Agent 1.2.371 or previous versions are not supported.
- The `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2` runbook uses VPC Reachability Analyzer to analyze the network connectivity between a source and a service endpoint. You are charged per analysis run between a source and destination. For more details, see Amazon VPC Pricing.
- The `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2` runbook is not available in all regions where Systems Manager is supported.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ssm:DescribeInstanceInformation
- ssm:SendCommand
- ssm:ListCommandInvocations
- ssm:ListCommands

> **Note**
>
> To run the child runbook AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2, add the permissions listed in this document.

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-TroubleshootWindowsUpdate in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

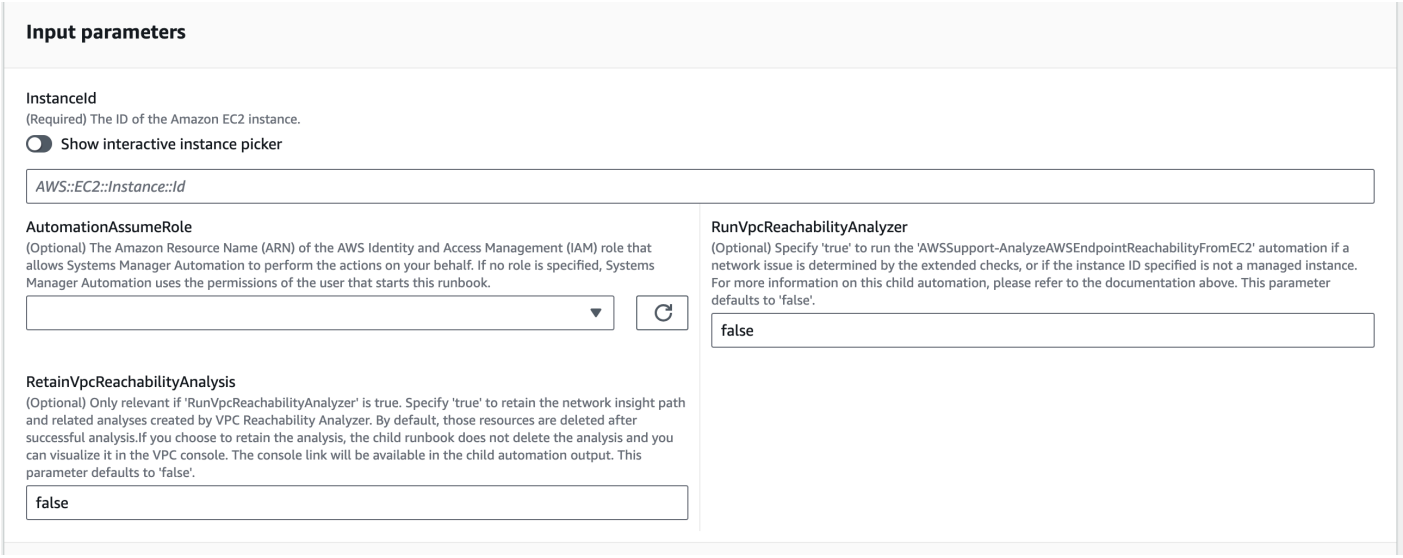
     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **InstanceId (Required):**

     Enter the ID of the Amazon EC2 instance where the Windows update failed.
   - **RunVpcReachabilityAnalyzer (Optional):**

     Specify true to run the AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2 automation if a network issue is determined by the extended checks or if the instance ID specified is not a managed instance. For more information on this child automation, refer to the documentation. The default value is false.

- **RetainVpcReachabilityAnalysis (Optional):**

  Only relevant if `RunVpcReachabilityAnalyzer` is `true`. Specify `true` to retain the network insight path and related analyses created by `Reachability Analyzer`. By default, those resources are deleted after successful analysis. If you choose to retain the analysis, the child runbook does not delete the analysis and you can visualize it in the Amazon VPC console. The console link will be available in the child automation output. The default value `false`.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

   - **`getWindowsServerAndSSMAgentVersion:`**

     Verifies that the target instance is managed by Amazon Systems Manager and gets details about the SSM Agent version and Windows version.

   - **`assertifInstanceIsSsmManaged:`**

     Ensures the Amazon EC2 instance is managed by Amazon Systems Manager (SSM), otherwise the automation ends.

   - **`CheckProxy:`**

     Checks for all proxy types for the Windows instance.

   - **`CheckPrerequisites:`**

Gets the SSM Agent version and Windows version, and determines if it is an Active Directory Domain Controller (DC). If the instance is a DC or the SSM Agent or Windows version is not supported, the runbook stops.

- **CheckDiskSpace:**

  Gets and validates the available disk space over the Windows instance if it is sufficient for performing the Windows update.

- **CheckPendingReboot:**

  Checks for any pending reboot over the Windows instance.

- **CheckS3Connectivity:**

  Checks if the instance can reach the Amazon S3 endpoints for `Patchbaseline`.

- **branchOnRunVpcReachabilityAnalyzer:**

  If `RunVpcReachabilityAnalyzer` is true, then it branches the automation to run deeper analysis for the debugging Amazon S3 connectivity.

- **GenerateEndpoints:**

  Generates an endpoint to have an extended connectivity check for the Amazon S3 endpoint.

- **analyzeAwsEndpointReachabilityFromEC2:**

  Calls the automation runbook, `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2`. to check the reachability of the selected instance to the required endpoints.

- **CheckWindowsUpdateServices:**

  Checks the Windows Update service status and start type.

- **CheckWindowsUpdateSettings:**

  Checks for Windows Update policies configured over the Windows instance.

- **CheckWSUSSettings:**

  Checks whether the Windows update is configured with WSUS or Microsoft Update Catalog and verifies connectivity.

- **CheckWUGlobalSettings:**

Checks the Windows Update global settings configured over the Windows instance.

- **`GenerateLogs:`**

  Downloads Windows Update logs and CBS logs onto the instance desktop and checks Windows event logs for failure.

- **`FinalReport:`**

  Generates a complete report of all steps.

7. After completed, review the Outputs section for the detailed results of the execution:

```
FinalReport.Results
"
==================================================Prerequisites Check==================================================
Result: ✅ [PASSED]
INFO: The target instance is not an Active Directory Domain Controller.
INFO: The platform 10.0.20348 is supported.
INFO: The SSM Agent version 3.2.1705.0 is supported.

====================================================Disk Space Check====================================================
Result: ✅ [PASSED]
INFO: Disk space on drive C: is recommended to run Windows updates.

==================================================Pending Reboot Check==================================================
Result: ✅ [PASSED]
INFO: There is no pending reboot.

================================================Amazon S3 Connectivity Check================================================
Result: ✅ [PASSED]
Calling GetDeployablePatchSnapshotForInstance API ...
VERBOSE: Invoking AWS Systems Manager operation 'GetDeployablePatchSnapshotForInstance' in region 'eu-west-1'
Downloading Windows Patching file...
Downloading Windows Patching file, attempt: 1/5...
INFO: Deployable Patch Snapshot downloaded successfully

==============================AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2==============================
Result: ✅ [PASSED]
Calling GetDeployablePatchSnapshotForInstance API ...
VERBOSE: Invoking AWS Systems Manager operation 'GetDeployablePatchSnapshotForInstance' in region 'eu-west-1'
Downloading Windows Patching file...
Downloading Windows Patching file, attempt: 1/5...
INFO: Deployable Patch Snapshot downloaded successfully

============================================Windows Update Services Status============================================
Result: ✅ [PASSED]
Getting Services Status and types for Windows Update...
The service 'Application Identity' is currently 'Stopped' and is set to 'Manual'
Starting the service: 'Application Identity'
Service 'Application Identity' started successfully
The service 'Background Intelligent Transfer Service' is currently 'Stopped' and is set to 'Manual'
Starting the service: 'Background Intelligent Transfer Service'
Service 'Background Intelligent Transfer Service' started successfully
INFO: The service 'Cryptographic Services' status is currently 'Running'
The service 'Windows Installer' is currently 'Stopped' and is set to 'Manual'
Starting the service: 'Windows Installer'
Service 'Windows Installer' started successfully
INFO: The service 'Windows Modules Installer' status is currently 'Running'
INFO: The service 'Windows Update' status is currently 'Running'

================================================Windows Proxy Settings================================================
Result: ✅ [PASSED]
No WinInet Proxy is set on the system
No Winhttp Proxy is set on the system
There is no proxy setting for SSM Agent
System Wide Environment HTTP Proxy is not set.
System Wide Environment HTTPS Proxy is not set.
System Wide Environment NO PROXY is not set.
There is no HTTP Proxy configured at local system account user environment.


============================================Windows Update Settings============================================
Result: ✅ [PASSED]
INFO: Windows Update (Policies): Never check for updates
INFO: To modify this setting is in Computer Configuration\\Adminisrative Template\\Windows Component\\Windows
Update\\Configure Automatic Updates. For more details please check this document: https://learn.microsoft.com/de-
de/security-updates/windowsupdateservices/18127451

============================================Windows Update Global Settings============================================
Result: ✅ [PASSED]
Windows Update Client has no restrictions

=========================================Copy of Windows Update and CBS logs=========================================
Result: ✅ [PASSED]
No errors found in Microsoft-Windows-WindowsUpdateClient events.
INFO: Logs copied to the C:\\Windows\\TEMP\\c176a507-d074-4402-8a5b-631dd643f33a folder
"
```

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation

- Setting up an Automation
- Support Automation Workflows landing page

Documentation related to the Amazon service

- Refer to the article, TroubleShoot Windows Update, for more information.

AWSSupport-UpgradeWindowsAWSDrivers

Description

The `AWSSupport-UpgradeWindowsAWSDrivers` runbook upgrades or repairs storage and network Amazon drivers on the specified EC2 instance. The runbook attempts to install the latest versions of Amazon drivers online by calling SSM Agent. If SSM Agent is not contactable, the runbook can perform an offline installation of the Amazon drivers if explicitly requested.

This runbook supports the following operating systems:

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows Server 2025

Note

Both the online and offline upgrade will create an AMI before attempting any operations, which will persist after the automation completes. It is your responsibility to secure access to the AMI, or to delete it. The online method restarts the instance as part of the upgrade process, while the offline method requires the provided EC2 instance be stopped and then started.

Important

If your instances connect to Amazon Systems Manager using VPC endpoints, this runbook will fail unless used in the us-east-1 Region. This runbook will also fail on a domain

controller. To update Amazon PV drivers on a domain controller, see Upgrade a Domain Controller (Amazon PV Upgrade).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AllowOffline

  Type: String

  Valid values: true | false

  Default: false

  Description: (Optional) Set it to true if you allow an offline drivers upgrade in case the online installation cannot be performed. Note: The offline method requires the provided EC2 instance be stopped and then started. Data stored in instance store volumes will be lost. The public IP address will change if you are not using an Elastic IP.

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ForceUpgrade

Type: String

Valid values: true | false

Default: false

Description: (Optional) Offline only - Set it to true if you allow the offline drivers upgrade to proceed even though your instance already has the latest drivers installed.

- InstanceId

  Type: String

  Description: (Required) ID of your EC2 instance for Windows Server.

- SubnetId

  Type: String

  Default: SelectedInstanceSubnet

  Description: (Optional) Offline only - The subnet ID for the EC2Rescue instance used to perform the offline drivers upgrade. If no subnet ID is specified, Systems Manager Automation will create a new VPC.

  > **Important**
  >
  > The subnet must be in the same Availability Zone as InstanceId, and it must allow access to the SSM endpoints.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

The EC2 instance receiving the command must at minimum have an IAM role that includes permissions for **ssm:StartAutomationExecution** and **ssm:SendCommand** to run the automation and send the command to the instance, plus **ssm:GetAutomationExecution** to be able to read the automation output. You can attach the `AmazonSSMManagedInstanceCore` Amazon managed policy to your IAM role to provide these permissions. We recommend, however, using the

Automation IAM role AmazonSSMAutomationRole for this purpose. For more information, see Use IAM to configure roles for Automation.

If you are performing an offline upgrade, see the permissions required by AWSSupport-StartEC2RescueWorkflow.

Document Steps

1. aws:assertAwsResourceProperty - Verifies the input instance is Windows.
2. aws:assertAwsResourceProperty - Verifies the input instance is a managed instance. If so, the online upgrade starts, otherwise the offline upgrade is evaluated.
   a. (Online upgrade) If the input instance is a managed instance:
      i. aws:createImage - Creates an AMI backup.
      ii. aws:createTags - Tags the AMI backup.
      iii. aws:runCommand - Installs ENA network driver via AWS-ConfigureAWSPackage.
      iv. aws:runCommand - Installs NVMe driver via AWS-ConfigureAWSPackage.
      v. aws:runCommand - Installs Amazon PV driver via AWS-ConfigureAWSPackage.
   b. (Offline upgrade) If the input instance is not a managed instance:
      i. aws:assertAwsResourceProperty - Verifies the AllowOffline flag is set to true. If so, the offline upgrade starts, otherwise the automation ends.
      ii. aws:changeInstanceState - Stop the source instance.
      iii. aws:changeInstanceState - Force-stop the source instance.
      iv. aws:createImage - Create an AMI backup of the source instance.
      v. aws:createTags - Tag the AMI backup of the source instance.
      vi. aws:executeAwsApi - Enable ENA for the instance
      vii. aws:assertAwsResourceProperty - Assert the ForceUpgrade flag.
      viii. Force offline upgrade) If **ForceUpgrade = true** then run aws:executeAutomation to invoke AWSSupport-StartEC2RescueWorkflow with the drivers force upgrade script. This installs the drivers regardless of the current version that is installed
      ix. (Offline upgrade) If **ForceUpgrade = false** then run aws:executeAutomation to invoke AWSSupport-StartEC2RescueWorkflow with the drivers upgrade script.

Outputs

preUpgradeBackup.ImageId

preOfflineUpgradeBackup.ImageId

`installAwsEnaNetworkDriverOnInstance.Output`

`installAWSNVMeOnInstance.Output`

`installAWSPVDriverOnInstance.Output`

upgradeDriversOffline.Output

forceUpgradeDriversOffline.Output

Amazon ECS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Elastic Container Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-CollectECSInstanceLogs
- AWS-InstallAmazonECSAgent
- AWS-ECSRunTask
- AWSSupport-TroubleshootECSContainerInstance
- AWSSupport-TroubleshootECSTaskFailedToStart
- AWS-UpdateAmazonECSAgent

`AWSSupport-CollectECSInstanceLogs`

Description

The `AWSSupport-CollectECSInstanceLogs` runbook collects operating system and Amazon Elastic Container Service (Amazon ECS) related log files from an Amazon Elastic Compute Cloud (Amazon EC2) instance to help you troubleshoot common Amazon ECS issues. While the automation is collecting the associated log files, changes are made to the file system. These changes include the creation of temporary directories and a log directory, the copying of log files to these directories, and compressing the log files into an archive.

If you specify a value for the `LogDestination` parameter, the target instance must have the Amazon Command Line Interface (Amazon CLI) for Linux instances or Amazon Tools for Windows PowerShell for Windows instances installed. The automation evaluates the policy status of the Amazon Simple Storage Service (Amazon S3) bucket you specify. To help with the security of the logs gathered from your Amazon EC2 instance, if the policy status `isPublic` is set to `true` , or if the access control list (ACL) grants `READ|WRITE` permissions to the `All Users` Amazon S3 predefined group, the logs are not uploaded. Additionaly, if the provided bucket is not available in your account, the logs are not uploaded. For more information about Amazon S3 predefined g roups, see Amazon S3 predefined groups in the *Amazon Simple Storage Service User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ECSInstanceId

  Type: String

  Description: (Required) The ID of the instance you want to collect logs from. The instance you specify must be managed by Systems Manager.
- LogDestination

Type: String

Description: (Optional) The Amazon S3 bucket in your Amazon Web Services account to upload the archived logs to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:ListCommandInvocations`
- `ssm:ListCommands`
- `ssm:SendCommand`
- `ssm:DescribeInstanceInformation`

We recommend that the Amazon EC2 instance you specify in the `ECSInstanceId` parameter has an IAM role with the `AmazonSSMManagedInstanceCore` Amazon managed policy attached. To upload the log archive to the Amazon S3 bucket you specify in the `LogDestination` parameter, you must add following permissions:

- `s3:PutObject`
- `s3:ListBucket`
- `s3:GetBucketPolicyStatus`
- `s3:GetBucketAcl`

Document Steps

- `assertInstanceIsManaged` - Verifies whether the instance you specify in the `ECSInstanceId` parameter is managed by Systems Manager.
- `getInstancePlatform` - Gets information about the operating system (OS) platform of the instance specified in the `ECSInstanceId` parameter.
- `verifyInstancePlatform` - Branches the automation based on the OS platform.
- `runLogCollectionScriptOnLinux` - Gathers operating system and Amazon ECS related log files on Linux instances and creates an archive file in the `/var/log/collectECSlogs` directory.

- `runLogCollectionScriptOnWindows` - Gathers operating system and Amazon ECS related log files on Windows instances and creates an archive file in the `C:\ProgramData\collectECSlogs` directory.
- `verifyIfS3BucketProvided` - Verifies whether a value was specified for the `LogDestination` parameter.
- `runUploadScript` - Branches the automation step based on the OS platform.
- `runUploadScriptOnLinux` - Uploads the log archive to the Amazon S3 bucket specified in the `LogDestination` parameter and deletes the archived log file from OS.
- `runUploadScriptOnWindows` - Uploads the log archive to the Amazon S3 bucket specified in the `LogDestination` parameter and deletes the archived log file from OS.

`AWS-InstallAmazonECSAgent`

Description

The `AWS-InstallAmazonECSAgent` runbook installs the Amazon Elastic Container Service (Amazon ECS) agent on the Amazon Elastic Compute Cloud (Amazon EC2) instance you specify. This runbook only supports Amazon Linux and Amazon Linux 2 instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceIds

  Type: StringList

  Description: (Required) The IDs of the Amazon EC2 instances you want to install the Amazon ECS agent on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:GetCommandInvocation`
- `ec2:DescribeImages`
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeInstances`

Document Steps

`aws:executeScript` - Installs the Amazon ECS agent on the Amazon EC2 instances you specify in the `InstanceIds` parameter.

Outputs

InstallAmazonECSAgent.SuccessfulInstances - The ID of the instance where installation of the Amazon ECS agent succeeded.

InstallAmazonECSAgent.FailedInstances - The ID of the instance where installation of the Amazon ECS agent failed.

InstallAmazonECSAgent.InProgressInstances - The ID of the instance where installation of the Amazon ECS agent is in progress.

AWS-ECSRunTask

Description

The AWS-ECSRunTask runbook runs the Amazon Elastic Container Service (Amazon ECS) task that you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- capacityProviderStrategy

  Type: String

  Description: (Optional) The capacity provider strategy to use for the task.
- cluster

  Type: String

  Description: (Optional) The short name or ARN of the cluster to run your task on. If you do not specify a cluster, the default cluster is used.
- count

Type: String

Description: (Optional) The number of instantiations of the specified task to place on your cluster. You can specify up to 10 tasks for each request.

- enableECSManagedTags

  Type: Boolean

  Description: (Optional) Specifies whether to use Amazon ECS managed tags for the task. For more information, see Tagging your Amazon ECS resources in the *Amazon Elastic Container Service Developer Guide*.

- enableExecuteCommand

  Type: Boolean

  Description: (Optional) Determines whether to activate the execute command functionality for the containers in this task. If true, this activates execute command functionality on all containers in the task.

- group

  Type: String

  Description: (Optional) The name of the task group to associate with the task. The default value is the family name of the task definition. For example, `family:my-family-name`.

- launchType

  Type: String

  Valid values: EC2 | FARGATE | EXTERNAL

  Description: (Optional) The infrastructure to run your standalone task on.

- networkConfiguration

  Type: String

  Description: (Optional) The network configuration for the task. This parameter is required for task definitions that use the `awsvpc` network mode to receive their own elastic network interface, and it isn't supported for other network modes.

- overrides

Type: String

Description: (Optional) A list of container overrides in JSON format that specify the name of a container in the specified task definition and the overrides it should receive. You can override the default command for a container that's specified in the task definition or Docker image with a command override. You can also override existing environment variables that are specified in the task definition or Docker image on a container. Additionally, you can add new environment variables with an environment override.

- placementConstraints

  Type: String

  Description: (Optional) An array of placement constraint objects to use for the task. You can specify up to 10 constraints for each task including constraints in the task definition and those specified at runtime.

- placementStrategy

  Type: String

  Description: (Optional) The placement strategy objects to use for the task. You can specify a maximum of 5 strategy rules for each task.

- platformVersion

  Type: String

  Description: (Optional) The platform version that the task uses. A platform version is only specified for tasks hosted on Fargate. If a platform version isn't specified, the LATEST platform version is used.

- propagateTags

  Type: String

  Description: (Optional) Determines whether tags propagate from the task definition to the task. If no value is specified, the tags aren't propagated. Tags can only be propagated to the task during task creation.

- referenceId

  Type: String

Description: (Optional) The reference ID to use for the task. The reference ID can have a maximum length of 1024 characters.

- startedBy

  Type: String

  Description: (Optional) An optional tag specified when a task is started. This helps you identify which tasks belong to a specific job by filtering the results of a `ListTasks` API operation. Up to 36 letters (uppercase and lowercase), numbers, hyphens (-), and underscores (_) are allowed.

- tags

  Type: String

  Description: (Optional) Metadata that you want to apply to the task to help you categorize and organize tasks. Each tag consists of a user-defined key and value.

- taskDefinition

  Type: String

  Description: (Optional) The `family` and `revision` (`family:revision`) or full ARN of the task definition to run. If a revision isn't specified, the latest `ACTIVE` revision is used.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ecs:RunTask`

Document Steps

`aws:executeScript` - Runs the Amazon ECS task based on the values that you specify for the runbook input parameters.

AWSSupport-TroubleshootECSContainerInstance

Description

The AWSSupport-TroubleshootECSContainerInstance runbook helps you troubleshoot an Amazon Elastic Compute Cloud (Amazon EC2) instance that fails to register with an Amazon ECS cluster. This automation reviews whether the user data for the instance contains the correct cluster information, whether the instance profile contains the required permissions, and network configuration issues.

> **Important**
>
> To successfully run this automation, the state of your Amazon EC2 instance must be `running` , and the Amazon ECS cluster state must be `ACTIVE` .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterName

  Type: String

  Description: (Required) The name of the Amazon ECS cluster that the instance failed to register with.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance you want to troubleshoot.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeIamInstanceProfileAssociations`
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeVpcs`
- `iam:GetInstanceProfile`
- `iam:GetRole`
- `iam:SimulateCustomPolicy`
- `iam:SimulatePrincipalPolicy`

Document Steps

aws:executeScript: Reviews whether the Amazon EC2 instance meets the prerequisites needed to register with an Amazon ECS cluster.

AWSSupport-TroubleshootECSTaskFailedToStart

Description

The `AWSSupport-TroubleshootECSTaskFailedToStart` runbook helps you troubleshoot why an Amazon Elastic Container Service (Amazon ECS) task in an Amazon ECS cluster failed to start.

You must run this runbook in the same Amazon Web Services Region as your task that failed to start. The runbook analyzes the following common issues that can prevent a task from starting:

- Network connectivity to the configured container registry
- Missing IAM permissions required by the task execution role
- VPC endpoint connectivity
- Security group rule configuration
- Amazon Secrets Manager secrets references
- Logging configuration

> **Note**
>
> If the analysis determines that network connectivity needs to be tested, a Lambda function and requisite IAM role are created in your account. These resources are used to simulate the network connectivity of your failed task. The automation deletes these resources when they're no longer required. However, if the automation fails to delete the resources, you must do so manually.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ClusterName

  Type: String

  Description: (Required) The name of the Amazon ECS cluster where the task failed to start.

- CloudwatchRetentionPeriod

  Type: Integer

  Description: (Optional) The retention period, in days, for the Lambda function logs to be stored in Amazon CloudWatch Logs. This is only necessary if the analysis determines network connectivity needs to be tested.

  Valid values: 1 | 3 | 5 | 7 | 14 | 30 | 60 | 90

  Default: 30

- TaskId

  Type: String

  Description: (Required) The ID of the failed task. Use the most recently failed task.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudtrail:LookupEvents`
- `ec2:DeleteNetworkInterface`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeIamInstanceProfileAssociations`
- `ec2:DescribeSecurityGroups`

- ec2:DescribeNetworkAcls
- ec2:DescribeNetworkInterfaces
- ec2:DescribeRouteTables
- ec2:DescribeSubnets
- ec2:DescribeVpcEndpoints
- ec2:DescribeVpcs
- ecr:DescribeImages
- ecr:GetRepositoryPolicy
- ecs:DescribeContainerInstances
- ecs:DescribeServices
- ecs:DescribeTaskDefinition
- ecs:DescribeTasks
- iam:AttachRolePolicy
- iam:CreateRole
- iam:DeleteRole
- iam:DetachRolePolicy
- iam:GetInstanceProfile
- iam:GetRole
- iam:ListRoles
- iam:PassRole
- iam:SimulateCustomPolicy
- iam:SimulatePrincipalPolicy
- kms:DescribeKey
- lambda:CreateFunction
- lambda:DeleteFunction
- lambda:GetFunctionConfiguration
- lambda:InvokeFunction
- lambda:TagResource

- logs:DescribeLogGroups
- logs:PutRetentionPolicy
- secretsmanager:DescribeSecret
- ssm:DescribeParameters
- sts:GetCallerIdentity

Document Steps

- aws:executeScript - Verifies that the user or role who started the automation has the required IAM permissions. If you don't have sufficient permissions to use this runbook, the missing required permissions are included in the output of the automation.
- aws:branch - Branches based on whether you have permissions to all required actions for the runbook.
- aws:executeScript - Creates a Lambda function in your VPC if the analysis determines network connectivity needs to be tested.
- aws:branch - Branches based on the results of the previous step.
- aws:executeScript - Analyzes possible causes for the failure to start your task.
- aws:executeScript - Deletes resources created by this automation.
- aws:executeScript - Formats the output of the automation to return the results of the analysis to the console. You can review the analysis after this step before the automation completes.
- aws:branch - Branches based on whether the Lambda function and associated resources were created and need to be deleted.
- aws:sleep - Sleeps for 30 minutes so the elastic network interface for the Lambda function can be deleted.
- aws:executeScript - Deletes the Lambda function network interface.
- aws:executeScript - Formats the output of the Lambda function network interface deletion step.

AWS-UpdateAmazonECSAgent

Description

The AWS-UpdateAmazonECSAgent runbook updates the Amazon Elastic Container Service (Amazon ECS) agent on the Amazon Elastic Compute Cloud (Amazon EC2) instance you specify. This runbook only supports Amazon Linux and Amazon Linux 2 instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ClusterARN

  Type: StringList

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon ECS cluster your container instances is registered with.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution

- ssm:GetAutomationExecution
- ssm:GetCommandInvocation
- ec2:DescribeImages
- ec2:DescribeInstanceAttribute
- ec2:DescribeImage
- ec2:DescribeInstance
- ec2:DescribeInstanceAttribute
- ecs:DescribeContainerInstances
- ecs:DescribeClusters
- ecs:ListContainerInstances
- ecs:UpdateContainerAgent

Document Steps

aws:executeScript - Updates the Amazon ECS agent on the Amazon ECS cluster you specify in the ClusterARN parameters.

Outputs

UpdateAmazonECSAgent.UpdatedContainers - The ID of the instance where the update of the Amazon ECS agent succeeded.

UpdateAmazonECSAgent.FailedContainers - The ID of the instance where the update of the Amazon ECS agent failed.

UpdateAmazonECSAgent.InProgressContainers - The ID of the instance where the update of the Amazon ECS agent is in progress.

Amazon EFS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Elastic File System. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-CheckAndMountEFS

AWSSupport-CheckAndMountEFS

Description

The AWSSupport-CheckAndMountEFS runbook verifies the prerequisites to mount your Amazon Elastic File System (Amazon EFS) file system and mounts the file system on the Amazon Elastic Compute Cloud (Amazon EC2) instance you specify. This runbook supports mounting your Amazon EFS file system with the DNS name, or using the mount target's IP address.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Action

  Type: String

  Valid values: Check | CheckAndMount

  Description: (Required) Determines whether the runbook verifies prerequisites, or verifies prerequisites and mounts the file system.
- EfsId

Type: String

Description: (Required) The ID of the file system you want to mount.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance on which you want to mount the file system.

- MountOptions

  Type: String

  Description: (Optional) The options supported by the Amazon EFS mount helper that you want to use when mounting the file system. If you specify the `tls` option, verify stunnel has been upgraded on the target instance.

- MountPoint

  Type: String

  Description: (Optional) The directory where you want to mount the file system. If you specify the `Check` value for the `Action` parameter, this parameter should not be specified.

- MountTargetIP

  Type: String

  Description: (Optional) The mount target's IP address. Mounting by IP address works in environments where DNS is disabled, such as virtual private clouds (VPCs) with DNS hostnames disabled. Also, you can use this option if your environment uses a DNS provider other than Amazon Route 53 (Route 53).

- Region

  Type: String

  Description: (Required) The Amazon Web Services Region where the Amazon EC2 instance and file system are located.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:DescribeInstanceInformation`
- `ssm:DescribeInstanceProperties`
- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:GetDocument`
- `ssm:ListCommands`
- `ssm:ListCommandInvocations`
- `ssm:ListDocuments`
- `ssm:StartAutomationExecution`
- `iam:ListRoles`
- `ec2:DescribeInstances`
- `ec2:DescribeSecurityGroups`
- `elasticfilesystem:DescribeFileSystemPolicy`
- `elasticfilesystem:DescribeMountTargets`
- `elasticfilesystem:DescribeMountTargetSecurityGroups`
- `resource-groups:*`

Document Steps

- `aws:executeScript` - Gathers details about the Amazon EC2 instance you specify in the `InstanceId` parameter.
- `aws:executeScript` - Gathers details about the file system you specify in the `EfsId` parameter.
- `aws:executeScript` - Verifies the security group associated with the file system allows traffic on port 2049 from the Amazon EC2 instance you specify in the `InstanceId` parameter.

- `aws:assertAwsResourceProperty` - Verifies the Amazon EC2 instance you specify in the `InstanceId` parameter is managed by Systems Manager and that the status is `Online` .
- `aws:branch` - Branches based on the value you specify for the `Action` parameter.
- `aws:runCommand` - Verifies prerequisites for mounting the file system you specify in the `EfsId` parameter.
- `aws:runCommand` - Verifies prerequisites for mounting the file system you specify in the `EfsId` parameter, and mounts the file system on the Amazon EC2 instance you specify in the `InstanceId` parameter.

Amazon EKS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Elastic Kubernetes Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

`AWSSupport-CollectEKSInstanceLogs`

Description

The AWSSupport-CollectEKSInstanceLogs runbook gathers operating system and Amazon Elastic Kubernetes Service (Amazon EKS) related log files from an Amazon Elastic Compute Cloud (Amazon EC2) instance to help you troubleshoot common issues. While the automation is gathering the associated log files, changes are made to the file system structure including the creation of temporary directories, the copying of log files to the temporary directories, and compressing the log files into an archive. This activity can result in increased CPUUtilization on the EC2 instance. For more information about CPUUtilization , see Instance metrics in the *Amazon CloudWatch User Guide* .

If you specify a value for the LogDestination parameter, the automation evaluates the policy status of the Amazon Simple Storage Service (Amazon S3) bucket you specify. To help with the security of the logs gathered from your EC2 instance, if the policy status isPublic is set to true , or if the access control list (ACL) grants READ|WRITE permissions to the All Users Amazon S3 predefined group, the logs are not uploaded. For more information about Amazon S3 predefined groups, see Amazon S3 predefined groups in the *Amazon Simple Storage Service User Guide* .

> **Note**
>
> This automation requires at least 10 percent of available disk space on the root Amazon Elastic Block Store (Amazon EBS) volume attached to your EC2 instance. If there is not enough available disk space on the root volume, the automation stops.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EKSInstanceId

  Type: String

  Description: (Required) ID of the Amazon EKS EC2 instance you want to collect logs from.

- LogDestination

  Type: String

  Description: (Optional) The S3 bucket in your account to upload the archived logs to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:SendCommand`

We recommend that the EC2 instance receiving the command has an IAM role with the **AmazonSSMManagedInstanceCore** Amazon managed policy attached. To upload the log archive to the S3 bucket you specify in the `LogDestination` parameter, you must add the `s3:PutObject` permission.

Document Steps

- `aws:assertAwsResourceProperty` - Confirms the operating system of the value specified in the `EKSInstanceId` parameter is Linux.
- `aws:runCommand` - Gathers operating system and Amazon EKS related log files, compressing them into an archive in the `/var/log` directory.

- aws:branch - Confirms whether a value was specified for the LogDestination parameter.
- aws:runCommand - Uploads the log archive to the S3 bucket you specify in the LogDestination parameter.

AWS-CreateEKSClusterWithFargateProfile

Description

The AWS-CreateEKSClusterWithFargateProfile runbook creates an Amazon Elastic Kubernetes Service (Amazon EKS) cluster using an Amazon Fargate.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ClusterName

  Type: String

  Description: (Required) A unique name for the cluster.

- ClusterRoleArn

  Type: String

  Description: (Required) The ARN of the IAM role that provides permissions for the Kubernetes control plane to make calls to Amazon API operations on your behalf.
- FargateProfileName

  Type: String

  Description: (Required) The name of the Fargate profile.
- FargateProfileRoleArn

  Type: String

  Description: (Required) The ARN of the Amazon EKS Pod execution IAM role.
- FargateProfileSelectors

  Type: String

  Description: (Required) The selectors to match pods to the Fargate profile.
- SubnetIds

  Type: StringList

  Description: (Required) The IDs of the subnets you want to use for your Amazon EKS cluster. Amazon EKS creates elastic network interfaces in these subnets for communication between your nodes and the Kubernetes control plane. You must specify at least two subnet IDs.
- EKSEndpointPrivateAccess

  Type: Boolean

  Default: True

  Description: (Optional) Set this value to `True` to allow private access for your cluster's Kubernetes API server endpoint. If you enable private access, Kubernetes API requests from within your cluster's VPC use the private VPC endpoint. If you disable private access and you have nodes or Amazon Fargate pods in the cluster, then ensure that `publicAccessCidrs` include the necessary CIDR blocks for communication with the nodes or Fargate pods.
- EKSEndpointPublicAccess

Type: Boolean

Default: False

Description: (Optional) Set this value to `False` to disable public access to your cluster's Kubernetes API server endpoint. If you disable public access, your cluster's Kubernetes API server can only receive requests from within the VPC where it was launched.

- PublicAccessCIDRs

  Type: StringList

  Description: (Optional) The CIDR blocks that are allowed access to your cluster's public Kubernetes API server endpoint. Communication to the endpoint from addresses outside of the CIDR blocks that you specify is denied. If you've disabled private endpoint access and you have nodes or Fargate pods in the cluster, then ensure that you specify the necessary CIDR blocks.

- SecurityGroupIds

  Type: StringList

  Description: (Optional) Specify one or more security groups to associate with the elastic network interfaces created in your account by Amazon EKS.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `eks:CreateCluster`
- `eks:CreateFargateProfile`
- `eks:DescribeCluster`
- `eks:DescribeFargateProfile`

- iam:CreateServiceLinkedRole
- iam:GetRole
- iam:ListAttachedRolePolicies
- iam:PassRole

Document Steps

- CreateEKSCluster (aws:executeAwsApi) - Creates an Amazon EKS cluster.
- VerifyEKSClusterIsActive (aws:waitForAwsResourceProperty) - Verifies the cluster state is ACTIVE.
- CreateFargateProfile (aws:executeAwsApi) - Creates a Fargate for the cluster.
- VerifyFargateProfileIsActive (aws:waitForAwsResourceProperty) - Verifies the Fargate profile state is ACTIVE.

Outputs

CreateEKSCluster.CreateClusterResponse

Description: Response received from the CreateCluster API call.

CreateFargateProfile.CreateFargateProfileResponse

Description: Response received from the CreateFargateProfile API call.

AWS-CreateEKSClusterWithNodegroup

Description

The AWS-CreateEKSClusterWithNodegroup runbook creates an Amazon Elastic Kubernetes Service (Amazon EKS) cluster using a node group for capacity.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterName

  Type: String

  Description: (Required) A unique name for the cluster.
- ClusterRoleArn

  Type: String

  Description: (Required) The ARN of the IAM role that provides permissions for the Kubernetes control plane to make calls to Amazon API operations on your behalf.
- NodegroupName

  Type: String

  Description: (Required) A unique name for the node group.
- NodegroupRoleArn

  Type: String

  Description: (Required) The ARN of the IAM role to associate with your node group. The Amazon EKS worker node kubelet daemon makes calls to Amazon APIs on your behalf. Nodes receive permissions for these API calls through an IAM instance profile and associated policies. Before you can launch nodes and register them into a cluster, you must create an IAM role for those nodes to use when they are launched.

- SubnetIds

  Type: StringList

  Description: (Required) The IDs of the subnets you want to use for your Amazon EKS cluster. Amazon EKS creates elastic network interfaces in these subnets for communication between your nodes and the Kubernetes control plane. You must specify at least two subnet IDs.

- EKSEndpointPrivateAccess

  Type: Boolean

  Default: True

  Description: (Optional) Set this value to `True` to allow private access for your cluster's Kubernetes API server endpoint. If you enable private access, Kubernetes API requests from within your cluster's VPC use the private VPC endpoint. If you disable private access and you have nodes or Amazon Fargate pods in the cluster, then ensure that `publicAccessCidrs` include the necessary CIDR blocks for communication with the nodes or Fargate pods.

- EKSEndpointPublicAccess

  Type: Boolean

  Default: False

  Description: (Optional) Set this value to `False` to disable public access to your cluster's Kubernetes API server endpoint. If you disable public access, your cluster's Kubernetes API server can only receive requests from within the VPC where it was launched.

- PublicAccessCIDRs

  Type: StringList

  Description: (Optional) The CIDR blocks that are allowed access to your cluster's public Kubernetes API server endpoint. Communication to the endpoint from addresses outside of the CIDR blocks that you specify is denied. If you've disabled private endpoint access and you have nodes or Fargate pods in the cluster, then ensure that you specify the necessary CIDR blocks.

- SecurityGroupIds

  Type: StringList

Description: (Optional) Specify one or more security groups to associate with the elastic network interfaces created in your account by Amazon EKS.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeSubnets`
- `eks:CreateCluster`
- `eks:CreateNodegroup`
- `eks:DescribeCluster`
- `eks:DescribeNodegroup`
- `iam:CreateServiceLinkedRole`
- `iam:GetRole`
- `iam:ListAttachedRolePolicies`
- `iam:PassRole`

Document Steps

- CreateEKSCluster (aws:executeAwsApi) - Creates an Amazon EKS cluster.
- VerifyEKSClusterIsActive (aws:waitForAwsResourceProperty) - Verifies the cluster state is `ACTIVE`.
- CreateNodegroup (aws:executeAwsApi) - Creates a node group for the cluster.
- VerifyNodegroupIsActive (aws:waitForAwsResourceProperty) - Verifies the node group state is `ACTIVE`.

Outputs

- `CreateEKSCluster.CreateClusterResponse`: Response received from the `CreateCluster` API call.

- `CreateNodegroup.CreateNodegroupResponse`: Response received from the `CreateNodegroup` API call.

`AWS-DeleteEKSCluster`

Description

This runbook deletes the resources associated with an Amazon EKS cluster, including node groups and Fargate profiles. Optionally, you can choose to delete all self-managed nodes, the Amazon CloudFormation stacks used to create the nodes, and the VPC CloudFormation stack for your cluster. For more information about deleting a cluster, see Deleting a cluster in the *Amazon EKS User Guide*.

> **Note**
>
> If you have active services in your cluster that are associated with a load balancer, you must delete those services before deleting the cluster. If you don't, the system can't delete the load balancers. Use the following procedure to find and delete services before you run the `AWS-DeleteEKSCluster` runbook.

To locate and delete services in your cluster

1. Install the Kubernetes command line utility, `kubectl` . For more information, see Installing kubectl in the *Amazon EKS User Guide*.
2. Run the following command to list all services running in your cluster.

```
kubectl get svc --all-namespaces
```

3. Run the following command to delete any services that have an associated EXTERNAL-IP value. These services are fronted by a load balancer, and you must delete them in Kubernetes to allow the load balancer and associated resources to be properly released.

```
kubectl delete svc
service-name
```

You can now run the `AWS-DeleteEKSCluster` runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EKSClusterName

  Type: String

  Description: (Required) The name of the Amazon EKS Cluster to be deleted.

- VPCCloudFormationStack

  Type: String

  Description: (Optional) Amazon CloudFormation stack name for VPC for the EKS cluster being deleted. This deletes the Amazon CloudFormation stack for VPC and any resources created by the stack.

- VPCCloudFormationStackRole

  Type: String

Description: (Optional) The ARN of an IAM role that Amazon CloudFormation assumes to delete the VPC CloudFormation stack. Amazon CloudFormation uses the role's credentials to make calls on your behalf.

- SelfManagedNodeStacks

  Type: String

  Description: (Optional) Comma-separated list of Amazon CloudFormation stack names for self-managed nodes, This will delete the Amazon CloudFormation stacks for self-managed nodes.

- SelfManagedNodeStacksRole

  Type: String

  Description: (Optional) The ARN of an IAM role that Amazon CloudFormation assumes to delete the Self-managed Node Stacks. Amazon CloudFormation uses the role's credentials to make calls on your behalf.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `sts:AssumeRole`
- `eks:ListNodegroups`
- `eks:DeleteNodegroup`
- `eks:ListFargateProfiles`
- `eks:DeleteFargateProfile`
- `eks:DeleteCluster`
- `cfn:DescribeStacks`
- `cfn:DeleteStack`

Document Steps

- `aws:executeScript` - DeleteNodeGroups: Find and delete all node groups in the EKS cluster.

- `aws:executeScript` - DeleteFargateProfiles: Find and delete all Fargate profiles in the EKS cluster.
- `aws:executeScript` - DeleteSelfManagedNodes: Delete all self-managed nodes and the CloudFormation stacks used to create the nodes.
- `aws:executeScript` - DeleteEKSCluster: Delete EKS cluster.
- `aws:executeScript` - DeleteVPCCloudFormationStack: Delete the VPC CloudFormation stack.

AWS-MigrateToNewEKSSelfManagedNodeGroup

Description

The `AWS-MigrateToNewEKSSelfManagedNodeGroup` runbook helps you create a new Amazon Elastic Kubernetes Service (Amazon EKS) Linux node group to migrate your existing application to. For more information, see Migrating to a new node group in the Amazon EKS User Guide.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- OldStackName

Type: String

Description: (Required) The name or stack ID of your existing Amazon CloudFormation stack.

- NewStackName

  Type: String

  Description: (Optional) The name of the new Amazon CloudFormation stack that is created for your new node group. If you don't specify a value for this parameter, the stack name is created using the format: `NewNodeGroup-ClusterName-AutomationExecutionID`.

- ClusterControlPlaneSecurityGroup

  Type: String

  Description: (Optional) The ID of the security group you want nodes to use to communicate with the Amazon EKS control plane. If you don't specify a value for this parameter, the security group specified in your existing Amazon CloudFormation stack is used.

- NodeInstanceType

  Type: String

  Description: (Optional) The instance type that you want to use for the new node group. If you don't specify a value for this parameter, the instance type specified in your existing Amazon CloudFormation stack is used.

- NodeGroupName

  Type: String

  Description: (Optional) The name of your new node group. If you don't specify a value for this parameter, the node group name specified in your existing Amazon CloudFormation stack is used.

- NodeAutoScalingGroupDesiredCapacity

  Type: String

  Description: (Optional) The desired number of nodes to scale to when your new stack is created. This number must be greater than or equal to the `NodeAutoScalingGroupMinSize` value and less than or equal to the `NodeAutoScalingGroupMaxSize`. If you don't specify a value for this

parameter, the node group desired capacity specified in your existing Amazon CloudFormation stack is used.

- NodeAutoScalingGroupMaxSize

  Type: String

  Description: (Optional) The maximum number of nodes that your node group can scale out to. If you don't specify a value for this parameter, the node group maximum size specified in your existing Amazon CloudFormation stack is used.

- NodeAutoScalingGroupMinSize

  Type: String

  Description: (Optional) The minimum number of nodes that your node group can scale in to. If you don't specify a value for this parameter, the node group minimum size specified in your existing Amazon CloudFormation stack is used.

- NodeImageId

  Type: String

  Description: (Optional) The ID of the Amazon Machine Image (AMI) that you want the node group to use.

- NodeImageIdSSMParam

  Type: String

  Description: (Optional) The public Systems Manager parameter for the AMI that you want the node group to use.

- NodeVolumeSize

  Type: String

  Description: (Optional) The size of the root volume for your nodes in GiB. If you don't specify a value for this parameter, the node volume size specified in your existing Amazon CloudFormation stack is used.

- NodeVolumeType

  Type: String

Description: (Optional) The type of Amazon EBS volume you want to use for the root volume of your nodes. If you don't specify a value for this parameter, the volume type specified in your existing Amazon CloudFormation stack is used.

- KeyName

  Type: String

  Description: (Optional) The key pair you want to assign to your nodes. If you don't specify a value for this parameter, the key pair specified in your existing Amazon CloudFormation stack is used.

- Subnets

  Type: StringList

  Description: (Optional) A comma-separated list of the subnet IDs that you want to use for your new node group. If you don't specify a value for this parameter, the subnets specified in your existing Amazon CloudFormation stack is used.

- DisableIMDSv1

  Type: Boolean

  Description: (Optional) Specify `true` to disable Instance Metadata Service Version 1 (IMDSv1). By default, nodes support IMDSv1 and IMDSv2.

- BootstrapArguments

  Type: String

  Description: (Optional) Additional arguments you want to pass to the node bootstrap script.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:GetParameters`
- `autoscaling:CreateAutoScalingGroup`

- autoscaling:CreateOrUpdateTags
- autoscaling:DeleteTags
- autoscaling:DescribeAutoScalingGroups
- autoscaling:DescribeScalingActivities
- autoscaling:DescribeScheduledActions
- autoscaling:SetDesiredCapacity
- autoscaling:TerminateInstanceInAutoScalingGroup
- autoscaling:UpdateAutoScalingGroup
- cloudformation:CreateStack
- cloudformation:DescribeStackResource
- cloudformation:DescribeStacks
- cloudformation:UpdateStack
- ec2:AuthorizeSecurityGroupEgress
- ec2:AuthorizeSecurityGroupIngress
- ec2:CreateLaunchTemplateVersion
- ec2:CreateLaunchTemplate
- ec2:CreateSecurityGroup
- ec2:CreateTags
- ec2:DeleteLaunchTemplate
- ec2:DeleteSecurityGroup
- ec2:DescribeAvailabilityZones
- ec2:DescribeImages
- ec2:DescribeInstanceAttribute
- ec2:DescribeInstanceStatus
- ec2:DescribeInstances
- ec2:DescribeKeyPairs
- ec2:DescribeLaunchTemplateVersions
- ec2:DescribeLaunchTemplates
- ec2:DescribeSecurityGroups

- ec2:DescribeSubnets
- ec2:DescribeVpcs
- ec2:RevokeSecurityGroupEgress
- ec2:RevokeSecurityGroupIngress
- ec2:RunInstances
- ec2:TerminateInstances
- iam:AddRoleToInstanceProfile
- iam:AttachRolePolicy
- iam:CreateInstanceProfile
- iam:CreateRole
- iam:GetInstanceProfile
- iam:GetRole
- iam:PassRole

Document Steps

- DetermineParameterValuesForNewNodeGroup (aws:executeScript) - Gathers the parameter values to use for the new node group.
- CreateStack (aws:createStack) - Creates the Amazon CloudFormation stack for the new node group.
- GetNewStackNodeInstanceRole (aws:executeAwsApi) - Gets the node instance role.
- GetNewStackSecurityGroup (aws:executeAwsApi) - The step gets the node security group.
- AddIngressRulesToNewNodeSecurityGroup (aws:executeAwsApi) - Adds ingress rules to the newly created security group so it can accept traffic from the one assigned to your previous node group.
- AddIngressRulesToOldNodeSecurityGroup (aws:executeAwsApi) - Adds ingress rules to the previous security group so it can accept traffic from the one assigned to your newly created node group.
- VerifyStackComplete (aws:assertAwsResourceProperty) - Verifies the new stack status is CREATE_COMPLETE.

Outputs

DetermineParameterValuesForNewNodeGroup.NewStackParameters - The parameters used to create the new stack.

GetNewStackNodeInstanceRole.NewNodeInstanceRole - The node instance role for the new node group.

GetNewStackSecurityGroup.NewNodeSecurityGroup - The ID of the security group for the new node group.

DetermineParameterValuesForNewNodeGroup.NewStackName - The Amazon CloudFormation stack name for the new node group.

CreateStack.StackId - The Amazon CloudFormation stack ID for the new node group.

AWSPremiumSupport-TroubleshootEKSCluster

Description

The `AWSPremiumSupport-TroubleshootEKSCluster` runbook diagnoses common issues with an Amazon Elastic Kubernetes Service (Amazon EKS) cluster, underlying infrastructure, and provides recommended remediation steps.

> **Important**
>
> Access to `AWSPremiumSupport-*` runbooks requires either an Enterprise or Business Support Subscription. For more information, see Compare Amazon Support Plans .

If you specify a value for the `S3BucketName` parameter, the automation evaluates the policy status of the Amazon Simple Storage Service (Amazon S3) bucket you specify. To help with the security of the logs gathered from your EC2 instance, if the policy status `isPublic` is set to `true` , or if the access control list (ACL) grants `READ|WRITE` permissions to the `All Users` Amazon S3 predefined group, the logs are not uploaded. For more information about Amazon S3 predefined groups, see Amazon S3 predefined groups in the *Amazon Simple Storage Service User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterName

  Type: String

  Description: (Required) The name of the Amazon EKS cluster that you want to troubleshoot.
- S3BucketName

  Type: String

  Description: (Required) The name of the private Amazon S3 bucket where the report generated by the runbook should be uploaded.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceTypes`

- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeRouteTables
- ec2:DescribeNatGateways
- ec2:DescribeVpcs
- ec2:DescribeNetworkAcls
- iam:GetInstanceProfile
- iam:ListInstanceProfiles
- iam:ListAttachedRolePolicies
- eks:DescribeCluster
- eks:ListNodegroups
- eks:DescribeNodegroup
- autoscaling:DescribeAutoScalingGroups

In addition, the Amazon Identity and Access Management (IAM) policy attached to the user or role that starts the automation must allow the ssm:GetParameter operation to the following public Amazon Systems Manager parameters to get the latest recommended Amazon EKS Amazon Machine Image (AMI) for the worker nodes.

- arn:aws:ssm:::parameter/aws/service/eks/optimized-ami/*/amazon-linux-2/recommended/image_id
- arn:aws:ssm:::parameter/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-EKS_Optimized-*/image_id
- arn:aws:ssm:::parameter/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-EKS_Optimized-*/image_id
- arn:aws:ssm:::parameter/aws/service/ami-windows-latest/Windows_Server-1909-English-Core-EKS_Optimized-*/image_id
- arn:aws:ssm:::parameter/aws/service/eks/optimized-ami/*/amazon-linux-2-gpu/recommended/image_id

To upload the report generated by the runbook to an Amazon S3 bucket, the following permissions are required for the specified Amazon S3 bucket you specify.

- s3:GetBucketPolicyStatus
- s3:GetBucketAcl
- s3:PutObject

Document Steps

- aws:executeAwsApi - Gathers details for the specified Amazon EKS cluster.
- aws:executeScript - Gathers details of the Amazon Elastic Compute Cloud (Amazon EC2) instances, Auto Scaling groups, AMIs, and Amazon EC2 GPU graphic instance types.
- aws:executeScript - Gathers details of the virtual private cloud (VPC), subnets, network address translation (NAT) gateways, subnet routes, security groups and network access control lists (ACLs) of the Amazon EKS cluster.
- aws:executeScript - Gathers details of attached IAM instance profiles and role policies.
- aws:executeScript - Gathers details of the Amazon S3 bucket you specify in the S3BucketName parameter.
- aws:executeScript - Classifies the Amazon VPC subnets as public or private.
- aws:executeScript - Checks the Amazon VPC subnets for tags that are required as part of an Amazon EKS cluster.
- aws:executeScript - Checks the Amazon VPC subnets for the tags that are required for Elastic Load Balancing subnets.
- aws:executeScript - Checks if the worker node Amazon EC2 instances use the latest Amazon EKS optimized AMIs
- aws:executeScript - Checks if the Amazon VPC security groups attached to worker nodes for the tags that are required.
- aws:executeScript - Checks the Amazon EKS cluster and worker node Amazon VPC security group rules for the recommended ingress rules to the Amazon EKS cluster.
- aws:executeScript - Checks the Amazon EKS cluster and worker node Amazon VPC security group rules for the recommended egress rules from the Amazon EKS cluster.
- aws:executeScript - Checks the network ACL configuration of the Amazon VPC subnets.
- aws:executeScript - Checks if the worker node Amazon EC2 instances have the required managed policies.
- aws:executeScript - Checks if the Auto Scaling groups have the necessary tags for cluster autoscaling.

- `aws:executeScript` - Checks if the worker node Amazon EC2 instances are connected to the internet.
- `aws:executeScript` - Generates a report based on the outputs from the previous steps. If a value is specified for the `S3BucketName` parameter, the generated report is uploaded to the Amazon S3 bucket.

AWSSupport-TroubleshootEKSWorkerNode

Description

The `AWSSupport-TroubleshootEKSWorkerNode` runbook analyzes an Amazon Elastic Compute Cloud (Amazon EC2) worker node and Amazon Elastic Kubernetes Service (Amazon EKS) cluster to help you identify and troubleshoot common causes that prevent worker nodes from joining a cluster. The runbook outputs guidance to help you resolve any issues that are identified.

> **Important**
>
> To successfully run this automation, the state of your Amazon EC2 worker node must be `running` , and the Amazon EKS cluster state must be `ACTIVE` .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ClusterName

  Type: String

  Description: (Required) The name of the Amazon EKS cluster.

- WorkerID

  Type: String

  Description: (Required) The ID of the Amazon EC2 worker node that failed to join the cluster.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeDhcpOptions`
- `ec2:DescribeImages`
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`
- `ec2:DescribeNatGateways`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeVpcs`

- eks:DescribeCluster
- iam:GetInstanceProfile
- iam:GetRole
- iam:ListAttachedRolePolicies
- ssm:DescribeInstanceInformation
- ssm:ListCommandInvocations
- ssm:ListCommands
- ssm:SendCommand

Document Steps

- aws:assertAwsResourceProperty - Confirms that the Amazon EKS cluster you specify in the ClusterName parameter exists and is in an ACTIVE state.
- aws:assertAwsResourceProperty - Confirms that the Amazon EC2 worker node you specify in the WorkerID parameter exists and is in a running state.
- aws:executeScript - Runs a Python script that helps identify possible causes for the worker node failing to join the cluster.

AWS-UpdateEKSCluster

Description

The AWS-UpdateEKSCluster runbook helps you update your Amazon Elastic Kubernetes Service (Amazon EKS) cluster to the Kubernetes version that you want to use.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterName

  Type: String

  Description: (Required) The name of your Amazon EKS cluster.
- Version

  Type: String

  Description: (Required) The Kubernetes version that you want to update your cluster to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `eks:DescribeUpdate`
- `eks:UpdateClusterVersion`

Document Steps

- `aws:executeAwsApi` - Updates the Kubernetes version that is used by your Amazon EKS cluster.
- `aws:waitForAwsResourceProperty` - Waits for the update status to be `Successful`.

AWS-UpdateEKSManagedNodeGroup

Description

The AWS-UpdateEKSManagedNodeGroup runbook helps you update an Amazon Elastic Kubernetes Service (Amazon EKS) managed node group. You can either choose a Version or Configuration update.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterName

  Type: String

  Description: (Required) The name of the cluster whose node group you want to update.
- NodeGroupName

  Type: String

  Description: (Required) The name of the node group to update.
- UpdateType

  Type: String

  Valid values: Update Node Group Version | Update Node Group Configurations

Default: Update Node Group Version

Description: (Required) The type of update that you want to perform on the node group.

The following parameters apply only to the `Version` update type:

- AMIReleaseVersion

  Type: String

  Description: (Optional) The version of the Amazon EKS optimized AMI that you want to use. By default, the latest version is used.
- ForceUpgrade

  Type: Boolean

  Description: (Optional) If true, the update won't fail in response to a pod disruption budget violation.
- KubernetesVersion

  Type: String

  Description: (Optional) The Kubernetes version to update the node group to.
- LaunchTemplateId

  Type: String

  Description: (Optional) The ID of the launch template.
- LaunchTemplateName

  Type: String

  Description: (Optional) The name of the launch template.
- LaunchTemplateVersion

  Type: String

  Description: (Optional) The Amazon Elastic Compute Cloud (Amazon EC2) launch template version. This parameter is only valid if a node group was created from a launch template.

The following parameters apply only to the `Configuration` update type:

- AddOrUpdateNodeGroupLabels

  Type: StringMap

  Description: (Optional) Kubernetes labels that you want to add or update.
- AddOrUpdateKubernetesTaintsEffect

  Type: StringList

  Description: (Optional) The Kubernetes taints that you want to add or update.
- MaxUnavailableNodeGroups

  Type: Integer

  Default: 0

  Description: (Optional) The maximum number of nodes that are unavailable at once during a version update.
- MaxUnavailablePercentageNodeGroup

  Type: Integer

  Default: 0

  Description: (Optional) The percentage of nodes that are unavailable during a version update.
- NodeGroupDesiredSize

  Type: Integer

  Default: 0

  Description: (Optional) The number of nodes that the managed node group should maintain.
- NodeGroupMaxSize

  Type: Integer

  Default: 0

Description: (Optional) The maximum number of nodes that the managed node group can scale out to.

- NodeGroupMinSize

  Type: Integer

  Default: 0

  Description: (Optional) The minimum number of nodes that the managed node group can scale in to.

- RemoveKubernetesTaintsEffect

  Type: StringList

  Description: (Optional) The Kubernetes taints that you want to remove.

- RemoveNodeGroupLabels

  Type: StringList

  Description: (Optional) A comma-separated list of labels that you want to remove.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `eks:UpdateNodegroupConfig`
- `eks:UpdateNodegroupVersion`

Document Steps

- `aws:executeScript` - Updates an Amazon EKS cluster node group according to the values that you specify for the runbook input parameters.
- `aws:waitForAwsResourceProperty` - Waits for the cluster update status to be `Successful`.

AWS-UpdateEKSSelfManagedLinuxNodeGroups

Description

The AWS-UpdateEKSSelfManagedLinuxNodeGroups runbook updates self-managed managed node groups in your Amazon Elastic Kubernetes Service (Amazon EKS) cluster using an Amazon CloudFormation stack.

If your cluster uses auto scaling, we recommend scaling the deployment down to two replicas before using this runbook.

To scale a deployment to two replicas

1. Install the Kubernetes command line utility, kubectl. For more information, see Installing kubectl in the *Amazon EKS User Guide*.
2. Run the following command.

```
kubectl scale deployments/cluster-autoscaler --replicas=2 -n kube-system
```

3. Run the AWS-UpdateEKSSelfManagedLinuxNodeGroups runbook.
4. Scale the deployment back to the desired number of replicas by running the following command.

```
kubectl scale deployments/cluster-autoscaler --replicas=number -n kube-system
```

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterName

  Type: String

  Description: (Required) The name of the Amazon EKS cluster.
- NodeGroupName

  Type: String

  Description: (Required) The name of the managed node group.
- ClusterControlPlaneSecurityGroup

  Type: String

  Description: (Required) The ID of the control plane security group.
- DisableIMDSv1

  Type: Boolean

  Description: (Optional) Determines whether you want to allow Instance Metadata Service Version 1 (IMDSv1) and IMDSv2.
- KeyName

  Type: String

  Description: (Optional) The key name for the instances.
- NodeAutoScalingGroupDesiredCapacity

  Type: String

  Description: (Optional) The number of nodes that the node group should maintain.
- NodeAutoScalingGroupMaxSize

Type: String

Description: (Optional) The maximum number of nodes that the node group can scale out to.

- NodeAutoScalingGroupMinSize

  Type: String

  Description: (Optional) The minimum number of nodes that the node group can scale in to.

- NodeInstanceType

  Type: String

  Default: t3.large

  Description: (Optional) The instance type that you want to use for the node group.

- NodeImageId

  Type: String

  Description: (Optional) The ID of the Amazon Machine Image (AMI) that you want the node group to use.

- NodeImageIdSSMParam

  Type: String

  Default: /aws/service/eks/optimized-ami/1.21/amazon-linux-2/recommended/image_id

  Description: (Optional) The public Systems Manager parameter for the AMI that you want the node group to use.

- StackName

  Type: String

  Description: (Required) The name of the Amazon CloudFormation stack used to update the node group.

- Subnets

  Type: String

Description: (Required) A comma-separated list of the IDs for the subnets that you want your cluster to use.

- VpcId

  Type: String

  Default: Default

  Description: (Required) The virtual private cloud (VPC) where your cluster is deployed.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `eks:CreateCluster`
- `eks:CreateNodegroup`
- `eks:DeleteNodegroup`
- `eks:DeleteCluster`
- `eks:DescribeCluster`
- `eks:DescribeNodegroup`
- `eks:ListClusters`
- `eks:ListNodegroups`
- `eks:UpdateClusterConfig`
- `eks:UpdateNodegroupConfig`

Document Steps

- `aws:executeScript` - Updates an Amazon EKS cluster node group according to the values that you specify for the runbook input parameters.
- `aws:waitForAwsResourceProperty` - Waits for the Amazon CloudFormation stack update status to be returned.

AWSSupport-SetupK8sApiProxyForEKS

Description

The **AWSSupport-SetupK8sApiProxyForEKS** automation runbook provides a way to create an Amazon Lambda function that acts as a proxy for making control plane API calls to the Amazon Elastic Kubernetes Service cluster endpoint. It serves as a building block for runbooks which require making control plane API calls for automating tasks and troubleshooting issues with an Amazon EKS cluster.

Important

All the resources created by this automation are tagged so that they can be easily found. The tags used are:

- `AWSSupport-SetupK8sApiProxyForEKS`: true

Note

- The automation is a helper runbook and cannot be executed as a standalone runbook. It is invoked as a child automation for runbooks which require control plane API calls to Amazon EKS cluster.
- Please ensure to run `Cleanup` operation after usage to avoid incurring unwanted costs.

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ClusterName

  Type: String

  Description: (Required) The name of the Amazon Elastic Kubernetes Service cluster.

- Operation

  Type: String

  Description: (Required) Operation to perform: `Setup` provisions the Lambda function in the account, `Cleanup` will de-provision resources created as part of setup phase.

  Allowed Values: `Setup` | `Cleanup`

  Default: Setup

- LambdaRoleArn

  Type: String

  Description: (Optional) The ARN of the IAM role that allows the Amazon Lambda function to access the required AWS services and resources. If no role is specified, this Systems Manager Automation will create one IAM role for Lambda in your account with the name `Automation-K8sProxy-Role-<ExecutionId>` that includes the managed policies: `AWSLambdaBasicExecutionRole` and `AWSLambdaVPCAccessExecutionRole`.

How does it work?

The runbook performs the following steps:

- Validates that the automation is running as a child execution. The runbook will not work when invoked as a standalone runbook since it does not perform any meaningful work on its own.

- Checks for existing Amazon CloudFormation stack for the proxy Lambda function for the specified cluster.
  - If the stack exists, the existing infrastructure is re-used instead of re-creating it.
  - A reference counter is maintained using tags to ensure a runbook does not delete the infrastructure if it is being re-used by another runbook for the same cluster.
- Perform the operation type (`Setup`/`Cleanup`) specified for the invocation:
  - **Setup:** Creates or describes existing resources.

    Cleanup: Removes provisioned resources, if the infrastructure is not being used by any other runbook.

Required IAM Permissions

The `AutomationAssumeRole` parameter requires the following permissions given `LambdaRoleArn` is not passed:

- `cloudformation:CreateStack`
- `cloudformation:DescribeStacks`
- `cloudformation:DeleteStack`
- `cloudformation:UpdateStack`
- `ec2:CreateNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:DeleteNetworkInterface`
- `eks:DescribeCluster`
- `lambda:CreateFunction`
- `lambda:DeleteFunction`
- `lambda:ListTags`
- `lambda:GetFunction`
- `lambda:ListTags`

- lambda:TagResource
- lambda:UntagResource
- lambda:UpdateFunctionCode
- logs:CreateLogGroup
- logs:PutRetentionPolicy
- logs:TagResource
- logs:UntagResource
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- logs:ListTagsForResource
- iam:CreateRole
- iam:AttachRolePolicy
- iam:DetachRolePolicy
- iam:PassRole
- iam:GetRole
- iam:DeleteRole
- iam:TagRole
- iam:UntagRole
- tag:GetResources
- tag:TagResources

When LambdaRoleArn is provided, the automation does not need to create the role and the following permissions can be excluded:

- iam:CreateRole
- iam:DeleteRole
- iam:TagRole
- iam:UntagRole
- iam:AttachRolePolicy
- iam:DetachRolePolicy

Below is an example policy demonstrating permissions required for `AutomationAssumeRole` when `LambdaRoleArn` is not passed:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Action": [
    "tag:GetResources",
    "tag:TagResources",
    "ec2:CreateNetworkInterface",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVpcs",
    "ec2:DeleteNetworkInterface",
    "eks:DescribeCluster",
    "iam:GetRole",
    "cloudformation:DescribeStacks",
    "logs:DescribeLogGroups",
    "logs:DescribeLogStreams",
    "lambda:GetFunction",
    "lambda:ListTags",
    "logs:ListTagsForResource"
   ],
   "Resource": "*",
   "Effect": "Allow",
   "Sid": "AllowActionsWithoutConditions"
  },
  {
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
    }
   },
   "Action": "iam:CreateRole",
   "Resource": [
    "arn:<partition>:iam:::<account-id>::role/Automation-K8sProxy*"
   ],
   "Effect": "Allow",
   "Sid": "AllowCreateRoleWithRequiredTag"
  },
```

```
  {
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
    }
   },
   "Action": [
    "iam:DeleteRole",
    "iam:TagRole",
    "iam:UntagRole"
   ],
   "Resource": [
    "arn:<partition>:iam:::<account-id>::role/Automation-K8sProxy*"
   ],
   "Effect": "Allow",
   "Sid": "IAMActions"
  },
  {
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
    },
    "StringLike": {
     "iam:PolicyARN": [
      "arn:<partition>:iam::<partition>:policy/service-role/
AWSLambdaBasicExecutionRole",
      "arn:<partition>:iam::<partition>:policy/service-role/
AWSLambdaVPCAccessExecutionRole"
     ]
    }
   },
   "Action": [
    "iam:AttachRolePolicy",
    "iam:DetachRolePolicy"
   ],
   "Resource": [
    "arn:<partition>:iam:::<account-id>::role/Automation-K8sProxy*"
   ],
   "Effect": "Allow",
   "Sid": "AttachRolePolicy"
  },
  {
   "Condition": {
    "StringEquals": {
```

```
      "aws:ResourceTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
     }
    },
    "Action": [
     "lambda:CreateFunction",
     "lambda:DeleteFunction",
     "lambda:TagResource",
     "lambda:UntagResource",
     "lambda:UpdateFunctionCode"
    ],
    "Resource": "arn:<partition>:lambda::<region-id>:::<account-
id>::function:Automation-K8sProxy*",
    "Effect": "Allow",
    "Sid": "LambdaActions"
   },
   {
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
     }
    },
    "Action": [
     "cloudformation:CreateStack",
     "cloudformation:DeleteStack",
     "cloudformation:UpdateStack"
    ],
    "Resource": "arn:<partition>:cloudformation::<region-id>:::<account-id>::stack/
AWSSupport-SetupK8sApiProxyForEKS*",
    "Effect": "Allow",
    "Sid": "CloudFormationActions"
   },
   {
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
     }
    },
    "Action": [
     "logs:CreateLogGroup",
     "logs:CreateLogStream",
     "logs:PutLogEvents",
     "logs:PutRetentionPolicy",
     "logs:TagResource",
     "logs:UntagResource"
```

```
    ],
    "Resource": [
     "arn:<partition>:logs::<region-id>:::<account-id>::log-group:/aws/lambda/
Automation-K8sProxy*",
     "arn:<partition>:logs::<region-id>:::<account-id>::log-group:/aws/lambda/
Automation-K8sProxy*:*"
    ],
    "Effect": "Allow",
    "Sid": "LogsActions"
   },
   {
    "Condition": {
     "StringLikeIfExists": {
      "iam:PassedToService": "lambda.amazonaws.com"
     }
    },
    "Action": [
     "iam:PassRole"
    ],
    "Resource": [
     "arn:<partition>:iam:::<account-id>::role/Automation-K8sProxy-Role*"
    ],
    "Effect": "Allow",
    "Sid": "PassRoleToLambda"
   }
  ]
}
```

In case the `LambdaRoleArn` is passed, please ensure that it has AWSLambdaBasicExecutionRole policy attached to it for public cluster and additionally, AWSLambdaVPCAccessExecutionRole for private clusters.

Resources Created

The following resources are created during `Setup` operation:

1. Amazon Lambda function
2. IAM Role: Lambda execution role, if not provided.
3. CloudWatch Log Group (Lambda Logs)

Lambda function and execution role are retained until `Cleanup` operation is executed. Lambda log group will be retained for 30 days or until manually deleted.

Instructions

The runbook is a helper utility designed to be executed from within other runbooks as a child automation. It facilitates the creation of infrastructure enabling the parent runbook to make Amazon EKS K8s control plane API calls. In order to use the runbook, you can follow the below steps from the context of the parent automation.

1. **Setup Phase**: Invoke the automation using `aws:executeAutomation` action operation from the runbook that would like to make Amazon EKS K8s control plane API calls with operation set to `Setup`.

   Example of input parameters:

   ```
   {
     "AutomationAssumeRole": "<role-arn>",
     "ClusterName": "<eks-cluster-name>",
     "Operation": "Setup"
   }
   ```

   The output of the `aws:executeAutomation` step will contain the ARN of the proxy Lambda function.

2. **Using the Lambda Proxy**: Invoke the Lambda function inside the `aws:executeScript` action using `boto3`'s `Lambda.Client.invoke(...)` with a list of API call paths and bearer token. The Lambda function will perform HTTP `GET` calls to the specified path by passing the bearer token as part of authorization header.

   Example of Lambda invoke event:

   ```
   {
       "ApiCalls": ["/api/v1/pods/", ...],
       "BearerToken": "..."
   }
   ```

> **Note**
>
> The bearer token has to be generated as part of the parent automation script. You need to ensure the principal executing the parent runbook has read-only permission to the specified Amazon EKS cluster.

3. **Cleanup Phase**: Invoke the automation using `aws:executeAutomation` action operation from the runbook that would like to make Amazon EKS K8s control plane API calls with operation set to `Cleanup`.

   Example of input parameters:

```
{
  "AutomationAssumeRole": "<role-arn>",
  "ClusterName": "<eks-cluster-name>",
  "Operation": "Cleanup"
}
```

Automation Steps

1. **ValidateExecution**
   - Verifies that the automation is not running as a standalone execution.
2. **CheckForExistingStack**
   - Checks if a Amazon CloudFormation stack was already provisioned for the specified cluster name.
   - Returns stack existence status and whether it's safe to delete.
3. **BranchOnIsStackExists**
   - Decision step that branches based on stack existence.
   - Routes to either update existing stack name or proceed with operation branching.
4. **UpdateStackName**
   - Updates the `StackName` variable with the existing stack's name.
   - Only executed if stack already exists.
5. **BranchOnOperation**
   - Routes the automation based on the `Operation` parameter (`Setup` /`Cleanup`).

- For `Setup`: Routes to either create new stack or describe existing resources.
- For `Cleanup`: Proceeds to stack deletion if safe to delete.

6. **GetClusterNetworkConfig**
   - Describes the Amazon EKS cluster to obtain VPC configuration.
   - Retrieves endpoint, VPC ID, subnet IDs, security group ID, and CA data.
7. **ProvisionResources**
   - Creates a Amazon CloudFormation stack with required resources.
   - Provisions Lambda function with necessary networking configuration.
   - Tags all resources for tracking and management.
8. **DescribeStackResources**
   - Retrieves information about the created/existing stack.
   - Gets the ARN of the provisioned Lambda function.
9. **BranchOnIsLambdaDeploymentRequired**
   - Determines if Lambda code deployment is needed.
   - Only proceeds to deployment for newly created stacks.
10. **DeployLambdaFunctionCode**
    - Deploys the Lambda function code using the deployment package.
    - Updates the function with the proxy implementation.
11. **AssertLambdaAvailable**
    - Verifies that the Lambda function code update was successful.
    - Waits for the function to be in `Successful` state.
12. **PerformStackCleanup**
    - Deletes the Amazon CloudFormation stack and associated resources.
    - Executed during `Cleanup` operation or on failure of `Setup` operation.

Outputs

LambdaFunctionArn: ARN of the proxy Lambda function

References

Systems Manager Automation

- Run an automation
- Setting up an Automation
- Support Automation Workflows

AWSSupport-TroubleshootEbsCsiDriversForEks

Description

The AWSSupport-TroubleshootEbsCsiDriversForEks runbook helps troubleshoot issues with Amazon Elastic Block Store volume mounts in Amazon Elastic Kubernetes Service (Amazon EKS) and Amazon EBS Container Storage Interface (CSI) driver issues

> **Important**
>
> Currently the Amazon EBS CSI Driver running on Amazon Fargate is not supported.

How does it work?

The runbook AWSSupport-TroubleshootEbsCsiDriversForEks performs the following high-level steps:

- Verifies if the target Amazon EKS cluster exists and is in active state.
- Deploys necessary authentication resources for making Kubernetes API calls based on whether the addon is Amazon EKS-managed or self-managed.
- Performs Amazon EBS CSI controller health checks and diagnostics.
- Runs IAM permissions checks on node roles and service account roles.
- Diagnoses persistent volume creation issues for the specified application pod.
- Checks node-to-pod scheduling and examines pod events.
- Collects relevant Kubernetes and application logs, uploading them to the specified Amazon S3 bucket.
- Performs node health checks and verifies connectivity with Amazon EC2 endpoints.
- Reviews persistent volume block device attachments and mounting status.
- Cleans up the authentication infrastructure created during troubleshooting.
- Generates a comprehensive troubleshooting report combining all diagnostic results.

Note

- The Amazon EKS cluster's authentication mode must be set to either `API` or `API_AND_CONFIG_MAP`. We recommend using Amazon EKS Access entry. The runbook requires Kubernetes Role-based access control (RBAC) permissions to perform the necessary API calls.
- If you don't specify an IAM role for the Lambda function (`LambdaRoleArn` parameter), the automation creates a role named `Automation-K8sProxy-Role-<ExecutionId>` in your account. This role includes the managed policies `AWSLambdaBasicExecutionRole` and `AWSLambdaVPCAccessExecutionRole`.
- Some diagnostic steps require the Amazon EKS worker nodes to be Systems Manager managed instances. If the nodes aren't Systems Manager managed instances, steps that require Systems Manager access are skipped, but other checks continue.
- The automation includes a cleanup step that removes authentication infrastructure resources. This cleanup step runs even when previous steps fail, which helps prevent orphaned resources in your Amazon account.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeIamInstanceProfileAssociations`

- ec2:DescribeInstanceStatus
- ec2:GetEbsEncryptionByDefault
- eks:DescribeAddon
- eks:DescribeAddonVersions
- eks:DescribeCluster
- iam:GetInstanceProfile
- iam:GetOpenIDConnectProvider
- iam:GetRole
- iam:ListOpenIDConnectProviders
- iam:SimulatePrincipalPolicy
- s3:GetBucketLocation
- s3:GetBucketPolicyStatus
- s3:GetBucketPublicAccessBlock
- s3:GetBucketVersioning
- s3:ListBucket
- s3:ListBucketVersions
- ssm:DescribeInstanceInformation
- ssm:GetAutomationExecution
- ssm:GetDocument
- ssm:ListCommandInvocations
- ssm:ListCommands
- ssm:SendCommand
- ssm:StartAutomationExecution

Instructions

Follow these steps to configure the automation:

1. Create a SSM automation role TroubleshootEbsCsiDriversForEks-SSM-Role in your account. Verify that the trust relationship contains the following policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "ssm.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Attach the policy below to the IAM role to grant the required permissions to perform the specified actions on the specified resources.
   - If you are expecting to upload execution and resources logs to Amazon S3 bucket in same Amazon region, replace `arn:{partition}:s3:::BUCKET_NAME/*` as yours in `OptionalRestrictPutObjects`.
     - The Amazon S3 bucket should point to the correct Amazon S3 bucket if you will select `S3BucketName` in SSM execution.
     - This permission is optional if you don't specify `S3BucketName`
     - The Amazon S3 bucket must be private and in the same Amazon region where you execute the SSM automation.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "OptionalRestrictPutObjects",
            "Effect": "Allow",
            "Action": ["s3:PutObject"],
            "Resource": ["arn:{partition}:s3:::BUCKET_NAME/*"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeIamInstanceProfileAssociations",
                "ec2:DescribeInstanceStatus",
                "ec2:GetEbsEncryptionByDefault",
```

```
                "eks:DescribeAddon",
                "eks:DescribeAddonVersions",
                "eks:DescribeCluster",
                "iam:GetInstanceProfile",
                "iam:GetOpenIDConnectProvider",
                "iam:GetRole",
                "iam:ListOpenIDConnectProviders",
                "iam:SimulatePrincipalPolicy",
                "s3:GetBucketLocation",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketVersioning",
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "ssm:DescribeInstanceInformation",
                "ssm:GetAutomationExecution",
                "ssm:GetDocument",
                "ssm:ListCommandInvocations",
                "ssm:ListCommands",
                "ssm:SendCommand",
                "ssm:StartAutomationExecution"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SetupK8sApiProxyForEKSActions",
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStacks",
                "cloudformation:UpdateStack",
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "eks:DescribeCluster",
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:GetRole",
                "iam:TagRole",
```

```
                "iam:UntagRole",
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:GetFunction",
                "lambda:InvokeFunction",
                "lambda:ListTags",
                "lambda:TagResource",
                "lambda:UntagResource",
                "lambda:UpdateFunctionCode",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:ListTagsForResource",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:TagResource",
                "logs:UntagResource",
                "ssm:DescribeAutomationExecutions",
                "tag:GetResources",
                "tag:TagResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassRoleToAutomation",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:*:iam::*:role/TroubleshootEbsCsiDriversForEks-SSM-Role",
                "arn:*:iam::*:role/Automation-K8sProxy-Role-*"
            ],
            "Condition": {
                "StringLikeIfExists": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "ssm.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AttachRolePolicy",
            "Effect": "Allow",
```

```
            "Action": [
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "iam:ResourceTag/AWSSupport-SetupK8sApiProxyForEKS": "true"
                }
            }
        }
    ]
}
```

3. Grant the required permissions for Amazon EKS cluster RBAC (Role-Based Access Control). The recommended approach is to create an Access Entry in your Amazon EKS cluster.

   In the Amazon EKS console, navigate to your cluster. For Amazon EKS access entries, verify your access configuration is set to `API_AND_CONFIG_MAP` or `API`. For steps to configure authentication mode for access entries, see Setting up access entries.

   Choose **Create access entry**.

   - For *IAM principal ARN*, select the IAM role you created for SSM automation in the previous step.
   - For *Type*, select `Standard`.

4. Add an access policy:

   - For *Access scope*, select `Cluster`.
   - For *Policy name*, select `AmazonEKSAdminViewPolicy`.

   Choose **Add policy**.

   If you are not using access entries to manage Kubernetes API permissions, you must update the `aws-auth` ConfigMap and create a role binding between your IAM user or role. Ensure your IAM entity has the following read-only Kubernetes API permissions:

   - GET `/apis/apps/v1/namespaces/{namespace}/deployments/{name}`
   - GET `/apis/apps/v1/namespaces/{namespace}/replicasets/{name}`
   - GET `/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}`
   - GET `/api/v1/nodes/{name}`

- GET /api/v1/namespaces/{namespace}/serviceaccounts/{name}
- GET /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}
- GET /api/v1/persistentvolumes/{name}
- GET /apis/storage.k8s.io/v1/storageclasses/{name}
- GET /api/v1/namespaces/{namespace}/pods/{name}
- GET /api/v1/namespaces/{namespace}/pods
- GET /api/v1/namespaces/{namespace}/pods/{name}/log
- GET /api/v1/events

5. Run the automation AWSSupport-TroubleshootEbsCsiDriversForEks (console)
6. Select **Execute automation**.
7. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**
     - Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows SSM Automation to perform the actions on your behalf. The role needs to be added to your Amazon EKS cluster access entry or RBAC permission to allow Kubernetes API calls.
     - Type: AWS::IAM::Role::Arn
     - Example: TroubleshootEbsCsiDriversForEks-SSM-Role
   - **EksClusterName:**
     - Description: The name of the target Amazon Elastic Kubernetes Service (Amazon EKS) cluster.
     - Type: String
   - **ApplicationPodName:**
     - Description: The name of the Kubernetes application pod having issues with the Amazon EBS CSI driver.
     - Type: String
   - **ApplicationNamespace:**
     - Description: The Kubernetes namespace for the application pod having issues with the Amazon EBS CSI driver.
     - Type: String
   - **EbsCsiControllerDeploymentName (Optional):**

- Description: (Optional) The deployment name for the Amazon EBS CSI controller pod.
- Type: `String`
- Default: `ebs-csi-controller`

- **EbsCsiControllerNamespace (Optional):**
  - Description: (Optional) The Kubernetes namespace for the Amazon EBS CSI controller pod.
  - Type: `String`
  - Default: `kube-system`
- **S3BucketName (Optional):**
  - Description: (Optional) The target Amazon S3 bucket name where the troubleshooting logs will be uploaded.
  - Type: `AWS::S3::Bucket::Name`
- **LambdaRoleArn (Optional):**
  - Description: (Optional) The ARN of the IAM role that allows the Amazon Lambda function to access the required Amazon services and resources.
  - Type: `AWS::IAM::Role::Arn`

Select **Execute**.

8. After completed, review the *Outputs* section for the detailed results of the execution.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows

For more information on Amazon EBS CSI Driver, see Amazon EBS CSI Driver.

Elastic Beanstalk

Amazon Systems Manager Automation provides predefined runbooks for Amazon Elastic Beanstalk. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWSSupport-CollectElasticBeanstalkLogs

Description

The `AWSSupport-CollectElasticBeanstalkLogs` runbook gathers Amazon Elastic Beanstalk related log files from an Amazon Elastic Compute Cloud (Amazon EC2) Windows Server instance launched by Elastic Beanstalk to help you troubleshoot common issues. While the automation is gathering the associated log files, changes are made to the file system structure including the creation of temporary directories, the copying of log files to the temporary directories, and compressing the log files into an archive. This activity can result in increased `CPUUtilization` on the Amazon EC2 instance. For more information about `CPUUtilization` , see Instance metrics in the *Amazon CloudWatch User Guide* .

If you specify a value for the `S3BucketName` parameter, the automation evaluates the policy status of the Amazon Simple Storage Service (Amazon S3) bucket you specify. To help with the security of the logs gathered from your Amazon EC2 instance, if the policy status `isPublic` is set to `true` , or if the access control list (ACL) grants `READ|WRITE` permissions to the `All Users` Amazon S3 predefined group, the logs are not uploaded. For more information about Amazon S3 predefined groups, see Amazon S3 predefined groups in the *Amazon Simple Storage Service User Guide* .

If you do not specify a value for the `S3BucketName` parameter, the automation uploads the log bundle to the default Elastic Beanstalk Amazon S3 bucket in the Amazon Web Services Region where you run the automation. The directory is named according to the following structure, `elasticbeanstalk-` *`region`* `-` *`accountID`* . The *`region`* and *`accountID`* values will differ

based on the Region and Amazon Web Services account you run the automation in. The log bundle will be saved to the `resources/environments/logs/bundle/` *`environmentID`* `/` *`instanceID`* directory. The *`environmentID`* and *`instanceID`* values will differ based on your Elastic Beanstalk environment and the Amazon EC2 instance you're gathering logs from.

By default, the Amazon Identity and Access Management (IAM) instance profile attached to the Amazon EC2 instances of the Elastic Beanstalk environment has the required permissions to upload the bundle to the default Elastic Beanstalk Amazon S3 bucket for your environment. If you specify a value for the `S3BucketName` parameter, the instance profile attached to the Amazon EC2 instance must allow the `s3:GetBucketAcl` , `s3:GetBucketPolicy` , `s3:GetBucketPolicyStatus` , and `s3:PutObject` actions for the specified Amazon S3 bucket and path.

> **Note**
>
> This automation requires at least 500 MB of available disk space on the root Amazon Elastic Block Store (Amazon EBS) volume attached to your Amazon EC2 instance. If there is not enough available disk space on the root volume, the automation stops.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EnvironmentId

  Type: String

  Description: (Required) The ID of your Elastic Beanstalk environment you want to collect the log bundle from.

- InstanceId

  Type: String

  (Required) The ID of the Amazon EC2 instance in your Elastic Beanstalk environment you want to collect the log bundle from.

- S3BucketName

  Type: String

  (Optional) The Amazon S3 bucket you want to upload the archived logs to.

- S3BucketPath

  Type: String

  (Optional) The Amazon S3 bucket path you want to upload the log bundle to. This parameter is ignored if you do not specify a value for the `S3BucketName` parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:SendCommand`
- `ssm:DescribeInstanceInformation`
- `ec2:DescribeInstances`

Document Steps

- `aws:assertAwsResourceProperty` - Confirms the Amazon EC2 instance you specify in the `InstanceId` parameter is managed by Amazon Systems Manager.
- `aws:assertAwsResourceProperty` - Confirms the Amazon EC2 instance you specify in the `InstanceId` parameter is a Windows Server instance.
- `aws:runCommand` - Checks whether the instance is part of an Elastic Beanstalk environment, if there is sufficient disk space to bundle the logs, and whether the Amazon S3 bucket to which the logs would be uploaded to is public.
- `aws:runCommand` - Collects the log files and uploads the archive to the Amazon S3 bucket specified in the `S3BucketName` parameter or to the default bucket for your Elastic Beanstalk environment if a value is not specified.

AWSConfigRemediation-EnableElasticBeanstalkEnvironmentLogStreaming

Description

The `AWSConfigRemediation-EnableElasticBeanstalkEnvironmentLogStreaming` runbook enables logging on the Amazon Elastic Beanstalk (Elastic Beanstalk) environment you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- EnvironmentId

  Type: String

  Description: (Required) The ID of the Elastic Beanstalk environment that you want to enable logging on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticbeanstalk:DescribeConfigurationSettings`
- `elasticbeanstalk:DescribeEnvironments`
- `elasticbeanstalk:UpdateEnvironment`

Document Steps

- `aws:executeAwsApi` - Enables logging on the Elastic Beanstalk environment you specify in the `EnvironmentId` parameter.
- `aws:waitForAwsResourceProperty` - Waits for the status of the environment to change to `Ready` .
- `aws:executeScript` - Verifies logging has been enabled on the Elastic Beanstalk environment.

AWSConfigRemediation-EnableBeanstalkEnvironmentNotifications

Description

The `AWSConfigRemediation-EnableBeanstalkEnvironmentNotifications` runbook enables notifications for the Amazon Elastic Beanstalk (Elastic Beanstalk) environment you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- EnvironmentId

  Type: String

  Description: (Required) The ID of the Elastic Beanstalk environment that you want to enable notifications for.

- TopicArn

  Type: String

  Description: (Required) The ARN of the Amazon Simple Notification Service (Amazon SNS) topic you want to send notifications to.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- elasticbeanstalk:DescribeConfigurationSettings
- elasticbeanstalk:DescribeEnvironments
- elasticbeanstalk:UpdateEnvironment

Document Steps

- aws:executeAwsApi - Enables notifications for the Elastic Beanstalk environment you specify in the EnvironmentId parameter.
- aws:waitForAwsResourceProperty - Waits for the status of the environment to change to Ready .
- aws:executeScript - Verifies notifications have been enabled for the Elastic Beanstalk environment.

AWSSupport-TroubleshootElasticBeanstalk

Description

The AWSSupport-TroubleshootElasticBeanstalk runbook helps you troubleshoot the potential reasons why your Amazon Elastic Beanstalk environment is in a Degraded or Severe state. This automation checks the following Amazon resources associated with your Elastic Beanstalk environment:

- Configuration details for a load balancer, Amazon CloudFormation stack, Amazon EC2 Auto Scaling group, Amazon Elastic Compute Cloud (Amazon EC2) instances, and virtual private cloud (VPC).
- Network configuration issues with the associated security group rules, route tables, and network access control lists (ACLs) associated with your subnets.
- Verifies connectivity to the Elastic Beanstalk endpoints and public internet access.
- Verifies the status of the load balancer.
- Verifies the status of the Amazon EC2 instances.

- Retrieves a log bundle from your Elastic Beanstalk environment, and optionally uploads the files to Amazon Web Services Support.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ApplicationName

  Type: String

  Description: (Required) The name of your Elastic Beanstalk application.

- EnvironmentName

  Type: String

  Description: (Required) The name of your Elastic Beanstalk environment.

- AWSS3UploaderLink

  Type: String

Description: (Optional) A URL provided to you by Amazon Web Services Support to upload the log bundle from your Elastic Beanstalk environment to. This option is only available to customers who have purchased an Amazon Web Services Support plan, and have opened a Support case.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `autoscaling:Describe*`
- `cloudformation:Describe*`
- `cloudformation:Estimate*`
- `cloudformation:Get*`
- `cloudformation:List*`
- `cloudformation:Validate*`
- `cloudwatch:Describe*`
- `cloudwatch:Get*`
- `cloudwatch:List*`
- `ec2:Describe*`
- `elasticbeanstalk:Check*`
- `elasticbeanstalk:Describe*`
- `elasticbeanstalk:List*`
- `elasticbeanstalk:RetrieveEnvironmentInfo*`
- `elasticbeanstalk:RequestEnvironmentInfo*`
- `elasticloadbalancing:Describe*`
- `rds:Describe*`
- `s3:Get*`
- `s3:List*`
- `sns:Get*`
- `sns:List*`

Document Steps

- `aws:executeScript` - Verifies the Amazon Identity and Access Management (IAM) principal who started the automation has the requisite permissions to perform all of the actions defined in the runbook.
- `aws:branch` - Branches the workflow based on the results of the previous step.
- `aws:executeScript` - Collects information about the Elastic Beanstalk environment including the load balancer, Amazon CloudFormation stack, Auto Scaling group, Amazon EC2 instances, and VPC configuration.
- `aws:executeScript` - Checks for network connectivity issues with the route tables and ACLs associated with the subnets in your VPC.
- `aws:executeScript` - Checks for network connectivity issues with the security group rules associated with your Amazon EC2 instances.
- `aws:executeScript` - Verifies the status checks for the Amazon EC2 instances.
- `aws:executeScript` - Generates a link for a log bundle of your Elastic Beanstalk environment.
- `aws:executeScript` - Uploads log bundle to Amazon Web Services Support.
- `aws:executeScript` - Outputs a report of action items to help you troubleshoot issues that might be affecting the status of your Elastic Beanstalk environment.

Elastic Load Balancing

Amazon Systems Manager Automation provides predefined runbooks for Elastic Load Balancing. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-DropInvalidHeadersForALB
- AWS-EnableCLBAccessLogs
- AWS-EnableCLBConnectionDraining
- AWSConfigRemediation-EnableCLBCrossZoneLoadBalancing
- AWSConfigRemediation-EnableELBDeletionProtection
- AWSConfigRemediation-EnableLoggingForALBAndCLB
- AWSSupport-TroubleshootCLBConnectivity
- AWSConfigRemediation-EnableNLBCrossZoneLoadBalancing
- AWS-UpdateALBDesyncMitigationMode

- AWS-UpdateCLBDesyncMitigationMode

AWSConfigRemediation-DropInvalidHeadersForALB

Description

The AWSConfigRemediation-DropInvalidHeadersForALB runbook enables the application load balancer you specify to remove HTTP headers with invalid headers.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LoadBalancerArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the load balancer that you want to drop invalid headers.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- elasticloadbalancing:DescribeLoadBalancerAttributes
- elasticloadbalancing:ModifyLoadBalancerAttributes

Document Steps

- aws:executeAwsApi - Enables the drop invalid headers setting for the load balancer you specify in the LoadBalancerArn parameter.
- aws:executeScript - Verifies the drop invalid headers setting has been enabled on the load balancer you specify in the LoadBalancerArn parameter.

AWS-EnableCLBAccessLogs

Description

The AWS-EnableCLBAccessLogs runbook enables access logs for a Classic Load Balancer.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EmitInterval

  Type: Integer

  Valid values: 5 | 60

  Default: 60

  Description: (Optional) The interval for publishing the access logs in minutes.

- LoadBalancerNames

  Type: String

  Description: (Required) A comma separated list of Classic Load Balancers you want to enable access logs for.

- S3BucketName

  Type: String

  Description: (Required) The name of the Amazon Simple Storage Service (Amazon S3) bucket where the access logs are stored.

- S3BucketPrefix

  Type: String

  Description: (Optional) The logical hierarchy you created for your Amazon S3 bucket, for example `my-bucket-prefix/prod`. If the prefix is not provided, the log is placed at the root level of the bucket.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- `aws:executeAwsApi` - Enables access logs for the Classic Load Balancers you specify in the `LoadBalancerNames` parameter.

Outputs

EnableCLBAccessLogs.SuccessesLoadBalancers - List of load balancer names where access logs were successfully enabled.

EnableCLBAccessLogs.FailedLoadBalancers - MapList of load balancer names where enabling access logs failed and the reason for the failure.

AWS-EnableCLBConnectionDraining

Description

The `AWS-EnableCLBConnectionDraining` runbook enables connection draining on a Classic Load Balancer (CLB) to the specified timeout value. Connection drainings enables the CLB to complete in-flight requests made to instances that are deregistering or unhealthy with the specified timeout being the time it keeps connections alive before reporting the instance as deregistered. For more information about connection draining on CLBs, see Configure connection draining for your Classic Load Balancer in the *User Guide for Classic Load Balancers*.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- LoadBalancerName

  Type: String

  Description: (Required) The name of the load balancer you want to enable connection draining on.
- ConnectionTimeout

  Type: Integer

  Valid values: 1-3600

  Default: 300

  Description: (Required) The connection timeout value for the load balancer. The timeout value can be set between 1 and 3600 seconds.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- ModifyLoadBalancerConnectionDraining (aws:executeAwsApi): Enables connection draining and sets the specified timeout value for the load balancer you specify.

- VerifyLoadBalancerConnectionDrainingEnabled (aws:assertAwsResourceProperty): Verifies that connection draining is enabled for the load balancer.
- VerifyLoadBalancerConnectionDrainingTimeout (aws:assertAwsResourceProperty): Verifies that the connection timeout value for the load balancer matches the value you specified.

AWSConfigRemediation-EnableCLBCrossZoneLoadBalancing

Description

The `AWSConfigRemediation-EnableCLBCrossZoneLoadBalancing` runbook enables cross-zone load balancing for the Classic Load Balancer (CLB) you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LoadBalancerName

  Type: String

  Description: (Required) The name of the CLB that you want to enable cross-zone load balancing on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elb:DescribeLoadBalancerAttributes`
- `elb:ModifyLoadBalancerAttributes`

Document Steps

- `aws:executeAwsApi` - Enables cross-zone load balancing for the CLB you specify in the `LoadBalancerName` parameter.
- `aws:assertAwsResourceProperty` - Verifies cross-zone load balancing has been enabled on the CLB.

AWSConfigRemediation-EnableELBDeletionProtection

Description

The `AWSConfigRemediation-EnableELBDeletionProtection` runbook enables deletion protection for the elastic load balancer (ELB) you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LoadBalancerArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the ELB that you want to enable deletion protection on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:DescribeLoadBalancers`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- `aws:executeScript` - Enables deletion protection on the ELB you specify in the `LoadBalancerArn` parameter.

AWSConfigRemediation-EnableLoggingForALBAndCLB

Description

The `AWSConfigRemediation-EnableLoggingForALBAndCLB` runbook enables logging for the specified Amazon Application Load Balancer or a Classic Load Balancer (CLB).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LoadBalancerId

  Type: String

  Description: (Required) The Classic Load Balancer name or the Application Load Balancer ARN.
- S3BucketName

  Type: String

  Description: (Required) The Amazon S3 bucket name.
- S3BucketPrefix

  Type: String

  Description: (Optional) The logical hierarchy you created for your Amazon Simple Storage Service (Amazon S3) bucket, for example `my-bucket-prefix/prod` . If the prefix is not provided, the log is placed at the root level of the bucket.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- `aws:executeScript` - Enables and verifies the logging for the Classic Load Balancer or the Application Load Balancer.

AWSSupport-TroubleshootCLBConnectivity

Description

The `AWSSupport-TroubleshootCLBConnectivity` runbook help you troubleshoot connectivity issues between a Classic Load Balancer (CLB) and Amazon Elastic Compute Cloud (Amazon EC2) instances. Also, connectivity issues between a client and the CLB are reviewed. This runbook also reviews health checks for the CLB, verifies that best practices are being followed, and creates a troubleshooting dashboard for you. Optionally, you can upload the automation output to an Amazon Simple Storage Service (Amazon S3) bucket. However, this runbook does not support uploading output to S3 buckets that are publicly accessible. We recommend creating a temporary S3 bucket for this automation.

> **Important**
>
> Using this runbook might incur charges for the dashboard that is created. For more information, see Amazon CloudWatch Pricing

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InvestigationType

  Type: String

  Valid values: Best Practices | Connectivity Issues | Troubleshooting Dashboard

  Description: (Required) The operations you want the runbook to perform.
- LoadBalancerName

  Type: String

  Description: (Required) The name of the CLB.
- S3Location

  Type: String

  Description: (Optional) The name of the S3 bucket you want to send the automation results to. Publicly accessible buckets are not supported. If your S3 bucket uses server-side encryption, the user or role running this automation must have `kms:GenerateDataKey` permissions for the Amazon KMS key.
- S3LocationPrefix

  Type: String

Description: (Optional) The Amazon S3 key prefix (subfolder) you want to upload the automation output to. The format output is stored in the following format: amzn-s3-demo-bucket/*S3LocationPrefix*/{{*InvestigationType*}}_{{automation:*EXECUTION_ID*}}.txt.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`
- `elasticloadbalancing:DescribeLoadBalancers`
- `elasticloadbalancing:DescribeLoadBalancerPolicies`
- `elasticloadbalancing:DescribeInstanceHealth`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `iam:ListRoles`
- `cloudwatch:PutDashboard`
- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:DescribeInstanceInformation`
- `ssm:DescribeInstanceProperties`
- `ssm:GetDocument`
- `ssm:ListCommands`

- ssm:ListCommandInvocations
- ssm:ListDocuments
- ssm:SendCommand
- s3:GetBucketAcl
- s3:GetBucketPolicyStatus
- s3:GetPublicAccessBlock
- s3:PutObject

Document Steps

- aws:executeScript - Verifies that the CLB you specify in the LoadBalancerName parameter exists.
- aws:branch - Branches based on the value specified for the InvestigationType parameter.
- aws:executeScript - Performs connectivity checks to the CLB.
- aws:executeScript - Verifies that the CLB configuration adheres to Elastic Load Balancing best practices.
- aws:executeScript - Creates an Amazon CloudWatch dashboard for your CLB.
- aws:executeScript - Creates a text file with the results of the automation and uploads it to the Amazon S3 bucket you specify in the S3Location parameter.

Outputs

RunBestPractices.Summary

RunConnectivityChecks.Summary

CreateTroubleshootingDashboard.Output

UploadOutputToS3.Output

AWSConfigRemediation-EnableNLBCrossZoneLoadBalancing

Description

The AWSConfigRemediation-EnableNLBCrossZoneLoadBalancing runbook enables cross zone load balancing for the network load balancer (NLB) you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LoadBalancerArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the NLB that you want to enable cross zone load balancing on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- `aws:executeAwsApi` - Enables cross zone load balancing for the NLB you specify in the `LoadBalancerArn` parameter.
- `aws:executeScript` - Verifies cross zone load balancing has been enabled on the NLB.

AWS-UpdateALBDesyncMitigationMode

Description

The `AWS-UpdateALBDesyncMitigationMode` runbook will update the desync mitigation mode on an Application Load Balancer (ALB) to the specified mitigation mode. The desync mitigation mode determines how the load balancer handles requests that might pose a security risk to your application.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- LoadBalancerArn

Type: String

Description: (Required) The Amazon Resource Name (ARN) of the ALB that you want to modify the desync mitigation mode of.

- DesyncMitigationMode

  Type: String

  Valid values: monitor | defensive | strictest

  Description: (Required) The mitigation mode that you want the ALB to use. For information about desync mitigation modes, see Desync mitigation mode in the *User Guide for Application Load Balancers*.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticloadbalancing:DescribeLoadBalancers`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- VerifyLoadBalancerType (aws:assertAwsResourceProperty) - Verifies that the value specified for the `LoadBalancerArn` input parameter is for an application load balancer before proceeding to the next step.
- ModifyLoadBalancerDesyncMode (aws:executeAwsApi) - Updates the ALB to use the specified `DesyncMitigationMode`.
- VerifyLoadBalancerDesyncMitigationMode (aws:executeScript) - Verifies that the desync mitigation mode was updated for the target ALB.

Outputs

VerifyLoadBalancerDesyncMitigationMode.ModificationResult - Message payload of the script verifying the modification to your ALB.

AWS-UpdateCLBDesyncMitigationMode

Description

The AWS-UpdateCLBDesyncMitigationMode runbook will update the desync mitigation mode on an Classic Load Balancer (CLB) to the specified mitigation mode. The desync mitigation mode determines how the load balancer handles requests that might pose a security risk to your application.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- LoadBalancerName

  Type: String

  Description: (Required) The name of the CLB that you want to modify the desync mitigation mode of.

- DesyncMitigationMode

Type: String

Valid values: monitor | defensive | strictest

Description: (Required) The mitigation mode that you want the CLB to use. For information about desync mitigation modes, see Desync mitigation mode in the *User Guide for Application Load Balancers*.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `elasticloadbalancing:DescribeLoadBalancerAttributes`
- `elasticloadbalancing:ModifyLoadBalancerAttributes`

Document Steps

- ModifyLoadBalancerDesyncMode (aws:executeAwsApi) - Updates the CLB to use the specified `DesyncMitigationMode`.
- VerifyLoadBalancerDesyncMitigationMode (aws:executeScript) - Verifies that the desync mitigation mode was updated for the target CLB.

Outputs

VerifyLoadBalancerDesyncMitigationMode.ModificationResult - Message payload of the script verifying the modification to your CLB.

Amazon EMR

Amazon Systems Manager Automation provides predefined runbooks for Amazon EMR. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-AnalyzeEMRLogs
- AWSSupport-DiagnoseEMRLogsWithAthena

AWSSupport-AnalyzeEMRLogs

Description

This runbook helps identify errors while running a job on an Amazon EMR cluster. The runbook analyzes a list of defined logs on the file system and looks for a list of predefined keywords. These log entries are used to create Amazon CloudWatch Events events so you can take any needed actions based on the events. Optionally, the runbook publishes log entries to the Amazon CloudWatch Logs log group of your choosing. This runbook currently looks for the following errors and patterns in log files:

- container_out_of_memory – YARN container ran out of memory, running job may fail.
- yarn_nodemanager_health: CORE or TASK node is running low on disk space and will not be able to run tasks.
- node_state_change: CORE or TASK node is unreachable by the MASTER node.
- step_failure: An EMR Step has failed.
- no_core_nodes_running: No CORE nodes are currently running, cluster is unhealthy.
- hdfs_missing_blocks: There are missing HDFS blocks which could lead to data loss.
- hdfs_high_util: HDFS Utilization is high, which may affect jobs and cluster health.
- instance_controller_restart: Instance-Controller process has restarted. This process is essential for cluster health.
- instance_controller_restart_legacy: Instance-Controller process has restarted. This process is essential for cluster health.
- high_load: High Load Average detected, may affect node health reporting or result in timeouts or slowdowns.
- yarn_node_blacklisted: CORE or TASK node has been blacklisted by YARN from running tasks.
- yarn_node_lost: CORE or TASK node has been marked as LOST by YARN, possible connectivity issues.

Instances associated with the `ClusterID` that you specify must be managed by Amazon Systems Manager. You can run this automation once, schedule the automation to run at a specific time

interval, or remove a schedule created previously by an automation. This runbook supports Amazon EMR release versions 5.20 to 6.30.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- ClusterID

  Type: String

  Description: (Required) The ID of the cluster whose nodes logs you want to analyze.
- Operation

  Type: String

  Valid values: Run Once | Schedule | Remove Schedule

  Description: (Required) The operation to perform on the cluster.
- IntervalTime

  Type: String

Valid values: 5 minutes | 10 minutes | 15 minutes

Description: (Optional) The duration of time between running the automation. This parameter is only applicable if you specify `Schedule` for the `Operation` parameter.

- LogToCloudWatchLogs

  Type: String

  Valid values: yes | no

  Description: (Optional) If you specify `yes` for the value of this parameter, the automation creates a CloudWatch Logs log group with the name specified in the `CloudWatchLogGroup` parameter to store any matched log entries.

- CloudWatchLogGroup

  Type: String

  Description: (Optional) The name of the CloudWatch Logs log group you want to store any matched log entries in. This parameter is only applicable if you specify `yes` for the `LogToCloudWatchLogs` parameter.

- CreateLogInsightsDashboard

  Type: String

  Valid values: yes | no

  Description: (Optional) If you specify `yes` , CloudWatch dashboard is created if it does not already exist. This parameter is only applicable if you specify `yes` for the `LogToCloudWatchLogs` parameter.

- CreateMetricFilters

  Type: String

  Valid values: yes | no

  Description: (Optional) Specify `yes` if you want to create metric filters for the CloudWatch Logs log group. This parameter is only applicable if you specify `yes` for the `LogToCloudWatchLogs` parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetDocument`
- `ssm:ListDocuments`
- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:GetAutomationExecution`
- `ssm:DescribeInstanceInformation`
- `ssm:ListCommandInvocations`
- `ssm:ListCommands`
- `ssm:SendCommand`
- `iam:CreateRole`
- `iam:DeleteRole`
- `iam:GetRolePolicy`
- `iam:PutRolePolicy`
- `iam:DeleteRolePolicy`
- `iam:passrole`
- `cloudformation:DescribeStacks`
- `cloudformation:DeleteStack`
- `cloudformation:CreateStack`
- `events:DeleteRule`
- `events:RemoveTargets`
- `events:PutTargets`
- `events:PutRule`
- `events:DescribeRule`
- `logs:DescribeLogGroups`
- `logs:CreateLogGroup`
- `logs:PutMetricFilter`

- cloudwatch:PutDashboard
- elasticmapreduce:ListInstances
- elasticmapreduce:DescribeCluster

Document Steps

- aws:executeAwsApi - Gathers information about the Amazon EMR cluster specified in the ClusterID parameter.
- aws:branch - Branches based on input.
  - If the provided operation is Run Once or Schedule :
    - aws:assertAwsResourceProperty - Verifies the cluster is available.
    - aws:executeAwsApi - Gathers the IDs of all instances running in the cluster.
    - aws:assertAwsResourceProperty - Verifies the SSM Agent is running on all instances in the cluster.
    - aws:branch - Branches based on whether you specified to run the automation once or on a schedule.
      - If the provided operation is Run Once :
        - aws:branch - Branches based on the value specified in the LogToCloudWatchLogs parameter.
          - If LogToCloudWatchLogs value is yes :
            - aws:executeScript - Checks if a CloudWatch Logs log group with the name specified in parameter CloudWatchLogGroup already exists. If not, the group is created with the name specified.
            - aws:branch - Branches based on the value specified in the CreateMetricFilters parameter.
              - If CreateMetricFilters value is yes :
                - aws:executeAwsApi - 12 steps are ran for each metric filter
                - aws:branch - Branches based on the value specified in the CreateLogInsightsDashboard parameter.
                  - If CreateLogInsightsDashboard value is yes :
                    - aws:executeAwsApi - Creates a CloudWatch dashboard with the same name specified in the CloudWatchLogGroup parameter, if it does not already exist.

- If `CreateLogInsightsDashboard` value is no :
  - `aws:runCommand` - Runs a shell script to find log patterns on each instance in the cluster.
- If `CreateMetricFilters` value is no :
  - `aws:branch` - Branches based on the value specified in `CreateLogInsightsDashboard` parameter.
    - If `CreateLogInsightsDashboard` value is `yes` :
      - `aws:executeAwsApi` - Creates a CloudWatch dashboard with the same name specified in the `CloudWatchLogGroup` parameter, if it does not already exist.
    - If `CreateLogInsightsDashboard` value is no :
      - `aws:runCommand` - Runs a shell script to find log patterns on each instance in the cluster.
- If `LogToCloudWatchLogs` value is no :
  - `aws:executeAwsApi` - Runs a shell script to find log patterns on each instance in the cluster.
- If the provided operation is `Schedule` :
  - `aws:createStack` - Creates an Amazon EventBridge event that targets this runbook.
- If the provided operation is `Remove Schedule` :
  - `aws:executeAwsApi` - Verifies a schedule exists for the cluster.
  - `aws:deleteStack` - Deletes the schedule.

Outputs

GetClusterInformation.ClusterName

GetClusterInformation.ClusterState

ListingClusterInstances.InstanceIDs

CreatingScheduleCloudFormationStack.StackStatus

RemovingScheduleByDeletingScheduleCloudFormationStack.StackStatus

CheckIfLogGroupExists.output

FindLogPatternOnEMRNode.CommandId

AWSSupport-DiagnoseEMRLogsWithAthena

Description

The AWSSupport-DiagnoseEMRLogsWithAthena runbook helps diagnose Amazon EMR logs using Amazon Athena in integration with Amazon Glue Data Catalog. Amazon Athena is used to query the Amazon EMR log files for containers, node logs, or both, with optional parameters for specific date ranges or keyword-based searches.

The runbook can automatically retrieve the Amazon EMR log location for an existing cluster, or you can specify the Amazon S3 log location. To analyze the logs, the runbook:

- Creates an Amazon Glue database and executes Amazon Athena Data Definition Language (DDL) queries on the Amazon EMR Amazon S3 log location to create tables for cluster logs and a list of known issues.
- Executes Data Manipulation Language (DML) queries to search for known issue patterns in the Amazon EMR logs. The queries return a list of detected issues, their occurrence count, and the number of matched keywords by Amazon S3 file path.
- The results are uploaded to an Amazon S3 bucket you specify under the prefix saw_diagnose_EMR_known_issues.
- The runbook returns the Amazon Athena query results, highlighting findings, recommendations, and references to Amazon Knowledge Center (KC) articles sourced from a predefined subset.
- Upon completion or failure, the Amazon Glue database and the known issues files uploaded to the Amazon S3 bucket are deleted.

How does it work?

The AWSSupport-DiagnoseEMRLogsWithAthena perform analysis of Amazon EMR logs using Amazon Athena to detect errors and highlight findings, recommendations and relevant Knowledge Center articles.

The runbook performs the following steps:

- Get Amazon EMR cluster log location using cluster ID or input Amazon S3 location to retrieve log location and size.
- Provide Athena costs estimate based on log location size.

- Get approval to proceed by requesting approval from designated IAM principals before running Athena queries and continuing to the next steps.
- Upload known issues to the specified Amazon S3 bucket, creates an Amazon Glue database and tables.
- Execute Athena queries on the Amazon EMR logs data. Queries can search by date range, keywords, both criteria, or run without filters based on the provided inputs.
- Analyze results to highlight findings, recommendations, and relevant KC articles.
- Output links for Amazon Athena DML queries results.
- Clean up the environment by removing created database, tables, and uploaded known issues.

Document type

Automation

Owner

Amazon

Platforms

/

The AutomationAssumeRole parameter requires the following actions to successfully use the runbook:

- athena:GetQueryExecution
- athena:StartQueryExecution
- athena:GetPreparedStatement
- athena:CreatePreparedStatement
- glue:GetDatabase
- glue:CreateDatabase
- glue:DeleteDatabase
- glue:CreateTable
- glue:GetTable
- glue:DeleteTable
- elasticmapreduce:DescribeCluster

- s3:ListBucket
- s3:GetBucketVersioning
- s3:ListBucketVersions
- s3:GetBucketPublicAccessBlock
- s3:GetBucketPolicyStatus
- s3:GetObject
- s3:GetBucketLocation
- pricing:GetProducts
- pricing:GetAttributeValues
- pricing:DescribeServices
- pricing:ListPriceLists

Important

To restrict access to only the resources needed by this automation, attach the following policy to the IAM role that trusts the SSM Service. Replace the Partition, Region and Account with the appropriate values for the partition, region and account number where the run book is executed.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeCluster",
        "glue:GetDatabase",
        "athena:GetQueryExecution",
        "athena:StartQueryExecution",
        "athena:GetPreparedStatement",
        "athena:CreatePreparedStatement",
        "s3:ListBucket",
        "s3:GetBucketVersioning",
        "s3:ListBucketVersions",
        "s3:GetBucketPublicAccessBlock",
```

```
        "s3:GetBucketPolicyStatus",
        "s3:GetObject",
        "s3:GetBucketLocation",
        "pricing:GetProducts",
        "pricing:GetAttributeValues",
        "pricing:DescribeServices",
        "pricing:ListPriceLists"
      ],
      "Resource": "*"
    },
    {
      "Sid": "RestrictPutObjects",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:{Partition}:s3:::*/*/results/*",
        "arn:{partition}:s3:::*/*/saw_diagnose_emr_known_issues/*"
      ]
    },
    {
      "Sid": "RestrictDeleteAccess",
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": [
        "arn:{Partition}:s3:::*/*/saw_diagnose_emr_known_issues/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase",
        "glue:CreateDatabase",
        "glue:DeleteDatabase"
      ],
      "Resource": [
        "arn:{Partition}:glue:{Region}:{Account}:database/saw_diagnose_emr_database_*",
        "arn:{Partition}:glue:{Region}:{Account}:table/saw_diagnose_emr_database_*/*",
        "arn:{Partition}:glue:{Region}:{Account}:userDefinedFunction/
saw_diagnose_emr_database_*/*",
```

```
        "arn:{Partition}:glue:{Region}:{Account}:catalog"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateTable",
        "glue:GetTable",
        "glue:DeleteTable"
      ],
      "Resource": [
        "arn:{Partition}:glue:{Region}:{Account}:table/saw_diagnose_emr_database_*/
saw_diagnose_emr_known_issues",
        "arn:{Partition}:glue:{Region}:{Account}:table/saw_diagnose_emr_database_*/
saw_diagnose_emr_logs_table",
        "arn:{Partition}:glue:{Region}:{Account}:table/saw_diagnose_emr_database_*/
j_*",
        "arn:{Partition}:glue:{Region}:{Account}:database/saw_diagnose_emr_database_*",
        "arn:{Partition}:glue:{Region}:{Account}:catalog"
      ]
    }
  ]
}
```

Instructions

Follow these steps to configure the automation:

1. Navigate AWSSupport-DiagnoseEMRLogsWithAthena in the Amazon Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **ClusterID (Required):**

     The Amazon EMR cluster ID.

- **S3LogLocation (Optional):**

  The Amazon S3 Amazon EMR log location. Input the Path-style URL Amazon S3 location, for example: `s3://amzn-s3-demo-bucket/myfolder/j-1K48XXXXXXHCB/`. Provide this parameter if the Amazon EMR cluster has been terminated for more than `30` days.

- **S3BucketName (Required):**

  The Amazon S3 bucket name to upload a list of known issues, and the output of Amazon Athena queries. The bucket should have Block Public Access Enabled and be in the same Amazon region and account as the Amazon EMR cluster.

- **Approvers (Required):**

  The list of Amazon authenticated principals who are able to either approve or reject the action. You can specify principals by using any of the following formats: user name, user ARN, IAM role ARN, or IAM assume role ARN. The maximum number of approvers is 10.

- **FetchNodeLogsOnly (Optional):**

  If set to `true`, the automation diagnoses the Amazon EMR application containers logs. The default value is `false`.

- **FetchContainersLogsOnly (Optional):**

  If set to `true`, the automation diagnoses the Amazon EMR containers logs. The default value is `false`.

- **EndSearchDate (Optional):**

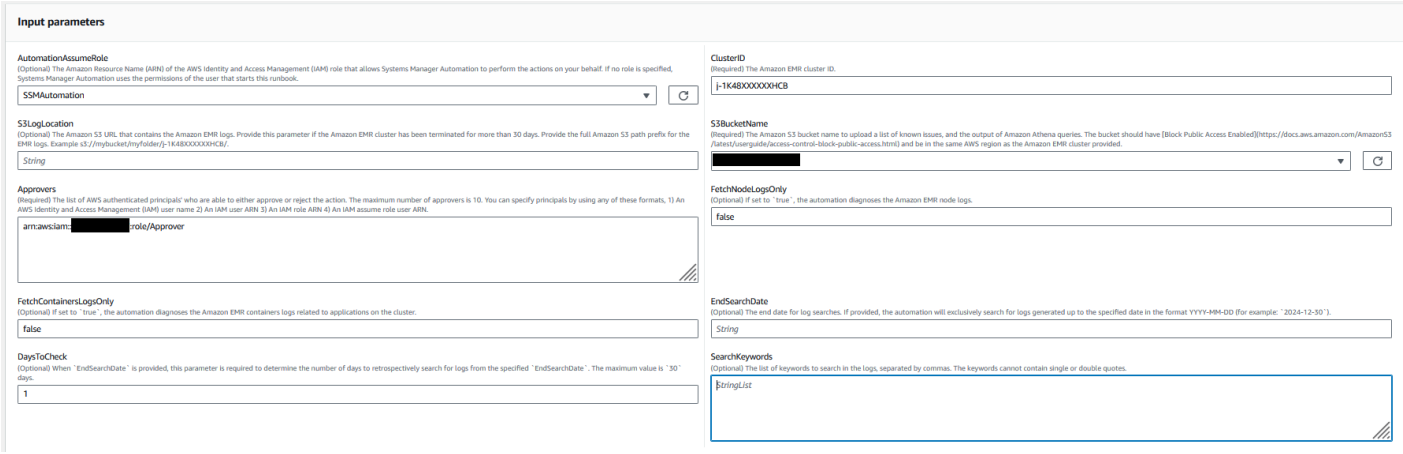
  The end date for log searches. If provided, the automation will exclusively search for logs generated up to the specified date in the format YYYY-MM-DD (for example: `2024-12-30`).

- **DaysToCheck (Optional):**

  When `EndSearchDate` is provided, this parameter is required to determine the number of days to retrospectively search for logs from the specified `EndSearchDate`. The maximum value is `30` days. The default value is `1`.

- **SearchKeywords (Optional):**

  The list of keywords to search in the logs, separated by commas. The keywords cannot contain single or double quotes.



4. Select **Execute.**
5. The automation initiates.
6. The document performs the following steps:
   - **getLogLocation:**

     Retrieves the Amazon S3 log location by querying the specified Amazon EMR Cluster ID. If the automation is unable to query the log location from the Amazon EMR cluster ID, the runbook uses the `S3LogLocation` input parameter.
   - **branchOnValidLog:**

     Verifies the Amazon EMR logs location. If the location is valid, proceed to estimate the Amazon Athena potential costs when executing queries on the Amazon EMR logs.
   - **estimateAthenaCosts:**

     Determines the size of Amazon EMR logs and provides a cost estimate for executing Athena scans on the log dataset. For non-commercial regions (non-Amazon partitions), this step just provides the log size without estimating costs. Costs can be calculated using the Athena pricing documentation in the specified region.
   - **approveAutomation:**

     Waits for the designated IAM principals approval to proceed with the next steps of the automation. The approve notification contains the estimated cost of Amazon Athena scan on the Amazon EMR logs, and details about the resources being provisioned by the automation.
   - **uploadKnownIssuesExecuteAthenaQueries:**

Uploads the predefined known issues to the Amazon S3 bucket specified in the `S3BucketName` parameter. Creates Amazon Glue database and tables. Executes Amazon Athena queries in the Amazon Glue database based on the input parameters.

- **getQueryExecutionStatus:**

  Waits until the Amazon Athena query execution is in `SUCCEEDED` state. The Amazon Athena DML query searches for errors and exceptions in Amazon EMR cluster logs.

- **analyzeAthenaResults:**

  Analyzes the Amazon Athena results to provide findings, recommendations, and Knowledge Center (KC) articles sourced from a predefined set of mappings.

- **getAnalyzeResultsQuery1ExecutionStatus:**

  Waits until the query execution is in `SUCCEEDED` state. The Amazon Athena DML query analyzes the results from the previous DML query. This analysis query will return matched exceptions with resolutions and KC articles

- **getAnalyzeResultsQuery2ExecutionStatus:**

  Waits until the query execution is in `SUCCEEDED` state. The Amazon Athena DML query analyzes the results from the previous DML query. This analysis query will return a list of exceptions/errors detected in each Amazon S3 log path.

- **printAthenaQueriesMessage:**

  Prints links for the Amazon Athena DML queries results.

- **cleanupResources:**

  Clean-ups resources by deleting the created Amazon Glue database and delete known issues files that were created in the Amazon EMR logs bucket.

7. After completed, review the Outputs section for the detailed results of the execution:

   Output provides three links for Athena query results:

   - List of all errors and frequently occurred exceptions found in the Amazon EMR cluster logs, along with the corresponding log locations (Amazon S3 prefix).
   - Summary of unique known exceptions matched in the Amazon EMR logs, along with recommended resolutions and KC articles to help in troubleshooting.

- Details on where specific errors and exceptions appear in the Amazon S3 log paths, to support further diagnosis.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- Refer toTroubleshooting Amazon EMR Clusters for more information

Amazon OpenSearch Service

Amazon Systems Manager Automation provides predefined runbooks for Amazon OpenSearch Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-DeleteOpenSearchDomain
- AWSConfigRemediation-EnforceHTTPSOnOpenSearchDomain
- AWSConfigRemediation-UpdateOpenSearchDomainSecurityGroups
- AWSSupport-TroubleshootOpenSearchRedYellowCluster
- AWSSupport-TroubleshootOpenSearchHighCPU

AWSConfigRemediation-DeleteOpenSearchDomain

Description

The AWSConfigRemediation-DeleteOpenSearchDomain runbook deletes the given Amazon OpenSearch Service domain using the DeleteDomain API.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- DomainName

  Type: String

  Allowed values: (\d{12}/)?[a-z]{1}[a-z0-9-]{2,28}

  Description: (Required) The name of the Amazon OpenSearch Service domain that you want to delete.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- es:DeleteDomain
- es:DescribeDomain

Document Steps

- aws:executeScript - Accepts the Amazon OpenSearch Service domain name as input, deletes it, and verifies the deletion.

AWSConfigRemediation-EnforceHTTPSOnOpenSearchDomain

Description

The AWSConfigRemediation-EnforceHTTPSOnOpenSearchDomain runbook enables EnforceHTTPS on a given Amazon OpenSearch Service domain using the UpdateDomainConfig API.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- DomainName

Type: String

Allowed values: (\d{12}/)?[a-z]{1}[a-z0-9-]{2,28}

Description: (Required) The name of the Amazon OpenSearch Service domain that you want to use to enforce HTTPS.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `es:DescribeDomain`
- `es:UpdateDomainConfig`

Document Steps

- `aws:executeScript` - Enables the `EnforceHTTPS` endpoint option on the Amazon OpenSearch Service domain you specify in the `DomainName` parameter.

AWSConfigRemediation-UpdateOpenSearchDomainSecurityGroups

Description

The `AWSConfigRemediation-UpdateOpenSearchDomainSecurityGroups` runbook updates the security group configuration on a given Amazon OpenSearch Service domain using the UpdateDomainConfig API.

> **Note**
>
> Amazon Security groups can only be applied to Amazon OpenSearch Service domains configured for Amazon Virtual Private Cloud (VPC) Access, and not to Amazon OpenSearch Service domains configured for Public Access.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- DomainName

  Type: String

  Description: (Required) The name of the Amazon OpenSearch Service domain that you want to use to update security groups.
- SecurityGroupList

  Type: StringList

  Description: (Required) The security group IDs that you want to assign to the Amazon OpenSearch Service domain.
- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `es:DescribeDomain`
- `es:UpdateDomainConfig`

Document Steps

- `aws:executeScript` - Updates the security group configuration on the Amazon OpenSearch Service domain you specify in the `DomainName` parameter.

AWSSupport-TroubleshootOpenSearchRedYellowCluster

Description

`AWSSupport-TroubleshootOpenSearchRedYellowCluster` automation runbook is used to identify the cause for red or yellow cluster health status and guide you through changing the cluster back to green.

How does it work?

The runbook `AWSSupport-TroubleshootOpenSearchRedYellowCluster` helps you troubleshoot the cause of red or yellow cluster and provides the next steps to resolve this issue by analyzing the cluster configuration and resource utilization.

The runbook performs the following steps:

- Calls the DescribeDomain API against the target domain to get the cluster configuration.
- Checks if the OpenSearch Service domain is internet-based (public) or Amazon Virtual Private Cloud (VPC)-based.
- Creates a public or Amazon VPC-based Amazon Lambda function depending on the cluster configuration. Note: The Lambda function contains the troubleshooting code that run the OpenSearch Service APIs against the cluster to determine why the cluster is in red or yellow state.

- Deletes the Lambda function.
- Displays the checks performed and the next recommended steps to resolve the red or yellow cluster issue.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudformation:CreateStack`
- `cloudformation:DescribeStacks`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DeleteStack`
- `lambda:CreateFunction`
- `lambda:DeleteFunction`
- `lambda:InvokeFunction`
- `lambda:GetFunction`
- `es:DescribeDomain`
- `es:DescribeDomainConfig`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:DescribeNetworkInterfaces`

- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:DescribeInstances
- ec2:AttachNetworkInterface
- cloudwatch:GetMetricData
- iam:PassRole

The LambdaExecutionRole parameter requires the following actions to successfully use the runbook:

- es:ESHttpGet
- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DeleteNetworkInterface

Overview of LambdaExecutionRole policy:

The following is an example of a Lambda function's execution role (Amazon Identity and Access Management (IAM) role) that grants the function permission to access Amazon services and resources required by this runbook. For more information, see Lambda execution role.

Note

The ec2:DescribeNetworkInterfaces, ec2:CreateNetworkInterface, and ec2:DeleteNetworkInterface are only required if your OpenSearch Service cluster is Amazon VPC-based to allow the Lambda function to create and manage the Amazon VPC network interfaces. For more information, see Connecting outbound networking to resources in a Amazon VPC and Lambda execution role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
                    "Action": "es:ESHttpGet",
                    "Resource": [
                        "arn:<partition>:es:<region>:<account-id>:domain/<domain-
name>/",
                        "arn:<partition>:es:<region>:<account-id>:domain/<domain-name>/
_cluster/health",
                        "arn:<partition>:es:<region>:<account-id>:domain/<domain-name>/
_cat/indices",
                        "arn:<partition>:es:<region>:<account-id>:domain/<domain-name>/
_cat/allocation",
                        "arn:<partition>:es:<region>:<account-id>:domain/<domain-name>/
_cluster/allocation/explain"
                    ]
                },
                {
                    "Condition": {
                        "ArnLikeIfExists": {
                            "ec2:Vpc": "arn:<partition>:ec2:<region>:<account-id>:vpc/
<vpc_id>"
                        }
                    },
                    "Action": [
                        "ec2:DeleteNetworkInterface",
                        "ec2:CreateNetworkInterface",
                        "ec2:DescribeNetworkInterfaces",
                        "ec2:UnassignPrivateIpAddresses",
                        "ec2:AssignPrivateIpAddresses"
                    ],
                    "Resource": "*",
                    "Effect": "Allow"
                }

            ]
        }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to the AWSSupport-TroubleshootOpenSearchRedYellowCluster in the Amazon Systems Manager console.
2. Select Execute automation.

3. For the input parameters enter the following:

   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

   - **LambdaExecutionRole (Required):**

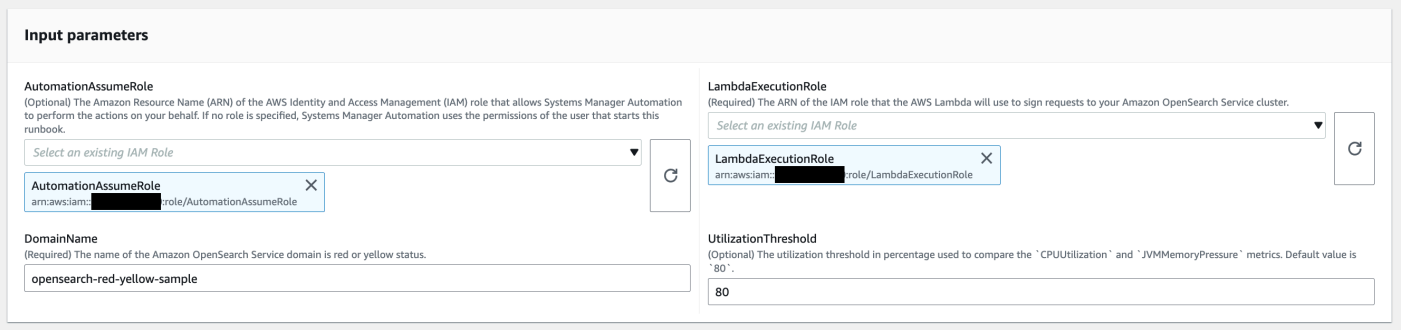
     The ARN of the IAM role that Lambda will use to sign requests to your Amazon OpenSearch Service cluster.

   - **DomainName (Required):**

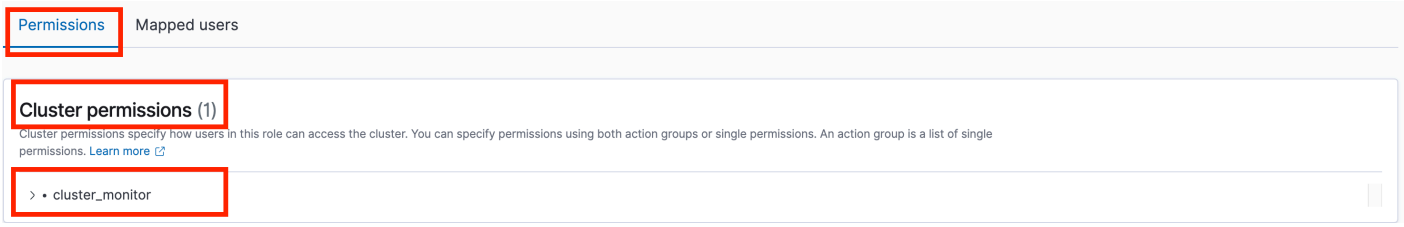
     The name of the OpenSearch Service domain with red or yellow cluster health status.

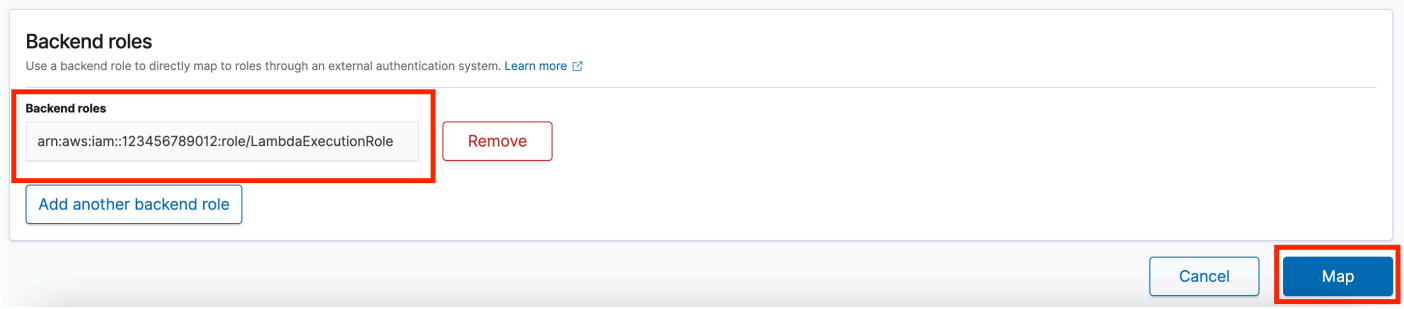
   - **UtilizationThreshold (Optional):**

     The utilization threshold percentage used to compare the CPUUtilization and JVMMemoryPressure metrics. Default value is 80.



4. If you have enabled fine-grained access control on an OpenSearch Service cluster, make sure that the `LambdaExecutionRole` role arn is mapped to a role with at least `cluster_monitor` permission.





5. Select Execute.
6. The automation initiates.
7. The automation runbook performs the following steps:
   - **GetClusterConfiguration:**

     Fetches the OpenSearch Service cluster configuration.
   - **CreateAWSLambdaFunctionStack:**

     Creates a temporary Lambda function in your account using Amazon CloudFormation. The Lambda function is used to run the OpenSearch Service APIs.
   - **WaitForAWSLambdaFunctionStack:**

     Waits for the CloudFormation stack to complete.
   - **GetClusterMetricsFromCloudWatch:**

     Gets the Amazon CloudWatch ClusterStatus, CPUUtilization, and JVMMemoryPressure OpenSearch Service cluster related metrics and its creation date.
   - **RunOpenSearchAPIs:**

     Uses the Lambda function to call the OpenSearch Service APIs and analyze the cluster metrics data to diagnose the cause for the red or yellow cluster status.
   - **DeleteAWSLambdaFunctionStack:**

     Deletes the Lambda function created by this automation in your account.
8. After completed, review the Outputs section for the detailed results of the execution.
   - **RootCause:**

     Provides an overview of the identified cause for cluster health to be in red or yellow state.
   - **IssueDescription:**

Provides details for why the cluster is in red or yellow state and possible steps to return the cluster to green state.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- Refer toTroubleshooting Amazon OpenSearch Service for more information

AWSSupport-TroubleshootOpenSearchHighCPU

Description

The AWSSupport-TroubleshootOpenSearchHighCPU runbook provides an automated solution to collect diagnostic data from an Amazon OpenSearch Service domain to troubleshoot high CPU issues.

How does it work?

The AWSSupport-TroubleshootOpenSearchHighCPU runbook helps to troubleshoot high CPU utilization in the Amazon OpenSearch Service domain.

The runbook performs the following steps:

- Runs the DescribeDomain API against the provided Amazon OpenSearch Service domain to get the cluster metadata.
- Checks whether the Amazon OpenSearch Service domain is public or Amazon VPC-based and with the help of Amazon CloudFormation, creates a public or Amazon VPC-based Amazon Lambda function.
- The Lambda function fetches diagnostic data from the Amazon OpenSearch Service domains.

- Uses an Amazon Step Functions state machine to orchestrate multiple Lambda function executions to gather more comprehensive data.
- Stores the collected data in an Amazon CloudWatch log group for 24 hours by default.
- Deletes the created resources, except the CloudWatch log group.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudformation:CreateStack`
- `cloudformation:CreateStack`
- `cloudformation:DescribeStacks`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DeleteStack`
- `lambda:CreateFunction`
- `lambda:DeleteFunction`
- `lambda:InvokeFunction`
- `lambda:GetFunction`
- `lambda:TagResource`
- `es:DescribeDomain`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`

- ec2:DescribeVpcs
- ec2:DescribeNetworkInterfaces
- ec2:CreateNetworkInterface
- ec2:DescribeInstances
- ec2:AttachNetworkInterface
- ec2:DeleteNetworkInterface
- logs:CreateLogGroup
- logs:PutRetentionPolicy
- logs:TagResource
- states:CreateStateMachine
- states:DeleteStateMachine
- states:StartExecution
- states:TagResource
- states:DescribeStateMachine
- states:DescribeExecution
- iam:PassRole
- iam:CreateRole
- iam:DeleteRole
- iam:GetRole
- iam:PutRolePolicy
- iam:DeleteRolePolicy
- ssm:DescribeAutomationExecutions
- ssm:GetAutomationExecution

The `LambdaExecutionRole` parameter requires the following actions to successfully use the runbook:

- es:ESHttpGet
- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DeleteNetworkInterface

- `logs:CreateLogStream`
- `logs:PutLogEvents`

The Lambda execution role grants the function permission to access Amazon services and resources required by this runbook. For more information, see Lambda execution role.

> **Note**
>
> The `ec2:DescribeNetworkInterfaces`, `ec2:CreateNetworkInterface`, and `ec2:DeleteNetworkInterface` are only required if your OpenSearch Service cluster is Amazon VPC-based to allow the Lambda function to create and manage the Amazon VPC network interfaces. For more information, see Connecting outbound networking to resources in a Amazon VPC and Lambda execution role.

Instructions

Follow these steps to configure the automation:

1. Navigate to the AWSSupport-TroubleshootOpenSearchHighCPU in the Amazon Systems Manager console.
2. Select Execute automation.
3. For the input parameters enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **DomainName (Required):**

     The name of the Amazon OpenSearch Service domain that you want to troubleshoot for high CPU issues.
   - **LambdaExecutionRoleForOpenSearch (Required):**

     The ARN of the IAM role to attach to the Lambda function. The Lambda function uses the credentials from this role to sign requests to the Amazon OpenSearch Service domain. If

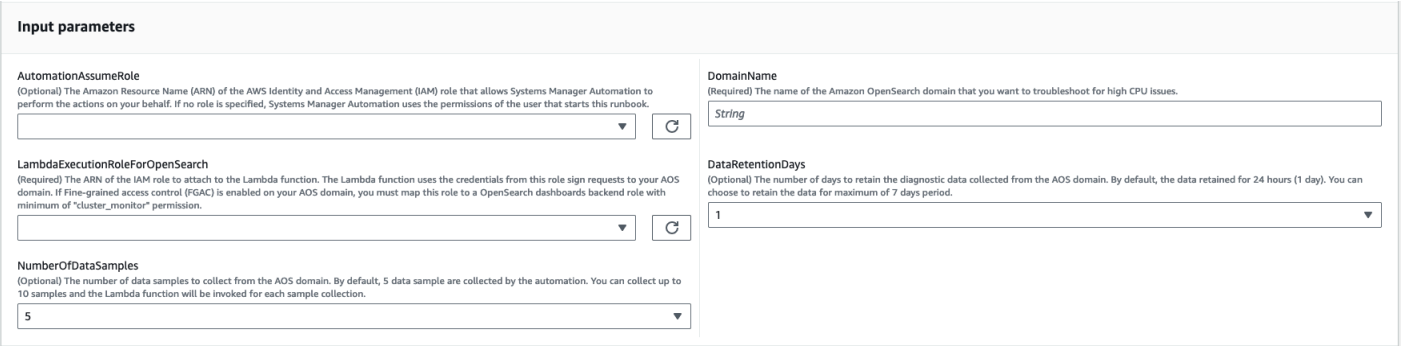
fine-grained access control is enabled on the Amazon OpenSearch Service domain, you must map this role to an OpenSearch Service Dashboards backend role with a minimum of "cluster_monitor" permission.

- **DataRetentionDays (Optional):**

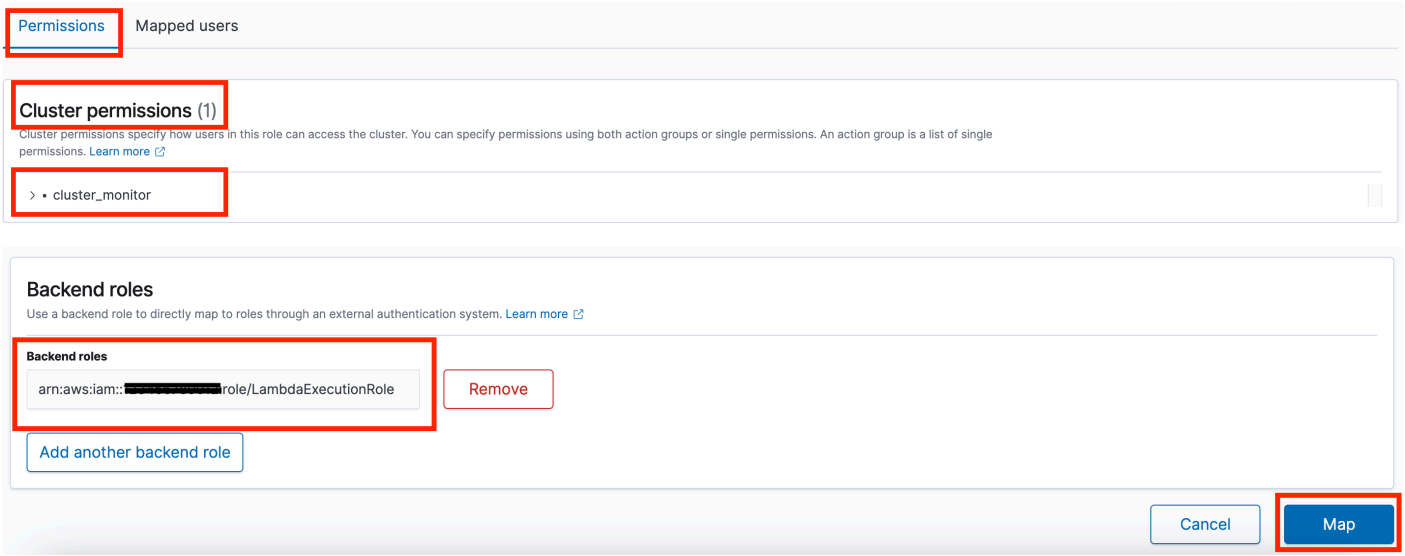
  The number of days to retain the diagnostic data collected from the Amazon OpenSearch Service domain. By default, the data is retained for 24 hours (one day). You can choose to retain the data for a maximum of up to 30 days.

- **NumberOfDataSamples (Optional):**

  The number of data samples to collect from the Amazon OpenSearch Service domain. By default, 5 data sample are collected. You can collect up to 10 samples and the Lambda function will be invoked for each sample collection.



4. If you have enabled fine-grained access control on an OpenSearch Service cluster, make sure that the `LambdaExecutionRole` role arn is mapped to a role with at least `cluster_monitor` permission.



5. Select Execute.

6. The automation initiates.
7. The automation runbook performs the following steps:
   - **checkConcurrency:**

     Ensures that there is only one execution of this runbook targeting the specified Amazon OpenSearch Service domain. If the runbook finds another execution targeting the same domain name, it returns an error and ends.
   - **getDomainConfig:**

     Gets the configuration details for the target OpenSearch Service domain.
   - **provisionResources:**

     Provisions the resources for data collection using Amazon CloudFormation.
   - **waitForStackCreation:**

     Waits for the Amazon CloudFormation stack to complete.
   - **describeStackResources:**

     Describes the Amazon CloudFormation stack and gets the ARN of the state machine.
   - **runStateMachine:**

     Invokes the data collector Lambda function one or more times by running a Step Functions state machine.
   - **describeErrorsFromStackEvents:**

     Describes errors from the Amazon CloudFormation stack for errors.
   - **unstageOpenSearchHighCPUAutomation:**

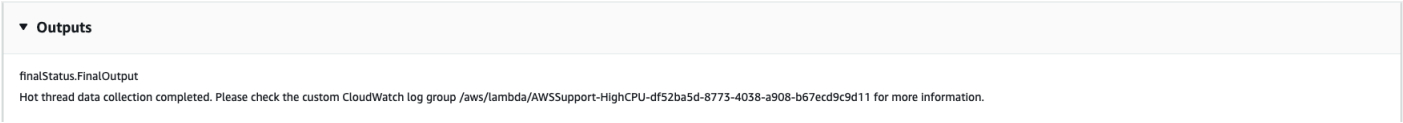
     Deletes the `AWSSupport-TroubleshootOpenSearchHighCPU` Amazon CloudFormation stack.
   - **describeErrorsFromStackDeletion:**

     Describes errors encountered while deleting the Amazon CloudFormation stack.
   - **finalStatus:**

     Returns the final output of the `AWSSupport-TroubleshootOpenSearchHighCPU` runbook.
8. After completed, review the Outputs section for the detailed results of the execution.

- **finalStatus.FinalOutput:**

  Provides the CloudWatch log group where the diagnostic data is stored.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- Refer toTroubleshooting Amazon OpenSearch Service for more information

EventBridge

Amazon Systems Manager Automation provides predefined runbooks for Amazon EventBridge. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-AddOpsItemDedupStringToEventBridgeRule
- AWS-DisableEventBridgeRule

AWS-AddOpsItemDedupStringToEventBridgeRule

Description

The `AWS-AddOpsItemDedupStringToEventBridgeRule` runbook adds a deduplication string for all Amazon Systems Manager OpsItems associated with an Amazon EventBridge rule. The

runbook doesn't add a deduplication string to the rule if one has already been applied. To learn more deduplication strings and OpsItems, see Reducing duplicate OpsItems in the *Amazon Systems Manager User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- DedupString

  Type: String

  Description: (Required) The deduplication string you want to add to the rule.

- RuleName

  Type: String

  Description: (Required) The name of the rule you want to add the deduplication string to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `events:ListTargetsByRule`
- `events:PutTargets`

Document Steps

- `aws:executeScript` - Adds a deduplication string to the EventBridge rule you specify in the `RuleName` parameter.

AWS-DisableEventBridgeRule

Description

The `AWS-DisableEventBridgeRule` runbook disables the Amazon EventBridge rule you specify.To learn more about EventBridge rules, see Amazon EventBridge rules in the *Amazon EventBridge User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- EventBusName

  Type: String

  Default: default

  Description: (Optional) The event bus associated with the rule you want to disable.

- RuleName

  Type: String

  Description: (Required) The name of the rule you want to disable.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `events:DisableRule`

Document Steps

- `aws:executeAwsApi` - Disables the EventBridge rule you specify in the `RuleName` parameter.

Amazon Glue

Amazon Systems Manager Automation provides predefined runbooks for Amazon Glue. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-TroubleshootGlueConnection

AWSSupport-TroubleshootGlueConnection

Description

The **AWSSupport-TroubleshootGlueConnections** runbook helps troubleshoot Amazon Glue connection issues. The target of the tested connection must be reached through a JDBC connection and can be either an Amazon Relational Database Service (Amazon RDS) cluster/instance, a Amazon Redshift cluster, or any other target accessible through JDBC. In the first two cases, the Reachability Analyzer tool is used to determine if the connectivity between the source (Amazon Glue) and the target (Amazon RDS or Amazon Redshift) is granted.

If the target of the connection is not Amazon RDS nor Amazon Redshift, connectivity is still tested by creating an Amazon Lambda function in the same subnet as the Amazon Glue connection (a network point of presence) and checking if the target name is resolvable and if it's reachable in the target port.

Important

In order to run the Reachability Analyzer checks, Elastic Network Interfaces will be created in each of the connection's datasource subnets. Please make sure you have enough free IPs on those subnets and that the consumption of one IP will not impact your workload before running this automation.

Important

All the resources created by this automation are tagged so that they can be easily found. The tags used are:

- `AWSSupport-TroubleshootGlueConnection`: true
- `AutomationExecutionId`: *`Amazon EC2 Systems Manager Execution Id`*

How does it work?

The runbook performs the following steps:

- Describes the Amazon Glue connection to get the source information (subnet and security groups) for the connectivity checks.
- Fetches the target information (subnet and security groups) from the datasource referenced in the JDBC URL or from the `DatasourceSecurityGroups` and `DatasourceSubnets` parameters if they are present.
- If the datasource present in the JDBC URL is a Amazon RDS instance or cluster or a Amazon Redshift cluster, this automation creates ENIs using both the source and target information gathered in the previous steps and uses Reachability Analyzer to perform a connectivity check between them.
- A Lambda function (network point of presence, in the context of this automation) is used to perform L4 connectivity and name resolution checks.
- The same Lambda function is used to perform the checks against the Amazon S3 endpoint.
- Policy Simulator is used to determine if the IAM role used in the connection has the needed permissions.
- The automation checks if the security group used by the connection has the expected configuration.
- A report is generated containing the possible causes for the failure in the test connection operation and/or also the succeeded tests that were performed.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudformation:CreateStack`

- cloudformation:DeleteStack
- ec2:CreateNetworkInsightsPath
- ec2:CreateNetworkInterface
- ec2:CreateTags
- ec2:DeleteNetworkInsightsAnalysis
- ec2:DeleteNetworkInsightsPath
- ec2:DeleteNetworkInterface
- ec2:StartNetworkInsightsAnalysis
- iam:AttachRolePolicy
- iam:CreateRole
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:DetachRolePolicy
- iam:PutRolePolicy
- iam:TagRole
- lambda:CreateFunction
- lambda:DeleteFunction
- lambda:TagResource
- logs:CreateLogGroup
- logs:DeleteLogGroup
- logs:PutRetentionPolicy
- logs:TagResource
- glue:GetConnection
- glue:GetDataCatalogEncryptionSettings
- cloudformation:DescribeStacks
- cloudformation:DescribeStackEvents
- ec2:DescribeDhcpOptions
- ec2:DescribeNetworkInsightsPaths
- ec2:DescribeNetworkInsightsAnalyses
- ec2:DescribeSecurityGroupRules

- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- ec2:DescribeVpcAttribute
- iam:GetRole
- iam:ListAttachedRolePolicies
- iam:SimulatePrincipalPolicy
- kms:DescribeKey
- lambda:InvokeFunction
- lambda:GetFunction
- s3:GetEncryptionConfiguration
- iam:PassRole

Important

In addition to the above mentioned actions, the AutomationAssumeRole should have the AmazonVPCReachabilityAnalyzerFullAccessPolicy as an attached managed policy so that the Reachability Analyzer tests are performed successfully.

Here is an example of a policy that could be granted for the AutomationAssumeRole:

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "TaggedAWSResourcesPermissions",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/AWSSupport-TroubleshootGlueConnection": "true"
                }
            },
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreateRole",
```

```
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:TagRole",
                "lambda:CreateFunction",
                "lambda:DeleteFunction",
                "lambda:TagResource",
                "logs:DeleteLogGroup",
                "logs:CreateLogGroup",
                "logs:PutRetentionPolicy",
                "logs:TagResource",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TaggedEC2ResourcesPermissions",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSSupport-TroubleshootGlueConnection": "true"
                }
            },
            "Action": [
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PutRolePolicy",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "iam:ResourceTag/AWSSupport-TroubleshootGlueConnection": "true"
                }
            },
            "Action": [
                "iam:PutRolePolicy",
                "iam:DeleteRolePolicy"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "InvokeFunction",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:*:lambda:*:*:function:point-of-presence-*"
        },
        {
            "Sid": "UnTaggedActions",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInsightsPath",
                "ec2:DeleteNetworkInsightsAnalysis",
                "ec2:DeleteNetworkInsightsPath",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags",
                "ec2:StartNetworkInsightsAnalysis",
                "glue:GetConnection",
                "glue:GetDataCatalogEncryptionSettings",
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackEvents",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeNetworkInsightsPaths",
                "ec2:DescribeNetworkInsightsAnalyses",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcAttribute",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "iam:SimulatePrincipalPolicy",
                "kms:DescribeKey",
                "lambda:GetFunction",
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:*:iam::*:role/point-of-presence-*",
```

```
            "Condition": {
                "StringLikeIfExists": {
                    "iam:PassedToService": "lambda.amazonaws.com"
                }
            }
        }
    ]
}
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-TroubleshootGlueConnection in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:

- **AutomationAssumeRole (Optional):**

  The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

- **TestConnectionRole (Required)**

  The Amazon Resource Name (ARN) of the IAM role that is used during the connection test.

- **ConnectionName (Required)**

  Amazon Glue failed test connection name you want to troubleshoot.

- **PersistReachabilityAnalyzerResults (Optional)**

  The flag informing if the results of the Reachability Analyzer execution should be kept or not. Default: `false`.

- **PointOfPresenceLogRetentionPeriod (Optional)**

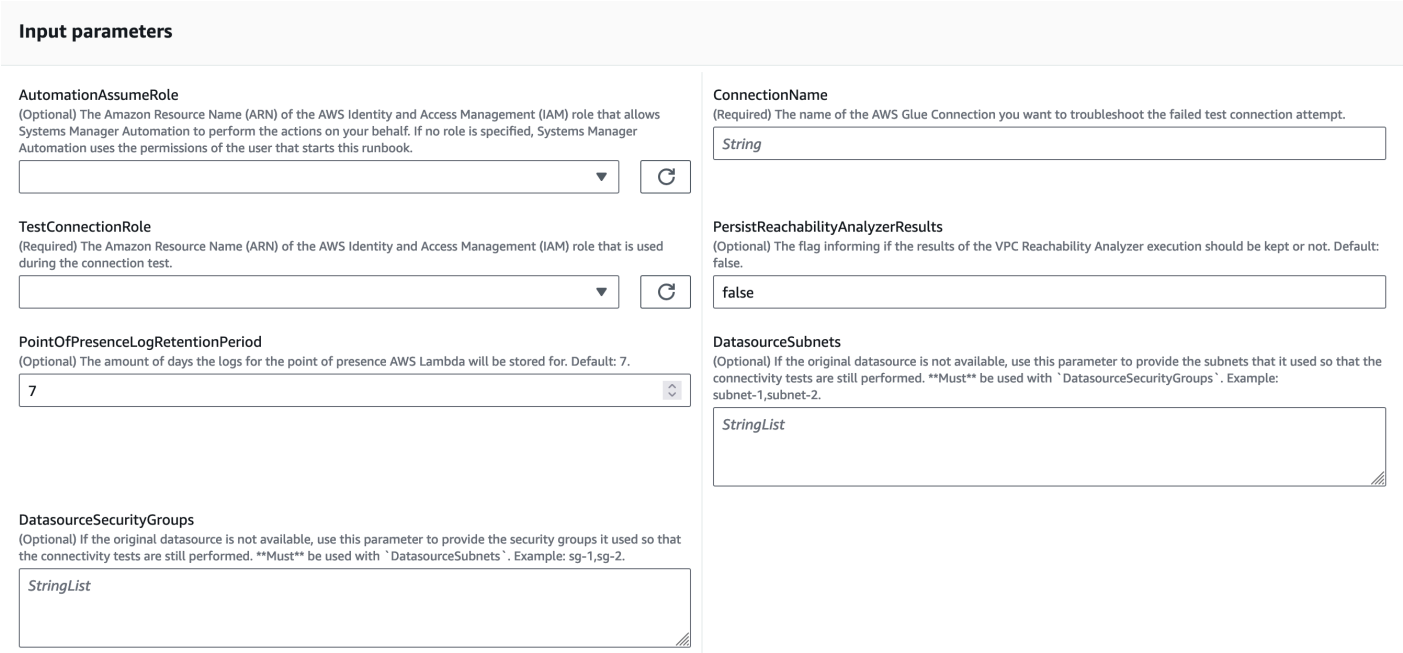
  The amount of days the logs for the point of presence Lambda will be stored for. Default: 7.

- **DatasourceSubnets (Optional)**

If the original datasource is not available, use this parameter to provide the subnets that it used so that the connectivity tests are still performed. **Must** be used with `DatasourceSecurityGroups`. Example: `subnet-1,subnet-2`.

- **DatasourceSecurityGroups (Optional)**

  If the original datasource is not available, use this parameter to provide the security groups it used so that the connectivity tests are still performed. **Must** be used with `DatasourceSubnets`. Example: `sg-1,sg-2`.



4. Select Execute.
5. The automation initiates.
6. The automation runbook performs the following steps:

   - **ParseInputs:**

     This step validates the combination of inputs. If both `DatasourceSecurityGroups` and `DatasourceSubnets` are provided, they are valid and returned as is. If none are provided, two empty lists are returned. If just one of them is provided, the step raises a `ValueException`.

   - **GetConnectionDetails:**

     This steps returns the details of the provided Amazon Glue connection.

   - **ParseSecurityGroupList:**

This step is used to concatenate the `SecurityGroupIdList` in a `String` for future utilization in this automation.

- **GetConnectionData:**

  Determines based on the JDBC URL, what type of connection between: `RedShift`, `RdsInstance`, `RdsCluster` and `Other`. In addition, returns the domain and port used in the JDBC connection, the connection's Amazon VPC and its domain name servers.

- **GetNetworkDetails:**

  Gets the subnet and security group information from the Amazon RDS or Amazon Redshift target.

- **CreateENITemplate:**

  Generates the Amazon CloudFormation template used to create the network interfaces that are used to test connectivity. This is required to run the Reachability Analyzer tool.

- **CreateENIStack:**

  Creates the Amazon CloudFormation stack from the template created in the previous step.

- **GetStackDetails:**

  Describes the Amazon CloudFormation stack created in the previous stack and retrieve the `SourceNetworkInterface` an `TargetNetworkInterfaces` information.

- **RunSourceToTargetCheck:**

  Runs checks between the source and target ENIs created in the previous step using the Reachability Analyzer tool.

- **DeleteENIStack:**

  Deletes the Amazon CloudFormation stack that creates Network Interfaces

- **CreateNetworkPointOfPresence:**

  Amazon CloudFormation creates the Lambda function used as network point of presence.

- **GetFunctionName:**

  Performs a Amazon CloudFormation describe stack API call to retrieve the name of the Lambda function created in the previous step.

- **RunEndpointChecks:**

Uses the network point of presence to determine if the endpoint present in the JDBC connection is resolvable and reachable in the declared port.

- **CheckS3Connectivity:**

  Checks the network connectivity from the Amazon Glue connection to the Amazon S3 service.

- **DeletePointOfPresence:**

  Deletes the Amazon CloudFormation stack that creates the network point of presence Lambda.

- **TestIAMRolePermissions:**

  Checks if the IAM role used for the test has the needed permissions to execute it.

- **CheckConnectionSecurityGroupReferencingRule:**

  Checks if the security group used in the Amazon Glue connection is allowing all ingress traffic from itself. It will return a list of the security groups without this rule, if any.

- **GenerateReport:**

  Generates a report containing a list o findings (possible reasons for the failure in the connection test) and next steps (attempts to resolve the connection test failure).

7. After completed, review the Outputs section for the detailed results of the execution:

- **Automation Results**

  In this section, you will find scenarios describing possible causes for the test connection operation to fail (findings) and how they can be fixed (next steps). If the automation cannot find the cause of the test failure, this will be informed in this section as well.

- **Successful Tests**

  In this section, you will find scenarios informing what has been successfully tested by this automation. Succeeded tests are useful in case the automation is not able to identify the cause of the test connection failure as they reduce the scope of the investigation by informing what is not contributing for the issue.

- **Automation Errors**

  In this section, you will find scenarios describing issues that happened during the automation, that may have limited the number of tests the automation could perform. The description of the scenario will inform which step has failed.

```
TroubleshootGlueConnection
_____________________________________________________________________________________________________________________
Automation Results
Bellow are possible causes for the issue being troubleshoot and their fixes.

------------------------------------------------------------

# Scenario: AWS Glue connection's VPC using Amazon Provided DNS
🔍 Findings:
  - The JDBC URL's endpoint dumy-instance.c3c3tj8ogvmj.us-east-1.rds.amazonaws.com could not be resolved from the AWS Glue
connection's subnet subnet-
  - Glue connection's VPC (vpc-[redacted]) is using AWS DNS Resolver but no issue could be found.
→ Next steps:
  - Please check the following links for further DNS troubleshooting:
- https://repost.aws/knowledge-center/vpc-find-cause-of-failed-dns-queries.
- https://repost.aws/knowledge-center/vpc-peering-troubleshoot-dns-resolution
- https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html#vpc-dns-support


# Scenario: Checking AWS Glue connection IAM role trust relationship
🔍 Findings:
  - The IAM role arn:aws:iam::[redacted]:role/SAW-Core-Test-AWSSupport-Trou-IncorrectGlueTestRole-lLu7FYAp9bfi used for
the AWS Glue connection test has no trust relationship with the AWS Glue service.
→ Next steps:
  - Please add the AWS Glue service to the arn:aws:iam::[redacted]:role/SAW-Core-Test-AWSSupport-Trou-
IncorrectGlueTestRole-lLu7FYAp9bfi trust relationship. More details on the following link:
- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/edit_trust.html


# Scenario: Glue test connection role IAM check
🔍 Findings:
  - The IAM role (arn:aws:iam::[redacted]:role/SAW-Core-Test-AWSSupport-Trou-IncorrectGlueTestRole-lLu7FYAp9bfi) used to
test the connection is not allowed to perform 'glue:TestConnection'.
→ Next steps:
  - Please add the 'glue:TestConnection' action to the role used for the AWS Glue connection test:
arn:aws:iam::[redacted]:role/SAW-Core-Test-AWSSupport-Trou-IncorrectGlueTestRole-lLu7FYAp9bfi.


_____________________________________________________________________________________________________________________
✓ Successful Tests
Below are the successful tests performed by the automation, for further troublehsooting.

------------------------------------------------------------

# Scenario: Datasource security group
  ✓ The target's security group(s) allow(s) inbound connections from the Glue connection.


# Scenario: Checking connection security group self referencing rule
  ✓ The AWS Glue connection has a self referencing inbound rule allowing all traffic.
```

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon FSx

Amazon Systems Manager Automation provides predefined runbooks for Amazon FSx. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-ValidateFSxWindowsADConfig

AWSSupport-ValidateFSxWindowsADConfig

Description

The AWSSupport-ValidateFSxWindowsADConfig runbook is used to validate the self-managed Active Directory (AD) configuration of an Amazon FSx for Windows File Server

How does it work?

The runbook AWSSupport-ValidateFSxWindowsADConfig executes the Amazon FSx validation script on the temporary Amazon Elastic Compute Cloud (Amazon EC2) Windows instance launched by the runbook on the Amazon FSx subnet. The script performs multiple checks to validate the network connectivity to self-managed AD/DNS servers and permissions of the Amazon FSx service account. The runbook can validate a failed or misconfigured Amazon FSx for Windows File Server or create a new Amazon FSx for Windows File Server with self-managed AD.

By default, the runbook creates the Amazon EC2 Windows instance, security group for Amazon Systems Manager (SSM) access, Amazon Identity and Access Management (IAM) role and policy using Amazon CloudFormation on the Amazon FSx subnet. If you want to run the script on an existing Amazon EC2 instance, provide the ID in the parameter InstanceId. On successful execution, it deletes the CloudFormation resources. However, to retain the resources, set the RetainCloudFormationStack parameter to true.

The CloudFormation template creates an IAM role on your behalf with required permissions to attach to the Amazon EC2 instance to run the Amazon FSx validation script. To specify an existing IAM instance profile for the temporary instance, use the InstanceProfileName parameter. The associated IAM role must contain the following permissions:

- ec2:DescribeSubnets and ec2:DescribeVpcs permissions and the Amazon Managed Policy AmazonSSMManagedInstanceCore.
- Permissions to get the Amazon FSx service account username and password from Systems Manager by calling the GetSecretValue API.
- Permissions to put object in the Amazon Simple Storage Service (Amazon S3) bucket for the script output.

Prerequisites

The subnet where the temporary Amazon EC2 instance is created (or the existing instance provided in the InstanceId parameter) must allow access to the Amazon Systems Manager, Amazon

Secrets Manager, and Amazon S3 endpoints in order to run the `AmazonFSxADValidation` script using SSM Run Command.

Amazon Secrets Manager setup

The validation script connects to the Microsoft AD domain by retrieving the Amazon FSx service account username and password with a runtime call to Secrets Manager. Follow the steps in Create an Amazon Secrets Manager secret to create a new Secrets Manager secret. Make sure that the username and password are stored using a key/value pair in the format `{"username":"EXAMPLE-USER","password":"EXAMPLE-PASSWORD"}`. Refer to Authentication and access control for Amazon Secrets Manager for information about securing access to secrets.

For more information about the tool, refer to the `TROUBLESHOOTING.md` and `README.md` files in the AmazonFSxADValidation file.

Runbook execution

Execute the runbook with Amazon FSx ID or AD parameters. Following is the runbook workflow:

- Gets the parameters from the Amazon FSx ID or uses the input AD parameters.
- Creates the temporary validation Amazon EC2 Windows instance on the Amazon FSx subnet, security group for SSM access, IAM role and policy (conditional) using CloudFormation. If the `InstanceId` parameter is specified, it is used.
- Downloads and executes the validation script on the target Amazon EC2 instance in Amazon FSx primary subnet.
- Provides the AD validation result code in the automation output. Additionally, the complete script output is uploaded to the Amazon S3 bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudformation:CreateStack`
- `cloudformation:DeleteStack`
- `cloudformation:DescribeStacks`
- `cloudformation:DescribeStackResources`
- `cloudformation:DescribeStackEvents`
- `ec2:CreateTags`
- `ec2:RunInstances`
- `ec2:TerminateInstances`
- `ec2:CreateLaunchTemplate`
- `ec2:DeleteLaunchTemplate`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeImages`
- `ec2:DescribeInstances`
- `ec2:DescribeLaunchTemplates`
- `ec2:DescribeLaunchTemplateVersions`
- `ec2:CreateSecurityGroup`
- `ec2:DeleteSecurityGroup`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:AuthorizeSecurityGroupEgress`
- `iam:CreateRole`
- `iam:CreateInstanceProfile`
- `iam:GetInstanceProfile`

- iam:getRolePolicy
- iam:DeleteRole
- iam:DeleteInstanceProfile
- iam:AddRoleToInstanceProfile
- iam:RemoveRoleFromInstanceProfile
- iam:AttachRolePolicy
- iam:DetachRolePolicy
- iam:PutRolePolicy
- iam:DeleteRolePolicy
- iam:GetRole
- iam:PassRole
- ssm:SendCommand
- ssm:StartAutomationExecution
- ssm:DescribeInstanceInformation
- ssm:DescribeAutomationExecutions
- ssm:GetDocument
- ssm:GetAutomationExecution
- ssm:DescribeAutomationStepExecutions
- ssm:ListCommandInvocations
- ssm:GetParameters
- ssm:ListCommands
- ssm:GetCommandInvocation
- fsx:DescribeFileSystems
- ds:DescribeDirectories
- s3:GetEncryptionConfiguration
- s3:GetBucketPublicAccessBlock
- s3:GetAccountPublicAccessBlock
- s3:GetBucketPolicyStatus
- s3:GetBucketAcl
- s3:GetBucketLocation

Example IAM Policy for the Automation Assume Role

```
                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Sid": "AllowDescribe",
                            "Effect": "Allow",
                            "Action": [
                                "ec2:DescribeSubnets",
                                "ec2:DescribeSecurityGroups",
                                "ec2:DescribeImages",
                                "ec2:DescribeInstances",
                                "ec2:DescribeLaunchTemplates",
                                "ec2:DescribeLaunchTemplateVersions",
                                "ssm:DescribeInstanceInformation",
                                "ssm:DescribeAutomationExecutions",
                                "ssm:DescribeAutomationStepExecutions",
                                "fsx:DescribeFileSystems",
                                "ds:DescribeDirectories"
                            ],
                            "Resource": "*"
                        },
                        {
                            "Sid": "CloudFormation",
                            "Effect": "Allow",
                            "Action": [
                                "cloudformation:DescribeStacks",
                                "cloudformation:DescribeStackResources",
                                "cloudformation:DescribeStackEvents",
                                "cloudformation:CreateStack",
                                "cloudformation:DeleteStack"
                            ],
                            "Resource": "arn:*:cloudformation:*:*:stack/AWSSupport-
ValidateFSxWindowsADConfig-*"
                        },
                        {
                            "Sid": "AllowCreateLaunchTemplate",
                            "Effect": "Allow",
                            "Action": [
                                "ec2:CreateLaunchTemplate",
                                "ec2:CreateTags"
                            ],
```

```
                "Resource": [
                    "arn:aws:ec2:*:*:launch-template/*"
                ]
            },
            {
                "Sid": "AllowEC2RunInstances",
                "Effect": "Allow",
                "Action": [
                    "ec2:RunInstances",
                    "ec2:CreateTags"
                ],
                "Resource": [
                    "arn:aws:ec2:*::image/*",
                    "arn:aws:ec2:*::snapshot/*",
                    "arn:aws:ec2:*:*:subnet/*",
                    "arn:aws:ec2:*:*:network-interface/*",
                    "arn:aws:ec2:*:*:security-group/*",
                    "arn:aws:ec2:*:*:key-pair/*",
                    "arn:aws:ec2:*:*:launch-template/*"
                ]
            },
            {
                "Sid": "AllowEC2RunInstancesWithTags",
                "Effect": "Allow",
                "Action": [
                    "ec2:RunInstances",
                    "ec2:CreateTags"
                ],
                "Resource": [
                    "arn:aws:ec2:*:*:instance/*",
                    "arn:aws:ec2:*:*:volume/*"
                ]
            },
            {
                "Sid": "EC2SecurityGroup",
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateSecurityGroup",
                    "ec2:RevokeSecurityGroupEgress",
                    "ec2:AuthorizeSecurityGroupEgress",
                    "ec2:CreateTags"
                ],
                "Resource": [
                    "arn:*:ec2:*:*:security-group/*",
```

```
                "arn:*:ec2:*:*:vpc/*"
            ]
        },
        {
            "Sid": "EC2Remove",
            "Effect": "Allow",
            "Action": [
                "ec2:TerminateInstances",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSecurityGroup"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:launch-template/*",
                "arn:*:ec2:*:*:security-group/*"
            ]
        },
        {
            "Sid": "IAMInstanceProfile",
            "Effect": "Allow",
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile"
            ],
            "Resource": "arn:*:iam::*:instance-profile/*"
        },
        {
            "Sid": "IAM",
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
                "iam:getRolePolicy",
                "iam:PutRolePolicy",
                "iam:DeleteRolePolicy",
                "iam:GetRole",
                "iam:TagRole"
            ],
            "Resource": "arn:*:iam::*:role/*"
```

```
                        },
                        {
                            "Sid": "SSM",
                            "Effect": "Allow",
                            "Action": [
                                "ssm:StartAutomationExecution",
                                "ssm:GetDocument",
                                "ssm:GetAutomationExecution",
                                "ssm:ListCommandInvocations",
                                "ssm:GetParameters",
                                "ssm:ListCommands",
                                "ssm:GetCommandInvocation"
                            ],
                            "Resource": "*"
                        },
                        {
                            "Sid": "SSMSendCommand",
                            "Effect": "Allow",
                            "Action": [
                                "ssm:SendCommand"
                            ],
                            "Resource": "arn:aws:ssm:*:*:document/AWS-
RunPowerShellScript"
                        },
                        {
                            "Sid": "SSMSendCommandOnlyFsxInstance",
                            "Effect": "Allow",
                            "Action": [
                                "ssm:SendCommand"
                            ],
                            "Resource": [
                                "arn:aws:ec2:*:*:instance/*"
                            ],
                            "Condition": {
                                "StringLike": {
                                    "ssm:resourceTag/CreatedBy": [
                                        "AWSSupport-ValidateFSxWindowsADConfig"
                                    ]
                                }
                            }
                        },
                        {
                            "Effect": "Allow",
                            "Action": "iam:PassRole",
```

```
                    "Resource": "*",
                    "Condition": {
                        "StringEquals": {
                            "iam:PassedToService": [
                                "ec2.amazonaws.com"
                            ]
                        }
                    }
                },
                {
                    "Effect": "Allow",
                    "Action": [
                        "s3:GetEncryptionConfiguration",
                        "s3:GetBucketPublicAccessBlock",
                        "s3:GetAccountPublicAccessBlock",
                        "s3:GetBucketPolicyStatus",
                        "s3:GetBucketAcl",
                        "s3:GetBucketLocation"
                    ],
                    "Resource": "*"
                }
            ]
        }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to [AWSSupport-ValidateFSxWindowsADConfig] in Systems Manager under Documents.
2. Select **Execute automation.**
3. To validate self-managed AD with an existing failed or misconfigured Amazon FSx, enter the following parameters:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **FSxId (Conditional):**

The Amazon FSx for Windows File Server ID. This is required to validate existing failed or misconfigured Amazon FSx.

- **SecretArn (Required):**

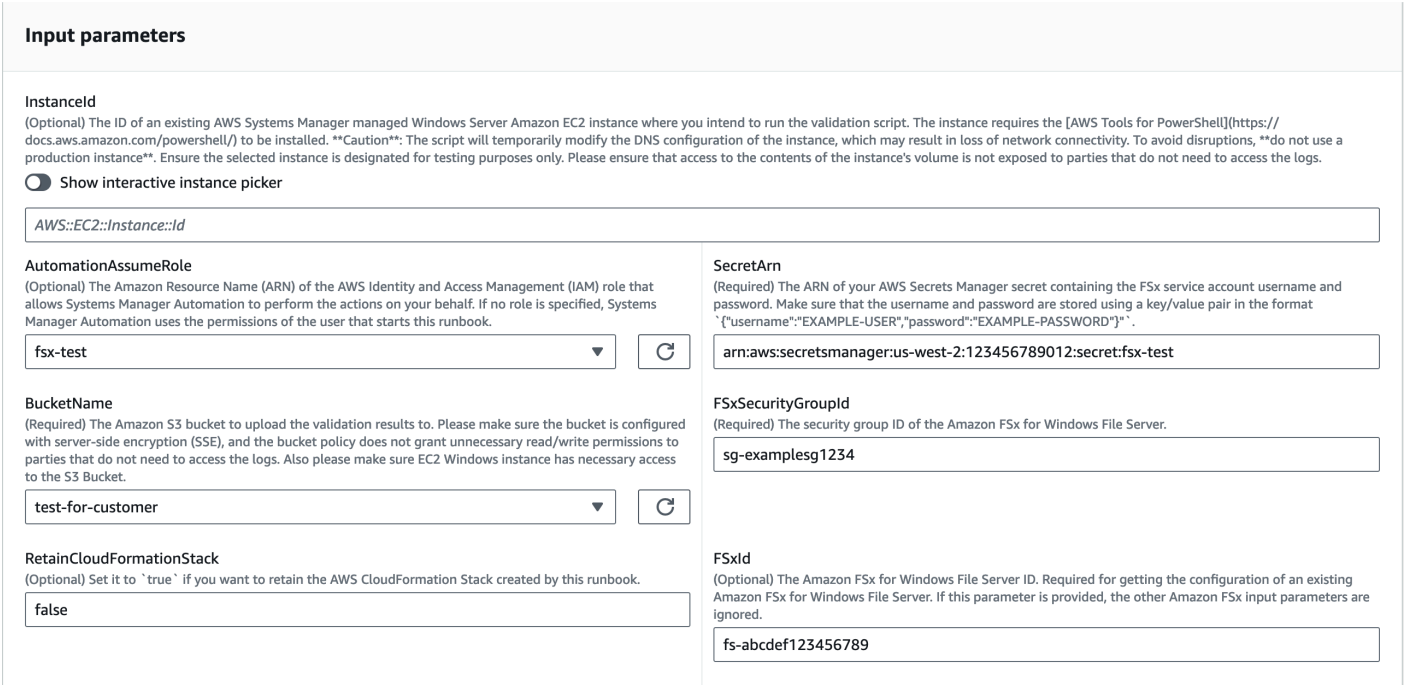
  The ARN of your Secrets Manager secret containing the Amazon FSx service account username and password. Make sure that the username and password are stored using a key/value pair in the format `{"username":"EXAMPLE-USER","password":"EXAMPLE-PASSWORD"}`. The CloudFormation stack creates the validation instance with permissions to perform `GetSecretValue` to this ARN.

- **FSxSecurityGroupId (Required):**

  The security group ID for the Amazon FSx for Windows File Server.

- **BucketName (Required):**

  The Amazon S3 bucket to upload the validation results to. Make sure that the bucket is configured with server-side encryption (SSE) and the bucket policy does not grant unnecessary read/write permissions to parties that do not need to access the logs. Also make sure that the Amazon EC2 Windows instance has necessary access to the Amazon S3 bucket.



4. To validate self-managed AD configuration for a new Amazon FSx creation, enter the following parameters:

   - **AutomationAssumeRole (Optional):**

The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

- **SecretArn (Required):**

  The ARN of your Secrets Manager secret containing the Amazon FSx service account username and password. Make sure that the username and password are stored using a key/value pair in the format `{"username":"EXAMPLE-USER","password":"EXAMPLE-PASSWORD"}`. The CloudFormation stack creates the validation instance with permissions to perform `GetSecretValue` to this ARN.

- **FSxSecurityGroupId (Required):**

  The security group ID for the Amazon FSx for Windows File Server.

- **BucketName (Required):**

  The Amazon S3 bucket to upload the validation results to. Make sure that the bucket is configured with server-side encryption (SSE) and the bucket policy does not grant unnecessary read/write permissions to parties that do not need to access the logs. Also make sure that the Amazon EC2 Windows instance has necessary access to the Amazon S3 bucket.

- **FSxPreferredSubnetId (Conditional):**

  The Amazon FSx for Windows File Server preferred subnet.

- **DomainName (Conditional):**

  The fully qualified domain name of your self-managed Microsoft AD domain.

- **DnsIpAddresses (Conditional):**

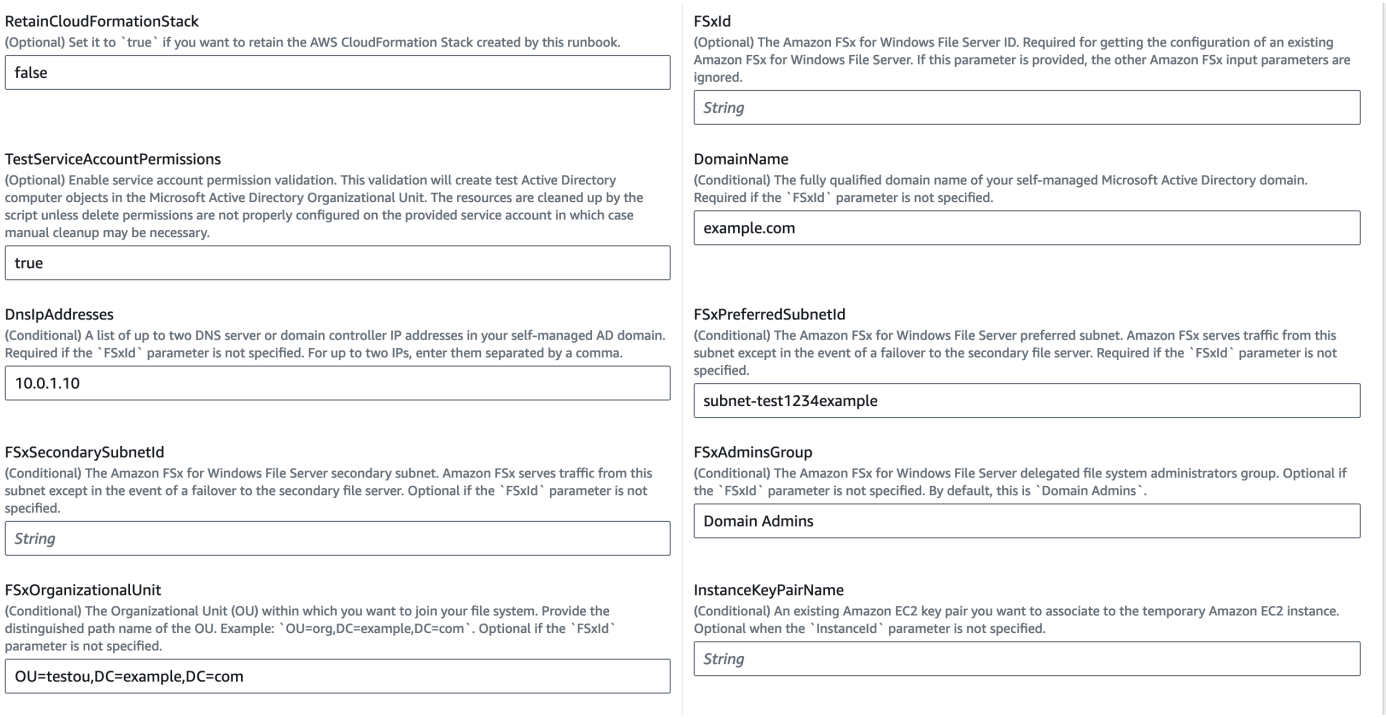
  A list of up to two DNS server or domain controller IP addresses in your self-managed AD domain. For up to two IPs, enter them separated by a comma.

- **FSxAdminsGroup (Conditional):**

  The Amazon FSx for Windows File Server delegated file system administrators group. By default, this is `Domain Admins`.

- **FSxOrganizationalUnit (Conditional):**

The Organizational Unit (OU) within which you want to join your file system. Provide the distinguished path name of the OU. Example: `OU=org,DC=example,DC=com`.



5. Select **Execute.**
6. The automation initiates.
7. The document performs the following steps:
   - **CheckBucketPublicStatus (aws:executeScript):**

     Checks if the target Amazon S3 bucket potentially grants read and/or write public access to its objects.
   - **BranchOnInputParameters (aws:branch):**

     Branches on the provided input parameters such as Amazon FSx ID or Amazon FSx parameters.
   - **AssertFileSystemTypeIsWindows (aws:assertAwsResourceProperty):**

     If Amazon FSx ID is provided, validates the file system type is Amazon FSx for Windows File Server.
   - **GetValidationInputs (aws:executeScript):**

Returns the self-managed Microsoft AD configuration required by the CloudFormation template to create the Amazon EC2 instance.

- **BranchOnInstanceId (aws:branch):**

  Branches on the provided input `InstanceId`. If `InstanceId` is provided, the validation script runs on the target Amazon EC2 instance from automation `step:RunValidationScript`.

- **CreateEC2InstanceStack (aws:createStack):**

  Creates the Amazon EC2 instance in the preferred subnet using Amazon CloudFormation where the `AmazonFSxADValidation` tool will be executed

- **DescribeStackResources (aws:executeAwsApi):**

  Describes the CloudFormation stack to get the temporary Amazon EC2 instance ID.

- **WaitForEC2InstanceToBeManaged (aws:waitForAwsResourceProperty):**

  Waits until the Amazon EC2 instance is managed by Systems Manager in order to run the validation script using SSM Run Command.

- **GetAmazonFSxADValidationAttachment (aws:executeAwsApi):**

  Gets the `AmazonFSxADValidation` tool URL from the runbook attachments.

- **RunValidationScript (aws:runCommand):**

  Runs the `AmazonFSxADValidation` tool on the temporary Amazon EC2 instance and stores the result in the Amazon S3 bucket specified in the `BucketName` parameter.

- **DescribeErrorsFromStackEvents (aws:executeScript):**

  Describes the CloudFormation stack events if the runbooks fails to create the stack.

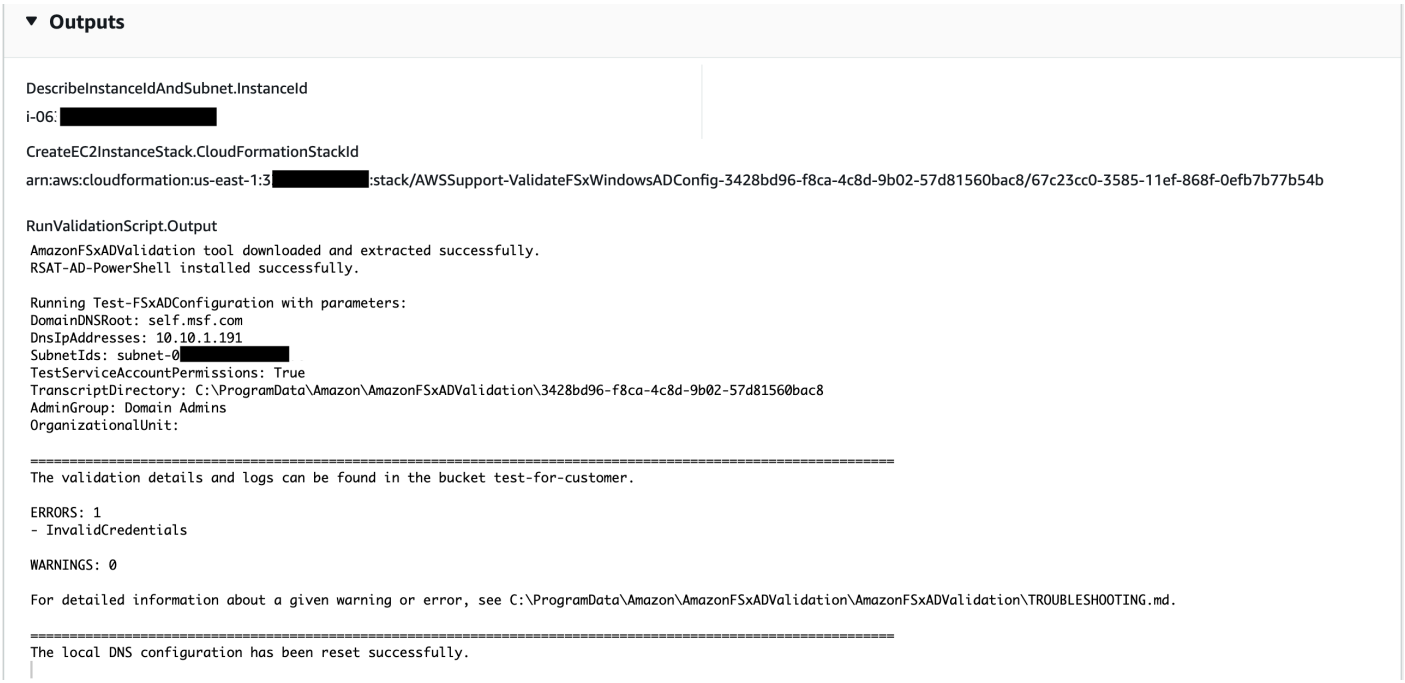
- **BranchOnRetainCloudFormationStack (aws:branch):**

  Branches on the `RetainCloudFormationStack` and `InstanceId` parameters to determine if the CloudFormation stack should be deleted.

- **DeleteCloudFormationStack (aws:deleteStack):**

  Deletes the Amazon CloudFormation stack.

8. After completed, review the Outputs section for the results of the execution:



The runbook will upload the results of the validation script execution to the Amazon S3 bucket.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- What is Amazon FSx for Windows File Server?
- Validating self-managed AD configuration for Amazon FSx for Windows File Server

GuardDuty

Amazon Systems Manager Automation provides predefined runbooks for Amazon GuardDuty. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWSConfigRemediation-CreateGuardDutyDetector

Description

The AWSConfigRemediation-CreateGuardDutyDetector runbook creates an Amazon GuardDuty (GuardDuty) detector in the Amazon Web Services Region where you run the automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution

- `ssm:GetAutomationExecution`
- `guardduty:CreateDetector`
- `guardduty:GetDetector`

Document Steps

- `aws:executeAwsApi` - Creates a GuardDuty detector.
- `aws:assertAwsResourceProperty` - Verifies the `Status` of the detector is `ENABLED` .

IAM

Amazon Systems Manager Automation provides predefined runbooks for Amazon Identity and Access Management. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-TroubleshootIAMAccessDeniedEvents
- AWS-AttachIAMToInstance
- AWS-DeleteIAMInlinePolicy
- AWSConfigRemediation-DeleteIAMRole
- AWSConfigRemediation-DeleteIAMUser
- AWSConfigRemediation-DeleteUnusedIAMGroup
- AWSConfigRemediation-DeleteUnusedIAMPolicy
- AWSConfigRemediation-DetachIAMPolicy
- AWSConfigRemediation-EnableAccountAccessAnalyzer
- AWSSupport-GrantPermissionsToIAMUser
- AWSConfigRemediation-RemoveUserPolicies
- AWSConfigRemediation-ReplaceIAMInlinePolicy
- AWSConfigRemediation-RevokeUnusedIAMUserCredentials
- AWSConfigRemediation-SetIAMPasswordPolicy
- AWSSupport-ContainIAMPrincipal

AWSSupport-TroubleshootIAMAccessDeniedEvents

Description

The AWSSupport-TroubleshootIAMAccessDeniedEvents automation runbook helps troubleshooting Amazon Identity and Access Management (IAM) access denied issues. The runbook queries CloudTrail for recent access denied events related to the specified IAM entity and Amazon service event source. It analyzes events within a configurable time window of up to 24 hours, processing up to 10 events per execution. Each identified access denied event is examined to help understand the context of the denial and the attempted actions. The automation analyzes both identity-based and resource-based IAM policies. For identity-based policies, it examines inline and managed policies attached to the IAM entity. For resource-based policies, it evaluates policies across multiple Amazon services including Amazon Simple Storage Service(Amazon S3), Amazon Key Management Service (Amazon KMS), Amazon Lambda, Amazon Simple Notification Service (Amazon SNS), Amazon Elastic Container Registry (Amazon ECR), Amazon API Gateway, CodeArtifact, Amazon Elastic File System (Amazon EFS), Amazon Simple Queue Service (Amazon SQS), Amazon Cloud9, Amazon OpenSearch Service, Amazon Signer, Amazon Serverless Application Repository, and Amazon Secrets Manager.

The runbook utilizes IAM policy simulation capabilities to evaluate these policies against the denied actions found in the CloudTrail events. The runbook leverages IAM's policy simulation capabilities through both SimulatePrincipalPolicy for IAM users and SimulateCustomPolicy for IAM roles to evaluate these policies against the denied actions found in the CloudTrail events. The automation outputs a report that helps identify the specific actions that were denied, differentiating between implicit and explicit denies, listing the policies responsible for access denials and provides explanations for each denial. The report also suggests potential resolutions, such as identifying missing allow statements or conflicting deny statements

How does it work?

The runbook performs the following steps:

- Describes and validates `RequesterARN` (role or user) to get information such as IAM entity type, and IAM Id.
- Fetches CloudTrail events associated with the `RequesterARN`, `EventSource`, and `ResourceARN` if provided.
- Analyzes the CloudTrail events to get the action that was performed when the Access Denied error was returned, then examines all the IAM policies such as inline and managed policies

attached to the IAM entity, as well as resource-based policies. It then simulates these policies against the actions found in the Access Denied errors from the CloudTrail events in question to determine the cause of the error.

- Outputs a report determining the type of Access Denied error, the policies responsible for the errors, and gives suggestions for potential solution to the error.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `apigateway:GetRestApis`
- `cloudtrail:LookupEvents`
- `cloud9:GetEnvironment`
- `codeartifact:GetRepositoryPermissionsPolicy`
- `ecr:GetRepositoryPolicy`
- `elasticfilesystem:GetFileSystemPolicy`
- `es:DescribeDomain`
- `iam:GetPolicy`
- `iam:GetPolicyVersion`
- `iam:GetRole`
- `iam:GetRolePolicy`
- `iam:GetUser`

- iam:GetUserPolicy
- iam:ListAttachedRolePolicies
- iam:ListAttachedUserPolicies
- iam:ListRolePolicies
- iam:ListUserPolicies
- iam:SimulatePrincipalPolicy
- iam:SimulateCustomPolicy
- kms:GetKeyPolicy
- lambda:GetPolicy
- secretsmanager:GetResourcePolicy
- serverlessrepo:GetApplication
- signer:GetSigningProfile
- sns:GetTopicAttributes
- ssm:StartAutomationExecution
- ssm:StopAutomationExecution
- sqs:GetQueueAttributes
- s3:GetBucketPolicy

Example Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:GetUser",
                "iam:GetRole",
                "iam:SimulatePrincipalPolicy",
                "iam:ListUserPolicies",
                "iam:ListRolePolicies",
                "iam:GetRolePolicy",
                "iam:ListAttachedRolePolicies",
                "iam:GetPolicy",
                "iam:GetUserPolicy",
```

```
                    "iam:GetPolicyVersion",
                    "iam:ListAttachedUserPolicies",
                    "ssm:StartAutomationExecution",
                    "ssm:StopAutomationExecution",
                    "cloudtrail:LookupEvents",
                    "iam:SimulateCustomPolicy"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Action": [
                    "s3:GetBucketPolicy",
                    "kms:GetKeyPolicy",
                    "lambda:GetPolicy",
                    "sns:GetTopicAttributes",
                    "ecr:GetRepositoryPolicy",
                    "apigateway:GET",
                    "codeartifact:GetRepositoryPermissionsPolicy",
                    "elasticfilesystem:GetFileSystemPolicy",
                    "sqs:GetQueueAttributes",
                    "cloud9:GetEnvironment",
                    "es:DescribeDomain",
                    "signer:GetSigningProfile",
                    "serverlessrepo:GetApplication",
                    "secretsmanager:GetResourcePolicy"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-TroubleshootIAMAccessDeniedEvents in Systems Manager under Documents.
2. Select **Execute automation.**
3. For the input parameters, enter the following:

- **AutomationAssumeRole (Optional):**
  - Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows SSM Automation to perform the actions on your behalf. The role needs to be added to your Amazon EKS cluster access entry or RBAC permission to allow Kubernetes API calls.
  - Type: `AWS::IAM::Role::Arn`
- **RequesterARN (Required):**
  - Description: (Required) The ARN of the IAM user or role for which you want to investigate the access permissions on a specific Amazon resource.
  - Type: String
  - Allow Pattern: `^arn:(aws|aws-cn|aws-us-gov|aws-iso(-[a-z])?):iam::[0-9]{12}:(role|user)\\/[\\w+\\/=,.@-]+$`
- **ResourceARN (Optional):**
  - Description: (Optional) The ARN of Amazon the resource for which the access denied is evaluated. The Amazon target resource should exist in the same region where the automation runbook is executed.
  - Type: String
  - Allow Pattern: `^$|^arn:(aws|aws-cn|aws-us-gov|aws-iso(-[a-z])?):([a-zA-Z0-9\\-]{1,63}):([a-z0-9\\-]{0,63})?:(\\d{12})?:([a-zA-Z0-9\\-_/:.]{1,1024})$`
- **EventSource (Required):**
  - Description: (Required) The Amazon API endpoint where the CloudTrail event originated. For example: `s3.amazonaws.com`.
  - Type: String
  - Allow Pattern: `^([a-zA-Z0-9.-]+)\\.amazonaws\\.com$`
- **EventName (Optional):**
  - Description: (Optional) The Amazon API action name associated with the CloudTrail event. For example: `s3:CreateBucket`.
  - Type: String
  - Allow Pattern: `^$|^[a-z0-9]+:[A-Za-z0-9]+$`
- **LookBackHours (Optional):**

- Description: (Optional) The number of hours to look back in the CloudTrail events when searching for `Access Denied` events. Valid range: 1 to 24 hours.
- Type: Integer
- Allow Pattern: `^([1-9]|1[0-9]|2[0-4])$`
- Default: 12

- **MaxEvents (Optional):**
  - Description: (Optional) The maximum number of CloudTrail `Access Denied` events returned when searching for events. Valid range: 1 to 5 events.
  - Type: Integer
  - Allow Pattern: `^([1-9]|1[0-9]|2[0-4])$`
  - Default: 3
- **UseContextEntries (Optional):**
  - Description: (Optional) If you specify `true`, the automation extracts details about the context of the API request from the CloudTrail event and include them for the IAM policy simulation.
  - Type: Boolean
  - Allow Pattern: `^([1-9]|1[0-9]|2[0-4])$`
  - Default: 3

4. Select **Execute**.
5. The automation initiates.
6. The document performs the following steps:
   - **ValidateRequesterArn**

     Validates and deconstructs the `RequesterArn` ARN, retrieving information about the target IAM user or role.
   - **GetCloudTrailEventsWithAccessDeniedError**

     Queries the CloudTrail events for recent `Access Denied` events related to the specified IAM entity and Amazon service `EventSource`.
   - **EvaluateIAMRequesterPolicies**

     Evaluates the IAM permissions of the requester IAM entity against the actions from CloudTrail events. This evaluation includes analyzing both identity-based and resource-based policies

associated with the requester. The automation utilizes IAM's policy simulation capabilities to assess these policies in the context of the denied actions identified in the CloudTrail events.

7. After completed, review the **Outputs** section for the detailed results of the execution:

- **PermissionEvaluationResults**

  Outputs a report that helps to identify the specific actions that were denied, differentiating between implicit and explicit denials. It also lists the policies responsible for access denials and provides explanations for each denial. The report also suggests potential resolutions, such as identifying missing allow statements or conflicting deny statements

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWS-AttachIAMToInstance

Description

Attach an Amazon Identity and Access Management (IAM) role to a managed instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ForceReplace

  Type: Boolean

  Description: (Optional) Flag to specify whether to replace the existing IAM profile or not.

  Default: true

- InstanceId

  Type: String

  Description: (Required) The ID of the instance on which you want to assign an IAM role.

- RoleName

  Type: String

  Description: (Required) The IAM role name to add to the managed instance.

Document Steps

1. `aws:executeAwsApi` - DescribeInstanceProfile - Find the IAM instance profile attached to the EC2 instance.
2. `aws:branch` - CheckInstanceProfileAssociations - Check the IAM instance profile attached to the EC2 instance.
   a. If an IAM instance profile is attached and `ForceReplace` is set to `true` :
      i. `aws:executeAwsApi` - DisassociateIamInstanceProfile - Disassociate the IAM instance profile from the EC2 instance.
   b. `aws:executeAwsApi` - ListInstanceProfilesForRole - List instance profiles for the IAM role provided.

c. `aws:branch` - CheckInstanceProfileCreated - Check if the IAM role provided has an associated instance profile.

   i. If the IAM role has an associated instance profile:

      A. `aws:executeAwsApi` - AttachIAMProfileToInstance - Attach the IAM instance profile role to the EC2 instance.

   i. If the IAM role does not have an associated instance profile:

      A. `aws:executeAwsApi` - CreateInstanceProfileForRole - Create an instance profile role for the specified IAM role.

      B. `aws:executeAwsApi` - AddRoleToInstanceProfile - Attach the instance profile role to the specified IAM role.

      C. `aws:executeAwsApi` - GetInstanceProfile - Get the instance profile data for the specified IAM role.

      D. `aws:executeAwsApi` - AttachIAMProfileToInstanceWithRetry - Attach the IAM instance profile role to the EC2 instance.

Outputs

AttachIAMProfileToInstanceWithRetry.AssociationId

GetInstanceProfile.InstanceProfileName

GetInstanceProfile.InstanceProfileArn

AttachIAMProfileToInstance.AssociationId

ListInstanceProfilesForRole.InstanceProfileName

ListInstanceProfilesForRole.InstanceProfileArn

AWS-DeleteIAMInlinePolicy

Description

The `AWS-DeleteIAMInlinePolicy` runbook deletes all Amazon Identity and Access Management (IAM) inline policies attached to the IAM identities you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- IamArns

  Type: String

  Description: (Required) A comma separated list of ARNs for the IAM identities you want to delete inline policies from. This list can include IAM users, groups, or roles.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `iam:DeleteGroupPolicy`
- `iam:DeleteRolePolicy`
- `iam:DeleteUserPolicy`
- `iam:ListGroupPolicies`
- `iam:ListRolePolicies`

- `iam:ListUserPolicies`

Document Steps

- `aws:executeScript` - Deletes the IAM inline policies attached to the targeted IAM identities.

AWSConfigRemediation-DeleteIAMRole

Description

The `AWSConfigRemediation-DeleteIAMRole` runbook deletes the Amazon Identity and Access Management (IAM) role you specify. This automation does not delete instance profiles associated with the IAM role, or service-linked roles.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- IAMRoleID

  Type: String

Description: (Required) The ID of the IAM role you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:DeleteRole`
- `iam:DeleteRolePolicy`
- `iam:GetRole`
- `iam:ListAttachedRolePolicies`
- `iam:ListInstanceProfilesForRole`
- `iam:ListRolePolicies`
- `iam:ListRoles`
- `iam:RemoveRoleFromInstanceProfile`

Document Steps

- `aws:executeScript` - Gathers the name of the IAM role you specify in the `IAMRoleID` parameter.
- `aws:executeScript` - Gathers policies and instance profiles associated with the IAM role.
- `aws:executeScript` - Deletes attached policies.
- `aws:executeScript` - Deletes the IAM role and verifies the role has been deleted.

AWSConfigRemediation-DeleteIAMUser

Description

The `AWSConfigRemediation-DeleteIAMUser` runbook deletes the Amazon Identity and Access Management (IAM) user you specify. This automation deletes or detaches the following resources associated with the IAM user:

- Access keys
- Attached managed policies
- Git credentials
- IAM group memberships
- IAM user password
- Inline policies
- Multi-factor authentication (MFA) devices
- Signing certificates
- SSH public keys

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- IAMUserId

  Type: String

  Description: (Required) The ID of the IAM user you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:DeactivateMFADevice`
- `iam:DeleteAccessKey`
- `iam:DeleteLoginProfile`
- `iam:DeleteServiceSpecificCredential`
- `iam:DeleteSigningCertificate`
- `iam:DeleteSSHPublicKey`
- `iam:DeleteVirtualMFADevice`
- `iam:DeleteUser`
- `iam:DeleteUserPolicy`
- `iam:DetachUserPolicy`
- `iam:GetUser`
- `iam:ListAttachedUserPolicies`
- `iam:ListAccessKeys`
- `iam:ListGroupsForUser`
- `iam:ListMFADevices`
- `iam:ListServiceSpecificCredentials`
- `iam:ListSigningCertificates`
- `iam:ListSSHPublicKeys`
- `iam:ListUserPolicies`
- `iam:ListUsers`
- `iam:RemoveUserFromGroup`

Document Steps

- `aws:executeScript` - Gathers the user name of the IAM user you specify in the `IAMUserId` parameter.

- `aws:executeScript` - Gathers access keys, certificates, credentials, MFA devices, and SSH keys associated with the IAM user.
- `aws:executeScript` - Gathers group memberships and policies for the IAM user.
- `aws:executeScript` - Deletes access keys, certificates, credentials, MFA devices, and SSH keys associated with the IAM user.
- `aws:executeScript` - Deletes group memberships and policies for the IAM user.
- `aws:executeScript` - Deletes the IAM user and verifies the user has been deleted.

`AWSConfigRemediation-DeleteUnusedIAMGroup`

Description

The `AWSConfigRemediation-DeleteUnusedIAMGroup` runbook deletes an IAM group that does not contain any users.

The `AWSConfigRemediation-DeleteUnusedIAMGroup` runbook deletes an IAM group that does not contain any users.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- GroupName

  Type: String

  Description: (Required) The name of the IAM group that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:DeleteGroup`
- `iam:DeleteGroupPolicy`
- `iam:DetachGroupPolicy`

Document Steps

- `aws:executeScript` - Removes managed and inline IAM policies attached to the target IAM group, and then deletes the IAM group.

AWSConfigRemediation-DeleteUnusedIAMPolicy

Description

The `AWSConfigRemediation-DeleteUnusedIAMPolicy` runbook deletes an Amazon Identity and Access Management (IAM) policy that is not attached to any users, groups, or roles.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- IAMResourceId

  Type: String

  Description: (Required) The resource identifier of the IAM policy that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `config:ListDiscoveredResources`
- `iam:DeletePolicy`
- `iam:DeletePolicyVersion`
- `iam:GetPolicy`
- `iam:ListEntitiesForPolicy`
- `iam:ListPolicyVersions`

Document Steps

- `aws:executeScript` - Deletes the policy you specify in the `IAMResourceId` parameter, and verifies the policy was deleted.

AWSConfigRemediation-DetachIAMPolicy

Description

The AWSConfigRemediation-DetachIAMPolicy runbook detaches the Amazon Identity and Access Management (IAM) policy you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- IAMResourceId

  Type: String

  Description: (Required) The ID of the IAM policy you want to detach.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution

- ssm:GetAutomationExecution
- config:GetResourceConfigHistory
- config:ListDiscoveredResources
- iam:DetachGroupPolicy
- iam:DetachRolePolicy
- iam:DetachUserPolicy
- iam:GetPolicy
- iam:ListEntitiesForPolicy

Document Steps

- aws:executeScript - Detaches the IAM policy from all resources.

AWSConfigRemediation-EnableAccountAccessAnalyzer

Description

The AWSConfigRemediation-EnableAccountAccessAnalyzer runbook creates an Amazon Identity and Access Management (IAM) Access Analyzer in your Amazon Web Services account. For information about Access Analyzer, see Using Amazon IAM Access Analyzer in the *IAM User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AnalyzerName

Type: String

Description: (Required) The name of the analyzer to create.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `access-analyzer:CreateAnalyzer`
- `access-analyzer:GetAnalyzer`

Document Steps

- `aws:executeAwsApi` - Creates an access analyzer for your account.
- `aws:waitForAwsResourceProperty` - Waits for the status of the access analyzer to be `ACTIVE` .
- `aws:assertAwsResourceProperty` - Confirms the status of the access analyzer is `ACTIVE` .

AWSSupport-GrantPermissionsToIAMUser

Description

This runbook grants the specified permissions to an IAM group (new or existing), and adds the existing IAM user to it. Policies you can choose from: Billing or Support . To enable billing access for IAM, remember to also activate IAM user and federated user access to the Billing and Cost Management pages .

Important

If you provide an existing IAM group, all current IAM users in the group receive the new permissions.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- IAMGroupName

  Type: String

  Default: ExampleSupportAndBillingGroup

  Description: (Required) Can be a new or existing group. Must comply with IAM Entity Name Limits .
- IAMUserName

  Type: String

Default: ExampleUser

Description: (Required) Must be an existing user.

- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role assumed by lambda.

- Permissions

  Type: String

  Valid values: SupportFullAccess | BillingFullAccess | SupportAndBillingFullAccess

  Default: SupportAndBillingFullAccess

  Description: (Required) Choose one of: `SupportFullAccess` grants full access to the Support center. `BillingFullAccess` grants full access to the Billing dashboard. `SupportAndBillingFullAccess` grants full access to both Support center and the Billing dashboard. More info on policies under Document details.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

The permissions required depend on how `AWSSupport-GrantPermissionsToIAMUser` is run.

Running as the currently logged in user or role

It is recommended you have the `AmazonSSMAutomationRole` Amazon managed policy attached, and the following additional permissions to be able to create the Lambda function and the IAM role to pass to Lambda:

```
{
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Action": [
                                "lambda:InvokeFunction",
```

```
                                "lambda:CreateFunction",
                                "lambda:DeleteFunction",
                                "lambda:GetFunction"
                            ],
                            "Resource":
 "arn:aws:lambda:*:ACCOUNTID:function:AWSSupport-*",
                            "Effect": "Allow"
                        },
                        {
                            "Effect" : "Allow",
                            "Action" : [
                                "iam:CreateGroup",
                                "iam:AddUserToGroup",
                                "iam:ListAttachedGroupPolicies",
                                "iam:GetGroup",
                                "iam:GetUser"
                            ],
                            "Resource" : [
                                "arn:aws:iam::*:user/*",
                                "arn:aws:iam::*:group/*"
                            ]
                        },
                        {
                            "Effect" : "Allow",
                            "Action" : [
                                "iam:AttachGroupPolicy"
                            ],
                            "Resource": "*",
                            "Condition": {
                                "ArnEquals": {
                                    "iam:PolicyArn": [
                                        "arn:aws:iam::aws:policy/job-function/Billing",
                                        "arn:aws:iam::aws:policy/AWSSupportAccess"
                                    ]
                                }
                            }
                        },
                        {
                            "Effect" : "Allow",
                            "Action" : [
                                "iam:ListAccountAliases",
                                "iam:GetAccountSummary"
                            ],
                            "Resource" : "*"
```

```
                        }
                    ]
                }
```

Using AutomationAssumeRole and LambdaAssumeRole

The user must have the **ssm:StartAutomationExecution** permissions on the runbook, and **iam:PassRole** on the IAM roles passed as **AutomationAssumeRole** and **LambdaAssumeRole** . Here are the permissions each IAM role needs:

```
AutomationAssumeRole

                    {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Action": [
                                    "lambda:InvokeFunction",
                                    "lambda:CreateFunction",
                                    "lambda:DeleteFunction",
                                    "lambda:GetFunction"
                                ],
                                "Resource":
 "arn:aws:lambda:*:ACCOUNTID:function:AWSSupport-*",
                                "Effect": "Allow"
                            }
                        ]
                    }
```

```
LambdaAssumeRole

                {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect" : "Allow",
                            "Action" : [
                                "iam:CreateGroup",
                                "iam:AddUserToGroup",
                                "iam:ListAttachedGroupPolicies",
                                "iam:GetGroup",
                                "iam:GetUser"
```

```
                        ],
                        "Resource" : [
                            "arn:aws:iam::*:user/*",
                            "arn:aws:iam::*:group/*"
                        ]
                    },
                    {
                        "Effect" : "Allow",
                        "Action" : [
                            "iam:AttachGroupPolicy"
                        ],
                        "Resource": "*",
                        "Condition": {
                            "ArnEquals": {
                                "iam:PolicyArn": [
                                    "arn:aws:iam::aws:policy/job-function/Billing",
                                    "arn:aws:iam::aws:policy/AWSSupportAccess"
                                ]
                            }
                        }
                    },
                    {
                        "Effect" : "Allow",
                        "Action" : [
                            "iam:ListAccountAliases",
                            "iam:GetAccountSummary"
                        ],
                        "Resource" : "*"
                    }
                ]
            }
```

Document Steps

1. `aws:createStack` - Run Amazon CloudFormation Template to create a Lambda function.
2. `aws:invokeLambdaFunction` - Run Lambda to set IAM permissions.
3. `aws:deleteStack` - Delete CloudFormation Template.

Outputs

configureIAM.Payload

AWSConfigRemediation-RemoveUserPolicies

Description

The AWSConfigRemediation-RemoveUserPolicies runbook deletes the Amazon Identity and Access Management (IAM) inline policies and detaches any managed policies attached to the user you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- IAMUserID

  Type: String

  Description: (Required) The ID of the user you want to remove policies from.
- PolicyType

  Type: String

  Valid values: All | Inline | Managed

Default: All

Description: (Required) The type of IAM policies you want to remove from the user.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:DeleteUserPolicy`
- `iam:DetachUserPolicy`
- `iam:ListAttachedUserPolicies`
- `iam:ListUserPolicies`
- `iam:ListUsers`

Document Steps

- `aws:executeScript` - Deletes and detaches IAM policies from the user you specify in the `IAMUserID` parameter.

AWSConfigRemediation-ReplaceIAMInlinePolicy

Description

The `AWSConfigRemediation-ReplaceIAMInlinePolicy` runbook replaces an inline Amazon Identity and Access Management (IAM) policy with a replicated managed IAM policy. For an inline policy attached to a user, group, or role, the inline policy permissions are cloned into a managed IAM policy. The managed IAM policy is added to the resource, and the inline policy is removed. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- InlinePolicyName

  Type: StringList

  Description: (Required) The inline IAM policy you want to replace.
- ResourceId

  Type: String

  Description: (Required) The ID of the IAM user, group, or role whose inline policy you want to replace.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:AttachGroupPolicy`
- `iam:AttachRolePolicy`

- iam:AttachUserPolicy
- iam:CreatePolicy
- iam:CreatePolicyVersion
- iam:DeleteGroupPolicy
- iam:DeleteRolePolicy
- iam:DeleteUserPolicy
- iam:GetGroupPolicy
- iam:GetRolePolicy
- iam:GetUserPolicy
- iam:ListGroupPolicies
- iam:ListRolePolicies
- iam:ListUserPolicies

Document Steps

- aws:executeScript - Replace the inline IAM policy with an Amazon replicated policy on the resource that you specify.

AWSConfigRemediation-RevokeUnusedIAMUserCredentials

Description

The AWSConfigRemediation-RevokeUnusedIAMUserCredentials runbook revokes unused Amazon Identity and Access Management (IAM) passwords and active access keys. This runbook also deactivates expired access keys, and deletes expired login profiles. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- IAMResourceId

  Type: String

  Description: (Required) The ID of the IAM resource you want to revoke unused credentials from.
- MaxCredentialUsageAge

  Type: String

  Default: 90

  Description: (Required) The number of days within which the credential must have been used.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:ListDiscoveredResources`
- `iam:DeleteAccessKey`
- `iam:DeleteLoginProfile`
- `iam:GetAccessKeyLastUsed`
- `iam:GetLoginProfile`

- iam:GetUser
- iam:ListAccessKeys
- iam:UpdateAccessKey

Document Steps

- aws:executeScript - Revokes IAM credentials for the user specified in the IAMResourceId parameter. Expired access keys are deactivated, and expired login profiles are deleted.

> **Note**
>
> Make sure to configure the MaxCredentialUsageAgeparameter of this remediation action to match the maxAccessKeyAge parameter of the Amazon Config rule you use to trigger this action: access-keys-rotated.

AWSConfigRemediation-SetIAMPasswordPolicy

Description

The AWSConfigRemediation-SetIAMPasswordPolicy runbook sets the Amazon Identity and Access Management (IAM) user password policy for your Amazon Web Services account.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- AllowUsersToChangePassword

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , all IAM users in your Amazon Web Services account can use the Amazon Web Services Management Console to change their passwords.

- HardExpiry

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , IAM users are prevented from resetting their passwords after their password expires.

- MaxPasswordAge

  Type: Integer

  Default: 0

  Description: (Optional) The number of days an IAM user's password is valid.

- MinimumPasswordLength

  Type: Integer

  Default: 6

  Description: (Optional) The minimum number of characters an IAM user's password can be.

- PasswordReusePrevention

  Type: Integer

  Default: 0

Description: (Optional) The number of previous passwords that an IAM user is prevented from reusing.

- RequireLowercaseCharacters

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , an IAM user's password must contain a lowercase character from the ISO basic Latin alphabet (a to z).

- RequireNumbers

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , an IAM user's password must contain a numeric character (0-9).

- RequireSymbols

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , an IAM user's password must contain a non-alphanumeric character (! @ # $ % ^ * ( ) _ + - = [ ] { } | ').

- RequireUppercaseCharacters

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , an IAM user's password must contain an uppercase character from the ISO basic Latin alphabet (A to Z).

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- iam:GetAccountPasswordPolicy
- iam:UpdateAccountPasswordPolicy

Document Steps

- aws:executeScript - Sets the IAM user password policy based on the values you specify for the runbook parameters for your Amazon Web Services account.

AWSSupport-ContainIAMPrincipal

Description

In the event of a security incident or a suspected compromise of an Amazon Identity and Access Management (IAM) User/Role or Amazon Identity Center (IDC) user, swift isolation of the affected identity is crucial while preserving its configuration for investigation. The AWSSupport-ContainIAMPrincipal runbook provides a structured, reversible approach to contain compromised IAM or IDC identities, effectively blocking their access to Amazon resources and preventing potential spread of the compromise.

This automated process enables investigation without permanent alteration of the identity's configuration, allowing for restoration of normal access when deemed appropriate. The containment process maintains the user or role within IAM or the user within IDC, while effectively isolating it from all network activities. This isolation prevents the contained identity resource from communicating with resources inside your Amazon Virtual Private Cloud or accessing internet resources. The containment is designed to be reversible, allowing for restoration of normal access when deemed appropriate.

How does it work?

The AWSSupport-ContainIAMPrincipal runbook implements a comprehensive containment process for IAM users, roles, and Identity Center users. When executed in Contain mode, it first validates all input parameters and performs security checks on the specified Amazon S3 bucket. It then gathers detailed information about the target IAM principal and applies appropriate containment measures based on the principal type. For IAM users, it disables access keys, removes console access, and attaches a deny policy. For IAM roles, it attaches a deny policy that revokes

permissions for sessions created before containment. For Identity Center users, it removes permission sets, group memberships, and applies a deny policy. Throughout the process, the runbook backs up the original configuration to an Amazon S3 bucket for potential restoration. When executed in `Restore` mode, it attempts to revert the principal to its pre-containment state using the backed-up configuration. The runbook includes a `DryRun` option to preview changes without applying them, and provides comprehensive reporting on both successful operations and failure scenarios.

Important

- **Use of Elevated Privileges:** This SSM document performs various operations that require elevated privileges, such as modifying IAM and IDC identity policies and applying quarantine configurations. These actions could potentially lead to a privilege escalation or impact other workloads that depend on the targeted identities. You should review the permissions granted to the role specified by the `AutomationAssumeRole` parameter and ensure they are appropriate for the intended use case. You can refer to the following Amazon documentation for more information on IAM permissions:
  - Identity and Access Management (IAM) Permissions
  - Amazon Systems Manager Automation Permissions
- **Workload Unavailability Risks:** This Systems Manager document performs isolation actions that could potentially cause unavailability or disruption to your workloads. When executed during a security event, it will restrict access to the affected resource by revoking Amazon API permissions from the specified IAM and IDC identities, preventing them from making any Amazon API calls or actions. This could impact any applications or services that depend on these identities.
- **Creation of Additional Resources:** The automation document may conditionally create additional resources, such as an Amazon Simple Storage Service (Amazon S3) bucket and Amazon S3 objects stored in them, depending on the execution parameters. These resources will incur additional charges based on your Amazon usage.
- **Restoration Risks:** If the *Action* parameter is set to `Restore`, this SSM document attempts to restore the IAM or IDC identity configuration to its original state. However, there is a risk that the restoration process may fail, leaving the IAM or IDC identity in an inconsistent state. The document provides instructions for manual restoration in case of such failures, but you should be prepared to handle potential issues during the restoration process.

It is recommended to review the runbook thoroughly, understand its potential impacts, and test it in a non-production environment before executing it in your production environment.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following permissions to successfully use the runbook:

- s3:GetBucketLocation
- s3:GetBucket
- s3:ListBucket
- s3:GetBucketPublicAccessBlocks
- s3:GetAccountPublicAccessBlocks
- s3:GetBucketPolicyStatus
- s3:GetBucketAcl
- s3:GetObject
- s3:CreateBucket
- s3:PutObject
- iam:GetUser
- iam:GetUserPolicy
- iam:GetRole

- iam:ListUserPolicies
- iam:ListAttachedUserPolicies
- iam:ListAccessKeys
- iam:ListMfaDevices
- iam:ListVirtualMFADevices
- iam:GetLoginProfile
- iam:GetPolicy
- iam:GetRolePolicy
- iam:ListPolicies
- iam:ListAttachedRolePolicies
- iam:ListRolePolicies
- iam:UpdateAccessKey
- iam:CreateAccessKey
- iam:DeleteLoginProfile
- iam:DeleteAccessKey
- iam:PutUserPolicy
- iam:DeleteUserPolicy
- iam:DeactivateMFADevice
- iam:AttachRolePolicy
- iam:AttachUserPolicy
- iam:DeleteRolePolicy
- iam:TagMFADevice
- iam:PutRolePolicy
- iam:TagPolicy
- iam:TagRole
- iam:TagUser
- iam:UntagUser
- iam:UntagRole
- organizations:ListAccounts
- sso:ListPermissionSetsProvisionedToAccount

- sso:GetInlinePolicyForPermissionSet
- sso:ListInstances
- sso-directory:SearchUsers
- sso:ListPermissionSets
- sso:ListAccountAssignments
- sso-directory:DescribeUser
- identitystore:ListUsers
- identitystore:ListGroups
- identitystore:IsMemberInGroups
- identitystore:ListGroupMemberships
- secretsmanager:CreateSecret
- secretsmanager:DeleteSecret
- sso:DeleteAccountAssignment
- sso:PutInlinePolicyToPermissionSet
- sso:CreateAccountAssignment
- sso:DeleteInlinePolicyFromPermissionSet
- sso:TagResource
- sso:UntagResource
- identitystore:DeleteGroupMembership
- identitystore:CreateGroupMembership

Here is an example of an IAM policy that grants the necessary permissions for the `AutomationAssumeRole`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3Permissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetBucket",
                "s3:ListBucket",
```

```
                "s3:GetBucketPublicAccessBlocks",
                "s3:GetAccountPublicAccessBlocks",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketAcl",
                "s3:GetObject",
                "s3:CreateBucket",
                "s3:PutObject"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMPermissions",
            "Effect": "Allow",
            "Action": [
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:GetRole",
                "iam:ListUserPolicies",
                "iam:ListAttachedUserPolicies",
                "iam:ListAccessKeys",
                "iam:ListMfaDevices",
                "iam:ListVirtualMFADevices",
                "iam:GetLoginProfile",
                "iam:GetPolicy",
                "iam:GetRolePolicy",
                "iam:ListPolicies",
                "iam:ListAttachedRolePolicies",
                "iam:ListRolePolicies",
                "iam:UpdateAccessKey",
                "iam:CreateAccessKey",
                "iam:DeleteLoginProfile",
                "iam:DeleteAccessKey",
                "iam:PutUserPolicy",
                "iam:DeleteUserPolicy",
                "iam:DeactivateMFADevice",
                "iam:AttachRolePolicy",
                "iam:AttachUserPolicy",
                "iam:DeleteRolePolicy",
                "iam:TagMFADevice",
                "iam:PutRolePolicy",
                "iam:TagPolicy",
                "iam:TagRole",
                "iam:TagUser",
                "iam:UntagUser",
```

```
                "iam:UntagRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OrganizationsPermissions",
            "Effect": "Allow",
            "Action": [
                "organizations:ListAccounts"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSOPermissions",
            "Effect": "Allow",
            "Action": [
                "sso:ListPermissionSetsProvisionedToAccount",
                "sso:GetInlinePolicyForPermissionSet",
                "sso:ListInstances",
                "sso-directory:SearchUsers",
                "sso:ListPermissionSets",
                "sso:ListAccountAssignments",
                "sso-directory:DescribeUser",
                "sso:DeleteAccountAssignment",
                "sso:PutInlinePolicyToPermissionSet",
                "sso:CreateAccountAssignment",
                "sso:DeleteInlinePolicyFromPermissionSet",
                "sso:TagResource",
                "sso:UntagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IdentityStorePermissions",
            "Effect": "Allow",
            "Action": [
                "identitystore:ListUsers",
                "identitystore:ListGroups",
                "identitystore:IsMemberInGroups",
                "identitystore:ListGroupMemberships",
                "identitystore:DeleteGroupMembership",
                "identitystore:CreateGroupMembership"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "SecretsManagerPermissions",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:DeleteSecret"
            ],
            "Resource": "*"
        }
    ]
}
```

Instructions

Follow these steps to configure the automation:

1. Navigate to the AWSSupport-ContainIAMPrincipal in the Amazon Systems Manager console.
2. Select **Execute automation.**
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**
     - Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
     - Type: `AWS::IAM::Role::Arn`
   - **PrincipalType (Required):**
     - Description: (Required) The Amazon IAM principal type: IAM user, IAM role, or Identity Center user.
     - Type: String
     - Allowed Values: `IAM user|IAM role|Identity Center user`
   - **PrincipalName (Required):**
     - Description: (Required) The name of the IAM principal. For Identity Center users, provide the username.
     - Type: String
     - Allowed Pattern: `^[a-zA-Z0-9\\.\\-_\\\\!*'()/+=,@]{1,1024}$`

- **Action (Required):**
  - Description: (Required) Select `Contain` to isolate the target IAM principal or `Restore` to try to restore the IAM principal to its original configuration from a previous backup.
  - Type: String
  - Allowed Values: `Contain|Restore`
- **DryRun (Optional):**
  - Description: (Optional) When set to `true`, the automation will not make any changes to the target IAM principal, instead it will output on what it would have attempted to change, detailing out on each step. Default value: `true`.
  - Type: Boolean
  - Allowed Values: `true|false`
- **ActivateDisabledKeys (Conditional):**
  - Description: (Conditional) If the input parameter Action is set to `Restore` and the PrincipalType is set to IAM user, this option determines if this automation should try to activate the associated access keys if deactivated. Please note that the integrity of a compromised access key cannot be verified. Amazon strongly recommends against reactivating a compromised key. Instead, it is advisable to generate new keys. Default value: `false`.
  - Type: Boolean
  - Allowed Values: `true|false`
- **BackupS3BucketName (Conditional):**
  - Description: (Conditional) The Amazon Amazon S3 bucket to backup the IAM principal configuration when the Action is set to `Contain` or to restore the configuration from when the Action is `Restore`. Note that if the specified Action is `Contain` and the runbook is not able to access the bucket or a value is not provided, a new bucket is created in your account with the name `awssupport-containiamprincipal-<random-string>`. If DryRun is set to `true` this parameter is required.
  - Type: `AWS::S3::Bucket::Name`
- **BackupS3KeyName (Conditional):**
  - Description: (Conditional) If Action is set to `Restore`, this specifies the Amazon Amazon S3 key the automation will use to try to restore the IAM principal configuration. The Amazon Amazon S3 key typically follows this format: `{year}/{month}/{day}/{hour}/`

{minute}/{automation_execution_id}.json. The key can be obtained from the output of a previous containment automation execution.

- Type: String
- Allowed Pattern: ^[a-zA-Z0-9\\.\\-_\\\\!*'()/]{0,1024}$

- **BackupS3BucketAccess (Conditional):**
  - Description: (Conditional) The ARN of the IAM users or roles that will be allowed access to the backup Amazon Amazon S3 bucket after running the containment actions. This parameter is required when Action is Contain. The AutomationAssumeRole, or in its absence the user under whose context the automation is running is automatically added to the list.
  - Type: StringList
  - Allowed Pattern: ^$|^arn:(aws|aws-cn|aws-us-gov|aws-iso(-[a-z])?):iam::[0-9]{12}:(role|user)\\/[\\w+\\/=,.@-]+$
- **TagIdentifier (Optional):**
  - Description: (Optional) Tag the IAM principal with a tag of your choice using the following format: Key=<EXAMPLE_KEY>,Value=<EXAMPLE_VALUE>. This option allows you to track the IAM principals that have been targeted by this runbook. **Note:** Tag keys and values are case-sensitive.
  - Type: String
  - Allowed Pattern: ^$|^[Kk][Ee][Yy]=[\\+\\-\\=\\.\\_\\:\\/@a-zA-Z0-9]{1,128},[Vv][Aa][Ll][Uu][Ee]=[\\+\\-\\=\\.\\_\\:\\/@a-zA-Z0-9]{0,128}$

4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:
   - **ValidateRequiredInputs**

     Validates the required automation input parameters based on the Action specified.
   - **CheckBackupS3BucketName**

     Checks if the target Amazon Amazon S3 bucket potentially grants read or write public access to its objects. In case of containment workflow, a new Amazon Amazon S3 bucket is created if the BackupS3BucketName bucket doesn't exist.
   - **BranchOnAction**

Branches the automation based on the value of the specified `Action`.

- **BranchOnPrincipalTypeAndDryRun**

  Branches the automation based on the type of IAM principal (IAM user, IAM role, or Identity Center user) and if it is running in `DryRun` mode.

- **BranchOnPrincipalTypeForContain**

  Branches the automation for the `Contain` action based and the IAM principal type (IAM user, IAM role, or Identity Center user) specified in the input.

- **GetIAMUser**

  Gets the creation time and username of the target IAM user.

- **GetIAMUserDetails**

  Gets and stores the configuration of the target IAM user, including inline policies, managed policies, access keys, MFA devices, and login profile.

- **UpdateS3KeyForUser**

  Updates the automation 'S3Key' variable from output of the step `GetIAMUserDetails`.

- **GetIAMRole**

  Gets the creation time, role name, and path of the target IAM role.

- **GetIAMRoleDetails**

  Gets and stores the configuration of the target IAM role, including inline policies and managed policies attached to the role.

- **UpdateS3KeyForRole**

  Updates the automation 'S3Key' variable from output of the step `GetIAMRoleDetails`.

- **GetIdentityStoreId**

  Gets the ID of the Amazon IAM Identity Center instance associated with the Amazon account.

- **GetIDCUser**

  Gets the user ID of the target Identity Center user using the Identity Store ID.

- **GatherIDCUserDetails**

Gets and stores the configuration of the target Identity Center user, including account assignments, associated permission sets, and inline policies.

- **UpdateS3KeyForIDCUser**

  Updates the automation 'S3Key' variable from output of the step `GatherIDCUserDetails`.

- **BranchOnIdentityContain**

  Branches the automation based on the value of `DryRun` and the IAM principal type for the `Contain` action.

- **BranchOnDisableAccessKeys**

  Branches the automation based on whether the IAM user has access keys that need to be disabled.

- **DisableAccessKeys**

  Disables the active IAM user access keys.

- **BranchOnDisableConsoleAccess**

  Branches based on whether the IAM user has Amazon Management Console access enabled or not.

- **DisableConsoleAccess**

  Removes the IAM user's password-based access to the Amazon Management Console.

- **AttachInlineDenyPolicyToUser**

  Attaches a deny policy to the IAM user to revoke permissions for older session tokens.

- **AttachInlineDenyPolicyToRole**

  Attaches a deny policy to the IAM role to revoke permissions for older session tokens.

- **RemovePermissionSets**

  Removes permission sets associated with the Identity Center user.

- **RemoveIDCUserFromIDCGroups**

  Removes the Identity Center user from Identity Center groups.

- **AttachInlineDenyPolicyToPermissionSet**

Attaches a deny policy to the permission sets associated with the Identity Center user.

- **BranchOnReactivateKeys**

  Branches the automation based on the `ActivateDisabledKeys` parameter during the restore process.

- **DetachInlineDenyPolicy**

  Removes the deny policy attached to the IAM role during the containment process.

- **DetachInlineDenyPolicyFromPermissionSet**

  Removes the deny policy attached to the permission sets during the containment process.

- **ReportContain**

  Outputs detailed information about the containment actions that would be performed when `DryRun` is set to `True`.

- **ReportRestore**

  Outputs detailed information about the restoration actions that would be performed when `DryRun` is set to `True`.

- **ReportContainFailure**

  Provides comprehensive instructions to manually restore the IAM principal's original configuration during a containment workflow failure scenario.

- **ReportRestoreFailure**

  Provides detailed instructions to manually complete the restoration of the IAM principal's original configuration during a restore workflow failure scenario.

7. After the execution completes, review the Outputs section for the detailed results of the execution:

   - **ContainIAMPrincipal.Output**

     Provides detailed information about the containment actions performed when Action is set to Contain and DryRun is set to False. Includes information about the backup location, applied deny policies, and modified configurations.

   - **RestoreIAMPrincipal.Output**

Provides detailed information about the restoration actions performed when Action is set to Restore and DryRun is set to False. Includes information about the restored configurations and any issues encountered during restoration.

- **ReportContain.Output**

  Outputs detailed information about the containment actions that would be performed when Action is set to Contain and DryRun is set to True. Includes a comparison of current and post-containment configurations.

- **ReportRestore.Output**

  Outputs detailed information about the restoration actions that would be performed when Action is set to Restore and DryRun is set to True. Shows the current configuration and the original configuration that would be restored.

- **ReportContainFailure.Output**

  Provides comprehensive instructions to manually restore the IAM principal's original configuration during a containment workflow failure scenario.

- **ReportRestoreFailure.Output**

  Provides detailed instructions to manually complete the restoration of the IAM principal's original configuration during a restore workflow failure scenario.

Outputs

After the execution completes, review the Outputs section for the detailed results:

- **ContainIAMPrincipal.Output**

  Provides detailed information about the containment actions performed when Action is set to Contain and DryRun is set to False. Includes information about the backup location, applied deny policies, and modified configurations.

- **RestoreIAMPrincipal.Output**

  Provides detailed information about the restoration actions performed when Action is set to Restore and DryRun is set to False. Includes information about the restored configurations and any issues encountered during restoration.

- **ReportContain.Output**

Outputs detailed information about the containment actions that would be performed when Action is set to Contain and DryRun is set to True. Includes a comparison of current and post-containment configurations.

- **ReportRestore.Output**

  Outputs detailed information about the restoration actions that would be performed when Action is set to Restore and DryRun is set to True. Shows the current configuration and the original configuration that would be restored.

- **ReportContainFailure.Output**

  Provides comprehensive instructions to manually restore the IAM principal's original configuration during a containment workflow failure scenario.

- **ReportRestoreFailure.Output**

  Provides detailed instructions to manually complete the restoration of the IAM principal's original configuration during a restore workflow failure scenario.

References

Systems Manager Automation

- Run this Automation (console)
- Running a simple automation
- Setting up Automation
- Support Automation Workflows

Amazon Kinesis Data Streams

Amazon Systems Manager Automation provides predefined runbooks for Amazon Kinesis Data Streams. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-EnableKinesisStreamEncryption

AWS-EnableKinesisStreamEncryption

Description

The AWS-EnableKinesisStreamEncryption runbook enables encryption on an Amazon Kinesis Data Streams (Kinesis Data Streams). Producer applications writing to an encrypted stream will encounter errors if they do not have access to the Amazon Key Management Service (Amazon KMS) key.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- KinesisStreamName

  Type: String

  Description: (Required) The name of the stream you want to enable encryption on.

- KeyId

  Type: String

Default: alias/aws/kinesis

Description: (Required) The customer-managed Amazon KMS key you want to use for encryption. This value can be a globally unique identifier, an ARN to either an alias or a key, or an alias name prefixed by "alias/". You can also use the Amazon managed key by using the default value for the parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `kinesis:DescribeStream`
- `kinesis:StartStreamEncryption`
- `kms:DescribeKey`

Document Steps

- VerifyKinesisStreamStatus (`aws:waitforAwsResourceProperty`) - Checks the status of the Kinesis Data Streams.
- EnableKinesisStreamEncryption (`aws:executeAwsApi`) - Enables encryption for the Kinesis Data Streams.
- VerifyKinesisStreamUpdateComplete (`aws:waitForAwsResourceProperty`) - Waits for the Kinesis Data Streams status to return to `ACTIVE`.
- VerifyKinesisStreamEncryption (`aws:assertAwsResourceProperty`) - Verifies encryption is enabled for the Kinesis Data Streams.

Amazon KMS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Key Management Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWSConfigRemediation-CancelKeyDeletion

Description

The `AWSConfigRemediation-CancelKeyDeletion` runbook cancels deletion of the Amazon Key Management Service (Amazon KMS) customer managed key that you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- KeyId

  Type: String

  Description: (Required) The ID of the customer managed key that you want to cancel deletion for.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `kms:CancelKeyDeletion`
- `kms:DescribeKey`

Document Steps

- `aws:executeAwsApi` - Cancels deletion for the customer managed key you specify in the `KeyId` parameter.
- `aws:assertAwsResourceProperty` - Confirms key deletion is disabled on your customer managed key.

AWSConfigRemediation-EnableKeyRotation

Description

The `AWSConfigRemediation-EnableKeyRotation` runbook enables automatic key rotation for the symmetric Amazon Key Management Service (Amazon KMS) customer managed key.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- KeyId

  Type: String

  Description: (Required) The ID of the customer managed key you want to enable automatic key rotation on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `kms:EnableKeyRotation`
- `kms:GetKeyRotationStatus`

Document Steps

- `aws:executeAwsApi` - Enables automatic key rotation on the customer managed key you specify in the `KeyId` parameter.
- `aws:assertAwsResourceProperty` - Confirms that automatic key rotation is enabled on your customer managed key.

Lambda

Amazon Systems Manager Automation provides predefined runbooks for Amazon Lambda. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-ConfigureLambdaFunctionXRayTracing
- AWSConfigRemediation-DeleteLambdaFunction
- AWSConfigRemediation-EncryptLambdaEnvironmentVariablesWithCMK
- AWSConfigRemediation-MoveLambdaToVPC
- AWSSupport-RemediateLambdaS3Event
- AWSSupport-TroubleshootLambdaInternetAccess
- AWSSupport-TroubleshootLambdaS3Event

AWSConfigRemediation-ConfigureLambdaFunctionXRayTracing

Description

The `AWSConfigRemediation-ConfigureLambdaFunctionXRayTracing` runbook enables Amazon X-Ray live tracing on the Amazon Lambda function you specify in the `FunctionName` parameter.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- FunctionName

  Type: String

  Description: (Required) The name or ARN of the Lambda function to enable tracing on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `lambda:UpdateFunctionConfiguration`
- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`

Document Steps

- `aws:executeAwsApi` - Enables X-Ray tracing on the Lambda function you specify in the `FunctionName` parameter.
- `aws:assertAwsResourceProperty` - Verifies that X-Ray tracing has been enabled on the Lambda function.

Outputs

UpdateLambdaConfig.UpdateFunctionConfigurationResponse - Response from the `UpdateFunctionConfiguration` API call.

AWSConfigRemediation-DeleteLambdaFunction

Description

The `AWSConfigRemediation-DeleteLambdaFunction` runbook deletes the Amazon Lambda function you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LambdaFunctionName

  Type: String

  Description: (Required) The name of the Lambda function that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `lambda:DeleteFunction`
- `lambda:GetFunction`

Document Steps

- `aws:executeAwsApi` - Deletes the Lambda function specified in the `LambdaFunctionName` parameter.
- `aws:executeScript` - Verifies the Lambda function has been deleted.

AWSConfigRemediation-EncryptLambdaEnvironmentVariablesWithCMK

Description

The AWSConfigRemediation-EncryptLambdaEnvironmentVariablesWithCMK runbook encrypts, at rest, the environment variables for the Amazon Lambda (Lambda) function you specify using an Amazon Key Management Service (Amazon KMS) customer managed key. This runbook should only be used as a baseline to ensure that your Lambda function's environment variables are encrypted according to minimum recommended security best practices. We recommend encrypting multiple functions with different customer managed keys.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- FunctionName

  Type: String

  Description: (Required) The name or ARN of the Lambda function whose environment variables you want to encrypt.
- KMSKeyArn

Type: String

Description: (Required) The ARN of the Amazon KMS customer managed key you want to use to encrypt your Lambda function's environment variables.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `lambda:GetFunctionConfiguration`
- `lambda:UpdateFunctionConfiguration`

Document Steps

- `aws:waitForAwsResourceProperty` - Waits for the `LastUpdateStatus` property to be `Successful` .
- `aws:executeAwsApi` - Encrypts the environment variables for the Lambda function you specify in the `FunctionName` parameter using the Amazon KMS customer managed key you specify in the `KMSKeyArn` parameter.
- `aws:assertAwsResourceProperty` - Confirms encryption is enabled on the environment variables for your Lambda function.

AWSConfigRemediation-MoveLambdaToVPC

Description

The `AWSConfigRemediation-MoveLambdaToVPC` runbook moves an Amazon Lambda (Lambda) function to an Amazon Virtual Private Cloud (Amazon VPC).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- FunctionName

  Type: String

  Description: (Required) The name of the Lambda function to move to an Amazon VPC.
- SecurityGroupIds

  Type: String

  Description: (Required) The security group IDs you want to assign to the elastic network interfaces (ENIs) associated with your Lambda function.
- SubnetIds

  Type: String

  Description: (Required) The subnet IDs you want to create the elastic network interfaces (ENIs) associated with your Lambda function.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`

- ssm:GetAutomationExecution
- lambda:GetFunction
- lambda:GetFunctionConfiguration
- lambda:UpdateFunctionConfiguration

Document Steps

- aws:executeAwsApi - Updates the Amazon VPC configuration for the Lambda function you specify in the FunctionName parameter.
- aws:waitForAwsResourceProperty - Waits for the Lambda function LastUpdateStatus to be successful .
- aws:executeScript - Verifies the Lambda function Amazon VPC configuration has been successfully updated.

AWSSupport-RemediateLambdaS3Event

Description

The AWSSupport-TroubleshootLambdaS3Event runbook provides an automated solution for the procedures outlined in the Amazon Knowledge Center articles Why doesn't my Amazon S3 event notification trigger my Lambda function? and Why do I get the error "Unable to validate the following destination configurations" when creating an Amazon S3 event notification to trigger my Lambda function? This runbook helps you identify and remediate why an Amazon Simple Storage Service (Amazon S3) event notification failed to trigger the Amazon Lambda function you specified. If the runbook output suggests validating and configuring your Lambda function concurrency, see Asynchronous invocation and Amazon Lambda Function scaling .

> **Note**
>
> "Unable to validate the following destination configurations" errors can also occur due to incorrect Amazon Simple Notification Service (Amazon SNS) and Amazon Simple Queue Service (Amazon SQS) Amazon S3 event configurations. This runbook only checks Lambda function configurations. If after using the runbook, you are still receiving the "Unable to validate the following destination configurations" error, please review any existing Amazon SNS and Amazon SQS Amazon S3 event configurations.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- LambdaFunctionArn

  Type: String

  Description: (Required) The ARN of the Lambda function.

- S3BucketName

  Type: String

  Description: (Required) The name of the Amazon S3 bucket whose event notifications triggers the Lambda function.

- Action

  Type: String

  Valid values: Troubleshoot | Remediate

Description: (Required) The action you want the runbook to perform. The `Troubleshoot` option helps identify any issues, but does not perform any mutating actions to resolve the issue. The `Remediate` option helps identify and attempts to resolve issues for you.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetDocument`
- `ssm:ListDocuments`
- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:GetAutomationExecution`
- `lambda:GetPolicy`
- `lambda:AddPermission`
- `s3:GetBucketNotification`

Document Steps

- `aws:branch` - Branches based on the input specified for the `Action` parameter.

  If the value specified is `Troubleshoot` :

  - `aws:executeAutomation` - Runs the `AWSSupport-TroubleshootLambdaS3Event` runbook.
  - `aws:executeAwsApi` - Checks the output of the `AWSSupport-TroubleshootLambdaS3Event` runbook that ran in the previous step.

  If the value specified is `Remediate` :

  - `aws:executeScript` - Runs a script to remediate the issues outlined in the Why doesn't my Amazon S3 event notification trigger my Lambda function? and Why do I get the error "Unable to validate the following destination configurations" when creating an Amazon S3 event notification to trigger my Lambda function? Knowledge Center articles.

Outputs

checkoutput.Output

remediatelambdas3event.Output

AWSSupport-TroubleshootLambdaInternetAccess

Description

The AWSSupport-TroubleshootLambdaInternetAccess runbook helps you troubleshoot internet access issues for a Amazon Lambda function that was launched into Amazon Virtual Private Cloud (Amazon VPC). Resources such as subnet routes, security groups rules, and network access control list (ACL) rules are reviewed to confirm outbound internet access is allowed.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- FunctionName

Type: String

Description: (Required) The name of the Lambda function you want to troubleshoot internet access for.

- destinationIp

  Type: String

  Description: (Required) The destination IP address you want to establish an outbound connection to.

- destinationPort

  Type: String

  Default: 443

  Description: (Optional) The destination port you want to establish an outbound connection on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `lambda:GetFunction`
- `ec2:DescribeRouteTables`
- `ec2:DescribeNatGateways`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkAcls`

Document Steps

- `aws:executeScript` - Verifies the configuration of various resources in your VPC where the Lambda function was launched.
- `aws:branch` - Branches based on whether the Lambda function specified is in a VPC or not.
- `aws:executeScript` - Reviews the route table routes for the subnet where the Lambda function was launched, and verifies that routes to a network address translation (NAT) gateway, and internet gateway are present. Confirms the Lambda function is not in a public subnet.

- `aws:executeScript` - Verifies the security group associated with the Lambda function allows outbound internet access based on the values specified for the `destinationIp` and `destinationPort` parameters.
- `aws:executeScript` - Verifies the ACL rules associated with the subnets of the Lambda function and the NAT gateway allow outbound internet access based on the values specified for the `destinationIp` and `destinationPort` parameters.

Outputs

checkVpc.vpc - The ID of the VPC where your Lambda function was launched.

checkVpc.subnet - The IDs of the subnets where your Lambda function was launched.

checkVpc.securityGroups - Security groups associated with the Lambda function.

checkNACL.NACL - Analysis message with resource names. `LambdaIp` refers to the private IP address of the elastic network interface for your Lambda function. The `LambdaIpRules` object is only generated for subnets that have a route to a NAT gateway. The following content is an example of the output.

```
{
   "subnet-1234567890":{
      "NACL":"acl-1234567890",
      "destinationIp_Egress":"Allowed",
      "destinationIp_Ingress":"notAllowed",
      "Analysis":"This NACL has an allow rule for Egress traffic but there is no
 Ingress rule. Please allow the destination IP / destionation port in Ingress rule",
      "LambdaIpRules":{
         "{LambdaIp}":{
            "Egress":"notAllowed",
            "Ingress":"notAllowed",
            "Analysis":"This is a NAT subnet NACL. It does not have ingress or egress
 rule allowed in it for Lambda's corresponding private ip {LambdaIp} Please allow this
 IP in your egress and ingress NACL rules"
         }
      }
   },
   "subnet-0987654321":{
      "NACL":"acl-0987654321",
      "destinationIp_Egress":"Allowed",
      "destinationIp_Ingress":"notAllowed",
```

```
      "Analysis":"This NACL has an allow rule for Egress traffic but there is no
 Ingress rule. Please allow the destination IP / destionation port in Ingress rule"
   }
}
```

checkSecurityGroups.secgrps - Analysis for the security group associated with your Lambda function. The following content is an example of the output.

```
{
   "sg-123456789":{
      "Status":"Allowed",
      "Analysis":"This security group has allowed destintion IP and port in its
 outbuond rule."
   }
}
```

checkSubnet.subnets - Analysis for the subnets in your VPC associated with your Lambda function. The following content is an example of the output.

```
{
   "subnet-0c4ee6cdexample15":{
      "Route":{
         "DestinationCidrBlock":"8.8.8.0/26",
         "NatGatewayId":"nat-00f0example69fdec",
         "Origin":"CreateRoute",
         "State":"active"
      },
      "Analysis":"This Route Table has an active NAT gateway path. Also, The NAT
 gateway is launched in public subnet",
      "RouteTable":"rtb-0b1fexample16961b"
   }
}
```

AWSSupport-TroubleshootLambdaS3Event

Description

The AWSSupport-TroubleshootLambdaS3Event runbook provides an automated solution for the procedures outlined in the Amazon Knowledge Center articles Why doesn't my Amazon S3 event notification trigger my Lambda function? and Why do I get the error "Unable to validate the following destination configurations" when creating an Amazon S3 event notification to trigger

my Lambda function? This runbook helps you identify why an Amazon Simple Storage Service (Amazon S3) event notification failed to trigger the Amazon Lambda function you specified. If the runbook output suggests validating and configuring your Lambda function concurrency, see Asynchronous invocation and Amazon Lambda Function scaling .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- LambdaFunctionArn

  Type: String

  Description: (Required) The ARN of the Lambda function that the Amazon S3 event notification triggers.

- S3BucketName

  Type: String

  Description: (Required) The name of the Amazon S3 bucket whose event notifications triggers the Lambda function.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `lambda:GetPolicy`
- `s3:GetBucketNotification`

Document Steps

- `aws:executeScript` - Runs the script to validate configuration settings for the Amazon S3 event notification. Validates the resource-based IAM policy for your Lambda function, and generates an Amazon Command Line Interface (Amazon CLI) command to add the needed permissions if the required permissions are missing from the policy. Validates other Lambda functions resource policies which are part of event notifications for the same S3 bucket and generates an Amazon CLI command as output if the required permissions are missing.

Outputs

lambdaS3Event.output

Amazon Managed Workflows for Apache Airflow

Amazon Systems Manager Automation provides predefined runbooks for Amazon Managed Workflows for Apache Airflow. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-TroubleshootMWAAEnvironmentCreation

`AWSSupport-TroubleshootMWAAEnvironmentCreation`

Description

The `AWSSupport-TroubleshootMWAAEnvironmentCreation` runbook provides information to debug Amazon Managed Workflows for Apache Airflow (Amazon MWAA) environment creation issues, and perform checks along with the documented reasons on a best effort basis to help identify the failure.

How does it work?

The runbook performs the following steps:

- Retrieves the details of the Amazon MWAA environment.
- Verifies the execution role permissions.
- Checks if the environment has permissions to use the provided Amazon KMS key for logging, and if the required CloudWatch log group exists.
- Parses the logs in the provided log group to locate any errors.
- Checks the network configuration to verify if the Amazon MWAA environment has access to the required endpoints.
- Generates a report with the findings.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `airflow:GetEnvironment`
- `cloudtrail:LookupEvents`
- `ec2:DescribeNatGateways`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`

- ec2:DescribeSubnets
- ec2:DescribeVpcEndpoints
- iam:GetPolicy
- iam:GetPolicyVersion
- iam:GetRolePolicy
- iam:ListAttachedRolePolicies
- iam:ListRolePolicies
- iam:SimulateCustomPolicy
- kms:GetKeyPolicy
- kms:ListAliases
- logs:DescribeLogGroups
- logs:FilterLogEvents
- s3:GetBucketAcl
- s3:GetBucketPolicyStatus
- s3:GetPublicAccessBlock
- s3control:GetPublicAccessBlock
- ssm:StartAutomationExecution
- ssm:GetAutomationExecution

Instructions

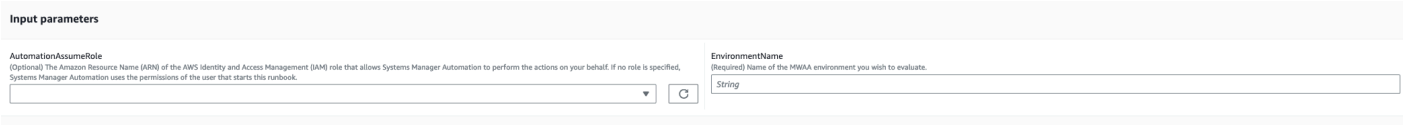
Follow these steps to configure the automation:

1. Navigate to AWSSupport-TroubleshootMWAAEnvironmentCreation in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

- **EnvironmentName (Required):**

  Name of the Amazon MWAA environment you wish to evaluate.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

- **`GetMWAAEnvironmentDetails:`**

  Retrieves the details of the Amazon MWAA environment. If this step fails, the automation process will halt and show as `Failed`.

- **`CheckIAMPermissionsOnExecutionRole:`**

  Verifies that the execution role has the required permissions for Amazon MWAA, Amazon S3, CloudWatch Logs, CloudWatch, and Amazon SQS resources. If it detects a customer managed Amazon Key Management Service (Amazon KMS) key, the automation validates the key's required permissions. This step employs the `iam:SimulateCustomPolicy` API to ascertain if the automation execution role meets all required permissions.

- **`CheckKMSPolicyOnKMSKey:`**

  Checks if the Amazon KMS key policy allows the Amazon MWAA environment to use the key for encrypting CloudWatch Logs. If the Amazon KMS key is Amazon-managed, the automation skips this check.

- **`CheckIfRequiredLogGroupsExists:`**

  Checks if the required CloudWatch log groups for the Amazon MWAA environment exist. If not, the automation checks CloudTrail for `CreateLogGroup` and `DeleteLogGroup` events. This step also checks for `CreateLogGroup` events.

- **`BranchOnLogGroupsFindings:`**

  Branches based on the existence of CloudWatch log groups related to the Amazon MWAA environment. If at least one log group exists, the automation parses it to locate errors. If no log groups are present, the automation skips the next step.

- **`CheckForErrorsInLogGroups:`**

Parses the CloudWatch log groups to locate errors.

- **`GetRequiredEndPointsDetails:`**

  Retrieves the service endpoints utilized by the Amazon MWAA environment.

- **`CheckNetworkConfiguration:`**

  Verifies that the Amazon MWAA environment's network configuration meets the requirements, including checks on security groups, network ACLs, subnets, and route table configurations.

- **`CheckEndpointsConnectivity:`**

  Invokes the `AWSSupport-ConnectivityTroubleshooter` child automation to validate the Amazon MWAA's connectivity to the required endpoints.

- **`CheckS3BlockPublicAccess:`**

  Checks whether the Amazon MWAA environment's Amazon S3 bucket has `Block Public Access` enabled and also reviews the account's overall Amazon S3 Block Public Access settings.

- **`GenerateReport:`**

  Gathers information from the automation and prints the result or output of each step.

7. After completed, review the Outputs section for the detailed results of the execution:

- **Checking the Amazon MWAA environment execution role permissions:**

  Verifies if the execution role has the required permissions for Amazon MWAA, Amazon S3, CloudWatch Logs, CloudWatch, and Amazon SQS resources. If a Customer Managed Amazon KMS key is detected, the automation validates the key's required permissions.

- **Checking the Amazon MWAA environment Amazon KMS key policy:**

  Verifies whether the execution role possesses the necessary permissions for Amazon MWAA, Amazon S3, CloudWatch Logs, CloudWatch, and Amazon SQS resources. Additionally, if a Customer Managed Amazon KMS key is detected, the automation checks for the key's required permissions.

- **Checking the Amazon MWAA environment CloudWatch logs groups:**

Checks whether the required CloudWatch Log Groups for the Amazon MWAA environment exist. If they do not, the automation then checks CloudTrail to locate `CreateLogGroup` and `DeleteLogGroup` events.

- **Checking the Amazon MWAA environment Route Tables:**

  Checks whether the Amazon VPC route tables in the Amazon MWAA environment are properly configured.

- **Checking the Amazon MWAA environment Security Groups:**

  Checks if the Amazon MWAA environment Amazon VPC security groups are properly configured.

- **Checking the Amazon MWAA environment Network ACLs:**

  Checks whether the Amazon VPC security groups in the Amazon MWAA environment are properly configured.

- **Checking the Amazon MWAA environment Subnets:**

  Verifies whether the Amazon MWAA environment's subnets are private.

- **Checking the Amazon MWAA environment required endpoints connectivity:**

  Verifies whether the Amazon MWAA environment can access the required endpoints. For this purpose, the automation invokes the `AWSSupport-ConnectivityTroubleshooter` automation.

- **Checking the Amazon MWAA environment Amazon S3 bucket:**

  Checks whether the Amazon MWAA environment's Amazon S3 bucket has `Block Public Access` enabled and also reviews the account's Amazon S3 Block Public Access settings.

- **Checking the Amazon MWAA environment CloudWatch logs groups errors:**

  Parses the existing CloudWatch log groups of the Amazon MWAA environment to locate errors.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Neptune

Amazon Systems Manager Automation provides predefined runbooks for Amazon Neptune. For more information about runbooks, see Working with runbooks . For information about how to view runbook content, see View runbook content .

Topics

AWS-EnableNeptuneDbAuditLogsToCloudWatch

Description

The `AWS-EnableNeptuneDbAuditLogsToCloudWatch` runbook helps you send audit logs for an Amazon Neptune DB cluster to Amazon CloudWatch Logs.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- DbClusterResourceId

  Type: String

  Description: (Required) The resource ID of the Neptune DB cluster you want to enable audit logs for.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `neptune:DescribeDBCluster`
- `neptune:ModifyDBCluster`
- `rds:DescribeDBClusters`
- `rds:ModifyDBCluster`

Document Steps

- GetNeptuneDbClusterIdentifier (`aws:executeAwsApi`) - Returns the ID of the Neptune DB cluster.
- VerifyNeptuneDbEngine (`aws:assertAwsResourceProperty`) - Verifies the Neptune DB engine type is `neptune`.
- EnableNeptuneDbAuditLogs (`aws:executeAwsApi`) - Enables audit logs for the Neptune DB cluster to be sent CloudWatch Logs.
- VerifyNeptuneDbStatus (`aws:waitAwsResourceProperty`) - Verifies the Neptune DB cluster status is `available`.
- VerifyNeptuneDbAuditLogs (`aws:executeScript`) - Verifies that audit logs were successfully configured to send to CloudWatch Logs.

AWS-EnableNeptuneDbBackupRetentionPeriod

Description

The AWS-EnableNeptuneDbBackupRetentionPeriod runbook helps you enable automated backups with a backup retention period between 7 and 35 days for an Amazon Neptune DB cluster.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- DbClusterResourceId

  Type: String

  Description: (Required) The resource ID of the Neptune DB cluster you want to enable backups for.

- BackupRetentionPeriod

  Type: Integer

Valid values: 7-35

Description: (Required) The number of days backups are retained.

- PreferredBackupWindow

  Type: String

  Description: (Optional) A daily time period of at least 30 minutes when backups are made. The value must be in Universal Time Coordinated (UTC) and use the format: hh24:mm-hh24:mm. The backup retention period can't conflict with the preferred maintenance window.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:GetAutomationExecution
- ssm:StartAutomationExecution
- neptune:DescribeDBCluster
- neptune:ModifyDBCluster
- rds:DescribeDBClusters
- rds:ModifyDBCluster

Document Steps

- GetNeptuneDbClusterIdentifier (aws:executeAwsApi) - Returns the ID of the Neptune DB cluster.
- VerifyNeptuneDbEngine (aws:assertAwsResourceProperty) - Verifies the Neptune DB engine type is neptune.
- VerifyNeptuneDbStatus (aws:waitAwsResourceProperty) - Verifies the Neptune DB cluster status is available.
- ModifyNeptuneDbRetentionPeriod (aws:executeAwsApi) - Sets the retention period for the Neptune DB cluster.
- VerifyNeptuneDbBackupsEnabled (aws:executeScript) - Verifies the retention period and backup window were successfully set.

AWS-EnableNeptuneClusterDeletionProtection

Description

The AWS-EnableNeptuneClusterDeletionProtection runbook enables deletion protection for the Amazon Neptune cluster you specify.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DbClusterResourceId

  Type: String

  Description: (Required) The ID of the Neptune cluster you want to enable deletion protection on.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:GetAutomationExecution

- ssm:StartAutomationExecution
- neptune:DescribeDBCluster
- neptune:ModifyDBCluster
- rds:DescribeDBClusters
- rds:ModifyDBCluster

Document Steps

- GetNeptuneDbClusterIdentifier (aws:executeAwsApi) - Returns the ID of the Neptune DB cluster.
- VerifyNeptuneDbEngine (aws:assertAwsResourceProperty) - Verifies the engine type of the specified DB cluster is neptune.
- VerifyNeptuneStatus (aws:waitForAwsResourceProperty) - Verifies that status of the cluster is available.
- EnableNeptuneDbDeletionProtection (aws:executeAwsApi) - Enables deletion protection on the Neptune DB cluster.
- VerifyNeptuneDbDeletionProtection (aws:assertAwsResourceProperty) - Verifies deletion protection is enabled on the DB cluster.

Outputs

- EnableNeptuneDbDeletionProtection.EnableNeptuneDbDeletionProtectionResponse - The output from the API operation.

Amazon RDS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Relational Database Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-CreateEncryptedRdsSnapshot
- AWS-CreateRdsSnapshot
- AWSConfigRemediation-DeleteRDSCluster

- AWSConfigRemediation-DeleteRDSClusterSnapshot
- AWSConfigRemediation-DeleteRDSInstance
- AWSConfigRemediation-DeleteRDSInstanceSnapshot
- AWSConfigRemediation-DisablePublicAccessToRDSInstance
- AWSConfigRemediation-EnableCopyTagsToSnapshotOnRDSCluster
- AWSConfigRemediation-EnableCopyTagsToSnapshotOnRDSDBInstance
- AWSConfigRemediation-EnableEnhancedMonitoringOnRDSInstance
- AWSConfigRemediation-EnableMinorVersionUpgradeOnRDS
- AWSConfigRemediation-EnableMultiAZOnRDSInstance
- AWSConfigRemediation-EnablePerformanceInsightsOnRDSInstance
- AWSConfigRemediation-EnableRDSClusterDeletionProtection
- AWSConfigRemediation-EnableRDSInstanceBackup
- AWSConfigRemediation-EnableRDSInstanceDeletionProtection
- AWSConfigRemediation-ModifyRDSInstancePortNumber
- AWSSupport-ModifyRDSSnapshotPermission
- AWSPremiumSupport-PostgreSQLWorkloadReview
- AWS-RebootRdsInstance
- AWSSupport-ShareRDSSnapshot
- AWS-StartRdsInstance
- AWS-StartStopAuroraCluster
- AWS-StopRdsInstance
- AWSSupport-TroubleshootConnectivityToRDS
- AWSSupport-TroubleshootRDSIAMAuthentication
- AWSSupport-ValidateRdsNetworkConfiguration

AWS-CreateEncryptedRdsSnapshot

Description

The `AWS-CreateEncryptedRdsSnapshot` runbook creates an encrypted snapshot from an unencrypted Amazon Relational Database Service (Amazon RDS) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- DBInstanceIdentifier

  Type: String

  Description: (Required) The ID of the Amazon RDS instance you want to create a snapshot of.

- DBSnapshotIdentifier

  Type: String

  Description: (Optional) The name template for the Amazon RDS snapshot. The default name template is *`DBInstanceIdentifier-yyyymmddhhmmss`*.

- EncryptedDBSnapshotIdentifier

  Type: String

  Description: (Optional) The name for the encrypted snapshot. The default name is the value you specify for the `DBSnapshotIdentifier` parameter appended with `-encrypted`.

- InstanceTags

  Type: String

  Description: (Optional) Tags to add to the DB instance. (Example: Key=tagKey1,Value=tagValue1;Key=tagKey2,Value=tagValue2)'
- KmsKeyId

  Type: String

  Default: `alias/aws/rds`

  Description: (Optional) The ARN, key ID, or the key alias of the of the customer managed key you want to use to encrypt the snapshot.
- SnapshotTags

  Type: String

  Description: (Optional) Tags to add to the snapshot. (Example: Key=tagKey1,Value=tagValue1;Key=tagKey2,Value=tagValue2)'

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `rds:AddTagsToResource`
- `rds:CopyDBSnapshot`
- `rds:CreateDBSnapshot`
- `rds:DeleteDBSnapshot`
- `rds:DescribeDBSnapshots`

Document Steps

- `aws:executeScript` - Creates a snapshot of the DB instance you specify in the `DBInstanceIdentifier` parameter.
- `aws:executeScript` - Verifies the snapshot created in the previous step exists and is `available`.

- `aws:executeScript` - Copies the previously created snapshot to an encrypted snapshot.
- `aws:executeScript` - Verifies the encrypted snapshot created in the previous step exists.

Outputs

CopyRdsSnapshotToEncryptedRdsSnapshot.EncryptedSnapshotId - The ID of the encrypted Amazon RDS snapshot.

AWS-CreateRdsSnapshot

Description

Create an Amazon Relational Database Service (Amazon RDS) snapshot for an Amazon RDS instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DBInstanceIdentifier

  Type: String

Description: (Required) The DBInstanceId ID of the RDS Instance to create Snapshot from.

- DBSnapshotIdentifier

  Type: String

  Description: (Optional) The DBSnapshotIdentifier ID of the RDS snapshot to create.

- InstanceTags

  Type: String

  Description: (Optional) Tags to create for instance.

- SnapshotTags

  Type: String

  Description: (Optional) Tags to create for snapshot.

Document Steps

createRDSSnapshot – Creates the RDS snapshot and returns the snapshot ID.

verifyRDSSnapshot – Checks that the snapshot created in the previous step exists.

Outputs

createRDSSnapshot.SnapshotId – The ID of the created snapshot.

AWSConfigRemediation-DeleteRDSCluster

Description

The `AWSConfigRemediation-DeleteRDSCluster` runbook deletes the Amazon Relational Database Service (Amazon RDS) cluster you specify. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DBClusterId

  Type: String

  Description: (Required) The resource identifier for the DB cluster you want to enable deletion protection on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `rds:DeleteDBCluster`
- `rds:DeleteDBInstance`
- `rds:DescribeDBClusters`

Document Steps

- `aws:executeScript` - Deletes the DB cluster you specify in the `DBClusterId` parameter.

AWSConfigRemediation-DeleteRDSClusterSnapshot

Description

The AWSConfigRemediation-DeleteRDSClusterSnapshot runbook deletes the given Amazon Relational Database Service (Amazon RDS) cluster snapshot.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DBClusterSnapshotId

  Type: String

  Description: (Required) The Amazon RDS cluster snapshot identifier to be deleted.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution

- ssm:GetAutomationExecution
- rds:DeleteDBClusterSnapshot
- rds:DescribeDBClusterSnapshots

Document Steps

- aws:branch - Checks if the cluster snapshot is in the available state. If it is not available, the flow ends.
- aws:executeAwsApi - Deletes the given Amazon RDS cluster snapshot using the database (DB) cluster snapshot identifier.
- aws:executeScript - Verifies that the given Amazon RDS cluster snapshot was deleted.

AWSConfigRemediation-DeleteRDSInstance

Description

The AWSConfigRemediation-DeleteRDSInstance runbook deletes the Amazon Relational Database Service (Amazon RDS) instance you specify. When you delete a database (DB) instance, all automated backups for that instance are deleted and can't be recovered. Manual DB snapshots are not deleted. If the DB instance you want to delete is in the failed , incompatible-network , or incompatible-restore state, you must set the SkipFinalSnapshot parameter to true .

> **Note**
>
> If the DB instance you want to delete is in an Amazon Aurora DB cluster, the runbook will not delete the DB instance if it is a read replica and the only instance in the DB cluster.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DbiResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to delete.
- SkipFinalSnapshot

  Type: Boolean

  Default: false

  Description: (Optional) If set to `true` , a final snapshot is not created before the DB instance is deleted.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DeleteDBInstance`
- `rds:DescribeDBInstances`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance name from the value you specify in the `DbiResourceId` parameter.

- aws:branch - Branches based on the value you specify in the SkipFinalSnapshot parameter.
- aws:executeAwsApi - Deletes the DB instance you specify in the DbiResourceId parameter.
- aws:executeAwsApi - Deletes the DB instance you specify in the DbiResourceId parameter after the final snapshot is created.
- aws:assertAwsResourceProperty - Verifies the DB instance was deleted.

AWSConfigRemediation-DeleteRDSInstanceSnapshot

Description

The AWSConfigRemediation-DeleteRDSInstanceSnapshot runbook deletes the Amazon Relational Database Service (Amazon RDS) instance snapshot you specify. Only snapshots in the available state are deleted. This runbook does not support deleting snapshots from Amazon Aurora database instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DbSnapshotId

Type: String

Description: (Required) The ID of the snapshot you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DeleteDBSnapshot`
- `rds:DescribeDBSnapshots`

Document Steps

- `aws:executeAwsApi` - Gathers the state of the snapshot specified in the `DbSnapshotId` parameter.
- `aws:assertAwsResourceProperty` - Confirms the state of the snapshot is `available` .
- `aws:executeAwsApi` - Deletes the snapshot specified in the `DbSnapshotId` parameter.
- `aws:executeScript` - Verifies the snapshot has been deleted.

AWSConfigRemediation-DisablePublicAccessToRDSInstance

Description

The `AWSConfigRemediation-DisablePublicAccessToRDSInstance` runbook disables public accessibility for the Amazon Relational Database Service (Amazon RDS) database (DB) instance that you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DbiResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance that you want to disable public accessibility for.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance identifier from the DB instance resource identifier.
- `aws:assertAwsResourceProperty` - Verifies the DB instances is in an `AVAILABLE` state.
- `aws:executeAwsApi` - Disables public accessibility on your DB instance.

- `aws:waitForAwsResourceProperty` - Waits for the DB instance to change to a `MODIFYING` state.
- `aws:waitForAwsResourceProperty` - Waits for the DB instance to change to an `AVAILABLE` state.
- `aws:assertAwsResourceProperty` - Confirms public accessibility is disabled on the DB instance.

AWSConfigRemediation-EnableCopyTagsToSnapshotOnRDSCluster

Description

The `AWSConfigRemediation-EnableCopyTagsToSnapshotOnRDSCluster` runbook enables the `CopyTagsToSnapshot` setting on the Amazon Relational Database Service (Amazon RDS) cluster you specify. Enabling this setting copies all tags from the DB cluster to snapshots of the DB cluster. The default is not to copy them. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- ApplyImmediately

  Type: Boolean

  Default: false

Description: (Optional) If you specify `true` for this parameter, the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the DB cluster.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- DbClusterResourceId

  Type: String

  Description: (Required) The resource identifier for the DB cluster you want to enable the `CopyTagsToSnapshot` setting on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `rds:DescribeDBClusters`
- `rds:ModifyDBCluster`

Document Steps

- `aws:executeAwsApi` - Gathers the DB cluster identifier from the DB cluster resource identifier.
- `aws:assertAwsResourceProperty` - Confirms the DB cluster is in an `AVAILABLE` state.
- `aws:executeAwsApi` - Enables the `CopyTagsToSnapshot` setting on your DB cluster.
- `aws:assertAwsResourceProperty` - Confirms the `CopyTagsToSnapshot` setting is enabled on your DB cluster.

AWSConfigRemediation-EnableCopyTagsToSnapshotOnRDSDBInstance

Description

The AWSConfigRemediation-EnableCopyTagsToSnapshotOnRDSDBInstance runbook enables the CopyTagsToSnapshot setting on the Amazon Relational Database Service (Amazon RDS) instance you specify. Enabling this setting copies all tags from the DB instance to snapshots of the DB instance. The default is not to copy them. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- ApplyImmediately

  Type: Boolean

  Default: false

  Description: (Optional) If you specify `true` for this parameter, the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the PreferredMaintenanceWindow setting for the DB instance.

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- DbiResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to enable the `CopyTagsToSnapshot` setting on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance identifier from the DB instance resource identifier.
- `aws:assertAwsResourceProperty` - Confirms the DB instance is in an `AVAILABLE` state.
- `aws:executeAwsApi` - Enables the `CopyTagsToSnapshot` setting on your DB instance.
- `aws:assertAwsResourceProperty` - Confirms the `CopyTagsToSnapshot` setting is enabled on your DB instance.

AWSConfigRemediation-EnableEnhancedMonitoringOnRDSInstance

Description

The `AWSConfigRemediation-EnableEnhancedMonitoringOnRDSInstance` runbook enables Enhanced Monitoring on the Amazon RDS database instance you specify. For information on Enhanced Monitoring, see Enhanced Monitoring in the *Amazon RDS User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- MonitoringInterval

  Type: Integer

  Valid values: 1 | 5 | 10 | 15 | 30 | 60

  Description: (Required) The interval in seconds when Enhanced Monitoring metrics are collected from the DB instance.

- MonitoringRoleArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the IAM role that allows Amazon RDS to send Enhanced Monitoring metrics to Amazon CloudWatch Logs.

- ResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to enable Enhanced Monitoring on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance identifier from the DB instance resource identifier.
- `aws:assertAwsResourceProperty` - Confirms the DB Instance is in an `AVAILABLE` state.
- `aws:executeAwsApi` - Enables Enhanced Monitoring on your DB instance.
- `aws:executeScript` - Confirms that Enhanced Monitoring is enabled on your DB instance.

AWSConfigRemediation-EnableMinorVersionUpgradeOnRDS

Description

The `AWSConfigRemediation-EnableMinorVersionUpgradeOnRDS` runbook enables the `AutoMinorVersionUpgrade` setting on the Amazon RDS database instance you specify. Enabling this setting means that minor version upgrades are applied automatically to the DB instance during the maintenance window.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DbiResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to the `AutoMinorVersionUpgrade` setting on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance identifier from the DB instance resource identifier.

- aws:assertAwsResourceProperty - Confirms the DB Instance is in an AVAILABLE state.
- aws:executeAwsApi - Enables the AutoMinorVersionUpgrade setting on your DB instance.
- aws:executeScript - Confirms that the AutoMinorVersionUpgrade setting is enabled on your DB instance.

AWSConfigRemediation-EnableMultiAZOnRDSInstance

Description

The AWSConfigRemediation-EnableMultiAZOnRDSInstance runbook changes your Amazon Relational Database Service (Amazon RDS) database (DB) instance to a Multi-AZ deployment. Changing this setting doesn't result in an outage. The change is applied during the next maintenance window unless you set the ApplyImmediately parameter to true .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- ApplyImmediately

  Type: Boolean

  Default: false

  Description: (Optional) If you specify true for this parameter, the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the PreferredMaintenanceWindow setting for the DB instance.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- DbiResourceId

  Type: String

  Description: (Required) The Amazon Web Services Region-unique, immutable identifier for the DB instance to enable the `MultiAZ` setting.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`
- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`

Document Steps

- `aws:executeAwsApi` - Retrieves the DB instance name using the value provided in the `DBInstanceId` parameter.
- `aws:executeAwsApi` - Verifies the `DBInstanceStatus` is `available` .
- `aws:branch` - Checks whether the `MultiAZ` is already set to `true` on the DB instance you specify in the `DbiResourceId` parameter.
- `aws:executeAwsApi` - Changes the `MultiAZ` setting to `true` on the DB instance you specify in the `DbiResourceId` parameter.
- `aws:assertAwsResourceProperty` - Verifies the `MultiAZ` is set to `true` on the DB instance you specify in the `DbiResourceId` parameter.

AWSConfigRemediation-EnablePerformanceInsightsOnRDSInstance

Description

The AWSConfigRemediation-EnablePerformanceInsightsOnRDSInstance runbook enables Performance Insights on the Amazon RDS DB instance you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DbiResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to enable Performance Insights on.
- PerformanceInsightsKMSKeyId

  Type: String

  Default: alias/aws/rds

Description: (Optional) The Amazon Resource Name (ARN), key ID, or the key alias of the Amazon Key Management Service (Amazon KMS) customer managed key you want Performance Insights to use to encrypt all potentially sensitive data. If you enter the key alias for this parameter, prefix the value with **`alias/`** . If you do not specify a value for this parameter, the Amazon managed key is used.

- PerformanceInsightsRetentionPeriod

  Type: Integer

  Valid values: 7, 731

  Default: 7

  Description: (Optional) The number of days you want to retain Performance Insights data.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `kms:CreateGrant`
- `kms:DescribeKey`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance identifier from the DB instance resource identifier.
- `aws:assertAwsResourceProperty` - Confirms the DB instance status is `available` .
- `aws:executeAwsApi` - Gathers the ARN of the Amazon KMS customer managed key specified in the `PerformanceInsightsKMSKeyId` parameter.
- `aws:branch` - Checks whether a value is already assigned to the `PerformanceInsightsKMSKeyId` property of the DB instance.

- `aws:executeAwsApi` - Enables Performance Insights on the DB instance you specify in the `DbiResourceId` parameter.
- `aws:assertAwsResourceProperty` - Confirms the value specified for the `PerformanceInsightsKMSKeyId` parameter was used to enable encryption for Performance Insights on the DB instance.
- `aws:assertAwsResourceProperty` - Confirms Performance Insights is enabled on the DB instance.

AWSConfigRemediation-EnableRDSClusterDeletionProtection

Description

The `AWSConfigRemediation-EnableRDSClusterDeletionProtection` runbook enables deletion protection on the Amazon Relational Database Service (Amazon RDS) cluster you specify. Amazon Config must be enabled in the Amazon Web Services Region where you run this automation.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ClusterId

Type: String

Description: (Required) The resource identifier for the DB cluster you want to enable deletion protection on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `config:GetResourceConfigHistory`
- `rds:DescribeDBClusters`
- `rds:ModifyDBCluster`

Document Steps

- `aws:executeAwsApi` - Gathers the DB cluster name from the DB cluster resource identifier.
- `aws:assertAwsResourceProperty` - Verifies the DB cluster status is `available` .
- `aws:executeAwsApi` - Enables deletion protection on the DB cluster you specify in the `ClusterId` parameter.
- `aws:assertAwsResourceProperty` - Verifies deletion protection has been enabled on the DB cluster.

AWSConfigRemediation-EnableRDSInstanceBackup

Description

The `AWSConfigRemediation-EnableRDSInstanceBackup` runbook enables backups for the Amazon Relational Database Service (Amazon RDS) database instance you specify. This runbook does not support enabling backups for Amazon Aurora database instances.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- ApplyImmediately

  Type: Boolean

  Default: false

  Description: (Optional) If you specify `true` for this parameter, the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the DB instance.
- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- BackupRetentionPeriod

  Type: Integer

  Valid values: 1-35

  Description: (Required) The number of days that backups are retained.
- DbiResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to enable backups for.
- PreferredBackupWindow

Type: String

Description: (Optional) The daily time range (in UTC) during which backups are created.

Constraints:

- Must be in the format hh24:mi-hh24:mi
- Must be in Coordinated Universal Time (UTC)
- Must not conflict with the preferred maintenance window
- Must be at least 30 minutes

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeScript` - Gathers the DB instance identifier from the DB instance resource identifier. Enables backups for your DB instance. Confirms backups are enabled on the DB instance.

AWSConfigRemediation-EnableRDSInstanceDeletionProtection

Description

The `AWSConfigRemediation-EnableRDSInstanceDeletionProtection` runbook enables deletion protection on the Amazon RDS database instance you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- ApplyImmediately

  Type: Boolean

  Default: false

  Description: (Optional) If you specify `true` for this parameter, the modifications in this request and any pending modifications are asynchronously applied as soon as possible, regardless of the `PreferredMaintenanceWindow` setting for the DB instance.
- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DbInstanceResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance you want to enable deletion protection on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`

- ssm:GetAutomationExecution
- rds:DescribeDBInstances
- rds:ModifyDBInstance

Document Steps

- aws:executeAwsApi - Gathers the DB instance identifier from the DB instance resource identifier.
- aws:executeAwsApi - Enables deletion protection on your DB instance.
- aws:assertAwsResourceProperty - Confirms deletion protection is enabled on the DB instance.

AWSConfigRemediation-ModifyRDSInstancePortNumber

Description

The AWSConfigRemediation-ModifyRDSInstancePortNumber runbook modifies the port number on which the Amazon Relational Database Service (Amazon RDS) instance accepts connections. Running this automation will restart the database.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- PortNumber

  Type: String

  Description: (Optional) The port number you want the DB instance to accept connections on.

- RDSDBInstanceResourceId

  Type: String

  Description: (Required) The resource identifier for the DB instance whose inbound port number you want to modify.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DescribeDBInstances`
- `rds:ModifyDBInstance`

Document Steps

- `aws:executeAwsApi` - Gathers the DB instance identifier from the DB instance resource identifier.
- `aws:assertAwsResourceProperty` - Confirms the DB Instance is in an `AVAILABLE` state.
- `aws:executeAwsApi` - Modifies the inbound port number on which your DB instance accepts connections.
- `aws:waitForAwsResourceProperty` - Waits for the DB Instance to be in a `MODIFYING` state.
- `aws:waitForAwsResourceProperty` - Waits for the DB Instance to be in in an `AVAILABLE` state.

AWSSupport-ModifyRDSSnapshotPermission

Description

The AWSSupport-ModifyRDSSnapshotPermission runbook helps you modify permissions for multiple Amazon Relational Database Service (Amazon RDS) snapshots. Using this runbook, you can make snapshots Public or Private and share them with other Amazon Web Services accounts. Snapshots encrypted with a default KMS key can't be shared with other accounts using this runbook.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AccountIds

  Type: StringList

  Default: none

  Description: (Optional) The IDs of the accounts you want to share snapshots with. This parameter is required if you enter No for the value of the Private parameter.

- AccountPermissionOperation

  Type: String

  Valid values: add | remove

  Default: none

  Description: (Optional) The type of operation to perform.
- Private

  Type: String

  Valid values: Yes | No

  Description: (Required) Enter No for the value if you want to share snapshots with specific accounts.
- SnapshotIdentifiers

  Type: StringList

  Description: (Required) The names of the Amazon RDS snapshots whose permission you want to modify.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `rds:DescribeDBSnapshots`
- `rds:ModifyDBSnapshotAttribute`

Document Steps

1. `aws:executeScript` - Verifies the IDs of the snapshots provided in the `SnapshotIdentifiers` parameter. After verifying the IDs, the script checks for encrypted snapshots and outputs a list if any are found.

2. aws:branch - Branches the automation based on the value you enter for the Private parameter.
3. aws:executeScript - Modifies permissions of the snapshots specified to share it with the accounts specified.
4. aws:executeScript - Modifies permissions of the snapshots to change them from Public to Private .

Outputs

ValidateSnapshots.EncryptedSnapshots

SharewithOtherAccounts.Result

MakePrivate.Result

MakePrivate.Commands

AWSPremiumSupport-PostgreSQLWorkloadReview

Description

The AWSPremiumSupport-PostgreSQLWorkloadReview runbook captures multiple snapshots of your Amazon Relational Database Service (Amazon RDS) PostgreSQL database usage statistics. The statistics captured are required for an Amazon Web Services Support Proactive Services expert to perform an operational review. The statistics are collected using a set of custom SQL and shell scripts. These scripts are downloaded to a temporary Amazon Elastic Compute Cloud (Amazon EC2) instance in your Amazon Web Services account that is created by this runbook. The runbook requires you to provide credentials using an Amazon Secrets Manager secret containing a username and password key-value pair. The username must have permissions to query the standard PostgreSQL statistics views and functions.

This runbook automatically creates the following Amazon resources in your Amazon Web Services account using an Amazon CloudFormation stack. You can monitor the stack creation using the Amazon CloudFormation console.

- A virtual private cloud (VPC) and an Amazon EC2 instance launched in a private subnet of the VPC with optional connectivity to the internet using a NAT gateway.
- An Amazon Identity and Access Management (IAM) role that is attached to the temporary Amazon EC2 instance with permissions to retrieve the Secrets Manager secret value. The role also

provides permissions to upload files to an Amazon Simple Storage Service (Amazon S3) bucket of your choice, and optionally to an Amazon Web Services Support case.

- A VPC peering connection to allow connectivity between your DB instance and the temporary Amazon EC2 instance.
- Systems Manager, Secrets Manager, and Amazon S3 VPC endpoints that are attached to the temporary VPC.
- A maintenance window with registered tasks that periodically start and stop the temporary Amazon EC2 instance, run data collection scripts, and upload files to an Amazon S3 bucket. An IAM role is also created for the maintenance window that provides permissions to perform the registered tasks.

When the runbook completes, the Amazon CloudFormation stack that is used to create the necessary Amazon resources is deleted and the report is uploaded to the Amazon S3 bucket of your choice, and optionally an Amazon Web Services Support case.

> **Note**
>
> By default, the root Amazon EBS volume of the temporary Amazon EC2 instance is preserved. You can override this option by setting the `EbsVolumeDeleteOnTermination` parameter to `true`.

Prerequisites

- **Enterprise Support subscription** This runbook and the Proactive Services Workload Diagnostics and Reviews require an Enterprise Support Subscription. Before using this runbook, contact your Technical Account Manager (TAM) or Specialist TAM (STAM) for instructions. For more information, see Amazon Web Services Support Proactive Services.
- **Account and Amazon Web Services Region quotas** Be sure you have not reached the maximum number of Amazon EC2 instances or VPCs that you can create in your account and Region where you use this runbook. If you need to request a limit increase, see the Service limit increase form.
- **Database configuration**
  1. The database you specify in the `DatabaseName` parameter should have the `pg_stat_statements` extension configured. If you have not configured `pg_stat_statements` in `shared_preload_libraries`, then you must edit the value in the DB Parameter Group and apply the changes. Changes to the parameter

`shared_preload_libraries` requires you to reboot your DB instance. For more information, see Working with parameter groups. Adding `pg_stat_statements` to `shared_preload_libraries` will add some performance overhead. However, this is useful for tracking performance of individual statements. For more information about the `pg_stat_statements` extension, see the PostgreSQL documentation. If you don't configure the `pg_stat_statements` extension or if the extension is not present in the database being used for statistics collection, the statement level analysis will not be presented in the operational review.

2. Make sure that `track_counts` and `track_activities` parameters are not turned off. If these parameters are turned off in the DB Parameter Group, no meaningful statistics will be available. Changing these parameters will require you to reboot your DB instance. For more information, see Working with parameters on your Amazon RDS for PostgreSQL DB instance.
3. If the `track_io_timing` parameter is turned off, the I/O level statistics will not be included in the operational review. Changing `track_io_timing` will require you to reboot your DB instance and will incur additional performance overhead depending on the DB instance workload. Despite the performance overhead for critical workloads, this parameter provides useful information related to I/O time per query.

Billing and charges Your Amazon Web Services account will be charged for the costs associated with the temporary Amazon EC2 instance, associated Amazon EBS volume, the NAT gateway, and the data transferred while this automation is running. By default, this runbook creates a `t3.micro` Amazon Linux 2 instance to collect the statistics. The runbook starts and stops the instance between steps to reduce costs.

Data security and governance This runbook collects statistics by querying the PostgreSQL statistics views and functions. Make sure the credentials provided in the `SecretId` parameter only allow read-only permissions to the statistics views and functions. As part of the automation, the collection scripts are uploaded to your Amazon S3 bucket and can be located in `s3://`*`amzn-s3-demo-bucket`*`/`*`automation execution id`*`/queries/`.

These scripts collect data that is used by an Amazon Specialist to review key performance indicators at object level. The script collects information such as table name, schema name, and index name. If any of this information contains sensitive information like revenue indicators, username, email address, or any other personally identifiable information, then we recommend that you discontinue with this workload review. Contact your Amazon TAM to discuss an alternative approach for the workload review.

Make sure you have the necessary approval and clearance to share the statistics and metadata collected by this automation with Amazon.

Security considerations If you set the `UpdateRdsSecurityGroup` parameter to `yes`, the runbook updates the security group associated with your DB instance to allow inbound traffic from the temporary Amazon EC2 instance's private IP address.

If you set the `UpdateRdsRouteTable` parameter to `yes`, the runbook updates the route table associated with the subnet your DB instance is running in to allow traffic to the temporary Amazon EC2 instance through the VPC peering connection.

User creation To allow the collection script to connect to your Amazon RDS database, you must set up a user with permissions to read the statistic views. Then you must store the credentials in Secrets Manager. We recommend creating a new dedicated user for this automation. Creating a separate user allows you to audit and track activities performed by this automation.

1. Create a new user.

```
psql -h <database_connection_endpoint> -p <database_port> -U <admin_user>
-c "CREATE USER <user_name> PASSWORD '<password>';"
```

2. Ensure that this user can only make read-only connections.

```
psql -h <database_connection_endpoint> -p <database_port> -U <admin_user>
-c "ALTER USER <user_name> SET default_transaction_read_only=true;"
```

3. Set user level limits.

```
psql -h <database_connection_endpoint> -p <database_port> -U <admin_user>
-c "ALTER USER <user_name> SET work_mem=4096;"

psql -h <database_connection_endpoint> -p <database_port> -U <admin_user>
-c "ALTER USER <user_name> SET statement_timeout=10000;"

psql -h <database_connection_endpoint> -p <database_port>
-U <admin_user> -c "ALTER USER <user_name> SET
idle_in_transaction_session_timeout=60000;"
```

4. Grant `pg_monitor` permissions to the new user so it can access the DB statistics. (The `pg_monitor` role is a member of `pg_read_all_settings`, `pg_read_all_stats`, and `pg_stat_scan_table`.)

```
psql -h <database_connection_endpoint> -p <database_port> -U <admin_user>
-c "GRANT pg_monitor to <user_name>;"
```

Permissions added to the temporary Amazon EC2 instance profile by this Systems Manager Automation The following permissions are added to the IAM role associated with the temporary Amazon EC2 instance. The `AmazonSSMManagedInstanceCore` managed policy is also associated with the IAM role to allow the Amazon EC2 instance to be managed by Systems Manager.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeTags"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/automation execution id/*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:region:account id:secret:secret id",
            "Effect": "Allow"
        },
        {
            "Action": [
```

```
                "support:AddAttachmentsToSet",
                "support:AddCommunicationToCase",
                "support:DescribeCases"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Permissions added to the temporary maintenance window by this Systems Manager Automation The following permissions are automatically added to the IAM role associated with the Maintenance Windows tasks. The Maintenance Windows tasks starts, stops, and sends commands to the temporary Amazon EC2 instance.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ssm:GetAutomationExecution",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations",
                "ssm:GetCommandInvocation",
                "ssm:GetCalendarState",
                "ssm:CancelCommand",
                "ec2:DescribeInstanceStatus"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "ssm:SendCommand",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ssm:StartAutomationExecution"
            ],
            "Resource": [
                "arn:aws:ec2:region:account id:instance/temporary instance id",
                "arn:aws:ssm:*:*:document/AWS-RunShellScript",
                "arn:aws:ssm:*:*:automation-definition/AWS-StopEC2Instance:$DEFAULT",
                "arn:aws:ssm:*:*:automation-definition/AWS-StartEC2Instance:$DEFAULT"
```

```
            ],
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "ssm.amazonaws.com"
                }
            },
            "Action": "iam:PassRole",
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DBInstanceIdentifier

  Type: String

Description: (Required) The ID of your DB instance.

- DatabaseName

  Type: String

  Description: (Required) The database name hosted on your DB instance.

- SecretId

  Type: String

  Description: (Required) The ARN of your Secrets Manager secret containing the username and password key value pair. The Amazon CloudFormation stack creates an IAM policy with permissions for the `GetSecretValue` operation to this ARN. The credentials are used to allow the temporary instance to collect the database statistics. Contact your TAM or STAM to discuss the minimum required permissions.

- Acknowledge

  Type: String

  Description: (Required) Enter **`yes`** if you acknowledge that this runbook will create temporary resources in your account to collect statistics from your DB instance. We recommend contacting your TAM or STAM before running this automation.

- SupportCase

  Type: String

  Description: (Optional) The Amazon Web Services Support case number provided by your TAM or STAM. If provided, the runbook updates the case and attaches the data collected. This option requires the temporary Amazon EC2 instance to have internet connectivity to access the Amazon Web Services Support API endpoint. You must set the `AllowVpcInternetAccess` parameter to `true`. The case subject must contain the phrase `AWSPremiumSupport-PostgreSQLWorkloadReview`.

- S3BucketName

  Type: String

  Description: (Required) The Amazon S3 bucket name in your account where you want to upload the data collected by the automation. Verify the bucket policy does not grant any unnecessary

read or write permissions to principals that do not need access to the contents of the bucket. We recommend creating a new temporary Amazon S3 bucket for the purpose of this automation. The runbook provides permissions to the `s3:PutObject` API operation to the IAM role attached to the temporary Amazon EC2 instance. The uploaded files will be located in `s3://`*`bucket name`*`/`*`automation execution id`*`/`.

- InstanceType

  Type: String

  Description: (Optional) The type of the temporary Amazon EC2 instance that will run the custom SQL and shell scripts.

  Valid values: t2.micro | t2.small | t2.medium | t2.large | t3.micro | t3.small | t3.medium | t3.large

  Default: t3.micro

- VpcCidr

  Type: String

  Description: (Optional) The IP address range in CIDR notation for the new VPC (for example, `172.31.0.0/16`). Make sure you select a CIDR that does not overlap or match any existing VPC with connectivity to your DB instance. The smallest VPC you can create uses a /28 subnet mask, and the largest VPC uses a /16 subnet mask.

  Default: 172.31.0.0/16

- StackResourcesNamePrefix

  Type: String

  Description: (Optional) The Amazon CloudFormation stack resources name prefix and tag. The runbook creates the Amazon CloudFormation stack resources using this prefix as part of the name and tag applied to the resources. The structure for the tag key-value pair is *`StackResourcesNamePrefix`*`:{{automation:EXECUTION_ID}}`.

  Default: AWSPostgreSQLWorkloadReview

- Schedule

  Type: String

Description: (Optional) The maintenance window schedule. Specifies how often the maintenance window runs the tasks. The default value is every `1 hour`.

Valid values: 15 minutes | 30 minutes | 1 hour | 2 hours | 4 hours | 6 hours | 12 hours | 1 day | 2 days | 4 days

Default: 1 hour

- Duration

  Type: Integer

  Description: (Optional) The maximum duration, in minutes, you want to allow the automation to run. The maximum duration supported is 8,640 minutes (6 days). The default value is 4,320 minutes (3 days).

  Valid values: 30-8640

  Default: 4320

- UpdateRdsRouteTable

  Type: String

  Description: (Optional) If set to `true`, the runbook updates the route table associated with the subnet your DB instance runs in. An IPv4 route is added to route traffic to the temporary Amazon EC2 instance private IPV4 address through the newly created VPC peering connection.

  Valid values: true | false

  Default: false

- AllowVpcInternetAccess

  Type: String

  Description: (Optional) If set to `true`, the runbook creates a NAT gateway to provide internet connectivity to the temporary Amazon EC2 instance to communicate with the Amazon Web Services Support API endpoint. You can leave this parameter as `false` if you only want the runbook to upload the output to your Amazon S3 bucket.

  Valid values: true | false

Default: false

- UpdateRdsSecurityGroup

  Type: String

  Description: (Optional) If set to `true`, the runbook updates the security group associated with your DB instance to allow traffic from the temporary instance's private IP address.

  Valid values: false | true

  Default: false
- EbsVolumeDeleteOnTermination

  Type: String

  Description: (Optional) If set to `true`, the temporary Amazon EC2 instance's root volume is deleted after the runbook completes and deletes the Amazon CloudFormation stack.

  Valid values: false | true

  Default: false

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `cloudformation:CreateStack`
- `cloudformation:DeleteStack`
- `cloudformation:DescribeStackEvents`
- `cloudformation:DescribeStackResource`
- `cloudformation:DescribeStacks`
- `cloudformation:UpdateStack`
- `ec2:AcceptVpcPeeringConnection`
- `ec2:AllocateAddress`
- `ec2:AssociateRouteTable`
- `ec2:AssociateVpcCidrBlock`

- ec2:AttachInternetGateway
- ec2:AuthorizeSecurityGroupEgress
- ec2:AuthorizeSecurityGroupIngress
- ec2:CreateEgressOnlyInternetGateway
- ec2:CreateInternetGateway
- ec2:CreateNatGateway
- ec2:CreateRoute
- ec2:CreateRouteTable
- ec2:CreateSecurityGroup
- ec2:CreateSubnet
- ec2:CreateTags
- ec2:CreateVpc
- ec2:CreateVpcEndpoint
- ec2:CreateVpcPeeringConnection
- ec2:DeleteEgressOnlyInternetGateway
- ec2:DeleteInternetGateway
- ec2:DeleteNatGateway
- ec2:DeleteRoute
- ec2:DeleteRouteTable
- ec2:DeleteSecurityGroup
- ec2:DeleteSubnet
- ec2:DeleteTags
- ec2:DeleteVpc
- ec2:DeleteVpcEndpoints
- ec2:DescribeAddresses
- ec2:DescribeEgressOnlyInternetGateways
- ec2:DescribeImages
- ec2:DescribeInstances
- ec2:DescribeInstanceStatus

- ec2:DescribeInternetGateways
- ec2:DescribeNatGateways
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcEndpoints
- ec2:DescribeVpcPeeringConnections
- ec2:DescribeVpcs
- ec2:DetachInternetGateway
- ec2:DisassociateRouteTable
- ec2:DisassociateVpcCidrBlock
- ec2:ModifySubnetAttribute
- ec2:ModifyVpcAttribute
- ec2:RebootInstances
- ec2:ReleaseAddress
- ec2:RevokeSecurityGroupEgress
- ec2:RevokeSecurityGroupIngress
- ec2:StartInstances
- ec2:StopInstances
- ec2:RunInstances
- ec2:TerminateInstances
- iam:AddRoleToInstanceProfile
- iam:AttachRolePolicy
- iam:CreateInstanceProfile
- iam:CreateRole
- iam:DeleteInstanceProfile
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:DetachRolePolicy

- iam:GetInstanceProfile
- iam:GetRole
- iam:GetRolePolicy
- iam:PassRole
- iam:PutRolePolicy
- iam:RemoveRoleFromInstanceProfile
- iam:TagPolicy
- iam:TagRole
- rds:DescribeDBInstances
- s3:GetAccountPublicAccessBlock
- s3:GetBucketAcl
- s3:GetBucketPolicyStatus
- s3:GetBucketPublicAccessBlock
- s3:ListBucket
- ssm:AddTagsToResource
- ssm:CancelMaintenanceWindowExecution
- ssm:CreateDocument
- ssm:CreateMaintenanceWindow
- ssm:DeleteDocument
- ssm:DeleteMaintenanceWindow
- ssm:DeregisterTaskFromMaintenanceWindow
- ssm:DescribeAutomationExecutions
- ssm:DescribeDocument
- ssm:DescribeInstanceInformation
- ssm:DescribeMaintenanceWindowExecutions
- ssm:GetCalendarState
- ssm:GetDocument
- ssm:GetMaintenanceWindowExecution
- ssm:GetParameters

- ssm:ListCommandInvocations
- ssm:ListCommands
- ssm:ListTagsForResource
- ssm:RegisterTaskWithMaintenanceWindow
- ssm:RemoveTagsFromResource
- ssm:SendCommand
- support:AddAttachmentsToSet
- support:AddCommunicationToCase
- support:DescribeCases

Document Steps

1. aws:assertAwsResourceProperty - Confirms the DB instance is in the available state.
2. aws:executeAwsApi - Gathers details about the DB instance.
3. aws:executeScript - Checks if the Amazon S3 bucket specified in the S3BucketName allows anonymous, or public read or write access permissions.
4. aws:executeScript - Gets the Amazon CloudFormation template content from the Automation runbook attachment that is used to create the temporary Amazon resources in your Amazon Web Services account.
5. aws:createStack - Creates the Amazon CloudFormation stack resources.
6. aws:waitForAwsResourceProperty - Waits until the Amazon EC2 instance created by the Amazon CloudFormation template is running.
7. aws:executeAwsApi - Gets the IDs for the temporary Amazon EC2 instance and VPC peering connection created by Amazon CloudFormation.
8. aws:executeAwsApi - Gets the IP address for the temporary Amazon EC2 instance to configure connectivity with your DB instance.
9. aws:executeAwsApi - Tags the Amazon EBS volume attached to the temporary Amazon EC2 instance.
10. aws:waitForAwsResourceProperty - Waits until the temporary Amazon EC2 instance passes status checks.
11. aws:waitForAwsResourceProperty - Waits until the temporary Amazon EC2 instance is managed by Systems Manager. If this step times out or fails, then the runbook reboots the instance.

a. `aws:executeAwsApi` - Reboots the temporary Amazon EC2 instance if the previous step failed or timed out.

b. `aws:waitForAwsResourceProperty` - Waits until the temporary Amazon EC2 instance is managed by Systems Manager after reboot.

12 `aws:runCommand` - Installs the metadata collector application requirements on the temporary Amazon EC2 instance.

13 `aws:runCommand` - Configures access to your DB instance by creating a configuration file on the temporary Amazon EC2 instance.

14 `aws:executeAwsApi` - Creates a maintenance window to periodically run the metadata collector application using Run Command. The maintenance window starts and stops the instance between commands.

15 `aws:waitForAwsResourceProperty` - Waits until the maintenance window created by the Amazon CloudFormation template is ready.

16 `aws:executeAwsApi` - Gets the IDs for the maintenance window and change calendar created by Amazon CloudFormation.

17 `aws:sleep` - Waits until the end date of the maintenance window.

18 `aws:executeAwsApi` - Turns off the maintenance window.

19 `aws:executeScript` - Gets the results of the tasks run during the maintenance window.

20 `aws:waitForAwsResourceProperty` - Waits for the maintenance window to finish the last task before continuing.

21 `aws:branch` - Branches the workflow based on whether you provided a value for the `SupportCase` parameter.

a. `aws:changeInstanceState` - Starts the temporary Amazon EC2 instance and waits for status checks to pass before uploading the report.

b. `aws:waitForAwsResourceProperty` - Waits until the temporary Amazon EC2 instance is managed by Systems Manager. If this step timeouts or fail, then the runbook reboots the instance.

i. `aws:executeAwsApi` - Reboots the temporary Amazon EC2 instance if the previous step failed or timed out.

ii. `aws:waitForAwsResourceProperty` - Waits until the temporary Amazon EC2 instance is managed by Systems Manager after reboot.

c. `aws:runCommand` - Attaches the metadata report to the Amazon Web Services Support case if you provided a value for the `SupportCase` parameter. The script compresses and splits the

report into 5 MB files. The maximum number of files the script attaches to a Amazon Web Services Support case is 12.

22 `aws:changeInstanceState` - Stops the temporary Amazon EC2 instance in case the Amazon CloudFormation stack fails to delete.

23 `aws:executeAwsApi` - Describes the Amazon CloudFormation stack events if the runbooks fails to create or update the Amazon CloudFormation stack.

24 `aws:waitForAwsResourceProperty` - Waits until the Amazon CloudFormation stack is in a terminal status before deleting.

25 `aws:executeAwsApi` - Deletes the Amazon CloudFormation stack excluding the maintenance window. The root Amazon EBS volume associated with the temporary Amazon EC2 instance is preserved if the `EbsVolumeDeleteOnTermination` parameter value was set to `false`.

AWS-RebootRdsInstance

Description

The `AWS-RebootRdsInstance` runbook reboots an Amazon Relational Database Service (Amazon RDS) DB instance if it isn't already rebooting.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon RDS DB instance that you want to reboot.

Document Steps

RebootInstance - Reboots the DB instance if it is not already rebooting.

WaitForAvailableState - Waits for the DB instance to complete the reboot process.

Outputs

This automation has no outputs.

AWSSupport-ShareRDSSnapshot

Description

The `AWSSupport-ShareRDSSnapshot` runbook provides an automated solution for the procedure outlined in the Knowledge Center article How can I share an encrypted Amazon RDS DB snapshot with another account? If your Amazon Relational Database Service (Amazon RDS) snapshot was encrypted using the default Amazon managed key, you cannot share the snapshot. In this case, you must copy the snapshot using a customer managed key, and then share the snapshot with the target account. This automation performs these steps using the value you specify in the `SnapshotName` parameter, or the latest snapshot found for the selected Amazon RDS DB instance or cluster.

> **Note**
>
> If you do not specify a value for the `KMSKey` parameter, the automation creates a new Amazon KMS customer managed key in your account that is used to encrypt the snapshot.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AccountIds

  Type: StringList

  Description: (Required) Comma-separated list of account IDs to share the snapshot with.
- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Database

  Type: String

  Description: (Required) The name of the Amazon RDS DB instance or cluster whose snapshot you want to share. This parameter is optional if you specify a value for the `SnapshotName` parameter.
- KMSKey

  Type: String

  Description: (Optional) The full Amazon Resource Name (ARN) of the Amazon KMS customer managed key used to encrypt the snapshot.

- SnapshotName

  Type: String

  Description: (Optional) The ID of the DB cluster or instance snapshot that you want to use.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `rds:DescribeDBInstances`
- `rds:DescribeDBSnapshots`
- `rds:CopyDBSnapshot`
- `rds:ModifyDBSnapshotAttribute`

The `AutomationAssumeRole` requires the following actions to successfully start the runbook for a DB cluster.

- `ssm:StartAutomationExecution`
- `rds:DescribeDBClusters`
- `rds:DescribeDBClusterSnapshots`
- `rds:CopyDBClusterSnapshot`
- `rds:ModifyDBClusterSnapshotAttribute`

The IAM role used to run the automation must be added as a key user to use the KMS key specified in the `ARNKmsKey` parameter. For information about adding key users to a KMS key, see Changing a key policy in the *Amazon Key Management Service Developer Guide* .

The `AutomationAssumeRole` requires the following additional actions to successfully start the runbook if you do not specify a value for the `KMSKey` parameter.

- `kms:CreateKey`
- `kms:ScheduleKeyDeletion`
- `kms:CreateGrant`

- kms:DescribeKey

Document Steps

1. aws:executeScript - Checks whether a value was provided for the KMSKey parameter, and creates a Amazon KMS customer managed key if no value is found.
2. aws:branch - Checks whether a value was provided for the SnapshotName parameter, and branches accordingly.
3. aws:executeAwsApi - Checks whether the snapshot provided is from a DB instance.
4. aws:executeScript - Formats the SnapshotName parameter replacing colons with a hyphen.
5. aws:executeAwsApi - Copies the snapshot using the specified KMSKey .
6. aws:waitForAwsResourceProperty - Waits for the copy snapshot operation to complete.
7. aws:executeAwsApi - Shares the new snapshot with the AccountIds specified.
8. aws:executeAwsApi - Checks whether the snapshot provided is from a DB cluster.
9. aws:executeScript - Formats the SnapshotName parameter replacing colons with a hyphen.
10. aws:executeAwsApi - Copies the snapshot using the specified KMSKey .
11. aws:waitForAwsResourceProperty - Waits for the copy snapshot operation to complete.
12. aws:executeAwsApi - Shares the new snapshot with the AccountIds specified.
13. aws:executeAwsApi - Checks whether the value provided for the Database parameter is a DB instance.
14. aws:executeAwsApi - Checks whether the value provided for the Database parameter is a DB cluster.
15. aws:executeAwsApi - Retrieves a list of snapshots for the specified Database .
16. aws:executeScript - Determines the latest snapshot available from the list assembled in the previous step.
17. aws:executeAwsApi - Copies the DB instance snapshot using the specified KMSKey .
18. aws:waitForAwsResourceProperty - Waits for the copy snapshot operation to complete.
19. aws:executeAwsApi - Shares the new snapshot with the AccountIds specified.
20. aws:executeAwsApi - Retrieves a list of snapshots for the specified Database .
21. aws:executeScript - Determines the latest snapshot available from the list assembled in the previous step.

22`aws:executeAwsApi` - Copies the DB instance snapshot using the specified `KMSKey` .

23`aws:waitForAwsResourceProperty` - Waits for the copy snapshot operation to complete.

24`aws:executeAwsApi` - Shares the new snapshot with the `AccountIds` specified.

25`aws:executeScript` - Deletes the Amazon KMS customer managed key created by the automation if you did not specify a value for the `KMSKey` parameter and the automation fails.

AWS-StartRdsInstance

Description

Start an Amazon Relational Database Service (Amazon RDS) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

Description: (Required) ID of the Amazon RDS instance to start.

AWS-StartStopAuroraCluster

Description

This runbook starts or stops an Amazon Aurora cluster.

> **Note**
>
> To start a cluster it must be in a `stopped` status. To stop a cluster it must be in an `available` status. This runbook can't be used to start or stop a cluster that is an Aurora Serverless cluster, an Aurora multi-master cluster, part of an Aurora global database, or a cluster that uses Aurora parallel query.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- ClusterName

  Type: String

  Description: (Required) The name of the Aurora cluster you want to stop or start.

- Action

  Type: String

  Valid values: Start | Stop

  Default: Start

  Description: (Required) The name of the Aurora cluster you want to stop or start.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `rds:DescribeDBClusters`
- `rds:StartDBCluster`
- `rds:StopDBCluster`

Document Steps

- `aws:executeScript` - Starts or stops the cluster based on the values you specify for the.

Outputs

StartStopAuroraCluster.ClusterName - The name of the Aurora cluster

StartStopAuroraCluster.CurrentStatus - The current status of the Aurora cluster

StartStopAuroraCluster.Message - Details of the automation

AWS-StopRdsInstance

Description

Stop an Amazon Relational Database Service (Amazon RDS) instance.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) ID of the Amazon RDS instance to stop.

AWSSupport-TroubleshootConnectivityToRDS

Description

The `AWSSupport-TroubleshootConnectivityToRDS` runbook diagnoses connectivity issues between an EC2 instance and an Amazon Relational Database Service instance. The automation ensures the DB instance is available, and then checks the associated security group rules, network access control lists (network ACLs), and route tables for potential connectivity issues.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DBInstanceIdentifier

  Type: String

  Description: (Required) The DB instance ID to test connectivity to.
- SourceInstance

  Type: String

  Allowed pattern: ^i-[a-z0-9]{8,17}$

  Description: (Required) The ID of the EC2 instance to test connectivity from.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`

- ec2:DescribeNetworkAcls
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- rds:DescribeDBInstances

Document Steps

- aws:assertAwsResourceProperty - Confirms the DB instance status is available .
- aws:executeAwsApi - Gets information about the DB instance.
- aws:executeAwsApi - Gets information about the DB instance network ACLs.
- aws:executeAwsApi - Gets the DB instance subnet CIDR.
- aws:executeAwsApi - Gets information about the EC2 instance.
- aws:executeAwsApi - Gets information about the EC2 instance network ACLs.
- aws:executeAwsApi - Gets information about the security groups associated with the EC2 instance.
- aws:executeAwsApi - Gets information about the security groups associated with the DB instance.
- aws:executeAwsApi - Gets information about the route tables associated with the EC2 instance.
- aws:executeAwsApi - Gets information about the main route table associated with the Amazon VPC for the EC2 instance.
- aws:executeAwsApi - Gets information about the route tables associated with the DB instance.
- aws:executeAwsApi - Gets information about the main route table associated with the Amazon VPC for the DB instance.
- aws:executeScript - Evaluates security group rules.
- aws:executeScript - Evaluates network ACLs.
- aws:executeScript - Evaluates route tables.
- aws:sleep - Ends the automation.

Outputs

getRDSInstanceProperties.DBInstanceIdentifier - The DB instance used in the automation.

getRDSInstanceProperties.DBInstanceStatus - The current status of the DBInstance.

evalSecurityGroupRules.SecurityGroupEvaluation - Results from comparing the `SourceInstance` security group rules to the DB instance security group rules.

evalNetworkAclRules.NetworkAclEvaluation - Results from comparing the `SourceInstance` network ACLs to the DB instance network ACLs.

evalRouteTableEntries.RouteTableEvaluation - Results from comparing the `SourceInstance` route table to the DB instance routes.

`AWSSupport-TroubleshootRDSIAMAuthentication`

Description

The `AWSSupport-TroubleshootRDSIAMAuthentication` helps troubleshoot Amazon Identity and Access Management (IAM) authentication for Amazon RDS for PostgreSQL, Amazon RDS for MySQL, Amazon RDS for MariaDB, Amazon Aurora PostgreSQL, and Amazon Aurora MySQL instances. Use this runbook to verify the configuration required for IAM authentication with an Amazon RDS instance or Aurora Cluster. It also provides steps to rectify the connectivity issues to the Amazon RDS Instance or Aurora Cluster.

> **Important**
>
> This runbook does not support Amazon RDS for Oracle or Amazon RDS for Microsoft SQL Server.

> **Important**
>
> If a source Amazon EC2 Instance is provided and the target Database is Amazon RDS, a child automation `AWSSupport-TroubleshootConnectivityToRDS` is invoked to troubleshoot TCP connectivity. The output also provides commands you can run on your Amazon EC2 instance or source machine to connect to the Amazon RDS instances using IAM authentication.

How does it work?

This runbook consists of six steps:

- **Step 1: validateInputs:** Validates the inputs to the automation.
- **Step 2: branchOnSourceEC2Provided:** Verifies if a source Amazon EC2 Instance ID is provided in the input parameters.
- **Step 3: validateRDSConnectivity:** Validates Amazon RDS connectivity from the source Amazon EC2 instance if provided.
- **Step 4: validateRDSIAMAuthentication:** Validates if the IAM Authentication feature is enabled.
- **Step 5: validateIAMPolicies:** Verifies if the required IAM permissions are present in the IAM user/role provided.
- **Step 6: generateReport:** Generates a report of the results of the previously executed steps.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- RDSType

  Type: String

Description: (Required): Select the type of relational database to which you are trying to connect and authenticate.

Allowed Values: `Amazon RDS` or `Amazon Aurora Cluster.`

- DBInstanceIdentifier

  Type: String

  Description: (Required) The identifier of the target Amazon RDS Database Instance or Aurora Database Cluster.

  Allowed Pattern: `^[A-Za-z0-9]+(-[A-Za-z0-9]+)*$`

  Max Characters: 63
- SourceEc2InstanceIdentifier

  Type: `AWS::EC2::Instance::Id`

  Description: (Optional) The Amazon EC2 Instance ID if you are connecting to the Amazon RDS Database Instance from an Amazon EC2 Instance running in the same account and region. Do not specify this parameter if the source is not an Amazon EC2 instance or if the target Amazon RDS type is an Aurora Database Cluster.

  Default: `""`
- DBIAMRoleName

  Type: String

  Description: (Optional) The IAM role name being used for IAM-based authentication. Provide only if the parameter `DBIAMUserName` is not provided, otherwise leave it empty. Either `DBIAMRoleName` or `DBIAMUserName` must be provided.

  Allowed Pattern: `^[a-zA-Z0-9+=,.@_-]{1,64}$|^$`

  Max Characters: 64

  Default: `""`
- DBIAMUserName

  Type: String

Description: (Optional) The IAM user name used for IAM-based authentication. Provide only if the `DBIAMRoleName` parameter is not provided, otherwise leave it empty. Either `DBIAMRoleName` or `DBIAMUserName` must be provided.

Allowed Pattern: `^[a-zA-Z0-9+=,.@_-]{1,64}$|^$`

Max Characters: 64

Default: ""

- DBUserName

  Type: String

  Description: (Optional) The database user name mapped to an IAM role/user for IAM-based authentication within the database. The default option `*` evaluates if the `rds-db:connect` permission is allowed for all users in the Database.

  Allowed Pattern: `^[a-zA-Z0-9+=,.@*_-]{1,64}$`

  Max Characters: 64

  Default: *

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeInstances`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `iam:GetPolicy`
- `iam:GetRole`
- `iam:GetUser`
- `iam:ListAttachedRolePolicies`

- iam:ListAttachedUserPolicies
- iam:ListRolePolicies
- iam:ListUserPolicies
- iam:SimulatePrincipalPolicy
- rds:DescribeDBClusters
- rds:DescribeDBInstances
- ssm:DescribeAutomationStepExecutions
- ssm:GetAutomationExecution
- ssm:StartAutomationExecution

Instructions

1. Navigate to the AWSSupport-TroubleshootRDSIAMAuthentication in the Amazon Systems Manager Console.
2. Select **Execute Automation**
3. For input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **RDSType (Required):**

     Select the type of Amazon RDS to which you are trying to connect and authenticate. Choose from the two allowed values: `Amazon RDS` or `Amazon Aurora Cluster.`
   - **DBInstanceIdentifier (Required):**

     Enter the identifier of the target Amazon RDS Database Instance or the Aurora Cluster to which you are trying to connect and use IAM credentials for authentication.
   - **SourceEc2InstanceIdentifier (Optional):**

     Provide the Amazon EC2 Instance ID if you are connecting to the Amazon RDS Database Instance from an Amazon EC2 Instance present in the same account and region. Leave blank if the source is not Amazon EC2 or if the target Amazon RDS type is an Aurora Cluster.

- **DBIAMRoleName (Optional):**

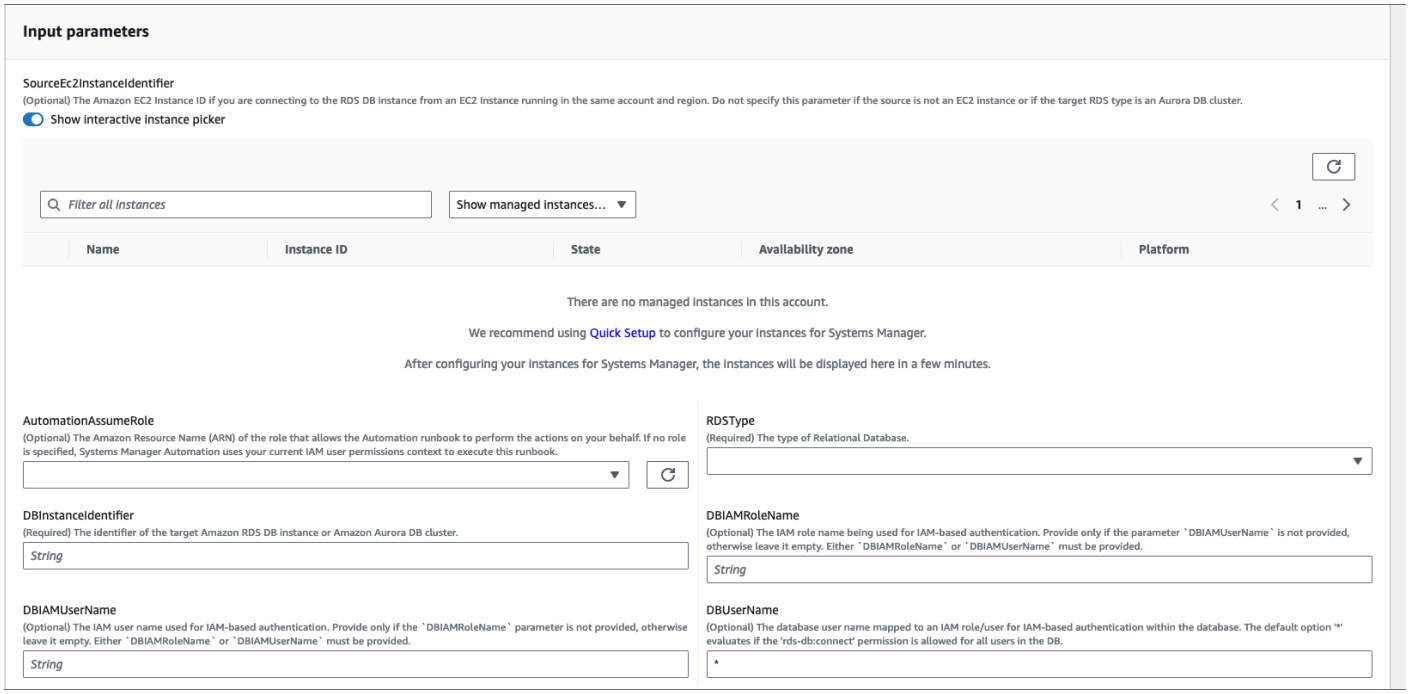
  Enter the IAM Role name used for IAM-Based authentication. Provide only if `DBIAMUserName` is not provided; otherwise, leave blank. Either `DBIAMRoleName` or `DBIAMUserName` must be provided.

- **DBIAMUserName (Optional):**

  Enter the IAM User used for IAM-Based authentication. Provide only if `DBIAMRoleName` is not provided, otherwise, leave blank. Either `DBIAMRoleName` or `DBIAMUserName` must be provided.

- **DBUserName (Optional):**

  Enter the database user mapped to an IAM role/user for IAM-Based authentication within the database. The default option `*` is used to evaluate; nothing is provided in this field.



4. Select **Execute**.
5. Notice that the automation initiates.
6. The document performs the following steps:

   - **Step 1: validateInputs:**

     Validates the inputs to the automation - `SourceEC2InstanceIdentifier` (optional), `DBInstanceIdentifier` or `ClusterID`, and `DBIAMRoleName` or `DBIAMUserName`. It verifies if the input parameters entered are present in your account and region. It also

verifies if the user entered one of the IAM parameters (for example, `DBIAMRoleName` or `DBIAMUserName`). Additionally, it performs other verifications, such as if the Database mentioned is in Available status.

- **Step 2: branchOnSourceEC2Provided:**

  Verifies if Source Amazon EC2 is provided in the input parameters and the Database is Amazon RDS. If yes, it proceeds to step 3. If not, it skips step 3, which is Amazon EC2-Amazon RDS Connectivity validation and proceeds to step 4.

- **Step 3: validateRDSConnectivity:**

  If the Source Amazon EC2 is provided in the input parameters and the Database is Amazon RDS, step 2 initiates step 3. In this step, the child automation `AWSSupport-TroubleshootConnectivityToRDS` is invoked to validate Amazon RDS connectivity from source Amazon EC2. The child automation runbook `AWSSupport-TroubleshootConnectivityToRDS` verifies if the required network configurations (Amazon Virtual Private Cloud [Amazon VPC], Security Groups, Network Access Control List [NACL], Amazon RDS availability) are in place so that you can connect from the Amazon EC2 instance to the Amazon RDS instance.

- **Step 4: validateRDSIAMAuthentication:**

  Validates if the IAM Authentication feature is enabled on the Amazon RDS instance or Aurora Cluster.

- **Step 5: validateIAMPolicies:**

  Verifies if the required IAM permissions are present in the IAM user/role passed to enable the IAM credentials to authenticate into the Amazon RDS instance for the specified Database User (if any).

- **Step 6: generateReport:**

  Obtains all the information from the previous steps and prints the result or the output of each step. It also lists the steps to refer to and perform, to connect to the Amazon RDS instance using the IAM credentials.

7. When the automation is complete, review the **Outputs** section for the detailed results:

- **Checking the IAM User/Role permission to connect to Database:**

Verifies if the required IAM permissions are present in the IAM user/role passed to enable the IAM credentials to authenticate into the Amazon RDS Instance for the specified Database User (if any).

- **Checking IAM-Based Authentication Attribute for the Database:**

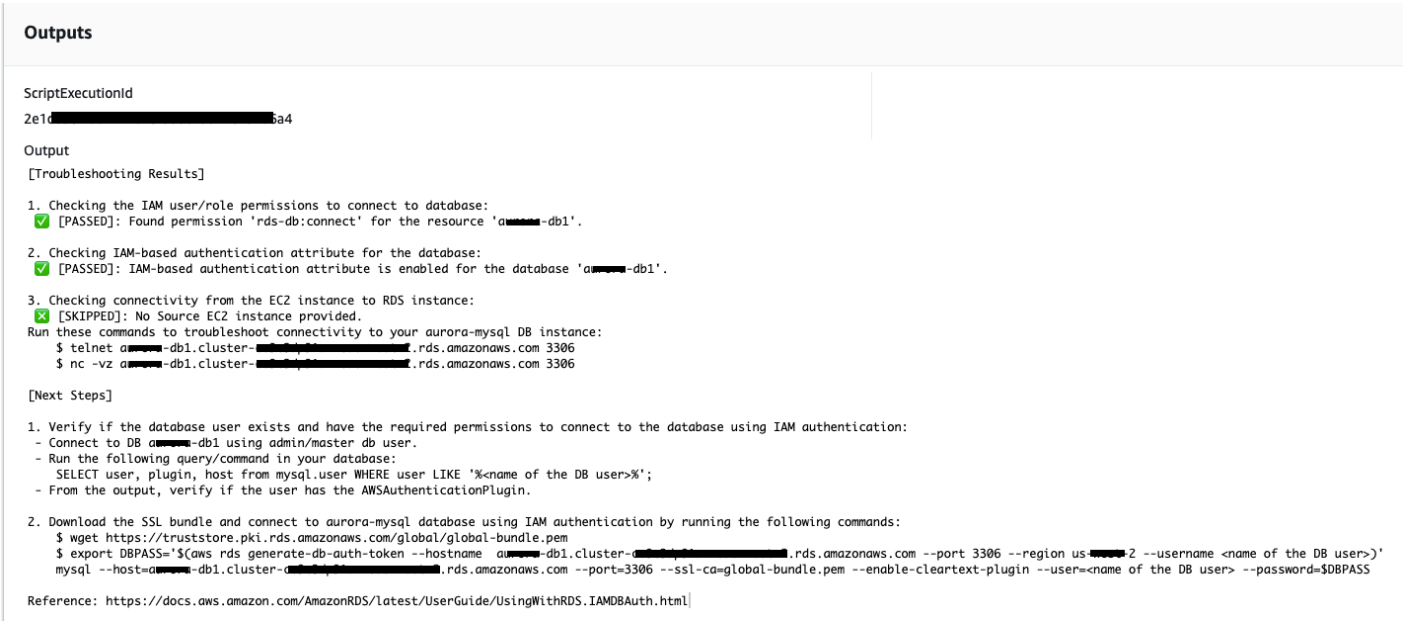
  Verifies if the feature of the IAM authentication is enabled for the specified Amazon RDS Database/Aurora Cluster.

- **Checking Connectivity from Amazon EC2 Instance to Amazon RDS Instance:**

  Verifies if the required network configurations (Amazon VPC, Security Groups, NACL, Amazon RDS availability) are in place so that you can connect from the Amazon EC2 instance to the Amazon RDS Instance.

- **Next Steps:**

  Lists the commands and steps to refer to and perform, to connect to the Amazon RDS Instance using the IAM credentials.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation

- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-ValidateRdsNetworkConfiguration

Description

`AWSSupport-ValidateRdsNetworkConfiguration` automation helps to avoid incompatible-network state for your existing Amazon Relational Database Service (Amazon RDS) / Amazon Aurora / Amazon DocumentDB instance before you perform `ModifyDBInstance` or `StartDBInstance` operation. If the instance is already in incompatible-network state, the runbook will provide the reason.

How does it work?

This runbook determines if your Amazon RDS database instance will go into incompatilble-network state, or if it has, determine the reason it's in incompatible-network state.

The runbook performs the following checks against your Amazon RDS database instance:

- Amazon Elastic Network Interface (ENI) quota per region.
- All subnets in the database Subnet Group exist.
- There are sufficient free IP addresses available for the subnet(s).
- (For publicly accessible Amazon RDS instances) Settings of VPC attributes (`enableDnsSupport` and `enableDnsHostnames`).

> **Important**
>
> When using this document against Amazon Aurora / Amazon DocumentDB clusters, ensure that you use `DBInstanceIdentifier` instead of `ClusterIdentifier`. Otherwise, the document will fail in the first step.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `rds:DescribeDBInstances`
- `servicequotas:GetServiceQuota`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeSubnets`

Sample policy:

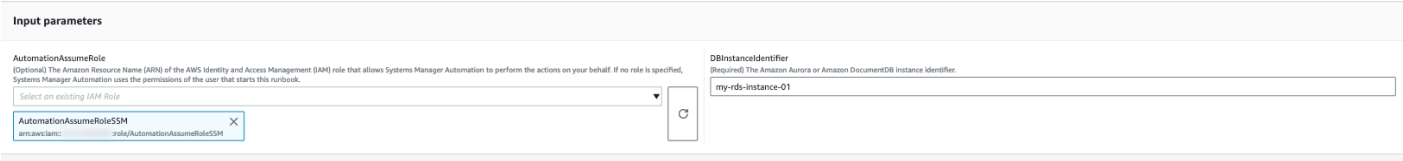
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateRdsNetwork",
            "Effect": "Allow",
            "Action": [
                "rds:DescribeDBInstances",
                "servicequotas:GetServiceQuota",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeSubnets"
            ],
            "Resource": [
                "arn:aws:rds:{Region}:{Account}:db:{DbInstanceName}"
            ]
        }
    ]
}
```

Instructions

1. Navigate to the AWSSupport-ValidateRdsNetworkConfiguration in the Amazon Systems Manager Console.
2. Select **Execute Automation**
3. For input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **DBInstanceIdentifier (Required):**

     Enter the Amazon Relational Database Service Instance Identifier.



4. Select **Execute**.
5. Notice that the automation initiates.
6. The document performs the following steps:
   - **Step 1: assertRdsState:**

     Checks if the provided instance identifier exists and has any of the following states: `available`, `stopped`, or `incompatible-network`.
   - **Step 2: gatherRdsInformation:**

     Gathers required information about the Amazon RDS instance to use later in the automation.
   - **Step 3: checkEniQuota:**

     Checks for the current available quota of Amazon ENI for the region.
   - **Step 4: validateVpcAttributes:**

Validates that the DNS parameters (`enableDnsSupport` and `enableDnsHostnames`) of the Amazon VPC are set to true (or not if the Amazon RDS instance is `PubliclyAccessible`).

- **Step 5: validateSubnetAttributes:**

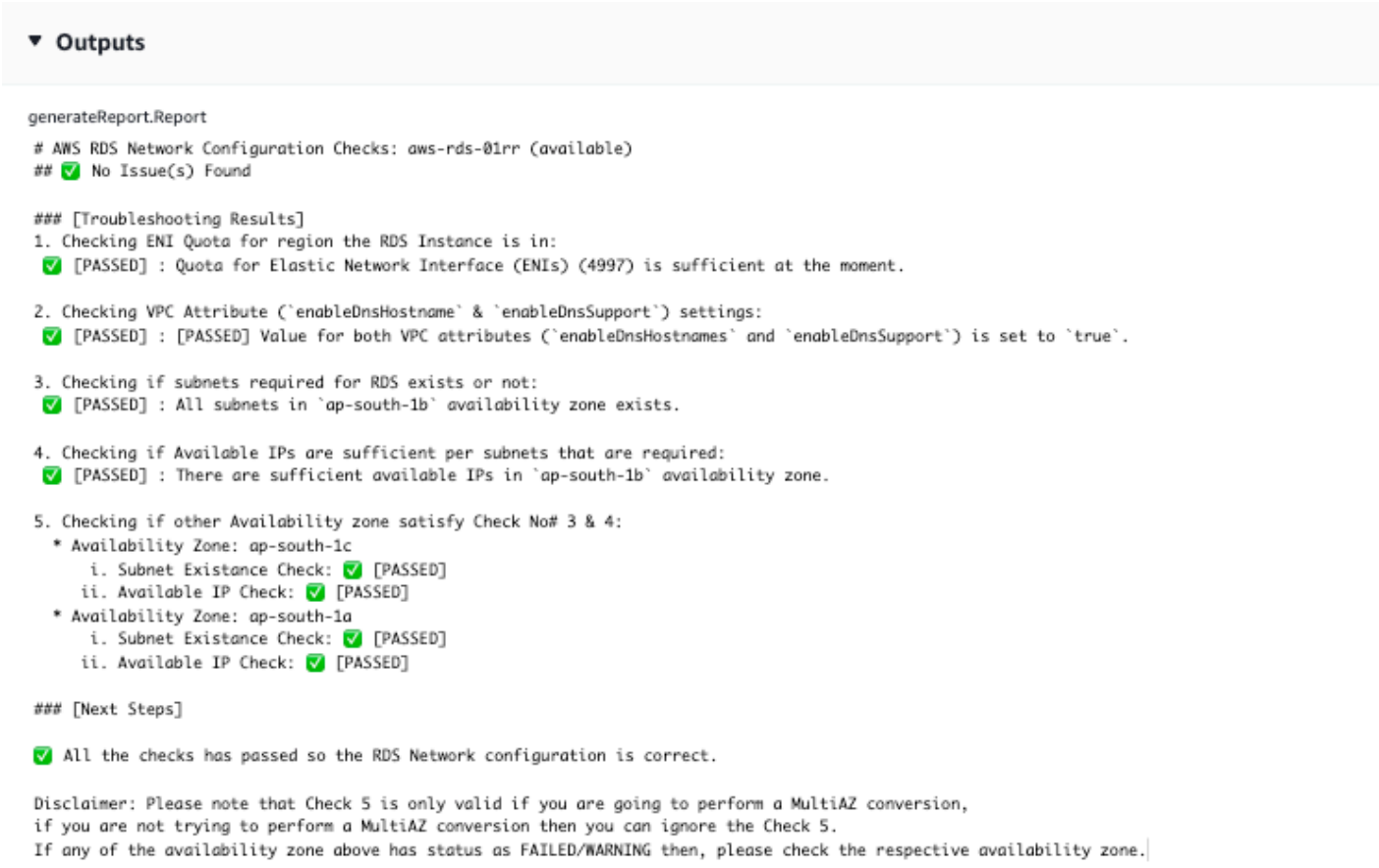
  Validates the existence of subnets in the `DBSubnetGroup` and checks for available IPs for each subnet.

- **Step 6: generateReport:**

  Obtains all the information from the previous steps and prints the result or the output of each step. It also lists the steps to refer to and perform, to connect to the Amazon RDS instance using the IAM credentials.

7. When the automation is complete, review the **Outputs** section for the detailed results:

   Amazon RDS instance with valid network configuration:



▼ Outputs

generateReport.Report

```
# AWS RDS Network Configuration Checks: aws-rds-01rr (available)
## ✅ No Issue(s) Found

### [Troubleshooting Results]
1. Checking ENI Quota for region the RDS Instance is in:
 ✅ [PASSED] : Quota for Elastic Network Interface (ENIs) (4997) is sufficient at the moment.

2. Checking VPC Attribute (`enableDnsHostname` & `enableDnsSupport`) settings:
 ✅ [PASSED] : [PASSED] Value for both VPC attributes (`enableDnsHostnames` and `enableDnsSupport`) is set to `true`.

3. Checking if subnets required for RDS exists or not:
 ✅ [PASSED] : All subnets in `ap-south-1b` availability zone exists.

4. Checking if Available IPs are sufficient per subnets that are required:
 ✅ [PASSED] : There are sufficient available IPs in `ap-south-1b` availability zone.

5. Checking if other Availability zone satisfy Check No# 3 & 4:
  * Availability Zone: ap-south-1c
      i. Subnet Existance Check: ✅ [PASSED]
     ii. Available IP Check: ✅ [PASSED]
  * Availability Zone: ap-south-1a
      i. Subnet Existance Check: ✅ [PASSED]
     ii. Available IP Check: ✅ [PASSED]

### [Next Steps]

✅ All the checks has passed so the RDS Network configuration is correct.

Disclaimer: Please note that Check 5 is only valid if you are going to perform a MultiAZ conversion,
if you are not trying to perform a MultiAZ conversion then you can ignore the Check 5.
If any of the availability zone above has status as FAILED/WARNING then, please check the respective availability zone.
```

   Amazon RDS instance with incorrect network configuration (VPC attribute enableDnsHostnames is set to false):

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- How do I resolve issues with an Amazon RDS database that is in an incompatible-network state?
- How do I resolve issues with an Amazon DocumentDB instance that is in an incompatible-network state?

Amazon Redshift

Amazon Systems Manager Automation provides predefined runbooks for Amazon Redshift. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

`AWSConfigRemediation-DeleteRedshiftCluster`

Description

The `AWSConfigRemediation-DeleteRedshiftCluster` runbook deletes the Amazon Redshift cluster you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ClusterIdentifier

  Type: String

  Description: (Required) The ID of the Amazon Redshift cluster that you want to delete.
- SkipFinalClusterSnapshot

  Type: Boolean

  Default: false

  Description: (Optional) If set to `false` , the automation creates a snapshot before deleting the Amazon Redshift cluster. If set to `true` , a final cluster snapshot is not created.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DeleteCluster`
- `redshift:DescribeClusters`

Document Steps

- `aws:branch` - Branches based on the value you specify for the `SkipFinalClusterSnapshot` parameter.
- `aws:executeAwsApi` - Deletes the Amazon Redshift cluster specified in the `ClusterIdentifier` parameter.
- `aws:assertAwsResourceProperty` - Verifies the Amazon Redshift cluster has been deleted.

AWSConfigRemediation-DisablePublicAccessToRedshiftCluster

Description

The AWSConfigRemediation-DisablePublicAccessToRedshiftCluster runbook disables public accessibility for the Amazon Redshift cluster that you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster that you want to disable public accessibility for.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- redshift:DescribeClusters
- redshift:ModifyCluster

Document Steps

- aws:executeAwsApi - Disables public accessibility for the cluster specified in the ClusterIdentifier parameter.
- aws:waitForAwsResourceProperty - Waits for the state of the cluster to change to available .
- aws:assertAwsResourceProperty - Confirms the public accessibility setting is disabled on the cluster.

AWSConfigRemediation-EnableRedshiftClusterAuditLogging

Description

The AWSConfigRemediation-EnableRedshiftClusterAuditLogging runbook enables audit logging for the Amazon Redshift cluster you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- BucketName

  Type: String

  Description: (Required) The name of the Amazon Simple Storage Service (Amazon S3) bucket you want to upload logs to.

- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster you want to enable audit logging on.

- S3KeyPrefix

  Type: String

  Description: (Optional) The Amazon S3 key prefix (subfolder) you want to upload logs to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeLoggingStatus`
- `redshift:EnableLogging`
- `s3:GetBucketAcl`
- `s3:PutObject`

Document Steps

- `aws:branch` - Branches based on whether a value was specified for the `S3KeyPrefix` parameter.

- `aws:executeAwsApi` - Enables audit logging on the cluster specified in the `ClusterIdentifier` parameter.
- `aws:assertAwsResourceProperty` - Verifies audit logging was enabled on the cluster.

AWSConfigRemediation-EnableRedshiftClusterAutomatedSnapshot

Description

The `AWSConfigRemediation-EnableRedshiftClusterAutomatedSnapshot` runbook enables automated snapshots for the Amazon Redshift cluster you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- AutomatedSnapshotRetentionPeriod

  Type: Integer

  Valid values: 1-35

Description: (Required) The number of days that automated snapshots are retained.

- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster you want to enable automated snapshots on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeClusters`
- `redshift:ModifyCluster`

Document Steps

- `aws:executeAwsApi` - Enables automation snapshots on the cluster specified in the `ClusterIdentifier` parameter.
- `aws:waitForAwsResourceProperty` - Waits for the state of the cluster to change to `available` .
- `aws:executeScript` - Confirms automated snapshots were enabled on the cluster.

AWSConfigRemediation-EnableRedshiftClusterEncryption

Description

The `AWSConfigRemediation-EnableRedshiftClusterEncryption` runbook enables encryption on the Amazon Redshift cluster you specify using an Amazon Key Management Service (Amazon KMS) customer managed key. This runbook should only be used as a baseline to ensure that your Amazon Redshift clusters are encrypted according to minimum recommended security best practices. We recommend encrypting multiple clusters with different customer managed

keys. This runbook cannot change the Amazon KMS customer managed key used on an already encrypted cluster. To change the Amazon KMS customer managed key used to encrypt a cluster, you must first disable encryption on the cluster.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster you want to enable encryption on.

- KMSKeyARN

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon KMS customer managed key you want to use to encrypt the cluster's data.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeClusters`
- `redshift:ModifyCluster`

Document Steps

- `aws:executeAwsApi` - Enables encryption on the Amazon Redshift cluster specified in the `ClusterIdentifier` parameter.
- `aws:assertAwsResourceProperty` - Verifies encryption has been enabled on the cluster.

AWSConfigRemediation-EnableRedshiftClusterEnhancedVPCRouting

Description

The `AWSConfigRemediation-EnableRedshiftClusterEnhancedVPCRouting` runbook enables enhanced virtual private cloud (VPC) routing for the Amazon Redshift cluster you specify. For information about enhanced VPC routing, see Amazon Redshift enhanced VPC routing in the *Amazon Redshift Management Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster you want to enable enhanced VPC routing on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeClusters`
- `redshift:ModifyCluster`

Document Steps

- `aws:executeAwsApi` - Enables enhanced VPC routing on the cluster specified in the `ClusterIdentifier` parameter.
- `assertAwsResourceProperty` - Confirms enhanced VPC routing was enabled on the cluster.

AWSConfigRemediation-EnforceSSLOnlyConnectionsToRedshiftCluster

Description

The `AWSConfigRemediation-EnforceSSLOnlyConnectionsToRedshiftCluster` runbook requires incoming connections to use SSL for the Amazon Redshift cluster you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster you want to enable enhanced VPC routing on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeClusters`
- `redshift:DescribeClusterParameters`
- `redshift:ModifyClusterParameterGroup`

Document Steps

- `aws:executeAwsApi` - Gathers parameter details from the cluster specified in the `ClusterIdentifier` parameter.
- `aws:executeAwsApi` - Enables the `require_ssl` setting on the cluster specified in the `ClusterIdentifier` parameter.
- `aws:assertAwsResourceProperty` - Confirms the `require_ssl` setting was enabled on the cluster.
- `aws:executeScript` - Verifies the `require_ssl` setting for the cluster.

AWSConfigRemediation-ModifyRedshiftClusterMaintenanceSettings

Description

The `AWSConfigRemediation-ModifyRedshiftClusterMaintenanceSettings` runbook modifies the maintenance settings for the Amazon Redshift cluster you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AllowVersionUpgrade

  Type: Boolean

  Description: (Required) If set to `true` , major version upgrades are applied automatically to the cluster during the maintenance window.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- AutomatedSnapshotRetentionPeriod

  Type: Integer

  Valid values: 1-35

  Description: (Required) The number of days automated snapshots are retained.
- ClusterIdentifier

  Type: String

  Description: (Required) The unique identifier of the cluster you want to enable enhanced VPC routing on.
- PreferredMaintenanceWindow

  Type: String

  Description: (Required) The weekly time range (in UTC) during which system maintenance can occur.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeClusters`
- `redshift:ModifyCluster`

Document Steps

- `aws:executeAwsApi` - Modifies the maintenance settings for the cluster specified in the `ClusterIdentifier` parameter.
- `aws:assertAwsResourceProperty` - Confirms the modified maintenance settings were configured for the cluster.

AWSConfigRemediation-ModifyRedshiftClusterNodeType

Description

The `AWSConfigRemediation-ModifyRedshiftClusterNodeType` runbook modifies the node type and number of nodes for the Amazon Redshift cluster you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Databases

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- Classic

  Type: Boolean

  Description: (Optional) If set to `true` , the resize operation uses the classic resize process.
- ClusterIdentifier

Type: String

Description: (Required) The unique identifier of the cluster whose node type you want to modify.

- ClusterType

  Type: String

  Valid values: single-node | multi-node

  Description: (Required) The type of cluster you want to assign to your cluster.

- NodeType

  Type: String

  Valid values: ds2.xlarge | ds2.8xlarge | dc1.large | dc1.8xlarge | dc2.large | dc2.8xlarge | ra3.4xlarge | ra3.16xlarge

  Description: (Required) The type of node you want to assign to your cluster.

- NumberOfNodes

  Type: Integer

  Valid values: 2-100

  Description: (Optional) The number of nodes you want to assign to your cluster. If your cluster is a `single-node` type, do not specify a value for this parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `redshift:DescribeClusters`
- `redshift:ResizeCluster`

Document Steps

- `aws:executeScript` - Modifies the node type and number of nodes for the cluster specified in the `ClusterIdentifier` parameter.

Amazon S3

Amazon Systems Manager Automation provides predefined runbooks for Amazon Simple Storage Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-ArchiveS3BucketToIntelligentTiering
- AWS-ConfigureS3BucketLogging
- AWS-ConfigureS3BucketVersioning
- AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock
- AWSConfigRemediation-ConfigureS3PublicAccessBlock
- AWS-CreateS3PolicyToExpireMultipartUploads
- AWS-DisableS3BucketPublicReadWrite
- AWS-EnableS3BucketEncryption
- AWS-EnableS3BucketKeys
- AWSConfigRemediation-RemovePrincipalStarFromS3BucketPolicy
- AWSConfigRemediation-RestrictBucketSSLRequestsOnly
- AWSSupport-TroubleshootS3PublicRead
- AWSSupport-EmptyS3Bucket
- AWSSupport-TroubleshootS3EventNotifications
- AWSSupport-ContainS3Resource

`AWS-ArchiveS3BucketToIntelligentTiering`

Description

The `AWS-ArchiveS3BucketToIntelligentTiering` runbook creates, or replaces, an intelligent tiering configuration for the Amazon Simple Storage Service (Amazon S3) bucket you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- BucketName

  Type: String

  Description: (Required) The name of the S3 bucket you want to create an intelligent tiering configuration for.

- ConfigurationId

  Type: String

  Description: (Required) The ID for the intelligent tiering configuration. This can be a new configuration ID, or the ID of an existing configuration.

- NumberOfDaysToArchive

  Type: String

  Valid values: 90-730

Description: (Required) The number of consecutive days after an object in your bucket is eligible to be transitioned to the Archive Access tier.

- NumberOfDaysToDeepArchive

  Type: String

  Valid values: 180-730

  Description: (Required) The number of consecutive days after an object in your bucket is eligible to be transitioned to the Deep Archive Access tier.

- S3Prefix

  Type: String

  Description: (Optional) The key name prefix of the objects you want to apply the configuration to.

- Tags

  Type: MapList

  Description: (Optional) Metadata assigned to the objects you want to apply the configuration to. Tags consist of a user-defined key and value.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `s3:GetIntelligentTieringConfiguration`
- `s3:PutIntelligentTieringConfiguration`

Document Steps

- PutsBucketIntelligentTieringConfiguration (aws:executeScript) - Creates or updates an Amazon S3 Intelligent-Tiering configuration for the specified bucket.

- VerifyBucketIntelligentTieringConfiguration (aws:assertAwsResourceProperty) - Verifies the S3 Bucket Intelligent Configuration was applied to the specified bucket.

AWS-ConfigureS3BucketLogging

Description

Enable logging on an Amazon Simple Storage Service (Amazon S3) bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BucketName

  Type: String

  Description: (Required) The name of the Amazon S3 Bucket for which you want to configure logging.
- GrantedPermission

  Type: String

Valid values: FULL_CONTROL | READ | WRITE

Description: (Required) Logging permissions assigned to the grantee for the bucket.

- GranteeEmailAddress

  Type: String

  (Optional) Email address of the grantee.

- GranteeId

  Type: String

  Description: (Optional) The canonical user ID of the grantee.

- GranteeType

  Type: String

  Valid values: CanonicalUser | AmazonCustomerByEmail | Group

  Description: (Required) Type of grantee.

- GranteeUri

  Type: String

  Description: (Optional) URI of the grantee group.

- TargetBucket

  Type: String

  Description: (Required) Specifies the bucket where you want Amazon S3 to store server access logs. You can have your logs delivered to any bucket that you own. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case you should choose a different TargetPrefix for each source bucket so that the delivered log files can be distinguished by key.

- TargetPrefix

  Type: String

  Default: /

Description: (Optional) Specifies a prefix for the keys under which the log files will be stored.

AWS-ConfigureS3BucketVersioning

Description

Configure versioning for an Amazon Simple Storage Service (Amazon S3) bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BucketName

  Type: String

  Description: (Required) The name of the Amazon S3 bucket you want to configure versioning for.
- VersioningState

  Type: String

  Valid values: Enabled | Suspended

Default: Enabled

Description: (Optional) Applied to the VersioningConfiguration.Status. When set to 'Enabled', this process enables versioning for the objects in the bucket, all objects added to the bucket receive a unique version ID. When set to `Suspended` , this process disables versioning for the objects in the bucket. All objects added to the bucket receive the version ID `null` .

AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock

Description

The `AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock` runbook configures the Amazon Simple Storage Service (Amazon S3) public access block settings for an Amazon S3 bucket based on the values you specify in the runbook parameters.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- BlockPublicAcls

  Type: Boolean

Default: true

Description: (Optional) If set to `true` , Amazon S3 blocks public access control lists (ACLs) for the S3 bucket, and objects stored in the S3 bucket you specify in the `BucketName` parameter.

- BlockPublicPolicy

  Type: Boolean

  Default: true

  Description: (Optional) If set to `true` , Amazon S3 blocks public bucket policies for the S3 bucket you specify in the `BucketName` parameter.

- BucketName

  Type: String

  Description: (Required) The name of the S3 bucket you want to configure.

- IgnorePublicAcls

  Type: Boolean

  Default: true

  Description: (Optional) If set to `true` , Amazon S3 ignores all public ACLs for the S3 bucket you specify in the `BucketName` parameter.

- RestrictPublicBuckets

  Type: Boolean

  Default: true

  Description: (Optional) If set to `true` , Amazon S3 restricts public bucket policies for the S3 bucket you specify in the `BucketName` parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- s3:GetAccountPublicAccessBlock
- s3:PutAccountPublicAccessBlock
- s3:GetBucketPublicAccessBlock
- s3:PutBucketPublicAccessBlock

Document Steps

- aws:executeAwsApi - Creates or modifies the PublicAccessBlock configuration for the S3 bucket specified in the BucketName parameter.
- aws:executeScript - Returns the PublicAccessBlock configuration for the S3 bucket specified in the BucketName parameter, and verifies the changes were successfully made based on the values specified in the runbook parameters.

AWSConfigRemediation-ConfigureS3PublicAccessBlock

Description

The AWSConfigRemediation-ConfigureS3PublicAccessBlock runbook configures an Amazon Web Services account's Amazon Simple Storage Service (Amazon S3) public access block settings based on the values you specify in the runbook parameters.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AccountId

  Type: String

  Description: (Required) The ID of the Amazon Web Services account that owns the S3 bucket you are configuring.

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- BlockPublicAcls

  Type: Boolean

  Default: true

  Description: (Optional) If set to `true` , Amazon S3 blocks public access control lists (ACLs) for S3 buckets owned by the Amazon Web Services account you specify in the `AccountId` parameter.

- BlockPublicPolicy

  Type: Boolean

  Default: true

  Description: (Optional) If set to `true` , Amazon S3 blocks public bucket policies for S3 buckets owned by the Amazon Web Services account you specify in the `AccountId` parameter.

- IgnorePublicAcls

  Type: Boolean

  Default: true

  Description: (Optional) If set to `true` , Amazon S3 ignores all public ACLs for S3 buckets owned by the Amazon Web Services account you specify in the `AccountId` parameter.

- RestrictPublicBuckets

  Type: Boolean

Default: true

Description: (Optional) If set to `true` , Amazon S3 restricts public bucket policies for S3 buckets owned by the Amazon Web Services account you specify in the `AccountId` parameter.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `s3:GetAccountPublicAccessBlock`
- `s3:PutAccountPublicAccessBlock`

Document Steps

- `aws:executeAwsApi` - Creates or modifies the `PublicAccessBlock` configuration for the Amazon Web Services account specified in the `AccountId` parameter.
- `aws:executeScript` - Returns the `PublicAccessBlock` configuration for the Amazon Web Services account specified in the `AccountId` parameter, and verifies the changes were successfully made based on the values specified in the runbook parameters.

AWS-CreateS3PolicyToExpireMultipartUploads

Description

The `AWS-CreateS3PolicyToExpireMultipartUploads` runbook creates a lifecycle policy for a specified bucket that expires incomplete, multi-part uploads in progress after a defined number of days. This runbook merges the new lifecycle policy with any existing lifecycle bucket policies that already exist.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BucketName

  Type: String

  Description: (Required) The name of the S3 bucket you want to configure.
- DaysUntilExpire

  Type: Integer

  Description: (Required) The number of days Amazon S3 waits before permanently removing all parts of the upload.
- RuleId

  Type: String

  Description: (Required) The ID used to identify the lifeycle bucket rule. This must be a unique value.
- S3Prefix

  Type: String

  Description: (Optional) The key name prefix of the objects you want to apply the configuration to.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `s3:GetLifecycleConfiguration`
- `s3:PutLifecycleConfiguration`

Document Steps

- ConfigureExpireMultipartUploads (aws:executeScript) - Configures the lifecycle policy for the bucket.
- VerifyExpireMultipartUploads (aws:executeScript) - Verifies the lifecycle policy has been configured for the bucket.

Outputs

- `VerifyExpireMultipartUploads.VerifyExpireMultipartUploadsResponse`
- `VerifyExpireMultipartUploads.LifecycleConfigurationRule`

AWS-DisableS3BucketPublicReadWrite

Description

Use Amazon Simple Storage Service (Amazon S3) `Block Public Access` to disable read and write access for a public S3 bucket. For more information, see Using Amazon S3 Block Public Access in the *Amazon Simple Storage Service User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- S3BucketName

  Type: String

  Description: (Required) S3 bucket on which you want to restrict access.

AWS-EnableS3BucketEncryption

Description

Configures default encryption for an Amazon Simple Storage Service (Amazon S3) bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- BucketName

  Type: String

  Description: (Required) The name of the S3 bucket where you want to encrypt the contents.

- SSEAlgorithm

  Type: String

  Default: AES256

  Description: (Optional) Server-side encryption algorithm to use for the default encryption.

AWS-EnableS3BucketKeys

Description

The `AWS-EnableS3BucketKeys` runbook enables Bucket Keys on the Amazon Simple Storage Service (Amazon S3) bucket you specify. This bucket level key creates data keys for new objects during its lifecycle. If you don't specify a value for the `KmsKeyId` parameter, server-side encryption using Amazon S3 managed keys (SSE-S3) are used for the default encryption configuration.

> **Note**
>
> Amazon S3 Bucket Keys aren't supported for dual-layer server-side encryption with Amazon Key Management Service (Amazon KMS) keys (DSSE-KMS).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BucketName

  Type: String

  Description: (Required) The name of the S3 bucket you want to enable Bucket Keys for.
- KMSKeyId

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN), key ID, or the key alias of the Amazon Key Management Service (Amazon KMS) customer managed key you want to use for server-side encryption.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `s3:GetEncryptionConfiguration`
- `s3:PutEncryptionConfiguration`

Document Steps

- ChooseEncryptionType (aws:branch) - Evaluates the value provided for the `KmsKeyId` parameter to determine if SSE-S3 (AES256) or SSE-KMS will be used.
- PutBucketKeysKMS (aws:executeAwsApi) - Sets the `BucketKeyEnabled` property to `true` for the specified S3 bucket using the specified `KmsKeyId`.
- PutBucketKeysAES256 (aws:executeAwsApi) - Sets the `BucketKeyEnabled` property to `true` for the specified S3 bucket with AES256 encryption.
- VerifyS3BucketKeysEnabled (aws:assertAwsResourceProperty) - Verifies that the Bucket Keys are enabled on the target S3 bucket.

AWSConfigRemediation-RemovePrincipalStarFromS3BucketPolicy

Description

The `AWSConfigRemediation-RemovePrincipalStarFromS3BucketPolicy` runbook removes principal policy statements that have wildcards (`Principal: *` or `Principal: "AWS": *`) for `Allow` actions from your Amazon Simple Storage Service (Amazon S3) bucket policy. Policy statements with conditions are also removed.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- BucketName

  Type: String

  Description: (Required) The name of the Amazon S3 bucket whose policy you want to modify.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `s3:DeleteBucketPolicy`
- `s3:GetBucketPolicy`
- `s3:PutBucketPolicy`

Document Steps

- `aws:executeScript` - Modifies the bucket policy and verifies principal policy statements with wildcards have been removed from the Amazon S3 bucket you specify in the `BucketName` parameter.

AWSConfigRemediation-RestrictBucketSSLRequestsOnly

Description

The `AWSConfigRemediation-RestrictBucketSSLRequestsOnly` runbook creates an Amazon Simple Storage Service (Amazon S3) bucket policy statement that explicitly denies HTTP requests to the Amazon S3 bucket you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- BucketName

  Type: String

  Description: (Required) The name of the S3 bucket that you want to deny HTTP requests.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `s3:DeleteBucketPolicy`
- `s3:GetBucketPolicy`
- `s3:PutEncryptionConfiguration`
- `s3:PutBucketPolicy`

Document Steps

- `aws:executeScript` - Creates a bucket policy for the S3 bucket specified in the `BucketName` parameter that explicitly denies HTTP requests.

AWSSupport-TroubleshootS3PublicRead

Description

The `AWSSupport-TroubleshootS3PublicRead` runbook diagnoses issues reading objects from the public Amazon Simple Storage Service (Amazon S3) bucket you specify in the `S3BucketName` parameter. A subset of settings are also analyzed for objects in the S3 bucket.

Run this Automation (console)

Limitations

- This automation does not check for access points that allow public access to objects.
- This automation does not evaluate condition keys in the S3 bucket policy.
- If you're using Amazon Organizations, this automation does not evaluate service control policies to confirm that access to Amazon S3 is allowed.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- CloudWatchLogGroupName

  Type: String

  Description: (Optional) The Amazon CloudWatch Logs log group where you want to send the automation output. If a log group is not found that matches the value you specify, the automation will create a log group using this parameter value. The retention period for the log group created by this automation is 14 days.

- CloudWatchLogStreamName

  Type: String

  Description: (Optional) The CloudWatch Logs log stream where you want to send the automation output. If a log stream is not found that matches the value you specify, the automation will create a log stream using this parameter value. If you do not specify a value for this parameter, the automation will use the `ExecutionId` for the name of the log stream.

- HttpGet

  Type: Boolean

  Valid values: true | false

  Default: true

  Description: (Optional) If this parameter is set to `true` , the automation makes a partial HTTP request to the objects in the `S3BucketName` you specify. Only the first byte of the object is returned using the Range HTTP header.

- IgnoreBlockPublicAccess

  Type: Boolean

  Valid values: true | false

  Default: false

  Description: (Optional) If this parameter is set to `true` , the automation ignores the public access block settings of the S3 bucket you specify in the `S3BucketName` parameter. Changing this parameter from the default value is not recommended.

- MaxObjects

  Type: Integer

  Valid values: 1-25

  Default: 5

  Description: (Optional) The number of objects to analyze in the S3 bucket you specify in the `S3BucketName` parameter.

- S3BucketName

  Type: String

  Description: (Required) The name of the S3 bucket to troubleshoot.

- S3PrefixName

  Type: String

  Description: (Optional) The key name prefix of the objects you want to analyze in your S3 bucket. For more information, see Object keys in the *Amazon Simple Storage Service User Guide* .

- StartAfter

  Type: String

  Description: (Optional) The object key name where you want the automation to begin analyzing objects in your S3 bucket.

- ResourcePartition

  Type: String

  Valid values: `aws` | `aws-us-gov` | `aws-cn`

  Default: `aws`

  Description: (Required) The partition where your S3 bucket is located.

- Verbose

  Type: Boolean

  Valid values: true | false

Default: false

Description: (Optional) To return more detailed information during the automation, set this parameter to `true` . Only warning and error messages will be returned if the parameter is set to `false` .

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

The `logs:CreateLogGroup` , `logs:CreateLogStream` , and `logs:PutLogEvents` permissions are only required if you want the automation to send log data to CloudWatch Logs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:SimulateCustomPolicy",
                "iam:GetContextKeysForCustomPolicy",
                "s3:ListAllMyBuckets",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "s3:GetAccountPublicAccessBlock"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Effect": "Allow"
        },
        {
```

```
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketRequestPayment",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPolicy",
                "s3:GetBucketAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
            "Effect": "Allow"
        }
    ]
}
```

Document Steps

- `aws:assertAwsResourceProperty` - Confirms the S3 bucket exists, and is accessible.
- `aws:executeScript` - Returns the S3 bucket location and your canonical user ID.
- `aws:executeScript` - Returns the public access block settings for your account and the S3 bucket.
- `aws:assertAwsResourceProperty` - Confirms the S3 bucket payer is set to `BucketOwner` . If `Requester Pays` is enabled on the S3 bucket, the automation ends.
- `aws:executeScript` - Returns the S3 bucket policy status and determines whether it is considered public. For more information about public S3 buckets, see The meaning of "public" in the *Amazon Simple Storage Service User Guide* .
- `aws:executeAwsApi` - Returns the S3 bucket policy.
- `aws:executeAwsApi` - Returns all context keys found in the S3 bucket policy.
- `aws:assertAwsResourceProperty` - Confirms whether there is an explicit deny in the S3 bucket policy for the `GetObject` API action.
- `aws:executeAwsApi` - Returns the access control list (ACL) for the S3 bucket.
- `aws:executeScript` - Creates a CloudWatch Logs log group and log stream if you specify a value for the `CloudWatchLogGroupName` parameter.
- `aws:executeScript` - Based on the values you specify in the runbook input parameters, evaluates whether any of the S3 bucket settings gathered during the automation are preventing objects from being accessed by the public. This script performs the following functions:

- Evaluates public access block settings
- Returns objects from your S3 bucket based on the values you specify in the `MaxObjects` , `S3PrefixName` , and `StartAfter` parameters.
- Returns the S3 bucket policy to simulate a custom IAM policy for the objects returned from your S3 bucket.
- Performs a partial HTTP request to the returned objects if the `HttpGet` parameter is set to `true` . Only the first byte of the object is returned using the Range HTTP header.
- Checks the returned object's key name to confirm whether it ends with one or two periods. Object key names that end in periods can't be downloaded from the Amazon S3 console.
- Checks whether the returned object's owner matches the owner of the S3 bucket.
- Checks whether the object's ACL grants `READ` or `FULL_CONTROL` permissions to anonymous users.
- Returns tags associated with the object.
- Uses the simulated IAM policy to confirm whether there is an explicit deny for this object in the S3 bucket policy for the `GetObject` API action.
- Returns the object's metadata to confirm that the storage class is supported.
- Checks the object's server-side encryption settings to confirm whether the object is encrypted using a Amazon Key Management Service (Amazon KMS) customer managed key.

Outputs

AnalyzeObjects.bucket

AnalyzeObjects.object

AWSSupport-EmptyS3Bucket

Description

The `AWSSupport-EmptyS3Bucket` automation runbook empties an existing Amazon Simple Storage Service (Amazon S3) bucket by using a lifecycle expiration configuration rule.

> **Important**
>
> - Amazon S3 buckets with Multi-factor Authentication (MFA) enabled are not supported.

- The lifecycle rules modified by this runbook permanently delete all objects and their versions in the specified Amazon S3 bucket. You cannot recover permanently deleted objects. For more information, review Expiring Objects.

How does it work?

The runbook `AWSSupport-EmptyS3Bucket` performs the following high-level steps:

- Suspends bucket versioning, if enabled.
- Updates the bucket policy to deny any `s3:PutObject` API calls (to prevent new uploads while it is being emptied).
- Updates the lifecycle rules to delete all the objects according to the expiration days specified in the input parameters.

Note

- Object versions protected with Amazon S3 Object Lock are not deleted or overwritten by lifecycle configurations.
- The deletion process is asynchronous and may take time to complete after the runbook execution finishes.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

/

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

The AutomationAssumeRole parameter requires the following actions to successfully use the runbook:

- ssm:DescribeAutomationExecutions
- ssm:GetAutomationExecution
- s3:GetBucketVersioning
- s3:PutBucketVersioning
- s3:GetBucketPolicy
- s3:GetBucketLifecycleConfiguration
- s3:GetLifecycleConfiguration
- s3:PutBucketPolicy
- s3:PutBucketLifecycleConfiguration
- s3:PutLifecycleConfiguration
- s3:DeleteBucketPolicy
- s3:DeleteBucketLifecycle

Instructions

Follow these steps to configure the automation:

1. Navigate to `AWSSupport-EmptyS3Bucket` in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **S3BucketName:**

     The name of the Amazon S3 bucket you want to empty.

- **SNSTopicArn:**

  Provide the ARN of the Amazon SNS Topic for approval notification. This Amazon SNS topic is used to send approval notifications during required during the automation execution.

- **ApproverIAM:**

  Provide a list of AWS authenticated principals who are able to either approve or reject the action. The maximum number of approvers is 10. You can specify principals by using any of these formats, an Amazon Identity and Access Management (IAM) user name, an IAM user ARN, an IAM role ARN, or an IAM assume role user ARN.

- **MinimumRequiredApprovals (Optional):**

  The minimum number of approvals required to resume the automation. If you don't specify a value, the system defaults to 1. The value for this parameter must be a positive number. The value for this parameter can't exceed the number of approvers defined by the ApproverIAM parameter.

- **NoncurrentVersionExpirationDays (Optional):**

  Specify the number of days when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.

  - Default: 1
  - Maximum Value: 365

- **ExpirationDays (Optional):**

  Specify the expiration for the lifecycle of the object in the form days.

  - Default: 1
  - Maximum Value: 365

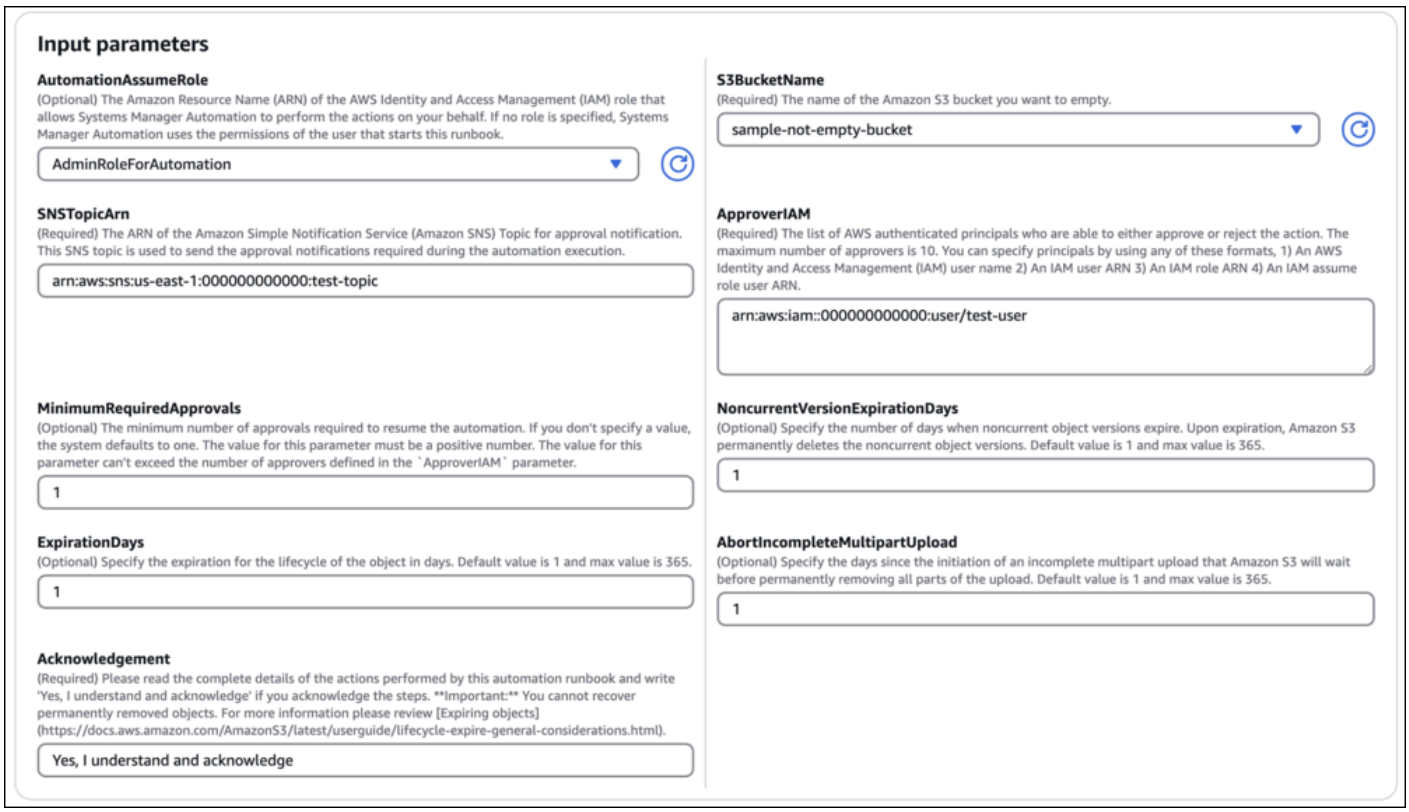
- **AbortIncompleteMultipartUpload (Optional):**

  Specify the days since the initiation of an incomplete multipart upload that Amazon S3 will wait before permanently removing all parts of the upload.

  - Default: 1
  - Maximum Value: 365

- **Acknowledgement:**

Please read the complete details of the actions performed by this automation runbook and provide consent `Yes, I understand and acknowledge` if you acknowledge the steps.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:
   - **`checkConcurrency`**:

     Ensures there is only one execution of this runbook targeting the specified Amazon S3 bucket. If the runbook finds another in progress execution targeting the same bucket name, it returns an error and ends.
   - **`getBucketVersioningConfiguration`**:

     Fetches the versioning status of the specified Amazon S3 bucket.
   - **`branchOnStoppingIfMFADeleteEnabled`** (conditional):

     Stops the automation if Multi-factor Authentication (MFA) is enabled on the specified Amazon S3 bucket.
   - **`approvalToMakeChangesToTheProvidedS3Bucket`**:

Waits for designated principals approval to disable bucket versioning and update the bucket policy and lifecycle rules configuration for the specified Amazon S3 bucket.

- **`branchOnBucketVersioningStatus`** (conditional):

  If versioning is enabled on the specified Amazon S3 bucket, disable it, otherwise continue to update bucket policy and lifecycle configuration.

- **`suspendBucketVersioning`**:

  Suspends the versioning state of the specified Amazon S3 bucket.

- **`updateBucketPolicyAndLifeCycleConfiguration`**:

  Adds or updates the bucket policy to deny all `s3:PutObject` requests and updates the lifecycle configuration to expire objects based on the user provided inputs parameters.

- **`branchOnFailingIfBucketPropertiesNotUpdated`** (conditional):

  Checks the status of the `updateBucketPolicyAndLifeCycleConfiguration` step and tries to revert the original bucket versioning state if changed by automation.

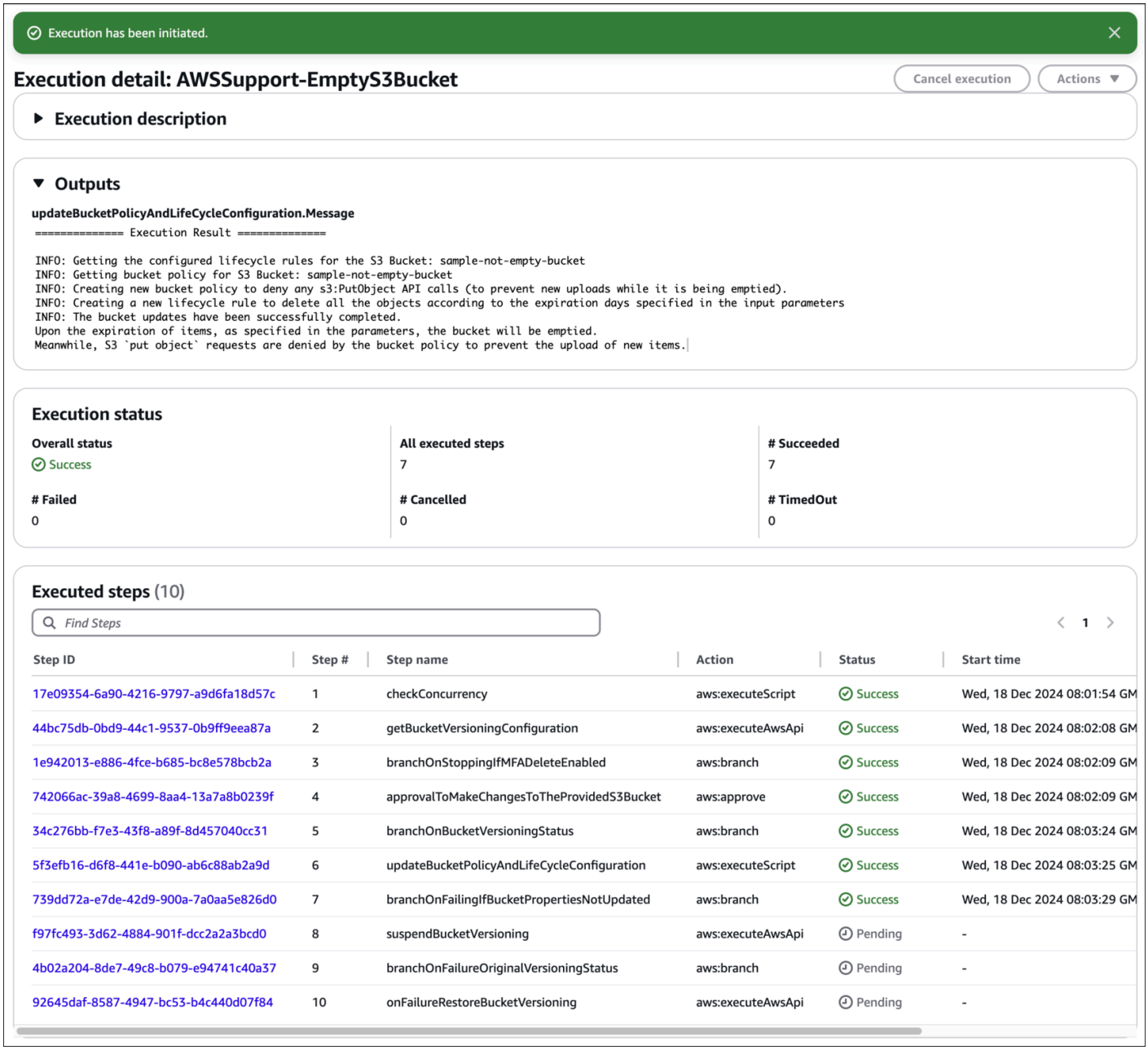
- **`branchOnFailureOriginalVersioningStatus`** (conditional):

  On failure, branches to determine the original versioning status. If was enabled and suspended by this automation, tries to enable it again.

- **`onFailureRestoreBucketVersioning`**

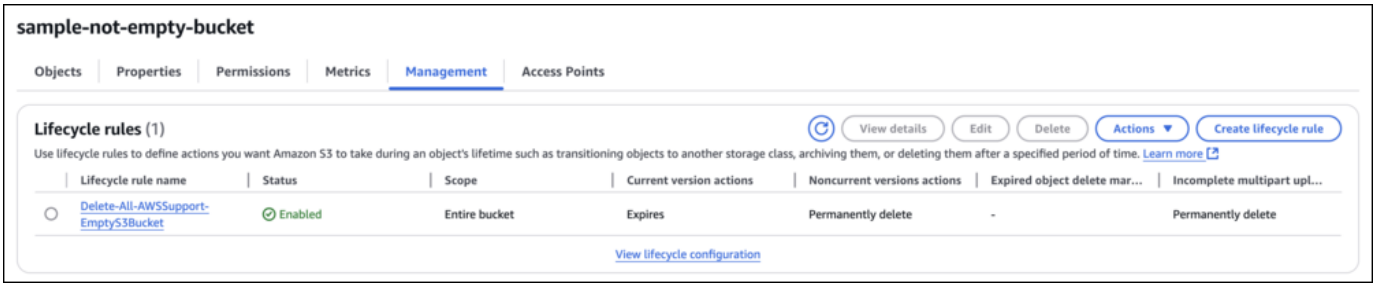
  Restores the enabled versioning state of the specified Amazon S3 bucket.

7. After completed, review the Outputs section for the detailed results of the execution:



- **Successful execution**

  This workflow updates the bucket's lifecycle rule. Objects will be deleted according to the `Delete-All-AWSSupport-EmptyS3-Bucket` lifecycle policy.



- **Failure execution**

  Partial deletion will not be performed. If execution fails, the lifecycle and other bucket settings are rolled back.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows

For more information on managing Amazon S3 buckets and objects, see Emptying a bucket.

`AWSSupport-TroubleshootS3EventNotifications`

Description

The `AWSSupport-TroubleshootS3EventNotifications` Amazon Systems Manager automation runbook helps troubleshoot Amazon Simple Storage Service (Amazon S3) Bucket Event Notifications configured with Amazon Lambda Functions, Amazon Simple Notification Service (Amazon SNS) Topics, or Amazon Simple Queue Service (Amazon SQS) Queues. It provides a configuration settings report of the different resources configured with the the Amazon S3 Bucket as a destination event notification.

How does it work?

The runbook performs the following steps:

- Checks if the Amazon S3 Bucket exists in the same account where `AWSSupport-TroubleshootS3EventNotifications` is executed.

- Fetches the destination resources (Amazon Lambda Function, or Amazon SNS Topic or Amazon SQS queue) configured as Event Notifications for the Amazon S3 Bucket using the GetBucketNotificationConfiguration API.
- Validates that the destination resource exists, then reviews the resource-based policy of the destination resources to determine if Amazon S3 is allowed to publish to the destination.
- If you encrypted the destination with an Amazon Key Management Service (Amazon KMS) key, the key policy is checked to determine if Amazon S3 access is allowed.
- Generates a report of all the destination resource checks.

Important

- This runbook can only evaluate event notification configurations if the Amazon S3 bucket owner is the same as the Amazon Web Services account owner where the automation runbook is being executed.
- Additionally, this runbook cannot evaluate policies on destination resources that are hosted in another Amazon Web Services account.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- S3BucketName

  Type: `AWS::S3::Bucket::Name`

  Description: (Required) The name of the Amazon S3 bucket configured with event notification(s).

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `s3:GetBucketLocation`
- `s3:ListAllMyBuckets`
- `s3:GetBucketNotification`
- `sqs:GetQueueAttributes`
- `sqs:GetQueueUrl`
- `sns:GetTopicAttributes`
- `kms:GetKeyPolicy`
- `kms:DescribeKey`
- `kms:ListAliases`
- `lambda:GetPolicy`
- `lambda:GetFunction`
- `iam:GetContextKeysForCustomPolicy`
- `iam:SimulateCustomPolicy`
- `iam:ListRoles`
- `ssm:DescribeAutomationStepExecutions`

Example IAM Policy for the Automation Assume Role

```
                    {
                        "Version": "2012-10-17",
                        "Statement": [
                            {
                                "Sid": "S3Permission",
                                "Effect": "Allow",
                                "Action": [
                                    "s3:GetBucketLocation",
                                    "s3:ListAllMyBuckets"
                                ],
                                "Resource": "*"
                            },
                            {
                                "Sid": "S3PermissionGetBucketNotification",
                                "Effect": "Allow",
                                "Action": [
                                    "s3:GetBucketNotification"
                                ],
                                "Resource": "arn:aws:s3::::<bucket-name>"
                            },
                            {
                                "Sid": "SQSPermission",
                                "Effect": "Allow",
                                "Action": [
                                    "sqs:GetQueueAttributes",
                                    "sqs:GetQueueUrl"
                                ],
                                "Resource": "arn:aws:sqs:<region>:123456789012:*"
                            },
                            {
                                "Sid": "SNSPermission",
                                "Effect": "Allow",
                                "Action": [
                                    "sns:GetTopicAttributes"
                                ],
                                "Resource": "arn:aws:sns:<region>:123456789012:*"
                            },
                            {
                            "Sid": "KMSPermission",
                                "Effect": "Allow",
                                "Action": [
                                    "kms:GetKeyPolicy",
                                    "kms:DescribeKey",
                                    "kms:ListAliases"
```

```
                                ],
                                "Resource": "arn:aws:kms:<region>:123456789012:key/
<key-id>"
                            },
                            {
                                "Sid": "LambdaPermission",
                                "Effect": "Allow",
                                "Action": [
                                    "lambda:GetPolicy",
                                    "lambda:GetFunction"
                                ],
                                "Resource":
 "arn:aws:lambda:<region>:123456789012:function:*"
                            },
                            {
                                "Sid": "IAMPermission",
                                "Effect": "Allow",
                                "Action": [
                                    "iam:GetContextKeysForCustomPolicy",
                                    "iam:SimulateCustomPolicy",
                                    "iam:ListRoles"
                                ],
                                "Resource": "*"
                            },
                            {
                                "Sid": "SSMPermission",
                                "Effect": "Allow",
                                "Action": [
                                    "ssm:DescribeAutomationStepExecutions"
                                ],
                                "Resource": "*"
                            }
                        ]
                    }
```

Instructions

Follow these steps to configure the automation:

1. Navigate to [AWSSupport-TroubleshootS3EventNotifications] in Systems Manager under Documents.
2. Select Execute automation.

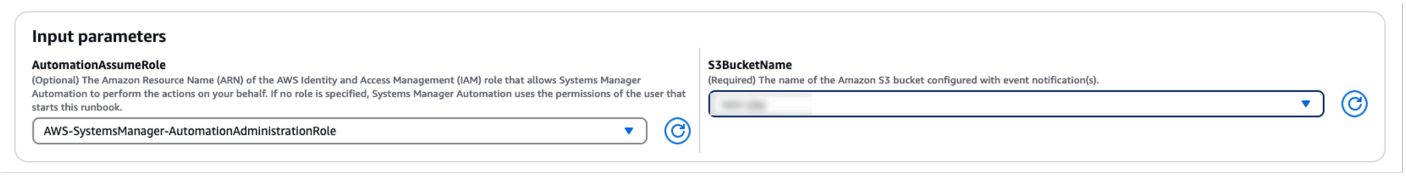
3. For the input parameters, enter the following:

   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.

   - **S3BucketName (Required):**

     The name of the Amazon S3 bucket configured with event notification(s).



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

   - **ValidateInputs**

     Validates Amazon S3 bucket provided belongs to the same account where the automation is executed and fetch the region the bucket is hosted.

   - **GetBucketNotificationConfiguration**

     Calls `GetBucketNotificationConfiguration` API to review Event Notifications configured with the Amazon S3 bucket and formats output.

   - **BranchOnSQSResourcePolicy**

     Branches on whether there are Amazon SQS resources in event notifications.

   - **ValidateSQSResourcePolicy**

     Validates resource policy on Amazon SQS Queue attributes has `sqs:SendMessage` permission for Amazon S3. If the Amazon SQS resource is encrypted, checks that encryption is not using default Amazon KMS key i.e. `aws/sqs` and checks that Amazon KMS key policy has permissions for Amazon S3.

   - **BranchOnSNSResourcePolicy**

     Branches on whether there are Amazon SNS resources in event notifications.

- **ValidateSNSResourcePolicy**

  Validates resource policy on Amazon SNS Topic attributes has `sns:Publish` permission for Amazon S3. If the Amazon SNS resource is encrypted, checks that encryption is not using default Amazon KMS key i.e. `aws/sns` and checks that Amazon KMS key policy has permissions for Amazon S3.

- **BranchOnLambdaFunctionResourcePolicy**

  Branches on whether there are Amazon Lambda functions in event notifications.

- **ValidateLambdaFunctionResourcePolicy**

  Validates resource policy on Amazon Lambda function has `lambda:InvokeFunction` permission for Amazon S3.

- **GenerateReport**

  Returns details of the runbook steps outputs, and recommendations to resolve any issue with the event notifications configured with the Amazon S3 bucket.

7. After completed, review the Outputs section for the detailed results of the execution:

- **Amazon SQS Event Notifications**

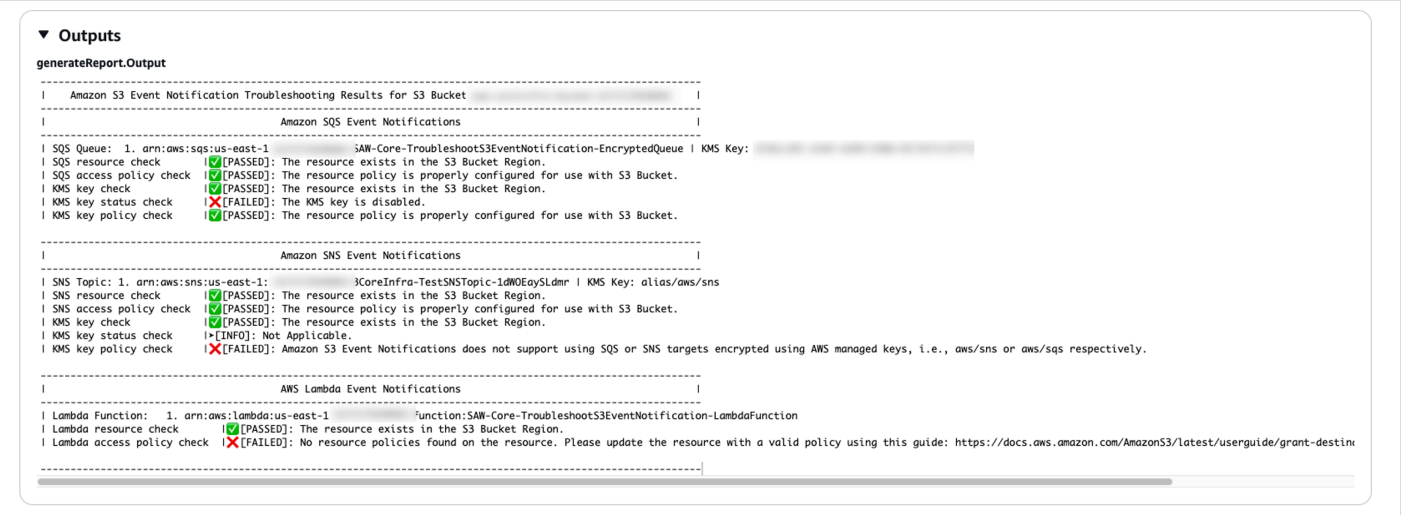
  If there are Amazon SQS destination notifications configured with the Amazon S3 bucket, a list of the Amazon SQS Queues is displayed alongside the results of the checks. The report includes Amazon SQS resource check, Amazon SQS access policy check, Amazon KMS key check, Amazon KMS key status check, and Amazon KMS key policy check.

- **Amazon SNS Event Notifications**

  If there are Amazon SNS destination notifications configured with the Amazon S3 bucket, a list of the Amazon SNS Topics is displayed alongside the results of the checks. The report includes Amazon SNS resource check, Amazon SNS access policy check, Amazon KMS key check, Amazon KMS key status check, and Amazon KMS key policy check.

- **Amazon Lambda Event Notifications**

  If there are Amazon Lambda destination notifications configured with the Amazon S3 bucket, a list of the Lambda functions is displayed alongside the results of the checks. The report includes Lambda resource check and Lambda access policy check.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-ContainS3Resource

Description

The `AWSSupport-ContainS3Resource` runbook provides an automated solution for the procedure outlined in the article Support Automation Workflow (SAW) Runbook: Contain a compromised Amazon Amazon S3 Bucket

Important

- This runbook performs various operations that require elevated privileges, such as modifying Amazon S3 bucket policies, tags, and public access configurations. These actions could potentially lead to privilege escalation or impact other workloads that depend on the targeted Amazon S3 bucket. You should review the permissions granted to the role specified by the `AutomationAssumeRole` parameter and

ensure they are appropriate for the intended use case. You can refer to the following Amazon documentation for more information on IAM permissions: Identity and Access Management (IAM) Permissions Amazon Amazon Systems Manager Automation Permissions.

- This runbook performs mutative actions that could potentially cause unavailability or disruption to your workloads. Specifically, the `Contain` action blocks all access to the specified Amazon S3 bucket, except for the roles specified in the `SecureRoles` parameter. This could impact any applications or services that rely on the targeted Amazon S3 bucket.
- During the `Contain` action, this runbook may create an additional Amazon S3 bucket (specified by the `BackupS3BucketName` parameter) to store the backup of the original bucket's configuration, if it does not already exist.
- If the `Action` parameter is set to `Restore`, this runbook attempts to restore the Amazon S3 bucket's configuration to its original state based on the backup stored in the `BackupS3BucketName` bucket. However, there is a risk that the restoration process may fail, leaving the Amazon S3 bucket in an inconsistent state. The runbook provides instructions for manual restoration in case of such failures, but you should be prepared to handle potential issues during the restoration process.

It is recommended to review the runbook thoroughly, understand its potential impacts, and test it in a non-production environment before executing it in your production environment.

How does it work?

This runbook operates differently based on the resource type and action:

- For Amazon S3 General Purpose Bucket `Containment`: The automation blocks public access to the bucket, disables ACL configuration, enforces Bucket Owner Object ownership, and puts a restrictive bucket policy denying all Amazon S3 actions to the bucket except for allow listed IAM Roles.
- For Amazon S3 General Purpose Object `Containment`: The automation blocks Public Access to bucket, disables ACL configuration, enforces Bucket Owner Object ownership, and puts a restrictive bucket policy denying all Amazon S3 actions on the object except for allow listed IAM Roles.

- For Amazon S3 Directory Bucket `Containment`: The automation puts a restrictive bucket policy denying all Amazon S3 actions to the bucket except for allow listed IAM Roles.
- For Amazon S3 General Purpose Bucket `Restore`: The automation restores the Block Public Access configuration, Bucket ACL configuration, Bucket Owner Object ownership and Bucket Policy to the initial configuration prior to containment.
- For Amazon S3 General Purpose Object `Restore`: The automation restores the Block Public Access configuration, Bucket ACL configuration, Object ACL Configuration, Bucket Owner Object ownership and Bucket Policy to the initial configuration prior to containment.
- For Amazon S3 Directory Bucket `Restore`: The automation restores the bucket policy to the initial configuration prior to containment.

Run this Automation (console)

Document Type

Automation

Owner

Amazon

Platform

/

Required IAM Permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- s3:CreateBucket
- s3:DeleteBucketPolicy
- s3:DeleteObjectTagging
- s3:GetAccountPublicAccessBlock
- s3:GetBucketAcl
- s3:GetBucketLocation
- s3:GetBucketOwnershipControls
- s3:GetBucketPolicy

- s3:GetBucketPolicyStatus
- s3:GetBucketTagging
- s3:GetEncryptionConfiguration
- s3:GetObject
- s3:GetObjectAcl
- s3:GetObjectTagging
- s3:GetReplicationConfiguration
- s3:ListBucket
- s3:PutAccountPublicAccessBlock
- s3:PutBucketACL
- s3:PutBucketOwnershipControls
- s3:PutBucketPolicy
- s3:PutBucketPublicAccessBlock
- s3:PutBucketTagging
- s3:PutBucketVersioning
- s3:PutObject
- s3:PutObjectAcl
- s3express:CreateSession
- s3express:DeleteBucketPolicy
- s3express:GetBucketPolicy
- s3express:PutBucketPolicy
- ssm:DescribeAutomationExecutions

Here is an example of an IAM policy that grants the necessary permissions for the `AutomationAssumeRole`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3Permissions",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:CreateBucket",
                "s3:DeleteBucketPolicy",
                "s3:DeleteObjectTagging",
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketOwnershipControls",
                "s3:GetBucketPolicy",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketTagging",
                "s3:GetEncryptionConfiguration",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:GetReplicationConfiguration",
                "s3:ListBucket",
                "s3:PutAccountPublicAccessBlock",
                "s3:PutBucketACL",
                "s3:PutBucketOwnershipControls",
                "s3:PutBucketPolicy",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketTagging",
                "s3:PutBucketVersioning",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3ExpressPermissions",
            "Effect": "Allow",
            "Action": [
                "s3express:CreateSession",
                "s3express:DeleteBucketPolicy",
                "s3express:GetBucketPolicy",
                "s3express:PutBucketPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "SSMPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAutomationExecutions"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

Instructions

Follow these steps to configure the automation:

1. Navigate to AWSSupport-ContainS3Resource in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:

- **BucketName (Required):**
  - Description: (Required) The name of the Amazon S3 bucket.
  - Type: `AWS::S3::Bucket::Name`
- **Action (Required):**
  - Description: (Required) Select `Contain` to isolate the Amazon S3 resource or `Restore` to try to restore the resource configuration to its original state from a previous backup.
  - Type: String
  - Allowed Values: `Contain|Restore`
- **DryRun (Optional):**
  - Description: (Optional) When set to true, the automation will not make any changes to the target Amazon S3 resource, instead it will output what it would have attempted to change. Default value: true.
  - Type: Boolean
  - Allowed Values: `true|false`
- **BucketKeyName (Optional):**
  - Description: (Optional) The key of the Amazon S3 object you want to contain or restore. Used during object level containment.
  - Type: String
  - Allowed Pattern: `^[a-zA-Z0-9\\.\\-_\\\\!*'()/]{0,1024}$`
- **BucketRestrictAccess (Conditional):**

- Description: (Conditional) The ARN of the IAM users or roles that will be allowed access to the target Amazon S3 resource after running the containment actions. This parameter is required when `Action` is set to `Contain`.
- Type: StringList
- Allowed Pattern: `^$|^arn:(aws|aws-cn|aws-us-gov|aws-iso(-[a-z])?):iam::[0-9]{12}:(role|user)\\/[\\w+\\/=,.@-]+$`

- **TagIdentifier (Optional):**
  - Description: (Optional) A tag in the format Key=BatchId,Value=78925 that will be added to the resources created or modified by this runbook during the containment workflow.
  - Type: String
  - Allowed Pattern: `^$|^[Kk][Ee][Yy]=[\\+\\-\\=\\.\\_\\:\\/@a-zA-Z0-9]{1,128},[Vv][Aa][Ll][Uu][Ee]=[\\+\\-\\=\\.\\_\\:\\/@a-zA-Z0-9]{0,128}$`
- **BackupS3BucketName (Conditional):**
  - Description: (Conditional) The Amazon S3 bucket to backup the target resource configuration when the `Action` is set to `Contain` or to restore the configuration from when the `Action` is set to `Restore`.
  - Type: `AWS::S3::Bucket::Name`
- **BackupS3KeyName (Conditional):**
  - Description: (Conditional) If `Action` is set to `Restore`, this specifies the Amazon S3 key the automation will use to try to restore the target resource configuration.
  - Type: String
  - Allowed Pattern: `^[a-zA-Z0-9\\.\\-_\\\\!*'()/]{0,1024}$`
- **BackupS3BucketAccess (Conditional):**
  - Description: (Conditional) The ARN of the IAM users or roles that will be allowed access to the backup Amazon S3 bucket after running the containment actions. This parameter is required when `Action` is `Contain`.
  - Type: StringList
  - Allowed Pattern: `^$|^arn:(aws|aws-cn|aws-us-gov|aws-iso(-[a-z])?):iam::[0-9]{12}:(role|user)\\/[\\w+\\/=,.@-]+$`
- **AutomationAssumeRole (Optional):**

- Description: (Optional) The Amazon Resource Name (ARN) of the IAM role that allows Systems Manager Automation to perform the actions on your behalf.
- Type: `AWS::IAM::Role::Arn`

4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

- **validateRequiredInputs**

  Validates the required automation input parameters based on the Action specified.

- **assertBucketExists**

  Checks if the target Amazon S3 bucket exists and is accessible.

- **backupBucketPreChecks**

  Checks if the backup Amazon S3 bucket potentially grants public read or write access to its objects.

- **backupTargetBucketMetadata**

  Describes the current configuration of the target Amazon S3 bucket and uploads the backup to the specified backup Amazon S3 bucket.

- **containBucket**

  Performs bucket level operations to contain the target Amazon S3 bucket.

- **BranchOnActionAndMode**

  Branches the automation based on the input parameters Action and DryRun.

- **RestoreInstanceConfiguration**

  Restores the Amazon S3 bucket configuration from the backup.

- **containFinalOutput**

  Consolidates containment activity in readable format.

- **ReportContain**

  Outputs dry run details for the containment actions.

- **ReportRestore**

Outputs dry run details for the restoring actions.

- **ReportRestoreFailure**

  Provides instructions to restore the Amazon S3 bucket original configuration during a restore workflow failure scenario.

- **ReportContainmentFailure**

  Provides instructions to restore the Amazon S3 bucket original configuration during a containment workflow failure scenario.

- **FinalOutput**

  Outputs the details of the containment actions.

7. After the execution completes, review the Outputs section for the detailed results of the execution:

   - **ContainFinalOutput.Output**

     Outputs the details of the containment actions performed by this runbook when `DryRun` is set to False.

   - **RestoreFinalOutput.Output**

     Outputs the details of the restore actions performed by this runbook when `DryRun` is set to False.

   - **ContainS3ResourceDryRun.Output**

     Outputs the details of the containment actions performed by this runbook when `DryRun` is set to True.

   - **RestoreS3ResourceDryRun.Output**

     Outputs the details of the restore actions performed by this runbook when `DryRun` is set to True.

   - **ReportContainmentFailure.Output**

     Provides instructions to restore the target Amazon S3 resource original configuration during a containment workflow failure scenario.

   - **ReportRestoreFailure.Output**

Provides instructions to restore the target Amazon S3 resource original configuration during a restore workflow failure scenario.

References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows

Amazon SES

Amazon Systems Manager Automation provides predefined runbooks for Amazon Simple Email Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSSupport-AnalyzeSESMessageSendingStatus

AWSSupport-AnalyzeSESMessageSendingStatus

Description

The `AWSSupport-AnalyzeSESMessageSendingStatus` automation runbook summarizes the email delivery status of undelivered email messages and gives you advice to solve why it was undelivered. The runbook retrieves Amazon Simple Email Service (Amazon SES) email sending events stored in an Amazon CloudWatch Logs group published by Amazon SES. For Amazon SES event publishing details, please refer to Monitoring using Amazon Simple Email Service event publishing. The runbook also provides a summary and the timeline of the email deliveries as well as recommendations which can potentially affect undelivered email messages. You can find those messages in the output section of each executions. Please note that this runbook can only troubleshoot the events after the event store deployment.

How does it work?

The runbook performs the following steps:

- Checks concurrent automation executions for the same CloudWatch Logs group.
- Analyze Amazon SES events corresponding to message IDs given by the automation parameter.
- Output delivery summaries to the output section of the automation execution.

> **Important**
>
> - Before executing this runbook, you have to store published Amazon SES events to a CloudWatch Logs log group specified by the automation parameter. This runbook only analyzes Amazon SES events stored in the log group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `logs:StartQuery`
- `logs:GetQueryResults`
- `ses:GetIdentityMailFromDomainAttributes`

- `ses:GetSendQuota`
- `ssm:DescribeAutomationExecutions`
- `ssm:GetAutomationExecution`

Instructions

Follow these steps to configure the automation:

1. Navigate to `AWSSupport-AnalyzeSESMessageSendingStatus` in Systems Manager under Documents.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **AutomationAssumeRole (Optional):**

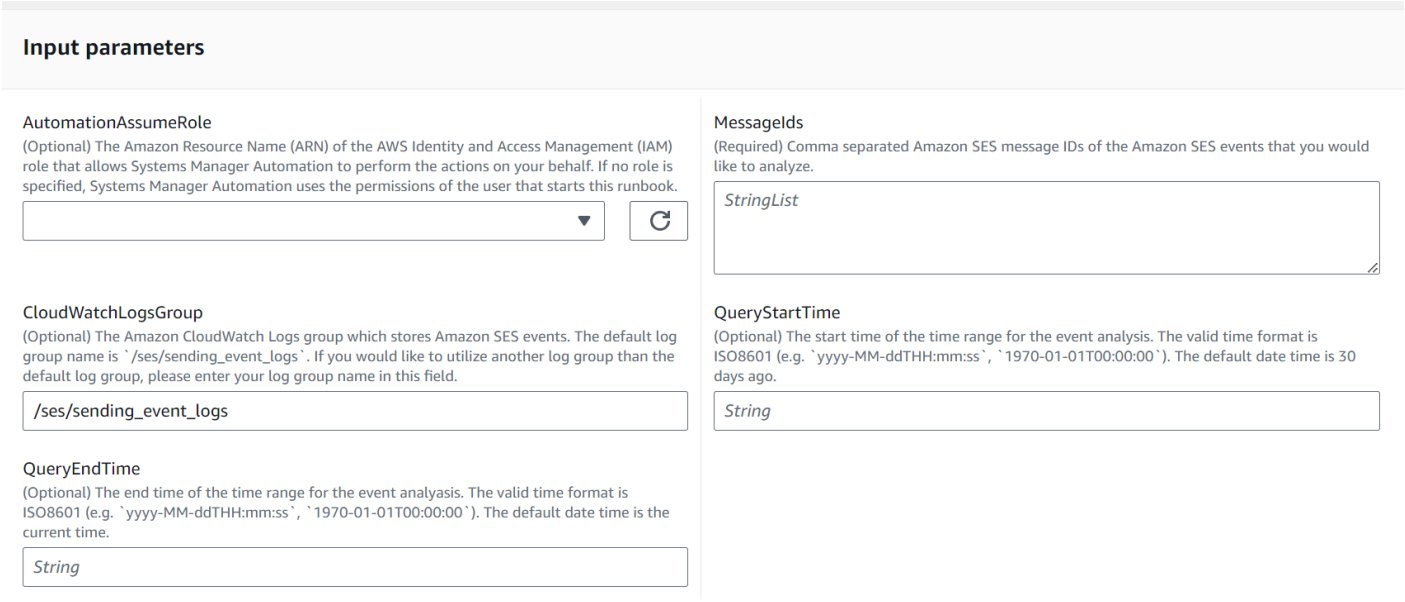
     The Amazon Resource Name (ARN) of the Amazon Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user who starts this runbook.
   - **MessageIds (Required)**

     Comma separated Amazon Simple Email Service message IDs of the Amazon Simple Email Service events that you would like to analyze.
   - **CloudWatchLogsGroup (Optional)**

     The Amazon CloudWatch Logs group which stores Amazon Simple Email Service events. The default log group name is `/ses/sending_event_logs`. If you would like to utilize another log group than the default log group, please enter your log group name in this field.",
   - **QueryStartTime (Optional)**

     The start time of the time range for the event analysis. The valid time format is ISO8601 (e.g. `yyyy-MM-ddTHH:mm:ss`, `1970-01-01T00:00:00`). The default date time is 30 days ago.
   - **QueryEndTime (Optional)**

     The end time of the time range for the event analyasis. The valid time format is ISO8601 (e.g. `yyyy-MM-ddTHH:mm:ss`, `1970-01-01T00:00:00`). The default date time is the current time.



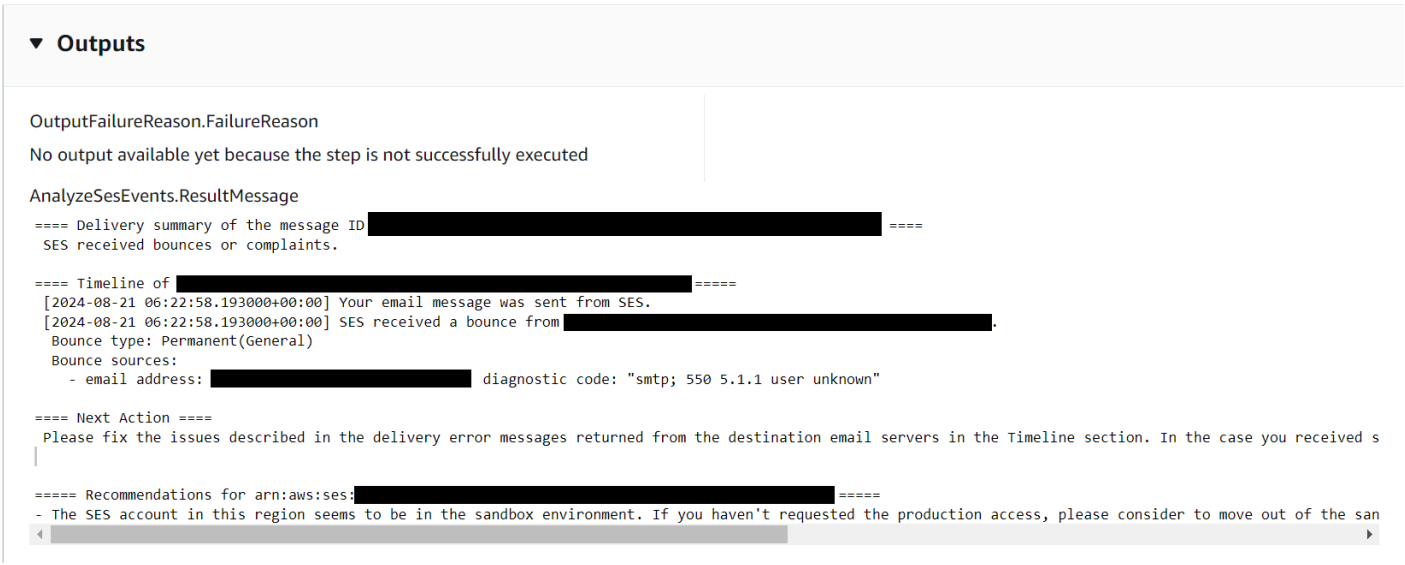
4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:
   - **`CheckConcurrency:`**

     Ensures that there is only one execution of this runbook targeting the Amazon CloudWatch Logs group. If the runbook finds another execution targeting the same log group, it returns an error and ends.
   - **`AnalyzeSesEvents:`**

     Analyze Amazon Simple Email Service events stored in the Amazon CloudWatch Logs group specified by the automation parameter.
   - **`OutputFailureReason:`**

     Output execution step failure messages when the `AnalyzeSESMessageSendingStatus` step failed.
7. After completed, review the Outputs section for the detailed results of the execution:
   - **Output of analysis on an undelivered email message because of a bounce**

     Output of an automation execution for an email message that didn't reach the destination mailbox because of a bounce.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

SageMaker AI

Amazon Systems Manager Automation provides predefined runbooks for Amazon SageMaker AI. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-DisableSageMakerNotebookRootAccess

AWS-DisableSageMakerNotebookRootAccess

Description

The `AWS-DisableSageMakerNotebookRootAccess` runbook disables root access on a Amazon SageMaker AI notebook instance. During the automation, the notebook instance is stopped to make the required changes. SageMaker AI Studio notebook instances aren't supported.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- NotebookInstanceName

  Type: String

  Description: (Required) The name of the SageMaker AI notebook instance to disable root access on.
- StartInstanceAfterUpdate

  Type: Boolean

  Default: true

  Description: (Optional) Determines whether the notebook instance is started after disabling root access. The default setting for this parameter is `true`. If set to `true`, the instance is started after

root access is disabled. If set to `false`, the instance is left in the `stopped` state after root access is disabled.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `sagemaker:DescribeNotebookInstance`
- `sagemaker:StartNotebookInstance`
- `sagemaker:StopNotebookInstance`
- `sagemaker:UpdateNotebookInstance`

Document Steps

- CheckNotebookInstanceStatus (aws:executeAwsApi): Checks the current status of the notebook instance.
- StopOrUpdateNotebookInstance (aws:branch): Branches based on the status of the notebook instance.
- StopNotebookInstance (aws:executeAwsApi): Starts the instance if the status is `stopped`.
- WaitForInstanceToStop (aws:waitForAwsResourceProperty): Verifies the instance is `stopped`.
- UpdateNotebookInstance (aws:executeAwsApi): Disables root access on the notebook instance.
- WaitForNotebookUpdate (aws:waitForAwsResourceProperty): Verifies root access has been disabled and the instance has a `stopped` status.
- ChooseInstanceStart (aws:branch): Branch based on whether the instance should be started.
- StartNotebookInstance (aws:executeAwsApi): Starts the notebook instance.
- VerifyNotebookInstanceStatus (aws:waitForAwsResourceProperty): Verifies if the instance is `available` before disabling root access.
- VerifyNotebookInstanceRootAccess (aws:assertAwsResourceProperty): Verifies the notebook instance root access setting is successfully disabled.

Secrets Manager

Amazon Systems Manager Automation provides predefined runbooks for Amazon Secrets Manager. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

`AWSConfigRemediation-DeleteSecret`

Description

The `AWSConfigRemediation-DeleteSecret` runbook deletes a secret and all of the versions stored in Amazon Secrets Manager. You can optionally specify the recovery window during which you can restore the secret. If you don't specify a value for the `RecoveryWindowInDays` parameter, the operation defaults to 30 days.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- RecoveryWindowInDays

  Type: Integer

  Valid values: 7-30

  Default: 30

  Description: (Optional) The number of days which you can restore the secret.

- SecretId

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the secret you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `secretsmanager:DeleteSecret`
- `secretsmanager:DescribeSecret`

Document Steps

- `aws:executeAwsApi` - Deletes the secret you specify in the `SecretId` parameter.
- `aws:executeScript` - Verifies the secret has been scheduled for deletion.

AWSConfigRemediation-RotateSecret

Description

The `AWSConfigRemediation-RotateSecret` runbook rotates a secret stored in Amazon Secrets Manager.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- RotationInterval

  Type: Interval

  Valid values: 1-365

  Description: (Required) The number of days between rotations of the secret.
- RotationLambdaArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Lambda funtion that can rotate the secret.
- SecretId

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the secret you want to rotate.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `lambda:InvokeFunction`
- `secretsmanager:DescribeSecret`
- `secretsmanager:RotateSecret`

Document Steps

- `aws:executeAwsApi` - Rotates the secret you specify in the `SecretId` parameter.
- `aws:executeScript` - Verifies rotation has been enabled on the secret.

Security Hub

Amazon Systems Manager Automation provides predefined runbooks for Amazon Security Hub. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSConfigRemediation-EnableSecurityHub

AWSConfigRemediation-EnableSecurityHub

Description

The `AWSConfigRemediation-EnableSecurityHub` runbook enables Amazon Security Hub (Security Hub) for the Amazon Web Services account and Amazon Web Services Region where you run the automation. For information about Security Hub, see What is Amazon Security Hub? in the *Amazon Security Hub User Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- EnableDefaultStandards

  Type: Boolean

  Default: true

  Description: (Required) If set to `true` , the default security standards designated by Security Hub are enabled.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `securityhub:DescribeHub`
- `securityhub:EnableSecurityHub`
- `ssm:StartAutomationExecution`

- `ssm:GetAutomationExecution`

Document Steps

- `aws:executeAwsApi` - Enables Security Hub in the current account and Region.
- `aws:executeAwsApi` - Verifies that Security Hub has been enabled.

Amazon Shield

Amazon Systems Manager Automation provides predefined runbooks for Amazon Shield. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWSPremiumSupport-DDoSResiliencyAssessment

`AWSPremiumSupport-DDoSResiliencyAssessment`

Description

The `AWSPremiumSupport-DDoSResiliencyAssessment`, Amazon Systems Manager automation runbook helps you to check DDoS vulnerabilities and configuration of resources in accordance with the Amazon Shield Advanced protection for your Amazon Web Services account. It provides a configuration settings report for resources that are vulnerable to Distributed Denial of Service (DDoS) attacks. It is used to collect, analyze, and assess the following resources: Amazon Route 53, Amazon Load Balancers, Amazon CloudFront distributions, Amazon Global Accelerator and Amazon Elastic IPs for their configuration settings in accordance with the recommended best practices for Amazon Shield Advanced Protection. The final configuration report is available in an Amazon S3 bucket of your choice as an HTML file.

How does it work?

This runbook contains a series of checks for the various types of resources that are enabled for public access and if they have protections configured as per the recommendations in the Amazon DDoS Best Practices Whitepaper. The runbook performs the following:

- Checks if a subscription to Amazon Shield Advanced is enabled.

- If enabled, it finds if there are any Shield Advanced protected resources.
- It finds all the global and regional resources in the Amazon Web Services account and checks if these are Shield protected.
- It requires the Resource Type parameters for assessment, Amazon S3 bucket name, and the Amazon S3 bucket Amazon Web Services account ID (S3BucketOwner).
- It returns the findings as an HTML report stored in the Amazon S3 bucket provided.

The input parameters `AssessmentType` decides if the checks on all resources will be performed. By default, the runbook checks for all types of resources. If only `GlobalResources` or `RegionalResources` parameter is selected, the runbook performs checks only on the selected resource types.

> **Important**
>
> - Access to `AWSPremiumSupport-*` runbooks requires an Enterprise or Business Support subscription. For more information, see Compare Amazon Web Services Support Plans.
> - This runbook requires an `ACTIVE` Amazon Shield Advanced subscription.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AssessmentType

  Type: String

  Description: (Optional) Determines the type of resources to be evaluated for DDoS resiliency assessment. By default, the runbook will evaluate both global and regional resources. For regional resources, the runbook describes all Application (ALB) and Network (NLB) load balancers as well as all the Auto Scaling group in your Amazon Web Services account/region.

  Valid values: `['Global Resources', 'Regional Resources', 'Global and Regional Resources']`

  Default: Global and Regional Resources

- S3BucketName

  Type: `AWS::S3::Bucket::Name`

  Description: (Required) The Amazon S3 bucket name where the report will be uploaded.

  Allowed Pattern: `^[0-9a-z][a-z0-9\-\.]{3,63}$`

- S3BucketOwnerAccount

  Type: String

  Description: (Optional) The Amazon Web Services account that owns the Amazon S3 bucket. Please specify this parameter if the Amazon S3 bucket belongs to a different Amazon Web Services account, otherwise you can leave this parameter empty.

  Allowed Pattern: `^$|^[0-9]{12,13}$`

- S3BucketOwnerRoleArn

  Type: `AWS::IAM::Role::Arn`

  Description: (Optional) The ARN of an IAM role with permissions to describe the Amazon S3 bucket and Amazon Web Services account block public access configuration if the bucket is in a

different Amazon Web Services account. If this parameter is not specified, the runbook uses the AutomationAssumeRole or the IAM user that starts this runbook (if AutomationAssumeRole is not specified). Please see the required permissions section in the runbook description.

Allowed Pattern: ^$|^arn:(aws|aws-cn|aws-us-gov|aws-iso|aws-iso-b):iam::[0-9]{12,13}:role/.*$

- S3BucketPrefix

  Type: String

  Description: (Optional) The prefix for the path inside Amazon S3 for storing the results.

  Allowed Pattern: ^[a-zA-Z0-9][-./a-zA-Z0-9]{0,255}$|^$

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- autoscaling:DescribeAutoScalingGroups
- cloudfront:ListDistributions
- ec2:DescribeAddresses
- ec2:DescribeNetworkAcls
- ec2:DescribeInstances
- elasticloadbalancing:DescribeLoadBalancers
- elasticloadbalancing:DescribeTargetGroups
- globalaccelerator:ListAccelerators
- iam:GetRole
- iam:ListAttachedRolePolicies
- route53:ListHostedZones
- route53:GetHealthCheck
- shield:ListProtections
- shield:GetSubscriptionState
- shield:DescribeSubscription
- shield:DescribeEmergencyContactSettings

- shield:DescribeDRTAccess
- waf:GetWebACL
- waf:GetRateBasedRule
- wafv2:GetWebACL
- wafv2:GetWebACLForResource
- waf-regional:GetWebACLForResource
- waf-regional:GetWebACL
- s3:ListBucket
- s3:GetBucketAcl
- s3:GetBucketLocation
- s3:GetBucketPublicAccessBlock
- s3:GetBucketPolicyStatus
- s3:GetBucketEncryption
- s3:GetAccountPublicAccessBlock
- s3:PutObject

Example IAM Policy for the Automation Assume Role

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetAccountPublicAccessBlock"
                ],
                "Resource": "*",
                "Effect": "Allow"
            },
            {
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation",
```

```
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketPolicyStatus",
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "arn:aws:s3:::<bucket-name>",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::<bucket-name>/*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "cloudfront:ListDistributions",
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:DescribeNetworkAcls",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeTargetGroups",
                "globalaccelerator:ListAccelerators",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "route53:ListHostedZones",
                "route53:GetHealthCheck",
                "shield:ListProtections",
                "shield:GetSubscriptionState",
                "shield:DescribeSubscription",
                "shield:DescribeEmergencyContactSettings",
                "shield:DescribeDRTAccess",
                "waf:GetWebACL",
                "waf:GetRateBasedRule",
                "wafv2:GetWebACL",
                "wafv2:GetWebACLForResource",
                "waf-regional:GetWebACLForResource",
                "waf-regional:GetWebACL"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
```

```
                                "Action": "iam:PassRole",
                                "Resource": "arn:aws:iam::123456789012:role/
<AutomationAssumeRole-Name>",
                                "Effect": "Allow"
                            }
                        ]
                    }
```

Instructions

1. Navigate to the AWSPremiumSupport-DDoSResiliencyAssessment in the Amazon Systems Manager Console.
2. Select **Execute Automation**
3. For input parameters, enter the following:

- **AutomationAssumeRole (Optional):**

  The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- **AssessmentType (Optional):**

  Determines the type of resources to evaluate for DDoS resiliency assessment. By default, the runbook evaluates both global and regional resources.

- **S3BucketName (Required):**

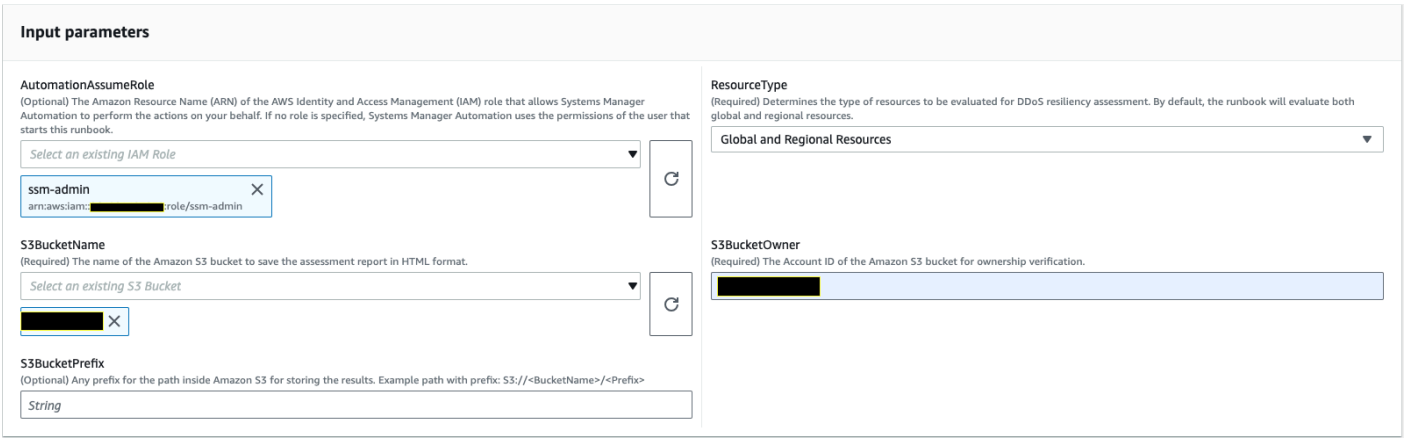
  The name of the Amazon S3 bucket to save the assessment report in HTML format.

- **S3BucketOwner (Optional):**

  The Amazon Web Services account ID of the Amazon S3 bucket for ownership verification. The Amazon Web Services account ID is required if the report needs to publish to a cross-account Amazon S3 bucket and optional if the Amazon S3 bucket is in the same Amazon Web Services account as automation initiation.

- **S3BucketPrefix (Optional):**

  Any prefix for the path inside Amazon S3 for storing the results.



4. Select **Execute**.
5. The automation initiates.
6. The document performs the following steps:

- **CheckShieldAdvancedState:**

  Checks if the Amazon S3 bucket specified in the "S3BucketName" allows anonymous, or public read or write access permissions, whether the bucket has encryption at rest enabled, and if the Amazon Web Services account ID provided in "S3BucketOwner" is the owner of the Amazon S3 bucket.

- **S3BucketSecurityChecks:**

  Checks if the Amazon S3 bucket specified in the "S3BucketName" allows anonymous, or public read or write access permissions, whether the bucket has encryption at rest enabled, and if the Amazon Web Services account ID provided in "S3BucketOwner" is the owner of the Amazon S3 bucket.

- **BranchOnShieldAdvancedStatus:**

  Branches document steps based on the Amazon Shield Advanced Subscription status and/or Amazon S3 Bucket Ownership status.

- **ShieldAdvancedConfigurationReview:**

  Reviews Shield Advanced configurations to ensure minimum required details are present. For example: IAM Access for Amazon Shield Response Team (SRT) Team, Contact List Details, and SRT Proactive Engagement Status.

- **ListShieldAdvancedProtections:**

Lists the Shield Protected Resources and creates a group of protected resources for each service.

- **BranchOnResourceTypeAndCount:**

  Branches document steps based on the value of Resource Type parameter and the number of Shield protected global resources.

- **ReviewGlobalResources:**

  Reviews the Shield Advanced protected Global resources like Route 53 Hosted Zones, CloudFront Distributions and Global Accelerators.

- **BranchOnResourceType:**

  Branches document steps based on the Resource type selections, if Global, Regional, or both.

- **ReviewRegionalResources:**

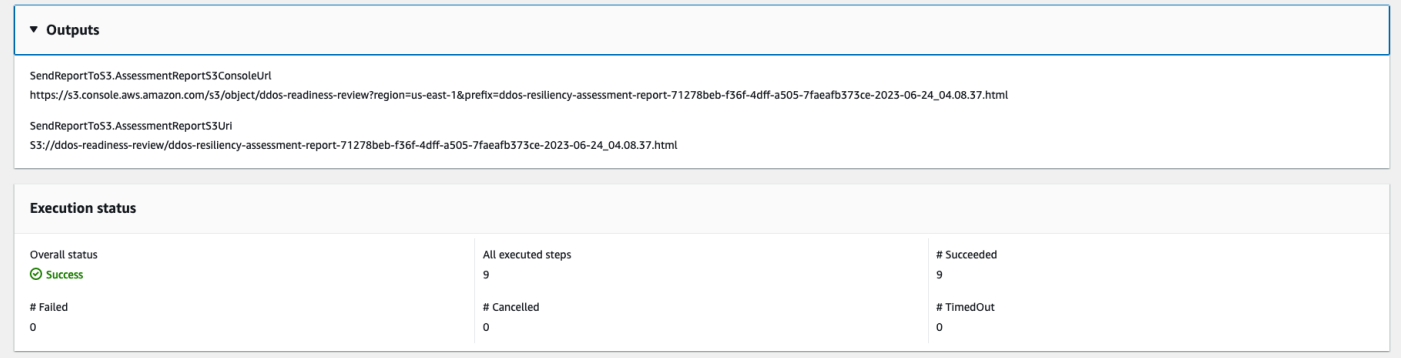
  Reviews the Shield Advanced protected Regional resources like Application Load Balancers, Network Load Balancers, Classic Load Balancers, Amazon Elastic Compute Cloud (Amazon EC2) Instances (Elastic IPs).

- **SendReportToS3:**

  Uploads the DDoS Assessment Report details to the Amazon S3 bucket.

7. After completed, the URI for the assessment report HTML file is provided in the Amazon S3 bucket:

S3 Console link and Amazon S3 URI for the Report on successful execution of the runbook



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- Amazon Shield Advanced

Amazon SNS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Simple Notification Service. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-EnableSNSTopicDeliveryStatusLogging
- AWSConfigRemediation-EncryptSNSTopic
- AWS-PublishSNSNotification

AWS-EnableSNSTopicDeliveryStatusLogging

Description

The `AWS-EnableSNSTopicDeliveryStatusLogging` runbook configures delivery status logging for a `HTTP`, Amazon Data Firehose, Lambda, `Platform application`, or Amazon Simple Queue Service (Amazon SQS) endpoint. This allow Amazon SNS to log failed alerts and a sample percentage of successful alert notifications to Amazon CloudWatch. If delivery status logging is already configured for the topic, the runbook replaces the existing configuration with the new values you specify for the input parameters.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- EndpointType

  Type: String

  Valid values:
  - HTTP
  - Firehose
  - Lambda
  - Application
  - SQS

  Description: (Required) The type of Amazon SNS topic endpoint you want to log delivery status notification messages for.
- TopicArn

  Type: String

  Description: (Required) The ARN of the Amazon SNS topic you want to configure delivery status logging for.
- SuccessFeedbackRoleArn

  Type: String

Description: (Required) The ARN of the IAM role which Amazon SNS uses to send logs for successful notification messages to CloudWatch.

- SuccessFeedbackSampleRate

  Type: String

  Valid values: 0-100

  Description: (Required) The percentage of successful messages to sample for the specified Amazon SNS topic.

- FailureFeedbackRoleArn

  Type: String

  Description: (Required) The ARN of the IAM role which Amazon SNS uses to send logs for failure notification messages to CloudWatch.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:PassRole`
- `sns:GetTopicAttributes`
- `sns:SetTopicAttributes`

Document Steps

- `aws:executeAwsApi` - Applies the value for the `SuccessFeedbackRoleArn` parameter to the Amazon SNS topic.
- `aws:executeAwsApi` - Applies the value for the `SuccessFeedbackSampleRate` parameter to the Amazon SNS topic.
- `aws:executeAwsApi` - Applies the value for the `FailureFeedbackRoleArn` parameter to the Amazon SNS topic.

- `aws:executeScript` - Confirms delivery status logging is enabled on the Amazon SNS topic.

Outputs

VerifyDeliveryStatusLoggingEnabled.GetTopicAttributesResponse - Response from the `GetTopicAttributes` API operations.

VerifyDeliveryStatusLoggingEnabled.VerifyDeliveryStatusLoggingEnabled - Message indicating successful verification of delivery status logging.

`AWSConfigRemediation-EncryptSNSTopic`

Description

The `AWSConfigRemediation-EncryptSNSTopic` runbook enables encryption on the Amazon Simple Notification Service (Amazon SNS) topic you specify using an Amazon Key Management Service (Amazon KMS) customer managed key. This runbook should only be used as a baseline to ensure that your Amazon SNS topics are encrypted according to minimum recommended security best practices. We recommend encrypting multiple topics with different customer managed keys.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- KmsKeyArn

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon KMS customer managed key you want to use to encrypt the Amazon SNS topic.

- TopicArn

  Type: String

  Description: (Required) The ARN of the Amazon SNS topic you want to encrypt.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `sns:GetTopicAttributes`
- `sns:SetTopicAttributes`

Document Steps

- `aws:executeAwsApi` - Encrypts the Amazon SNS topic you specify in the `TopicArn` parameter.
- `aws:assertAwsResourceProperty` - Confirms encryption is enabled on the Amazon SNS topic.

AWS-PublishSNSNotification

Description

Publish a notification to Amazon SNS.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Message

  Type: String

  Description: (Required) The message to include in the SNS notification.

- TopicArn

  Type: String

  Description: (Required) The ARN of the SNS topic to publish the notification to.

Amazon SQS

Amazon Systems Manager Automation provides predefined runbooks for Amazon Simple Queue Service (Amazon SQS). For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-EnableSQSEncryption

AWS-EnableSQSEncryption

Description

The AWS-EnableSQSEncryption runbook enables encryption at rest for an Amazon Simple Queue Service (Amazon SQS) queue. An Amazon SQS queue can be encrypted with Amazon SQS managed keys (SSE-SQS), or with Amazon Key Management Service (Amazon KMS) managed keys (SSE-KMS). The key that you assign to your queue must have a key policy that includes permissions for all principals that are authorized to use the queue. With encryption enabled, anonymous SendMessage and ReceiveMessage requests to the encrypted queue are rejected.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- QueueUrl

  Type: String

  Description: (Required) The URL of the Amazon SQS queue you want to enable encryption on.

- KmsKeyId

  Type: String

  Description: (Optional) The Amazon KMS key to use for encryption. This value can be a globally unique identifier, an ARN to either an alias or a key, or an alias name prefixed by "alias/". You can also use the Amazon managed key by specifying the alias aws/sqs.

- KmsDataKeyReusePeriodSeconds

  Type: String

  Valid values: 60-86400

  Default: 300

  Description: (Optional) The length of time, in seconds, an Amazon SQS queue can reuse a data key to encrypt or decrypt messages before calling Amazon KMS again.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `sqs:GetQueueAttributes`
- `sqs:SetQueueAttributes`

Document Steps

- SelectKeyType (`aws:branch`): Branches based on the key specified.
- PutAttributeSseKms (`aws:executeAwsApi`) - Updates the Amazon SQS queue to use the Amazon KMS key specified for encryption.
- PutAttributeSseSqs (`aws:executeAwsApi`) - Updates the Amazon SQS queue to use the default key for encryption.
- VerifySqsEncryptionKms (`aws:assertAwsResourceProperty`) - Verifies encryption is enabled on the Amazon SQS queue.

- VerifySqsEncryptionDefault (`aws:assertAwsResourceProperty`) - Verifies encryption is enabled on the Amazon SQS queue.

Step Functions

Amazon Systems Manager Automation provides predefined runbooks for Amazon Step Functions (Step Functions). For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-EnableStepFunctionsStateMachineLogging

AWS-EnableStepFunctionsStateMachineLogging

Description

The `AWS-EnableStepFunctionsStateMachineLogging` runbook enables or updates logging on the Amazon Step Functions state machine you specify. The minimum logging level must be set to `ALL`, `ERROR`, or `FATAL`.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Level

  Type: String

  Valid values: ALL | ERROR | FATAL

  Description: (Required) The URL of the Amazon SQS queue you want to enable encryption on.

- LogGroupArn

  Type: String

  Description: (Required) The ARN of the Amazon CloudWatch Logs log group you want to send state machine logs to.

- StateMachineArn

  Type: String

  Description: (Required) The ARN of the state machine you want enable logging on.

- IncludeExecutionData

  Type: Boolean

  Default: False

  Description: (Optional) Determines whether execution data is included in the logs.

- TracingConfiguration

  Type: Boolean

  Default: False

  Description: (Optional) Determines whether Amazon X-Ray tracing is enabled.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `states:DescribeStateMachine`
- `states:UpdateStateMachine`

Document Steps

- `EnableStepFunctionsStateMachineLogging (aws:executeAwsApi)` - Updates the specified state machine with the logging configuration specified.
- `VerifyStepFunctionsStateMachineLoggingEnabled (aws:assertAwsResourceProperty)` - Verifies logging was enabled for the specified state machine.

Outputs

- EnableStepFunctionsStateMachineLogging.Response - Response from the UpdateStateMachine API call.

Systems Manager

Amazon Systems Manager Automation provides predefined runbooks for Systems Manager. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-BulkDeleteAssociation
- AWS-BulkEditOpsItems
- AWS-BulkResolveOpsItems
- AWS-ConfigureMaintenanceWindows
- AWS-CreateManagedLinuxInstance
- AWS-CreateManagedWindowsInstance

- AWSConfigRemediation-EnableCWLoggingForSessionManager
- AWS-ExportOpsDataToS3
- AWS-ExportPatchReportToS3
- AWS-SetupInventory
- AWS-SetupManagedInstance
- AWS-SetupManagedRoleOnEC2Instance
- AWSSupport-TroubleshootManagedInstance
- AWSSupport-TroubleshootPatchManagerLinux
- AWSSupport-TroubleshootSessionManager

AWS-BulkDeleteAssociation

Description

The `AWS-BulkDeleteAssociation` runbook helps you to delete up to 50 Systems Manager State Manager associations at a time.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AssociationIds

  Type: StringList

  Description: (Required) A comma-separated list of the IDs of the associations you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DeleteAssociation`

Document Steps

- `aws:executeScript` - Deletes the associations you specify in the `AssociationIds` parameter.

AWS-BulkEditOpsItems

Description

The `AWS-BulkEditOpsItems` runbook helps you edit the status, severity, category, or priority of Amazon Systems Manager OpsItems. This automation can edit a maximum of 50 OpsItems at a time.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Category

  Type: String

  Valid values:

  - Availability
  - Cost
  - No change
  - Performance
  - Recovery
  - Security

  Default: No change

  Description: (Optional) The new category you want to specify for the edited OpsItems.

- OpsItemIds

  Type: StringList

  Description: (Required) A comma-separated list of OpsItems IDs you want to edit (for example, oi-XXXXXXXXXXXX,oi-XXXXXXXXXXXX).

- Priority

  Type: String

  Valid values:

  - No change
  - 1

- 2
- 3
- 4
- 5

Default: No change

Description: (Optional) The importance of the edited OpsItems in relation to other OpsItems in the system.

- Severity

  Type: String

  Valid values:

  - No change
  - 1
  - 2
  - 3
  - 4

  Default: No change

  Description: (Optional) The severity of the edited OpsItems.

- WaitTimeBetweenEditsInSecs

  Type: String

  Valid values: 0.0-2.0

  Default: 0.8

  Description: (Optional) The time the automation waits between calling the `UpdateOpsItems` operation.

- Status

  Type: String

  Valid values:

- InProgress
- No change
- Open
- Resolved

Default: No change

Description: (Optional) The new status of the edited OpsItems.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:UpdateOpsItem`

Document Steps

- `aws:executeScript` - Edits the OpsItems you specified in the `OpsItemIds` parameter based on the values you specify for the `Category` , `Priority` , `Severity` , and `Status` parameters.

AWS-BulkResolveOpsItems

Description

The `AWS-BulkResolveOpsItems` runbook resolves Amazon Systems Manager OpsItems that match the filter you specify. You can also specify an OpsItemId to add to the resolved OpsItems using the `OpsInsightsId` parameter. If you specify a value for the `S3BucketName` parameter, a result summary is sent to the Amazon Simple Storage Service (Amazon S3) bucket. To receive a notification once the result summary has been sent to the Amazon S3 bucket, specify a value for the `SnsTopicArn` parameter. This automation will resolve a maximum of 1,000 OpsItems at a time.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Filters

  Type: String

  Description: (Required) The key-value pairs of filters to return the OpsItems you want to resolve. For example, `[{"Key": "Status", "Values": ["Open"], "Operator": "Equal"}]`. To learn more about the options available for filtering OpsItems responses, see OpsItemFilters in the *Amazon Systems Manager API Reference* .

- OpsInsightId

  Type: String

  Description: (Optional) The related resource identifier you want to add to resolved OpsItems.

- S3BucketName

  Type: String

  Description: (Optional) The name of the Amazon S3 bucket you want to send the result summary to.

- SnsMessage

  Type: String

  Description: (Optional) The notification you want Amazon Simple Notification Service (Amazon SNS) to send when the automation completes.

- SnsTopicArn

  Type: String

  Description: (Optional) The ARN of the Amazon SNS topic you want to notify when the result summary has been sent to Amazon S3.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `s3:GetBucketAcl`
- `s3:PutObject`
- `sns:Publish`
- `ssm:DescribeOpsItems`
- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:UpdateOpsItem`

Document Steps

- `aws:executeScript` - Gathers and resolves the OpsItems based on the filters you specify. If you specified a value for the `OpsInsightId` parameter, the value is added as a related resource.
- `aws:executeScript` - If you specified a value for the `S3BucketName` parameter, a result summary is then sent to the Amazon S3 bucket.
- `aws:executeScript` - If you specified a value for the `SnsTopicArn` parameter, a notification is sent to the Amazon SNS topic after the result summary has been sent to Amazon S3 including the `SnsMessage` parameter value if specified.

AWS-ConfigureMaintenanceWindows

Description

The AWS-ConfigureMaintenanceWindows runbook helps you to enable or disable multiple Systems Manager maintenance windows.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- MaintenanceWindows

  Type: StringList

  Description: (Required) A comma-separated list of the IDs of the maintenance windows you want to enable or disable.
- MaintenanceWindowsStatus

  Type: String

  Valid values: "True" | "False"

Default: "False"

Description: (Required) Determines whether maintenance windows are enabled or disabled. Specify "True" to enable maintenance windows, and "False" to disable them.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:GetMaintenanceWindow`
- `ssm:UpdateMaintenanceWindow`

Document Steps

- `aws:executeScript` - Gathers the status of the maintenance windows you specify in the `MaintenanceWindows` parameter, and enables or disables the maintenance windows.

AWS-CreateManagedLinuxInstance

Description

Create an EC2 instance for Linux that is configured for Systems Manager.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux

Parameters

- AmiId

Type: String

Description: (Required) AMI ID to use for launching the instance.

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- GroupName

  Type: String

  Default: SSMSecurityGroupForLinuxInstances

  Description: (Required) Security group name to create.

- HttpTokens

  Type: String

  Valid values: optional | required

  Default: optional

  Description: (Optional) IMDSv2 uses token-backed sessions. Set the use of HTTP tokens to `optional` or `required` to determine whether IMDSv2 is optional or required.

- InstanceType

  Type: String

  Default: t2.medium

  Description: (Required) Type of instance to launch. Default is t2.medium.

- KeyPairName

  Type: String

  Description: (Required) Key pair to use when creating instance.

- RemoteAccessCidr

  Type: String

  Default: 0.0.0.0/0

  Description: (Required) Creates Security group with port for SSH(Port range 22) open to IPs specified by CIDR (default is 0.0.0.0/0). If the security group already exists it will not be modified and rules will not be changed.

- RoleName

  Type: String

  Default: SSMManagedInstanceProfileRole

  Description: (Required) Role name to create.

- StackName

  Type: String

  Default: CreateManagedInstanceStack{{automation:EXECUTION_ID}}

  Description: (Optional) Specify stack name used by this runbook

- SubnetId

  Type: String

  Default: Default

  Description: (Required) New instance will be deployed into this subnet or in the default subnet if not specified.

- VpcId

  Type: String

  Default: Default

  Description: (Required) New instance will be deployed into this Amazon Virtual Private Cloud (Amazon VPC) or in the default Amazon VPC if not specified.

AWS-CreateManagedWindowsInstance

Description

Create an EC2 instance for a Windows Server that is configured for Systems Manager.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Windows

Parameters

Parameters

- AmiId

  Type: String

  Default: `{{ssm:/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base}}`

  Description: (Required) AMI ID to use for launching the instance.

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- GroupName

Type: String

Default: SSMSecurityGroupForLinuxInstances

Description: (Required) Security group name to create.

- HttpTokens

  Type: String

  Valid values: optional | required

  Default: optional

  Description: (Optional) IMDSv2 uses token-backed sessions. Set the use of HTTP tokens to `optional` or `required` to determine whether IMDSv2 is optional or required.

- InstanceType

  Type: String

  Default: t2.medium

  Description: (Required) Type of instance to launch. Default is t2.medium.

- KeyPairName

  Type: String

  Description: (Required) Key pair to use when creating instance.

- RemoteAccessCidr

  Type: String

  Default: 0.0.0.0/0

  Description: (Required) Creates security group with port for RDP (Port range 3389) open to IPs specified by CIDR (default is 0.0.0.0/0). If the security group already exists it will not be modified and rules will not be changed.

- RoleName

  Type: String

Default: SSMManagedInstanceProfileRole

Description: (Required) Role name to create.

- StackName

  Type: String

  Default: CreateManagedInstanceStack{{automation:EXECUTION_ID}}

  Description: (Optional) Specify stack name used by this runbook

- SubnetId

  Type: String

  Default: Default

  Description: (Required) New instance will be deployed into this subnet or in the default subnet if not specified.

- VpcId

  Type: String

  Default: Default

  Description: (Required) New instance will be deployed into this Amazon Virtual Private Cloud (Amazon VPC) or in the default Amazon VPC if not specified.

AWSConfigRemediation-EnableCWLoggingForSessionManager

Description

The `AWSConfigRemediation-EnableCWLoggingForSessionManager` runbook enables Amazon Systems Manager Session Manager (Session Manager) sessions to store output logs to an Amazon CloudWatch (CloudWatch) log group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DestinationLogGroup

  Type: String

  Description: (Required) The name of the CloudWatch log group.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ssm:GetDocument`
- `ssm:UpdateDocument`
- `ssm:CreateDocument`
- `ssm:UpdateDefaultDocumentVersion`
- `ssm:DescribeDocument`

Document Steps

- `aws:executeScript` - Accepts the CloudWatch log group to update the document which stores Session Manager session output logs preferences, or creates one if it doesn't exist.

AWS-ExportOpsDataToS3

Description

This runbook retrieves a list of OpsData summaries in Amazon Systems Manager Explorer and exports them to an object in a specified Amazon Simple Storage Service (Amazon S3) bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- columnFields

  Type: StringList

  Description: (Required) Column fields to write to the output file.
- filters

  Type: String

Description: (Optional) Filters for the getOpsSummary request.

- resultAttribute

  Type: String

  Description: (Optional) The result attribute for getOpsSummary request.

- s3BucketName

  Type: String

  Description: (Required) S3 bucket where you want to download the output file.

- snsSuccessMessage

  Type: String

  Description: (Optional) Message to send when runbook finishes.

- snsTopicArn

  Type: String

  Description: (Required) Amazon Simple Notification Service (Amazon SNS) topic ARN to notify when the download completes.

- syncName

  Type: String

  Description: (Optional) The name of the resource data sync.

Document Steps

getOpsSummaryStep – Retrieves up to 5,000 ops summaries to export in a CSV file now.

Outputs

OpsData object – If the runbook runs successfully, you will find the exported OpsData object in your target S3 bucket.

AWS-ExportPatchReportToS3

Description

This runbook retrieves lists of patch summary data and patch details in Amazon Systems Manager Patch Manager and exports them to .csv files in a specified Amazon Simple Storage Service (Amazon S3) bucket.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- assumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that runs this document.
- s3BucketName

  Type: String

  Description: (Required) The S3 bucket where you want to download the output file.
- snsTopicArn

  Type: String

  Description: (Optional) The Amazon Simple Notification Service (Amazon SNS) topic Amazon Resource Name (ARN) to notify when the download completes.
- snsSuccessMessage

  Type: String

Description: (Optional) Text of the message to send when the runbook finishes.

- targets

  Type: String

  Description: (Required) The instance ID or a wildcard character (*) to indicate whether to report patch data for a specific instance or for all instances.

Document Steps

ExportReportStep – The action for this step depends on the value of the `targets` parameter. If `targets` is in the format of `instanceids=*` , the step retrieves up to 10,000 patch summaries for instances in your account and exports the data to a .csv file.

If `targets` is in the format `instanceids=<instance-id>` , the step retrieves both the patch summary and all the patches for the specified instance in your account and exports them to a .csv file.

Outputs

PatchSummary/Patches object – If the runbook runs successfully, the exported patch report object is downloaded to your target S3 bucket.

AWS-SetupInventory

Description

Create a Systems Manager Inventory association for one or more managed instances. The system collects metadata from your instances according to the schedule in the association. For more information, see Amazon Systems Manager Inventory .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- Applications

  Type: String

  Default: Enabled

  Description: (Optional) Collect metadata about installed applications.
- AssociatedDocName

  Type: String

  Default: `AWS-GatherSoftwareInventory`

  Description: (Optional) The name of the runbook used to collect Inventory from the managed instance.
- AssociationName

  Type: String

  Description: (Optional) A name for the Inventory association that will be assigned to the instance.
- AssocWaitTime

  Type: String

  Default: PT5M

  Description: (Optional) Amount of time that Inventory collection should pause when the Inventory association start time is reached. The time uses ISO 8601 format.
- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- `AwsComponents`

  Type: String

  Default: Enabled

  Description: (Optional) Collect metadata for Amazon Components like amazon-ssm-agent.

- CustomInventory

  Type: String

  Default: Enabled

  Description: (Optional) Collect custom inventory metadata.

- Files

  Type: String

  Description: (Optional) Collect metadata about files on your instances. For more information about how to collect this type of Inventory data, see Working with file and Windows registry inventory . Requires SSMAgent version 2.2.64.0 or later. Linux example: `[{"Path":"/usr/bin", "Pattern":["aws*", "*ssm*"],"Recursive":false}, {"Path":"/var/log", "Pattern":["amazon*.*"], "Recursive":true, "DirScanLimit":1000}] Windows example: [{"Path":"%PROGRAMFILES%", "Pattern":["*.exe"],"Recursive":true}]`

- InstanceDetailedInformation

  Type: String

  Default: Enabled

  Description: (Optional) Collect additional information about the instance, including the CPU model, speed, and the number of cores, to name a few.

- InstanceIds

  Type: String

  Default: *

Description: (Required) EC2 instances that you want to inventory.

- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.

- NetworkConfig

  Type: String

  Default: Enabled

  Description: (Optional) Collect metadata about network configurations.

- OutputS3BucketName

  Type: String

  Description: (Optional) Name of an Amazon S3 bucket where you want to write Inventory log data.

- OutputS3KeyPrefix

  Type: String

  Description: (Optional) An Amazon S3 key prefix (subfolder) where you want to write Inventory log data.

- OutputS3Region

  Type: String

  Description: (Optional) The name of the Amazon Web Services Region where the Amazon S3 exists.

- Schedule

  Type: String

  Default: cron(0 */30 * * * ? *)

Description: (Optional) A cron expression for the Inventory association schedule. The default is every 30 minutes.

- Services

  Type: String

  Default: Enabled

  Description: (Optional, Windows OS only, requires SSMAgent version 2.2.64.0 and above) Collect data for service configurations.

- WindowsRegistry

  Type: String

  Description: (Optional) Collect metadata about Microsoft Windows Registry keys. For more information about how to collect this type of Inventory data, see Working with file and Windows registry inventory . Requires SSM Agent version 2.2.64.0 or later. Example: [{"Path":"HKEY_CURRENT_CONFIG\System","Recursive":true},{"Path":"HKEY_LOCAL_MACHINE\SOFTWARE\Amazon\MachineImage", "ValueNames":["AMIName"]}]

- WindowsRoles

  Type: String

  Default: Enabled

  Description: (Optional) Collect information about Windows roles on the instance. Applies to Windows operating systems only. Requires SSMAgent version 2.2.64.0 or later.

- WindowsUpdates

  Type: String

  Default: Enabled

  Description: (Optional) Collect data about all Windows Updates on the instance.

AWS-SetupManagedInstance

Description

Configure an instance with an Amazon Identity and Access Management (IAM) role for Systems Manager access.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- InstanceId

  Type: String

  Description: (Required) ID of the EC2 instance to configure

- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.

- RoleName

Type: String

Default: SSMRoleForManagedInstance

Description: (Optional) The name of the IAM role for the EC2 instance. If this role does not exist, it will be created. When specifying this value, verify that the role contains the **AmazonSSMManagedInstanceCore** Managed Policy.

AWS-SetupManagedRoleOnEC2Instance

Description

Configure an instance with the SSMRoleForManagedInstance managed IAM role for Systems Manager access.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

Type: String

Description: (Required) ID of the EC2 instance to configure

- LambdaAssumeRole

  Type: String

  Description: (Optional) The ARN of the role that allows Lambda created by Automation to perform the actions on your behalf. If not specified a transient role will be created to run the Lambda function.

- RoleName

  Type: String

  Default: SSMRoleForManagedInstance

  Description: (Optional) The name of the IAM role for the EC2 instance. If this role does not exist, it will be created. When specifying this value, verify that the role contains the **AmazonSSMManagedInstanceCore** Managed Policy.

AWSSupport-TroubleshootManagedInstance

Description

The `AWSSupport-TroubleshootManagedInstance` runbook helps you to determine why an Amazon Elastic Compute Cloud (Amazon EC2) instance does not report as managed by Amazon Systems Manager. This runbook reviews the VPC configuration for the instance including security group rules, VPC endpoints, network access control list (ACL) rules, and route tables. It also confirms an Amazon Identity and Access Management (IAM) instance profile that contains the required permissions is attached to the instance.

> **Important**
>
> This automation runbook does not evaluate IPv6 rules.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance that is not reporting as managed by Systems Manager.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:DescribeAutomationExecutions`
- `ssm:DescribeAutomationStepExecutions`
- `ssm:DescribeInstanceInformation`
- `ssm:DescribeInstanceProperties`
- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`

- ssm:GetDocument
- ssm:ListDocuments
- ssm:StartAutomationExecution
- iam:ListRoles
- iam:GetInstanceProfile
- iam:ListAttachedRolePolicies
- ec2:DescribeInstances
- ec2:DescribeNetworkAcls
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeVpcEndpoints

Document Steps

- aws:executeScript - Gathers the PingStatus of the instance.
- aws:branch - Branches based on whether the instance is already reporting as managed by Systems Manager.
- aws:executeAwsApi - Gathers details about the instance including the VPC configuration.
- aws:executeScript - If applicable, gathers additional details related to VPC endpoints that have been deployed to use with Systems Manager, and confirms the security groups attached to the VPC endpoint allow inbound traffic on TCP port 443 from the instance.
- aws:executeScript - Checks whether the route table allows traffic to the VPC endpoint or public Systems Manager endpoints.
- aws:executeScript - Checks whether the network ACL rules allow traffic to the VPC endpoint or public Systems Manager endpoints.
- aws:executeScript - Checks whether outbound traffic to the VPC endpoint or public Systems Manager endpoints is allowed by the security group associated with the instance.
- aws:executeScript - Checks if the instance profile attached to the instance includes a managed policy that provides the required permissions.
- aws:branch - Branches based on the operating system of the instance.
- aws:executeScript - Provides reference to ssmagent-toolkit-linux shell script.

- `aws:executeScript` - Provides reference to `ssmagent-toolkit-windows` PowerShell script.
- `aws:executeScript` - Generates final output for the automation.
- `aws:executeScript` - If the `PingStatus` of the instance is `Online` , returns that the instance is already managed by Systems Manager.

AWSSupport-TroubleshootPatchManagerLinux

Description

The `AWSSupport-TroubleshootPatchManagerLinux` runbook troubleshoots common issues that can cause a patch failure on Linux-based managed nodes using Patch Manager, a tool in Amazon Systems Manager. The main goal of this runbook is to identify the patch command failure root cause and suggest a remediation plan.

How does it work?

The `AWSSupport-TroubleshootPatchManagerLinux` runbook considers the couple instance ID/Command ID provided by you for troubleshooting. If no Command ID is provided, it selects the latest failed patch command within the last 30 days on the provided instance. After checking the command status, the prerequisites fulfillment, and the OS distribution, the runbook downloads and runs a log analyzer package. The output includes the issue root cause as well as the needed action to fix the issue.

Document Type

Automation

Owner

Amazon

Platforms

- Amazon Linux 2 and AL2023
- Red Hat Enterprise Linux 8.X and 9.X
- Centos 8.X and 9.X
- SUSE 15.X

Parameters

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:SendCommand`
- `ssm:DescribeDocument`
- `ssm:GetCommandInvocation`
- `ssm:ListCommands`
- `ssm:DescribeInstanceInformation`
- `ssm:ListCommandInvocations`
- `ssm:GetDocument`
- `ssm:DescribeAutomationExecutions`
- `ssm:GetAutomationExecution`

Instructions

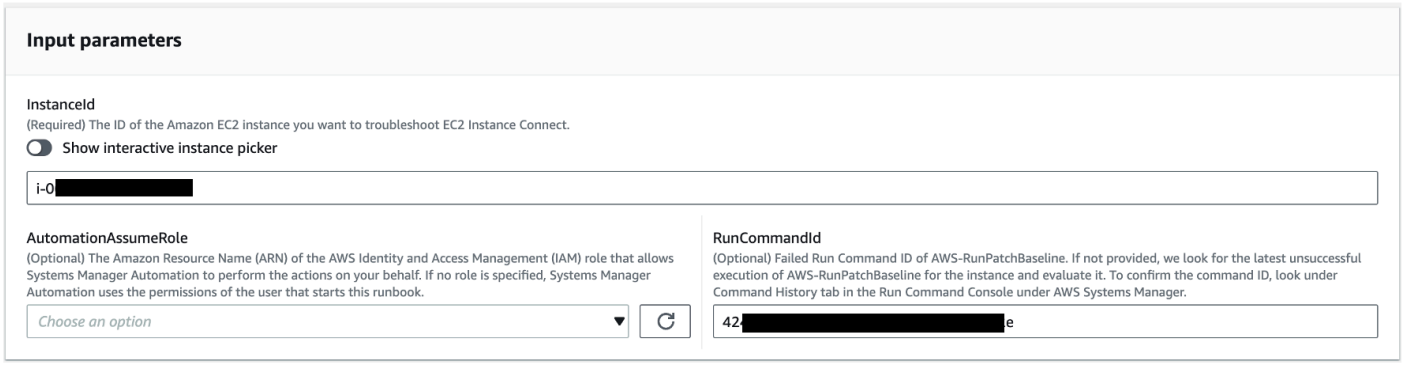
Follow these steps to configure the automation:

1. Navigate to the `AWSSupport-TroubleshootPatchManagerLinux` in the Amazon Systems Manager console.
2. Select Execute automation.
3. For the input parameters, enter the following:
   - **InstanceId (Required):**

     Use the interactive instance picker to choose the ID of the Linux Based SSM Managed Node (Amazon Elastic Compute Cloud (Amazon EC2) or Hybrid Activated server) that the patch command failed against, or manually enter the ID of the SSM Managed instance.
   - **AutomationAssumeRole (Optional):**

     Enter the ARN of the IAM role that allows Automation to perform actions on your behalf. If a role isn't specified, Automation uses the permissions of the user who starts this runbook.
   - **RunCommandId (Optional):**

Enter the Failed Run Command ID of the `AWS-RunPatchBaseline` document. If you don't provide a Command ID, the runbook will look for the latest failed patch command within the last 30 days on the selected instance.



4. Select Execute.
5. The automation initiates.
6. The document performs the following steps:

- **CheckConcurrency:**

  Ensures that there is only one execution of this runbook targeting the same instance. If the runbook finds another execution in progress targeting the same instance, it returns an error and ends.

- **ValidateCommandID:**

  Validates if the provided Command ID, as input parameter, was executed for the `AWS-RunPatchBaseline` SSM Document. If no Command ID is provided, the runbook will consider the latest failed execution of `AWS-RunPatchBaseline` within the last 30 days on the selected instance.

- **BranchOnCommandStatus:**

  Confirms that the status of the provided command is failed. Otherwise, the runbook ends the execution and generates a report stating that the provided command was successfully executed.

- **VerifyPrerequistes:**

  Confirms that the Prerequisites mentioned above are fulfilled.

- **GetPlatformDetails:**

  Retrieves the Operating System (OS) distribution and version.

- **GetDownloadURL:**

  Retrieves the download URL for the PatchManager Log Analyzer package.

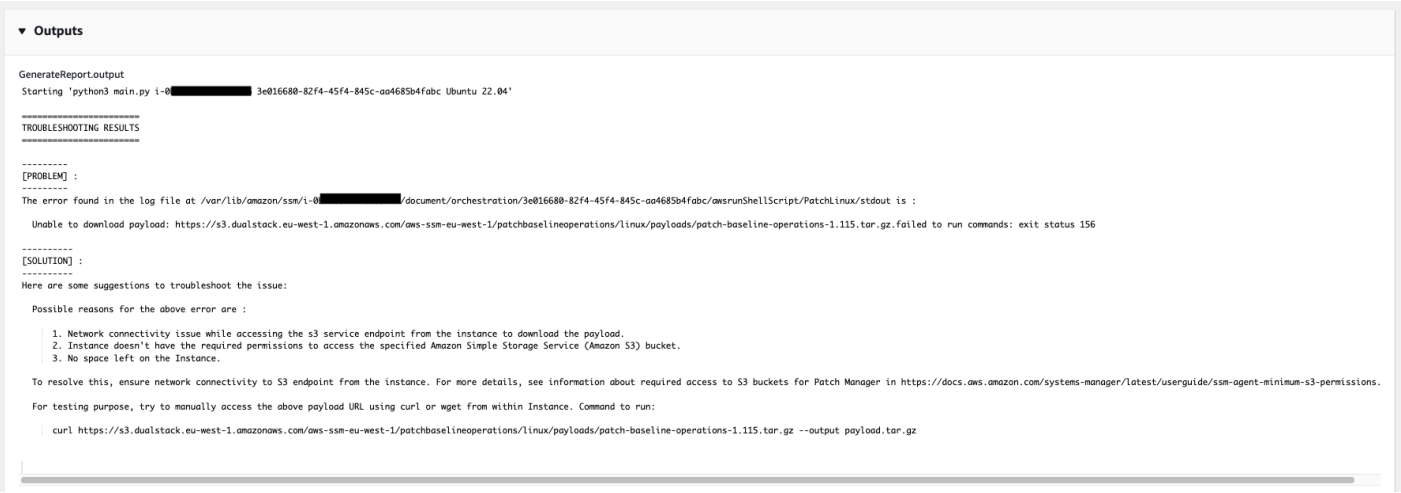
- **EvaluatePatchManagerLogs:**

  Downloads and executes the PatchManager Log Analyzer python package on the instance to evaluate the log file.

- **GenerateReport:**

  Generates a final report of the runbook execution that includes the identified problem and suggested solution.

7. After completed, review the Outputs section for the detailed results of the execution:



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

AWSSupport-TroubleshootSessionManager

Description

The `AWSSupport-TroubleshootSessionManager` runbook helps you troubleshoot common issues that prevent you from connecting to managed Amazon Elastic Compute Cloud (Amazon EC2) instances using Session Manager. Session Manager is a tool in Amazon Systems Manager. This runbook checks the following:

- Checks whether the instance is running and reporting as managed by Systems Manager.
- Runs the `AWSSupport-TroubleshootManagedInstance` runbook if the instance is not reporting as managed by Systems Manager.
- Checks the version of the SSM Agent installed on the instance.
- Checks whether an instance profile containing a recommended Amazon Identity and Access Management (IAM) policy for Session Manager is attached to the Amazon EC2 instance.
- Collects SSM Agent logs from the instance.
- Analyzes your Session Manager preferences.
- Runs the `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2` runbook to analyze the instance's connectivity to the endpoints for Session Manager, Amazon Key Management Service (Amazon KMS), Amazon Simple Storage Service (Amazon S3) and Amazon CloudWatch Logs (CloudWatch Logs).

Considerations

- Hybrid managed nodes are not supported.
- This runbook only checks whether a recommended managed IAM policy is attached to the instance profile. It does not analyze IAM or Amazon KMS permissions contained in your instance profile.

> **Important**
>
> The `AWSSupport-AnalyzeAWSEndpointReachabilityFromEC2` runbook uses VPC Reachability Analyzer to analyze the network connectivity between a source and a service endpoint. You are charged per analysis run between a source and destination. For more details, see Amazon VPC Pricing.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- InstanceId

  Type: String

  Description: (Required) The ID of the Amazon EC2 instance that you are unable to connect to using Session Manager.
- SessionPreferenceDocument

  Type: String

  Default: SSM-SessionManagerRunShell

  Description: (Optional) The name of your session preferences document. If you don't specify a custom session preferences document when starting sessions, use the default value.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:CreateNetworkInsightsPath`

- ec2:DeleteNetworkInsightsAnalysis
- ec2:DeleteNetworkInsightsPath
- ec2:StartNetworkInsightsAnalysis
- tiros:CreateQuery
- ec2:DescribeAvailabilityZones
- ec2:DescribeCustomerGateways
- ec2:DescribeDhcpOptions
- ec2:DescribeInstances
- ec2:DescribeInstanceStatus
- ec2:DescribeInternetGateways
- ec2:DescribeManagedPrefixLists
- ec2:DescribeNatGateways
- ec2:DescribeNetworkAcls
- ec2:DescribeNetworkInsightsAnalyses
- ec2:DescribeNetworkInsightsPaths
- ec2:DescribeNetworkInterfaces
- ec2:DescribePrefixLists
- ec2:DescribeRegions
- ec2:DescribeRouteTables
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeTransitGatewayAttachments
- ec2:DescribeTransitGatewayConnects
- ec2:DescribeTransitGatewayPeeringAttachments
- ec2:DescribeTransitGatewayRouteTables
- ec2:DescribeTransitGateways
- ec2:DescribeTransitGatewayVpcAttachments
- ec2:DescribeVpcAttribute
- ec2:DescribeVpcEndpoints

- ec2:DescribeVpcEndpointServiceConfigurations
- ec2:DescribeVpcPeeringConnections
- ec2:DescribeVpcs
- ec2:DescribeVpnConnections
- ec2:DescribeVpnGateways
- ec2:GetManagedPrefixListEntries
- ec2:GetTransitGatewayRouteTablePropagations
- ec2:SearchTransitGatewayRoutes
- elasticloadbalancing:DescribeListeners
- elasticloadbalancing:DescribeLoadBalancerAttributes
- elasticloadbalancing:DescribeLoadBalancers
- elasticloadbalancing:DescribeRules
- elasticloadbalancing:DescribeTags
- elasticloadbalancing:DescribeTargetGroups
- elasticloadbalancing:DescribeTargetHealth
- iam:GetInstanceProfile
- iam:ListAttachedRolePolicies
- iam:ListRoles
- iam:PassRole
- ssm:DescribeAutomationStepExecutions
- ssm:DescribeInstanceInformation
- ssm:GetAutomationExecution
- ssm:GetDocument
- ssm:ListCommands
- ssm:ListCommandInvocations
- ssm:SendCommand
- ssm:StartAutomationExecution
- tiros:GetQueryAnswer

- `tiros:GetQueryExplanation`

Document Steps

1. `aws:waitForAwsResourceProperty`: Waits up to 6 minutes for your target instance to pass status checks.
2. `aws:executeScript`: Parses the session preference document.
3. `aws:executeAwsApi`: Gets the ARN of the instance profile attached to your instance.
4. `aws:executeAwsApi`: Checks whether your instance is reporting as managed by Systems Manager.
5. `aws:branch`: Branches based on whether your instance is reporting as managed by Systems Manager.
6. `aws:executeScript`: Checks whether the SSM Agent installed on your instance supports Session Manager.
7. `aws:branch`: Branches based on the platform of your instance to collect `ssm-cli` logs.
8. `aws:runCommand`: Collects logs output from `ssm-cli` from a Linux or macOS instance.
9. `aws:runCommand`: Collects logs output from `ssm-cli` from a Windows instance.
10. `aws:executeScript`: Parses the `ssm-cli` logs.
11. `aws:executeScript`: Checks whether a recommended IAM policy is attached to the instance profile.
12. `aws:branch`: Determines whether to evaluate `ssmmessages` endpoint connectivity based on `ssm-cli` logs.
13. `aws:executeAutomation`: Evaluates whether the instance can connect to an `ssmmessages` endpoint.
14. `aws:branch`: Determines whether to evaluate Amazon S3 endpoint connectivity based on `ssm-cli` logs and your session preferences.
15. `aws:executeAutomation`: Evaluates whether the instance can connect to an Amazon S3 endpoint.
16. `aws:branch`: Determines whether to evaluate Amazon KMS endpoint connectivity based on `ssm-cli` logs and your session preferences.
17. `aws:executeAutomation`: Evaluates whether the instance can connect to an Amazon KMS endpoint.

18 `aws:branch`: Determines whether to evaluate CloudWatch Logs endpoint connectivity based on `ssm-cli` logs and your session preferences.

19 `aws:executeAutomation`: Evaluates whether the instance can connect to an CloudWatch Logs endpoint.

20 `aws:executeAutomation`: Runs the `AWSSupport-TroubleshootManagedInstance` runbook.

21 `aws:executeScript`: Compiles the output of the previous steps and outputs a report.

Ouputs

- `generateReport.EvalReport` - The results of the checks performed by the runbook in plain text.

Third-party

Amazon Systems Manager Automation provides predefined runbooks for third-party products and services. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-CreateJiraIssue
- AWS-CreateServiceNowIncident
- AWS-RunPacker

`AWS-CreateJiraIssue`

Description

Create an issue in Jira.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AssigneeName

  Type: String

  Description: (Optional) The username of the person the issue should be assigned to.
- DueDate

  Type: String

  Description: (Optional) The due date for the issue in yyyy-mm-dd format.
- IssueDescription

  Type: String

  Description: (Required) A detailed description of the issue.
- IssueSummary

  Type: String

  Description: (Required) A brief summary of the issue.
- IssueTypeName

  Type: String

  Description: (Required) The name of the type of issue you want to create (for example, Task, Sub-task, Bug, etc.).
- JiraURL

  Type: String

  Description: (Required) The url of the Jira instance.

- JiraUsername

  Type: String

  Description: (Required) The name of the user the issue will be created with.
- PriorityName

  Type: String

  Description: (Optional) The name of the priority of the issue.
- ProjectKey

  Type: String

  Description: (Required) The key of the project the issue should be created in.
- SSMParameterName

  Type: String

  Description: (Required) The name of an encrypted SSM Parameter containing the API key or password for the Jira user.

Document Steps

`aws:createStack` - Create CloudFormation stack to create Lambda IAM role and function.

`aws:invokeLambdaFunction` - Invoke Lambda function to create the Jira issue

`aws:deleteStack` - Delete the CloudFormation stack created.

Outputs

IssueId: ID of the newly created Jira issue

AWS-CreateServiceNowIncident

Description

Create an incident in the ServiceNow incident table.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Category

  Type: String

  Description: (Optional) The category of the incident.

  Valid values: None | Inquiry/Help | Software | Hardware | Network | Database

  Default Value: None
- Description

  Type: String

  Description: (Required) A detailed explanation on the incident.
- Impact

  Type: String

  Description: (Optional) The effect an incident has on business.

  Valid values: High | Medium | Low

Default Value: Low

- ServiceNowInstanceUsername

  Type: String

  Description: (Required) The name of the user the incident will be created with.

- ServiceNowInstancePassword

  Type: String

  Description: (Required) The name of an encrypted SSM Parameter containing the password for the ServiceNow user.

- ServiceNowInstanceURL

  Type: String

  Description: (Required) The URL of the ServiceNow instance

- ShortDescription

  Type: String

  Description: (Required) A brief description of the incident.

- Subcategory

  Type: String

  Description: (Optional) The subcategory of the incident.

  Valid values: None | Antivirus | Email | Internal Application | Operating System | CPU | Disk | Keyboard | Hardware | Memory | Monitor | Mouse | DHCP | DNS | IP Address | VPN | Wireless | DB2 | MS SQL Server | Oracle

  Default Value: None

Document Steps

Push_incident – Pushes the incident information to ServiceNow.

Outputs

Push_incident.incidentID – The created incident ID.

AWS-RunPacker

Description

This runbook uses the HashiCorp Packer tool to validate, fix, or build packer templates that are used to create machine images. This runbook uses Packer v1.7.2.

> **Note**
>
> If you specify a `vpc_id` value, you must also specify the `subnet_id` value of a public subnet. Unless you modify your subnet's IPv4 public addressing attribute, you must also set `associate_public_ip_address` to true.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- Force

Type: Boolean

Description: A Packer option to force a builder to run when artifacts from a previous build otherwise prevent a build from running.

- Mode

  Type: String

  Description: The mode, or command, in which to use Packer when validating against the template. Options include `Build` , `Validate` , and `Fix` .

- TemplateFileName

  Type: String

  Description: The name, or key, of the template file in the S3 bucket.

- TemplateS3BucketName

  Type: String

  Description: The name of the S3 bucket containing the packer template.

Document Steps

RunPackerProcessTemplate – Runs the selected mode against the template using the Packer tool.

Outputs

RunPackerProcessTemplate.output – The stdout from the Packer tool.

RunPackerProcessTemplate.fixed_template_key – The name of the template stored in an S3 bucket to use only when running in "Fix" mode.

RunPackerProcessTemplate.s3_bucket – The name of the S3 bucket that contains the fixed template to use only when running in "Fix" mode.

Amazon VPC

Amazon Systems Manager Automation provides predefined runbooks for Amazon Virtual Private Cloud. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWS-CloseSecurityGroup

Description

This runbook removes all ingress and egress rules from the security group you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- SecurityGroupId

  Type: String

  Description: (Required) The ID of the security group you want to close.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeSecurityGroups`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:RevokeSecurityGroupIngress`

Document Steps

- `aws:executeScript` - Removes all ingress and egress rules from the security group you specify in the `SecurityGroupId` parameter.

AWSSupport-ConfigureDNSQueryLogging

Description

The `AWSSupport-ConfigureDNSQueryLogging` runbook configures logging for DNS queries that originate in your virtual private cloud (VPC) or for Amazon Route 53 hosted zones. You can choose to publish query logs to Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3), or Amazon Data Firehose. For more information about query logging and resolver query logs, see Public DNS query logging and Resolver query logging .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- LogDestinationArn

Type: String

Description: (Optional) The ARN of the CloudWatch Logs group, Amazon S3 bucket or Firehose stream you want to send query logs to. Note that Route 53 public DNS query logging only supports CloudWatch Logs groups. If you do not specify a value for this parameter, the automation creates a CloudWatch Logs group with the format `AWSSupport-ConfigureDNSQueryLogging-{automation: EXECUTION_ID }` , and an IAM resource policy to publish the query logs. The CloudWatch Logs group created by the automation has a retention period of 14 days.

- QueryLogType

  Type: String

  Description: (Optional) The types of queries you want to log.

  Valid values: Public | Resolver/Private

  Default: Public

- ResourceId

  Type: String

  Description: (Required) The ID of the resource whose queries you want to log. If you specify `Public` for the `QueryLogType` parameter, the resource must be the ID of a Route 53 private hosted zone. If you specify `Resolver/Private` for the `QueryLogType` parameter, the resource must be the ID of a VPC.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeVpcs`
- `firehose:ListTagsForDeliveryStream`
- `firehose:PutRecord`
- `firehose:PutRecordBatch`
- `firehose:TagDeliveryStream`

- iam:AttachRolePolicy
- iam:CreatePolicy
- iam:CreateRole
- iam:CreateServiceLinkedRole
- iam:DeletePolicy
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:GetPolicy
- iam:GetRole
- iam:PassRole
- iam:PutRolePolicy
- iam:TagRole
- iam:UpdateRole
- logs:CreateLogDelivery
- logs:CreateLogGroup
- logs:DeleteLogDelivery
- logs:DeleteLogGroup
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- logs:DescribeResourcePolicies
- logs:ListLogDeliveries
- logs:PutResourcePolicy
- logs:PutRetentionPolicy
- logs:UpdateLogDelivery
- route53:CreateQueryLoggingConfig
- route53:DeleteQueryLoggingConfig
- route53:GetHostedZone
- route53resolver:AssociateResolverQueryLogConfig
- route53resolver:CreateResolverQueryLogConfig

- route53resolver:DeleteResolverQueryLogConfig
- s3:GetBucketAcl

Document Steps

- aws:executeScript - Verifies the resource you specify for the ResourceId parameter exists, and checks whether the resource type matches the required QueryLogType option.
- aws:executeScript - Verifies that the value you specify for the LogDestinationArn parameter matches the required QueryLogType .
- aws:executeScript - Verifies the required permissions for Route 53 to publish logs to the CloudWatch Logs log group, and creates the required IAM resource policy if it doesn't exist.
- aws:executeScript - Enables the DNS query logging on the selected destination.

AWSSupport-ConfigureTrafficMirroring

Description

The AWSSupport-ConfigureTrafficMirroring runbook configures traffic mirroring to help you troubleshoot connectivity issues between a load balancer and Amazon Elastic Compute Cloud (Amazon EC2) instances. Traffic mirroring copies inbound and outbound traffic from the network interfaces that are attached to your instances. To configure traffic mirroring, this runbook creates the required targets, filters, and sessions. By default, the runbook configures mirroring for all inbound and outbound traffic for all protocols except Amazon DNS. If you want to mirror traffic from specific sources and destinations, you can modify the inbound and outbound rules after the automation completes.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- SourceENI

  Type: String

  Description: (Required) The elastic network interface you want to configure traffic mirroring for.

- Target

  Type: String

  Description: (Required) The destination for the mirrored traffic. You must specify the ID of a network interface, a Network Load Balancer, or a Gateway Load Balancer endpoint. If you specify a Network Load Balancer, there must be UDP listeners on port 4789.

- SessionNumber

  Type: String

  Valid values: 1-32766

  Description: (Required) The number of the mirror session you want to use.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:CreateTrafficMirrorTarget`
- `ec2:CreateTrafficMirrorFilter`
- `ec2:CreateTrafficMirrorFilterRule`

- ec2:CreateTrafficMirrorSession
- ec2:DeleteTrafficMirrorSession
- ec2:DeleteTrafficMirrorFilter
- ec2:DeleteTrafficMirrorSession
- ec2:DeleteTrafficMirrorFilterRule
- iam:ListRoles
- ssm:GetAutomationExecution
- ssm:StartAutomationExecution

Document Steps

- aws:executeScript - Runs a script to create a target.
- aws:executeAwsApi - Creates a filter rule.
- aws:executeAwsApi - Creates a mirror filter rule for all inbound traffic.
- aws:executeAwsApi - Creates a mirror filter rule for all outbound traffic.
- aws:executeAwsApi - Creates a traffic mirror session.
- aws:executeAwsApi - Deletes the filter if filter or session creation fails.
- aws:executeAwsApi - Deletes the target if filter or session creation fails.

Outputs

CreateFilter.FilterId

CreateSession.SessionId

CreateTarget.TargetIDOutput

AWSSupport-ConnectivityTroubleshooter

Description

The AWSSupport-ConnectivityTroubleshooter runbook diagnoses connectivity issues between the following:

- Amazon resources within an Amazon Virtual Private Cloud (Amazon VPC)

- Amazon resources in different Amazon VPCs within the same Amazon Web Services Region that are connected using VPC peering
- Amazon resources in an Amazon VPC and an internet resource using an internet gateway
- Amazon resources in an Amazon VPC and an internet resource using a network address translation (NAT) gateway

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- DestinationIP

  Type: String

  Description: (Required) The IPv4 address of the resource you want to connect to.
- DestinationPort

  Type: String

  Default: true

Description: (Required) The port number you want to connect to on the destination resource.

- DestinationVpc

  Type: String

  Default: All

  Description: (Optional) The ID of the Amazon VPC you want to test connectivity to.

- SourceIP

  Type: String

  Description: (Required) The private IPv4 address of the Amazon resource in your Amazon VPC you want to test connectivity from.

- SourcePortRange

  Type: String

  Description: (Optional) The port range used by the Amazon resource in your Amazon VPC you want to test connectivity from.

- SourceVpc

  Type: String

  Default: All

  Description: (Optional) The ID of the Amazon VPC you want to test connectivity from.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeNatGateways`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSecurityGroups`

- ec2:DescribeVpcPeeringConnections

Document Steps

- aws:executeScript - Gathers details about the Amazon resource you specify in the SourceIP parameter.
- aws:executeScript - Determines the destination of network traffic from the Amazon resource using the routes gathered from the previous step.
- aws:branch - Branches based on the destination of the network traffic.
- aws:executeAwsApi - Gathers details about the destination resource.
- aws:executeScript - Confirms that the ID returned for the destination Amazon VPC matches the value specified, if any, in the DestinationVpc parameter.
- aws:executeAwsApi - Gathers the security group rules for the source and destination resources.
- aws:executeScript - Confirms whether the security group rules allow the needed traffic between the source and destination resources.
- aws:executeAwsApi - Gathers the network access control lists (NACLs) associated with the subnets for the source and destination resources.
- aws:executeScript - Confirms whether the NACLs allow the needed traffic between the source and destination resources.
- aws:executeScript - Confirms whether the source has a public IP address associated with the resource, if the route destination is an internet gateway.
- aws:executeAwsApi - Gathers the security group rules for the source resource.
- aws:executeScript - Confirms whether the security group rules allow the needed traffic from the source to the destination resource.
- aws:executeAwsApi - Gathers the NACLs associated with the subnet for the source resource.
- aws:executeScript - Confirms whether the NACLs allow the needed traffic from the source resource.
- aws:executeAwsApi - Gathers details about the NAT gateway.
- aws:executeAwsApi - Gathers the NACLs associated with the subnet for the NAT gateway.
- aws:executeScript - Confirms whether the NACLs allow the needed traffic from the subnet for the NAT gateway.

- `aws:executeScript` - Gathers the routes associated with the subnet for the NAT gateway.
- `aws:executeScript` - Confirms whether the NAT gateway has a route to an internet gateway.
- `aws:executeAwsApi` - Gathers details about the VPC peering connection.
- `aws:executeScript` - Confirms both VPCs are in the same Region and that the ID returned for the destination VPC matches the value specified, if any, in the `DestinationVpc` parameter.
- `aws:executeAwsApi` - Returns the subnet of the destination resource.
- `aws:executeScript` - Gathers the routes associated with the subnet for the peered VPC.
- `aws:executeScript` - Confirms whether the peered VPC has a route to the peering connection.
- `aws:executeScript` - Confirms whether traffic is allowed from the source resource if the destination is not supported by the automation.

AWSSupport-TroubleshootVPN

Description

The `AWSSupport-TroubleshootVPN` runbook helps you to trace and resolve errors in an Amazon Site-to-Site VPN connection. The automation includes several automated checks designed to trace `IKEv1` or `IKEv2` errors related to Amazon Site-to-Site VPN connection tunnels. The automation tries to match specific errors and its corresponding resolution form a list of common issues.

Note: This automation does not rectify the errors. It runs for the mentioned time range and scans the log group for errors in VPN CloudWatch logs group.

How does it work?

The runbook runs a parameter validation to confirm if the Amazon CloudWatch log group included in the input parameter exists, if there are any log streams in the log group that correspond to VPN tunnel logging, if VPN connection id exists, and if the Tunnel IP address exists. It makes Logs Insights API calls on your CloudWatch log group that are configured for VPN logging.

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- LogGroupName

  Type: String

  Description: (Required) The Amazon CloudWatch log group name configured for Amazon Site-to-Site VPN connection logging

  Allowed Pattern: `^[\.\-_/#A-Za-z0-9]{1,512}`
- VpnConnectionId

  Type: String

  Description: (Required) The Amazon Site-to-Site VPN connection id to be troubleshooted.

  Allowed Pattern: `^vpn-[0-9a-f]{8,17}$`
- TunnelAIPAddress

  Type: String

  Description: (Required) The tunnel number 1 IPv4 address associated with your Amazon Site-to-Site VPN.

  Allowed Pattern: `^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)[.]){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?){1}$`
- TunnelBIPAddress

  Type: String

Description: (Optional) The tunnel number 2 IPv4 address associated with your Amazon Site-to-Site VPN.

Allowed Pattern: `^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)[.]){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?){1}|^$`

- IKEVersion

  Type: String

  Description: (Required) Select what IKE Version you are using. Allowed values : IKEv1, IKEv2

  Valid values: `['IKEv1', 'IKEv2']`

- StartTimeinEpoch

  Type: String

  Description: (Optional) Start time for log analysis. You can either use StartTimeinEpoch/EndTimeinEpoch or LookBackPeriod for logs analysis

  Allowed Pattern: `^\d{10}|^$`

- EndTimeinEpoch

  Type: String

  Description: (Optional) End time for log analysis. You can either use StartTimeinEpoch/EndTimeinEpoch or LookBackPeriod for logs analysis. If given both StartTimeinEpoch/EndTimeinEpoch and LookBackPeriod then LookBackPeriod takes precedence

  Allowed Pattern: `^\d{10}|^$`

- LookBackPeriod

  Type: String

  Description: (Optional) Two digit time in hours to look back for log analysis. Valid range : 01 - 99. This value takes precedence if you also give StartTimeinEpoch and EndTime

  Allowed Pattern: `^(\d?[1-9]|[1-9]0)|^$`

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `logs:DescribeLogGroups`
- `logs:GetQueryResults`
- `logs:DescribeLogStreams`
- `logs:StartQuery`
- `ec2:DescribeVpnConnections`

Instructions

Note: This automation works on the CloudWatch log groups that is configured for your VPN tunnel logging, when the logging Output format is JSON.

Follow these steps to configure the automation:

1. Navigate to the AWSSupport-TroubleshootVPN in the Amazon Systems Manager console.
2. For the input parameters enter the following:
   - **AutomationAssumeRole (Optional):**

     The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
   - **LogGroupName (Required):**

     The Amazon CloudWatch log group name to be validated. This must be the CloudWatch log group which is configured for VPN to send logs to.
   - **VpnConnectionId (Required):**

     The Amazon Site-to-Site VPN connection id whose log group is traced for VPN error.
   - **TunnelAIPAddress (Required):**

     The tunnel A IP address associated with your Amazon Site-to-Site VPN connection.
   - **TunnelBIPAddress (Optional):**

     The tunnel B IP address associated with your Amazon Site-to-Site VPN connection.

- **IKEVersion (Required):**

  Select what IKEversion you are using. Allowed values : IKEv1, IKEv2.

- **StartTimeinEpoch (Optional):**

  The beginning of the time range to query for error. The range is inclusive, so the specified start time is included in the query. Specified as epoch time, the number of seconds since January 1, 1970, 00:00:00 UTC.

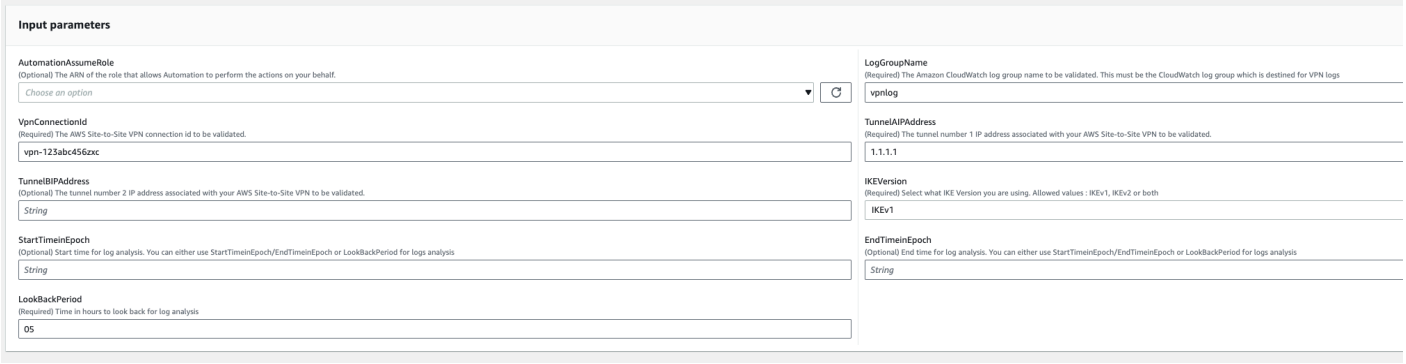
- **EndTimeinEpoch (Optional):**

  The end of the time range to query for errors. The range is inclusive, so the specified end time is included in the query. Specified as epoch time, the number of seconds since January 1, 1970, 00:00:00 UTC.

- **LookBackPeriod (Required):**

  Time in hours to look back to query for error.

Note: Configure a StartTimeinEpoch, EndTimeinEpoch, or LookBackPeriod to fix the time range for log analysis. Give a two-digit number in hours to check for errors in the past from the automation start time. Or, if the error is in the past within a specific time range, include StartTimeinEpoch and EndTimeinEpoch, instead of LookBackPeriod.



3. Select **Execute.**
4. The automation initiates.
5. The automation runbook performs the following steps:

   - **parameterValidation:**

     Runs a series of validation on input parameters included in automation.

   - **branchOnValidationOfLogGroup:**

Checks if log group mentioned in the parameter is valid. If invalid, it halts the further initiation of automation steps.

- **branchOnValidationOfLogStream:**

  Checks if log stream exists in the included CloudWatch log group. If invalid, it halts the further initiation of automation steps.

- **branchOnValidationOfVpnConnectionId:**

  Checks if the VPN Connection id included in the parameter is valid. If invalid, it halts the further initiation of automation steps.

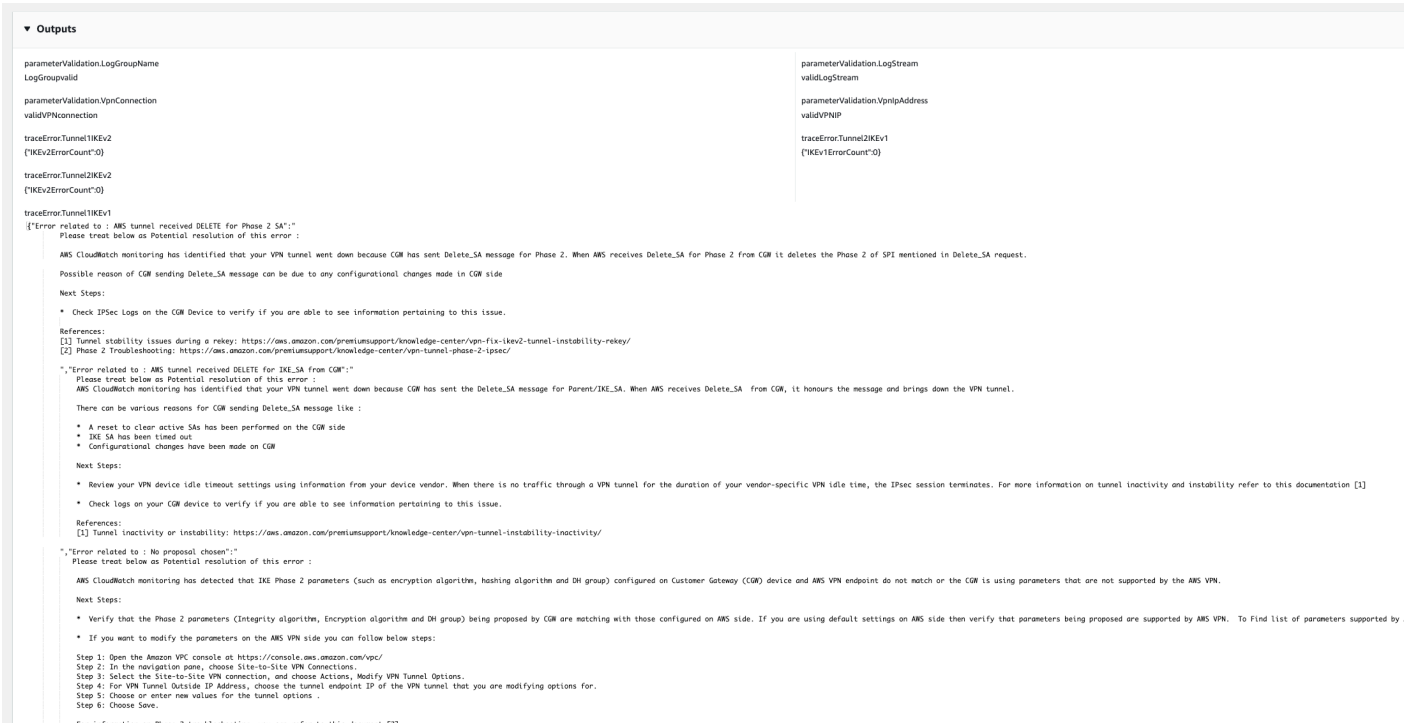
- **branchOnValidationOfVpnIp:**

  Checks if Tunnel IP address mentioned in parameter is valid or not. If invalid then it halts the further execution of automation steps.

- **traceError:**

  Makes a logs insight API call in your included CloudWatch log group and searches for the error related to IKEv1/IKEv2 along with a related suggested resolution.

6. After completed, review the Outputs section for the detailed results of the execution.



References

Systems Manager Automation

- Run this Automation (console)
- Run an automation
- Setting up an Automation
- Support Automation Workflows landing page

Amazon service documentation

- Contents of Site-to-Site VPN logs

AWSConfigRemediation-DeleteEgressOnlyInternetGateway

Description

The `AWSConfigRemediation-DeleteEgressOnlyInternetGateway` runbook deletes the egress-only internet gateway you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- EgressOnlyInternetGatewayId

  Type: String

  Description: (Required) The ID of the egress-only internet gateway that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DeleteEgressOnlyInternetGateway`
- `ec2:DescribeEgressOnlyInternetGateways`

Document Steps

- `aws:executeScript` - Deletes the egress-only internet gateway specified in the `EgressOnlyInternetGatewayId` parameter.
- `aws:executeScript` - Verifies the egress-only internet gateway has been deleted.

AWSConfigRemediation-DeleteUnusedENI

Description

The `AWSConfigRemediation-DeleteUnusedENI` runbook deletes an elastic network interface (ENI) that has an attachment status of `detached` .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- NetworkInterfaceId

  Type: String

  Description: (Required) The ID of the ENI that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DeleteNetworkInterface`
- `ec2:DescribeNetworkInterfaces`

Document Steps

- `aws:executeAwsApi` - Deletes the ENI you specify in the `NetworkInterfaceId` parameter.
- `aws:executeScript` - Verifies the ENI has been deleted.

AWSConfigRemediation-DeleteUnusedSecurityGroup

Description

The AWSConfigRemediation-DeleteUnusedSecurityGroup runbook deletes the security group you specify in the GroupId parameter. If you attempt to delete a security group that is associated with an Amazon Elastic Compute Cloud (Amazon EC2) instance, or is referenced by another security group, the automation fails. This automation does not delete a default security group.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- GroupId

  Type: String

  Description: (Required) The ID of the security group that you want to delete.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ec2:DescribeSecurityGroups
- ec2:DeleteSecurityGroup

Document Steps

- aws:executeAwsApi - Returns the security group name using the value you provide in the GroupId parameter.
- aws:branch - Confirms that the group name is not "default".
- aws:executeAwsApi - Deletes the security group specified in the GroupId parameter.
- aws:executeScript - Confirms the security group was deleted.

AWSConfigRemediation-DeleteUnusedVPCNetworkACL

Description

The AWSConfigRemediation-DeleteUnusedVPCNetworkACL runbook deletes a network access control list (ACL) that is not associated with a subnet.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- NetworkAclId

  Type: String

  Description: (Required) The ID of the network ACL that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DeleteNetworkAcl`
- `ec2:DescribeNetworkAcls`

Document Steps

- `aws:executeAwsApi` - Deletes the network ACL specified in the `NetworkAclId` parameter.
- `aws:executeScript` - Confirms the network ACL specified in the `NetworkAclId` parameter was deleted.

AWSConfigRemediation-DeleteVPCFlowLog

Description

The `AWSConfigRemediation-DeleteVPCFlowLog` runbook deletes the virtual private cloud (VPC) flow log you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- FlowLogId

  Type: String

  Description: (Required) The ID of the flow log that you want to delete.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DeleteFlowLogs`
- `ec2:DescribeFlowLogs`

Document Steps

- `aws:executeAwsApi` - Deletes the flow log you specify in the `FlowLogId` parameter.

- `aws:executeScript` - Verifies the flow log has been deleted.

AWSConfigRemediation-DetachAndDeleteInternetGateway

Description

The `AWSConfigRemediation-DetachAndDeleteInternetGateway` runbook detaches and deletes the internet gateway you specify. If any Amazon EC2 instances in your virtual private cloud (VPC) have elastic IP addresses or public IPv4 addresses associated with them, the runbook fails.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- InternetGatewayId

  Type: String

  Description: (Required) The ID of the internet gateway that you want to delete.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ec2:DeleteInternetGateway
- ec2:DescribeInternetGateways
- ec2:DetachInternetGateway

Document Steps

- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway and waits until the virtual private gateway's state property changes to available or times out.
- aws:executeAwsApi - Retrieves a specified virtual private gateway configuration.
- aws:branch - Branches based on the VpcAttachments.state parameter value.

- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway and waits until the virtual private gateway's VpcAttachments.state's property changes to attached or times out.
- aws:executeAwsApi - Accepts the ID of the virtual private gateway and the ID of the Amazon VPC as input, and detaches the virtual private gateway from the Amazon VPC.
- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway and waits until the virtual private gateway's VpcAttachments.state's property changes to detached or times out.

- aws:executeAwsApi - Accepts the ID of the virtual private gateway as input and deletes it.

- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway as input and verifies its deletion.

  aws:executeAwsApi - Gathers the VPC ID from the internet gateway ID.
- aws:executeAwsApi - Detaches the internet gateway ID from the VPC.
- aws:executeAwsApi - Deletes the internet gateway.

AWSConfigRemediation-DetachAndDeleteVirtualPrivateGateway

Description

The AWSConfigRemediation-DetachAndDeleteVirtualPrivateGateway runbook detaches and deletes a given Amazon Elastic Compute Cloud (Amazon EC2) virtual private gateway attached to a virtual private cloud (VPC) created with Amazon Virtual Private Cloud (Amazon VPC).

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- VpnGatewayId

  Type: String

  Description: (Required) The ID of the virtual private gateway to be deleted.

Required IAM permissions

The AutomationAssumeRole parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ec2:DeleteVpnGateway
- ec2:DetachVpnGateway
- ec2:DescribeVpnGateways

Document Steps

- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway and waits until the virtual private gateway's state property changes to available or times out.
- aws:executeAwsApi - Retrieves a specified virtual private gateway configuration.
- aws:branch - Branches based on the VpcAttachments.state parameter value.

- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway and waits until the virtual private gateway's VpcAttachments.state's property changes to attached or times out.
- aws:executeAwsApi - Accepts the ID of the virtual private gateway and the ID of the Amazon VPC as input, and detaches the virtual private gateway from the Amazon VPC.
- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway and waits until the virtual private gateway's VpcAttachments.state's property changes to detached or times out.

- aws:executeAwsApi - Accepts the ID of the virtual private gateway as input and deletes it.

- aws:waitForAwsResourceProperty - Accepts the ID of the virtual private gateway as input and verifies its deletion.

AWS-DisableIncomingSSHOnPort22

Description

The AWS-DisableIncomingSSHOnPort22 runbook removes rules that allow unrestricted incoming SSH traffic on TCP port 22 for security groups.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- SecurityGroupIds

  Type: String

  Description: (Required) A comma separated list of the IDs of the security groups you want to restrict SSH traffic for.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ec2:DescribeSecurityGroups`
- `ec2:RevokeSecurityGroupIngress`

Document Steps

- `aws:executeAwsApi` - Removes all rules allowing incoming SSH traffic on TCP port 22 from the security groups you specify in the `SecurityGroupIds` parameter.

Outputs

DisableIncomingSSHTemplate.RestrictedSecurityGroupIds - A list of the IDs of the security groups that had inbound SSH rules removed.

AWS-DisablePublicAccessForSecurityGroup

Description

This runbook disables default SSH and RDP ports that are opened to all IP addresses.

Important

This runbook fails with an "InvalidPermission.NotFound" error for security groups that meet both of the following criteria: 1) The security group is located in a non-default VPC; and 2) The inbound rules for the security group don't specify open ports using all four of the following patterns:

- `0.0.0.0/0`
- `::/0`
- `SSH or RDP port + 0.0.0.0/0`
- `SSH or RDP port + ::/0`

Note

This runbook is not available in the Amazon Web Services Regions located within China.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- GroupId

  Type: String

  Description: (Required) The ID of the security group for which the ports should be disabled.
- IpAddressToBlock

  Type: String

  Description: (Optional) Additional IPv4 addresses from which access should be blocked, in the format `1.2.3.4/32` .

AWSConfigRemediation-DisableSubnetAutoAssignPublicIP

Description

The `AWSConfigRemediation-DisableSubnetAutoAssignPublicIP` runbook disables the IPv4 public addressing attribute for the subnet you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- SubnetId

  Type: String

  Description: (Required) The ID of the subnet that you want to disable the auto-assign public IPv4 address attribute on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeSubnets`
- `ec2:ModifySubnetAttribute`

Document Steps

- `aws:executeAwsApi` - Disables the auto-assign public IPv4 address attribute for the subnet you specified in the `SubnetId` parameter.
- `aws:assertAwsResourceProperty` - Verifies the attribute has been disabled.

AWSSupport-EnableVPCFlowLogs

Description

The AWSSupport-EnableVPCFlowLogs runbook creates Amazon Virtual Private Cloud (Amazon VPC) Flow Logs for subnets, network interfaces, and VPCs in your Amazon Web Services account. If you create a flow log for a subnet or VPC, each elastic network interface in that subnet or Amazon VPC is monitored. Flow log data is published to the Amazon CloudWatch Logs log group or the Amazon Simple Storage Service (Amazon S3) bucket you specify. For more information about flow logs, see VPC Flow Logs in the *Amazon VPC User Guide* .

Important

Data ingestion and archival charges for vended logs apply when you publish flow logs to CloudWatch Logs or to Amazon S3. For more information, see Flow Logs pricing

Run this Automation (console)

Note

When selecting s3 as the log destination, ensure that the bucket policy allows the log delivery service access to the bucket. For more information see Amazon S3 bucket permissions for flow logs

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- DeliverLogsPermissionArn

  Type: String

  Description: (Optional) The ARN for the IAM role that permits Amazon Elastic Compute Cloud (Amazon EC2) to publish flow logs to the CloudWatch Logs log group in your account. If you specify `s3` for the `LogDestinationType` parameter, do not provide a value for this parameter. For more information, see Publish flow logs to CloudWatch Logs in the *Amazon VPC User Guide* .

- LogDestinationARN

  Type: String

  Description: (Optional) The ARN of the resource to which the flow log data is published. If `cloud-watch-logs` is specified for the `LogDestinationType` parameter, provide the ARN of the CloudWatch Logs log group you want to publish flow log data to. Alternatively, use `LogGroupName` instead. If `s3` is specified for the `LogDestinationType` parameter, you must specify the ARN of the Amazon S3 bucket you want to publish flow log data to for this parameter. You can also specify a folder in the bucket.

  > **Important**
  >
  > When choosing `s3` as the `LogDestinationType` you should ensure that the bucket selected follows Amazon S3 Bucket security best practices, and that you follow the data privacy laws for your organisation and geographic region.

- LogDestinationType

  Type: String

  Valid values: cloud-watch-logs | s3

Description: (Required) Determines where flow log data is published. If you specify `LogDestinationType` as `s3` , do not specify `DeliverLogsPermissionArn` or `LogGroupName` .

- LogFormat

  Type: String

  Description: (Optional) The fields to include in the flow log, and the order in which they should appear in the record. For a list of available fields, see Flow log records in the *Amazon VPC User Guide* . If you do not provide a value for this parameter, the flow log is created using the default format. If you specify this parameter, you must specify at least one field.

- LogGroupName

  Type: String

  Description: (Optional) The name of the CloudWatch Logs log group where flow log data is published. If you specify `s3` for the `LogDestinationType` parameter, do not provide a value for this parameter.

- ResourceIds

  Type: StringList

  Description: (Required) A comma-separated list of the IDs for the subnets, elastic network interfaces, or VPC for which you want to create a flow log.

- TrafficType

  Type: String

  Valid values: ACCEPT | REJECT | ALL

  Description: (Required) The type of traffic to log. You can log traffic that the resource accepts or rejects, or all traffic.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- ec2:CreateFlowLogs
- ec2:DeleteFlowLogs
- ec2:DescribeFlowLogs
- iam:AttachRolePolicy
- iam:CreateRole
- iam:CreatePolicy
- iam:DeletePolicy
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:GetPolicy
- iam:GetRole
- iam:TagRole
- iam:PassRole
- iam:PutRolePolicy
- iam:UpdateRole
- logs:CreateLogDelivery
- logs:CreateLogGroup
- logs:DeleteLogDelivery
- logs:DeleteLogGroup
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- s3:GetBucketLocation
- s3:GetBucketAcl
- s3:GetBucketPublicAccessBlock
- s3:GetBucketPolicyStatus
- s3:GetBucketAcl
- s3:ListBucket
- s3:PutObject

Sample Policy

```
        {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "SSM Execution Permissions",
                    "Effect": "Allow",
                    "Action": [
                        "ssm:StartAutomationExecution",
                        "ssm:GetAutomationExecution"
                    ],
                    "Resource": "*"
                },
                {
                    "Sid": "EC2 FlowLogs Permissions",
                    "Effect": "Allow",
                    "Action": [
                        "ec2:CreateFlowLogs",
                        "ec2:DeleteFlowLogs",
                        "ec2:DescribeFlowLogs"
                    ],
                    "Resource": "arn:{partition}:ec2:{region}:{account-id}:{instance|
subnet|vpc|transit-gateway|transit-gateway-attachment}/{resource ID}"
                },
                {
                    "Sid": "IAM CreateRole Permissions",
                    "Effect": "Allow",
                    "Action": [
                        "iam:AttachRolePolicy",
                        "iam:CreateRole",
                        "iam:CreatePolicy",
                        "iam:DeletePolicy",
                        "iam:DeleteRole",
                        "iam:DeleteRolePolicy",
                        "iam:GetPolicy",
                        "iam:GetRole",
                        "iam:TagRole",
                        "iam:PassRole",
                        "iam:PutRolePolicy",
                        "iam:UpdateRole"
                    ],
                    "Resource": [
```

```
                        "arn:{partition}:iam::{account-id}:role/{role name}",
                        "arn:{partition}:iam::{account-id}:role/
AWSSupportCreateFlowLogsRole"
                    ]
                },
                {
                    "Sid": "CloudWatch Logs Permissions",
                    "Effect": "Allow",
                    "Action": [
                        "logs:CreateLogDelivery",
                        "logs:CreateLogGroup",
                        "logs:DeleteLogDelivery",
                        "logs:DeleteLogGroup",
                        "logs:DescribeLogGroups",
                        "logs:DescribeLogStreams"
                    ],
                    "Resource": [
                        "arn:{partition}:logs:{region}:{account-id}:log-group:{log
 group name}",
                        "arn:{partition}:logs:{region}:{account-id}:log-group:{log
 group name}:*"
                    ]
                },
                {
                    "Sid": "S3 Permissions",
                    "Effect": "Allow",
                    "Action": [
                        "s3:GetBucketLocation",
                        "s3:GetBucketPublicAccessBlock",
                        "s3:GetAccountPublicAccessBlock",
                        "s3:GetBucketPolicyStatus",
                        "s3:GetBucketAcl",
                        "s3:ListBucket",
                        "s3:PutObject"
                    ],
                    "Resource": [
                        "arn:{partition}:s3:::{bucket name}",
                        "arn:{partition}:s3:::{bucket name}/*"
                    ]
                }
            ]
        }
```

Document Steps

- `aws:branch` - Branches based on the value specified for the `LogDestinationType` parameter.
- `aws:executeScript` - Checks if the target Amazon Simple Storage Service (Amazon S3) potentially grants **read** or **write** `public` access to its objects.
- `aws:executeScript` - Creates a log group if no value is specified for the `LogDestinationARN` parameter, and `cloud-watch-logs` is specified for the `LogDestinationType` parameter.
- `aws:executeScript` - Creates flow logs based on the values specified in the runbook parameters.

AWSConfigRemediation-EnableVPCFlowLogsToCloudWatch

Description

The `AWSConfigRemediation-EnableVPCFlowLogsToCloudWatch` runbook replaces an existing Amazon VPC flow log that publishes flow log data to Amazon Simple Storage Service (Amazon S3) with a flow log that publishes flow log data to the Amazon CloudWatch Logs (CloudWatch Logs) log group you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- DestinationLogGroup

  Type: String

  Description: (Required) The name of the CloudWatch Logs log group you want to publish flow log data to.

- DeliverLogsPermissionArn

  Type: String

  Description: (Required) The ARN of the Amazon Identity and Access Management (IAM) role you want to use that provides Amazon Elastic Compute Cloud (Amazon EC2) the requisite permissions to publish flow log data to CloudWatch Logs.

- FlowLogId

  Type: String

  Description: (Required) The ID of the flow log that publishes to Amazon S3 you want to replace.

- MaxAggregationInterval

  Type: Integer

  Valid values: 60 | 600

  Description: (Optional) The maximum interval of time, in seconds, during which a flow of packets is captured and aggregated into a flow log record.

- TrafficType

  Type: String

  Valid values: ACCEPT | REJECT | ALL

  Description: (Required) The type of flow log data you want to record and publish.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:CreateFlowLogs`
- `ec2:DeleteFlowLogs`
- `ec2:DescribeFlowLogs`

Document Steps

- `aws:executeAwsApi` - Gathers details about your VPC from the value you specify in the `FlowLogId` parameter.
- `aws:executeAwsApi` - Creates a flow log based on the values you specify for the runbook parameters.
- `aws:assertAwsResourceProperty` - Verifies the newly created flow log publishes to CloudWatch Logs.
- `aws:executeAwsApi` - Deletes the flow log that publishes to Amazon S3.
- `aws:executeScript` - Confirms the flow log that published to Amazon S3 was deleted.

AWSConfigRemediation-EnableVPCFlowLogsToS3Bucket

Description

The `AWSConfigRemediation-EnableVPCFlowLogsToS3Bucket` runbook *replaces* an existing Amazon VPC flow log that publishes flow log data to Amazon CloudWatch Logs (CloudWatch Logs) with a flow log that publishes flow log data to the Amazon Simple Storage Service (Amazon S3) bucket you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DestinationS3BucketArn

  Type: String

  Description: (Required) The ARN of the Amazon S3 bucket you want to publish flow log data to.
- FlowLogId

  Type: String

  Description: (Required) The ID of the flow log that publishes to CloudWatch Logs you want to replace.
- MaxAggregationInterval

  Type: Integer

  Valid values: 60 | 600

  Description: (Optional) The maximum interval of time, in seconds, during which a flow of packets is captured and aggregated into a flow log record.
- TrafficType

  Type: String

  Valid values: ACCEPT | REJECT | ALL

  Description: (Required) The type of flow log data you want to record and publish.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:CreateFlowLogs`
- `ec2:DeleteFlowLogs`
- `ec2:DescribeFlowLogs`

Document Steps

- `aws:executeAwsApi` - Gathers details about your VPC from the value you specify in the `FlowLogId` parameter.
- `aws:executeAwsApi` - Creates a flow log based on the values you specify for the runbook parameters.
- `aws:assertAwsResourceProperty` - Verifies the newly created flow log publishes to Amazon S3.
- `aws:executeAwsApi` - Deletes the flow log that publishes to CloudWatch Logs.
- `aws:executeScript` - Confirms the flow log that published to CloudWatch Logs was deleted.

AWS-ReleaseElasticIP

Description

Release the specified Elastic IP address using the allocation ID.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- AllocationId

  Type: String

  Description: (Required) The Allocation ID of the Elastic IP address.

AWS-RemoveNetworkACLUnrestrictedSSHRDP

Description

The AWS-RemoveNetworkACLUnrestrictedSSHRDP runbook removes all network access control list (ACL) rules from the specified network ACL that allow ingress traffic from all source addresses to default SSH and RDP ports. Rules that include port ranges that overlap with the default SSH and RDP ports aren't removed.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- NetworkAclId

  Type: String

  Description: (Required) The ID of the network ACL that you want to remove unrestricted rules that allow ingress traffic from all source addresses to default SSH and RDP ports.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DeleteNetworkAclEntry`
- `ec2:DescribeNetworkAcls`

Document Steps

- `aws:executeScript` - Removes all ingress rules that allow traffic from all source addresses from the security group you specified in the `SecurityGroupId` parameter.

Outputs

RemoveNaclEntriesAndVerify.VerificationMessage - Verification messages of the successfully deleted network ACL rules.

RemoveNaclEntriesAndVerify.RulesDeletedAndApiResponses - The network ACL rules that were deleted, and the `DeleteNetworkAclEntry` API operation responses.

AWSConfigRemediation-RemoveUnrestrictedSourceIngressRules

Description

The `AWSConfigRemediation-RemoveUnrestrictedSourceIngressRules` runbook removes all ingress rules from the security group you specify that allow traffic from all source addresses.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- SecurityGroupId

  Type: String

  Description: (Required) The ID of the security group that you want to remove ingress rules that allow traffic from all source addresses from.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeSecurityGroups`
- `ec2:RevokeSecurityGroupIngress`

Document Steps

- `aws:executeScript` - Removes all ingress rules that allow traffic from all source addresses from the security group you specified in the `SecurityGroupId` parameter.

AWSConfigRemediation-RemoveVPCDefaultSecurityGroupRules

Description

The `AWSConfigRemediation-RemoveVPCDefaultSecurityGroupRules` runbook removes all rules from the default security group of the virtual private cloud (VPC) you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- GroupId

  Type: String

  Description: (Required) The ID of the security group that you want to remove all rules from.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `ec2:DescribeSecurityGroups`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:RevokeSecurityGroupIngress`

Document Steps

- `aws:assertAwsResourceProperty` - Confirms the security group you specified in the `GroupId` parameter is named default.
- `aws:executeScript` - Removes all rules from the security group you specified in the `GroupId` parameter.

AWSSupport-SetupIPMonitoringFromVPC

Description

`AWSSupport-SetupIPMonitoringFromVPC` creates an Amazon Elastic Compute Cloud (Amazon EC2) instance in the specified subnet and monitors selected target IPs (IPv4 or IPv6) by continuously running ping, MTR, traceroute and tracetcp tests. The results are stored in Amazon

CloudWatch Logs logs, and metric filters are applied to quickly visualize latency and packet loss statistics in a CloudWatch dashboard.

Additional Information

The CloudWatch Logs data can be used for network troubleshooting and analysis of pattern/trends. Additionally, you can configure CloudWatch alarms with Amazon SNS notifications when packet loss and/or latency reach a threshold. The data can also be used when opening a case with Amazon Web Services Support, to help isolate an issue quickly and reduce time to resolution when investigating a network issue.

> **Note**
>
> To clean up resources created by `AWSSupport-SetupIPMonitoringFromVPC`, you can use the runbook `AWSSupport-TerminateIPMonitoringFromVPC` . For more information, see `AWSSupport-TerminateIPMonitoringFromVPC` .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on

your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- CloudWatchLogGroupNamePrefix

  Type: String

  Default: `/AWSSupport-SetupIPMonitoringFromVPC`

  Description: (Optional) Prefix used for each CloudWatch log group created for the test results.

- CloudWatchLogGroupRetentionInDays

  Type: String

  Valid values: 1 | 3 | 5 | 7 | 14 | 30 | 60 | 90 | 120 | 150 | 180 | 365 | 400 | 545 | 731 | 1827 | 3653

  Default: 7

  Description: (Optional) Number of days you want to keep the network monitoring results for.

- InstanceType

  Type: String

  Valid values: t2.micro | t2.small | t2.medium | t2.large | t3.micro | t3.small | t3.medium | t3.large | t4g.micro | t4g.small | t4g.medium | t4g.large

  Default: t3.micro

  Description: (Optional) The EC2 instance type for the EC2Rescue instance. Recommended size: t3.micro.

- SubnetId

  Type: String

  Description: (Required) The subnet ID for the monitor instance. Be aware that if you specify a private subnet, then you must make sure there is Internet access to allow the monitor instance to setup the test (meaning, install the CloudWatch Logs agent, interact with Systems Manager and CloudWatch).

- TargetIPs

  Type: String

Description: (Required) Comma separated list of IPv4s and/or IPv6s to monitor. No spaces allowed. Maximum size is 255 characters. Be aware that if you provide an invalid IP, then the automation will fail and rollback the test setup.

- TestInstanceSecurityGroupId

  Type: String

  Description: (Optional) The security group ID for the test instance. If not specified, the automation creates one during the instance creation. Make sure the security group allows outbound access to the monitoring IPs.

- TestInstanceProfileName

  Type: String

  Description: (Optional) The name of an existing IAM instance profile for the test instance. If not specified, the automation creates one during the instance creation. The role must have the following permissions: `logs:CreateLogStream`, `logs:DescribeLogGroups`, `logs:DescribeLogStreams`, and `logs:PutLogEvents` and the AWS Managed Policy `AmazonSSMManagedInstanceCore`.

- TestInterval

  Type: String

  Description: (Optional) The number of minutes between test intervals. The default value is `1` minute and the maximum is `10` minutes.

- RetainDashboardAndLogsOnDeletion

  Type: String

  Description: (Optional) Specify `False` to delete the Amazon CloudWatch dashboard and Logs when deleting the AWS CloudFormation stack. The default value is `True`. By default, the dashboard and logs are retained and will need to be manually deleted when they are no longer needed.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

Warning

It is recommended to pass `TestInstanceProfileName` parameter or ensure security guardrails in place to prevent misuse of mutable IAM permissions.

It is recommended that the user who runs the automation have the **AmazonSSMAutomationRole** IAM managed policy attached. In addition, the user must have the following policy attached to their user account, group, or role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:GetRole",
                "iam:GetInstanceProfile",
                "iam:DetachRolePolicy",
                "iam:AttachRolePolicy",
                "iam:PassRole",
                "iam:AddRoleToInstanceProfile",
                "iam:GetRolePolicy",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile",
                "iam:PutRolePolicy",
                "iam:TagRole"
            ],
            "Resource": [
                "arn:<partition>:iam::<account-id>:role/SetupIPMonitoringFromVPC*",
                "arn:<partition>:iam::<account-id>:instance-profile/
SetupIPMonitoringFromVPC*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:CreateStack",
```

```
                "cloudformation:CreateChangeSet",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeStackEvents",
                "cloudwatch:PutDashboard",
                "cloudwatch:DeleteDashboards",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:CreateSecurityGroup",
                "ec2:CreateLaunchTemplate",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeImages",
                "ec2:DescribeSubnets",
                "ec2:DescribeInstanceTypes",
                "ec2:DescribeVpcs",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSecurityGroup",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:CreateTags",
                "ec2:AssignIpv6Addresses",
                "ec2:DescribeTags",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeLaunchTemplates",
                "ec2:RevokeSecurityGroupEgress",
                "logs:CreateLogGroup",
                "logs:DeleteLogGroup",
                "logs:PutMetricFilter",
                "logs:PutRetentionPolicy",
                "logs:TagResource",
                "ssm:DescribeInstanceInformation",
                "ssm:GetParameter",
                "ssm:GetParameters",
                "ssm:SendCommand",
                "ssm:ListCommands",
                "ssm:ListCommandInvocations"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow"
        }
    ]
```

```
}
```

If the `TestInstanceProfileName` parameter is provided, the following IAM permissions are not required to execute the runbook:

- iam:CreateRole
- iam:CreateInstanceProfile
- iam:DetachRolePolicy
- iam:AttachRolePolicy
- iam:AddRoleToInstanceProfile
- iam:RemoveRoleFromInstanceProfile
- iam:DeleteRole
- iam:DeleteRolePolicy
- iam:DeleteInstanceProfile

Document Steps

1. **`aws:executeAwsApi`** - describe the provided subnet.
2. **`aws:branch`** - evaluate the TargetIPs input.

   (IPv6) If TargetIPs contains an IPv6:

   `aws:assertAwsResourceProperty` - check the provided subnet has an IPv6 pool associated
3. **`aws:executeScript`** - get the architecture of instance type and public parameter path for latest Amazon Linux 2 AMI.
4. **`aws:executeAwsApi`** - get the latest Amazon Linux 2 AMI from Parameter Store.
5. **`aws:executeAwsApi`** - create a security group for the test in the subnet's VPC.

   (Cleanup) If the security group creation fails:

   `aws:executeAwsApi` - delete the security group created by the automation, if it exists.
6. **`aws:executeAwsApi`** - allow all outbound traffic in the test security group.

   (Cleanup) If the security group egress rule creation fails:

aws:executeAwsApi - delete the security group created by the automation, if it exists.

7. **aws:executeAwsApi** - create an IAM role for the test EC2 instance

   (Cleanup) If the role creation fails:

   a. **aws:executeAwsApi** - delete the IAM role created by the automation, if it exists.

   b. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

8. **aws:executeAwsApi** - attach the AmazonSSMManagedInstanceCore managed policy

   (Cleanup) If the policy attachment fails:

   a. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation, if attached.

   b. **aws:executeAwsApi** - delete the IAM role created by the automation.

   c. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

9. **aws:executeAwsApi** - attach an inline policy to allow setting CloudWatch log group retentions and creating a CloudWatch dashboard

   (Cleanup) If the inline policy attachment fails:

   a. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation, if created.

   b. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

   c. **aws:executeAwsApi** - delete the IAM role created by the automation.

   d. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

10. **aws:executeAwsApi** - create an IAM instance profile.

    (Cleanup) If the instance profile creation fails:

    a. **aws:executeAwsApi** - delete the IAM instance profile created by the automation, if it exists.

    b. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

    c. **aws:executeAwsApi** - delete the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

    d. **aws:executeAwsApi** - delete the IAM role created by the automation.

e. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

11 **aws:executeAwsApi** - associate the IAM instance profile to the IAM role.

(Cleanup) If the instance profile and role association fails:

a. **aws:executeAwsApi** - remove the IAM instance profile from the role, if associated.

b. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

c. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

d. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

e. **aws:executeAwsApi** - delete the IAM role created by the automation.

f. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

12 **aws:sleep** - wait for the instance profile to become available.

13 **aws:runInstances** - create the test instance in the specified subnet, and with the instance profile created earlier attached.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

14 **aws:branch** - evaluate the TargetIPs input.

(IPv6) If TargetIPs contains an IPv6:

aws:executeAwsApi - assign an IPv6 to the test instance.

15 **aws:waitForAwsResourceProperty** - wait for the test instance to become a managed instance.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

16 **aws:runCommand** - install test pre-requisites:

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

17 **aws:runCommand** - validate the provided IPs are syntactically correct IPv4 and/or IPv6 addresses:

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

18 **aws:runCommand** - define the MTR test for each of the provided IPs.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

19 **aws:runCommand** - define the first ping test for each of the provided IPs.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

20 **aws:runCommand** - define the second ping test for each of the provided IPs.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

21 **aws:runCommand** - define the tracepath test for each of the provided IPs.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

22 **aws:runCommand** - define the traceroute test for each of the provided IPs.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

23 **aws:runCommand** - configure CloudWatch logs.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

24 **aws:runCommand** - schedule cronjobs to run each test every minute.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

25 **aws:sleep** - wait for the tests to generate some data.

26 **aws:runCommand** - set the desired CloudWatch log group retentions.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

27 **aws:runCommand** - set the CloudWatch log group metric filters.

(Cleanup) If the step fails:

a. **aws:changeInstanceState** - terminate the test instance.

b. **aws:executeAwsApi** - remove the IAM instance profile from the role.

c. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

d. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

e. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

f. **aws:executeAwsApi** - delete the IAM role created by the automation.

g. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

28 **aws:runCommand** - create the CloudWatch dashboard.

(Cleanup) If the step fails:

a. **aws:executeAwsApi** - delete the CloudWatch dashboard, if it exists.

b. **aws:changeInstanceState** - terminate the test instance.

c. **aws:executeAwsApi** - remove the IAM instance profile from the role.

d. **aws:executeAwsApi** - delete the IAM instance profile created by the automation.

e. **aws:executeAwsApi** - delete the CloudWatch inline policy from the role created by the automation.

f. **aws:executeAwsApi** - detach the AmazonSSMManagedInstanceCore managed policy from the role created by the automation.

g. **aws:executeAwsApi** - delete the IAM role created by the automation.

h. **aws:executeAwsApi** - delete the security group created by the automation, if it exists.

Outputs

createCloudWatchDashboards.Output - the URL of the CloudWatch dashboard.

createManagedInstance.InstanceIds - the test instance ID.

AWSSupport-TerminateIPMonitoringFromVPC

Description

AWSSupport-TerminateIPMonitoringFromVPC terminates an IP monitoring test previously started by AWSSupport-SetupIPMonitoringFromVPC . Data related to the specified test ID will be deleted.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- AutomationExecutionId

  Type: String

  Description: (Required) The automation execution ID from when you previously ran the AWSSupport-SetupIPMonitoringFromVPC runbook. All resources associated with this execution ID are deleted.

- InstanceId

  Type: String

  Description: (Required) The instance ID for the monitor instance.

- SubnetId

  Type: String

  Description: (Required) The subnet ID for the monitor instance.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

It is recommended that the user who runs the automation have the **AmazonSSMAutomationRole** IAM managed policy attached. In addition, the user must have the following policy attached to their user, group, or role:

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Action": [
            "iam:DetachRolePolicy",
            "iam:RemoveRoleFromInstanceProfile",
            "iam:DeleteRole",
            "iam:DeleteInstanceProfile",
            "iam:DeleteRolePolicy"
        ],
        "Resource": [
            "arn:aws:iam::An-AWS-Account-ID:role/AWSSupport/
SetupIPMonitoringFromVPC_*",
            "arn:aws:iam::An-AWS-Account-ID:instance-profile/AWSSupport/
SetupIPMonitoringFromVPC_*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "iam:DetachRolePolicy"
```

```
        ],
        "Resource": [
            "arn:aws:iam::aws:policy/service-role/AmazonSSMManagedInstanceCore"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "cloudwatch:DeleteDashboards"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "ec2:DescribeTags",
            "ec2:DescribeInstances",
            "ec2:DescribeSecurityGroups",
            "ec2:DeleteSecurityGroup",
            "ec2:TerminateInstances",
            "ec2:DescribeInstanceStatus"
        ],
        "Resource": [
            "*"
        ],
        "Effect": "Allow"
    ]
}
```

Document Steps

1. `aws:assertAwsResourceProperty` - check AutomationExecutionId and InstanceId are related to the same test.
2. `aws:assertAwsResourceProperty` - check SubnetId and InstanceId are related to the same test.
3. `aws:executeAwsApi` - retrieve the test security group.
4. `aws:executeAwsApi` - delete the CloudWatch dashboard.
5. `aws:changeInstanceState` - terminate the test instance.
6. `aws:executeAwsApi` - remove the IAM instance profile from the role.

7. `aws:executeAwsApi` - delete the IAM instance profile created by the automation.
8. `aws:executeAwsApi` - delete the CloudWatch inline policy from the role created by the automation.
9. `aws:executeAwsApi` - detach the **AmazonSSMManagedInstanceCore** managed policy from the role created by the automation.
10. `aws:executeAwsApi` - delete the IAM role created by the automation.
11. `aws:executeAwsApi` - delete the security group created by the automation, if it exists.

Outputs

None

Amazon WAF

Amazon Systems Manager Automation provides predefined runbooks for Amazon WAF. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-AddWAFRegionalRuleToRuleGroup
- AWS-AddWAFRegionalRuleToWebAcl
- AWSConfigRemediation-EnableWAFClassicLogging
- AWSConfigRemediation-EnableWAFClassicRegionalLogging
- AWSConfigRemediation-EnableWAFV2Logging

AWS-AddWAFRegionalRuleToRuleGroup

Description

The `AWS-AddWAFRegionalRuleToRuleGroup` runbook adds an existing Amazon WAF regional rule to a Amazon WAF regional rule group. Only Amazon WAF Classic regional rule groups are supported. Amazon WAF Classic regional rule groups can have a maximum of 10 rules.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- RuleGroupId

  Type: String

  Description: (Required) The ID of the rule group that you want to update.
- RulePriority

  Type: Integer

  Description: (Required) The priority for the new rule. Rule priority determines the order in which rules in a regional group are evaluated. Rules with a lower value have higher priority than rules with a higher value. The value must be a unique integer. If you add multiple rules to a regional rule group, the values don't have to be consecutive.
- RuleId

  Type: String

  Description: (Required) The ID for the rule that you want to add to your regional rule group.
- RuleAction

  Type: String

Description: (Required) Specifies the action that Amazon WAF takes when a web request matches the conditions of the rule.

Valid values: ALLOW | BLOCK | COUNT

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `waf-regional:GetChangeToken`
- `waf-regional:GetChangeTokenStatus`
- `waf-regional:ListActivatedRulesInRuleGroup`
- `waf-regional:UpdateRuleGroup`

Document Steps

- GetWAFChangeToken (aws:executeAwsApi) - Retrieves a Amazon WAF change token to ensure the runbook doesn't submit conflicting requests to the service.
- AddWAFRuleToWAFRegionalRuleGroup (aws:executeScript) - Adds the specified rule to the Amazon WAF regional rule group.
- VerifyChangeTokenPropagating (aws:waitForAwsResourceProperty) - Verifies the change token has a status of `PENDING` or `INSYNC`.
- VerifyRuleAddedToRuleGroup (aws:executeScript) - Verifies the specified Amazon WAF rule was added to the target regional rule group.

Outputs

- VerifyRuleAddedToRuleGroup.VerifyRuleAddedToRuleGroupResponse - Output of the step verifying that the new rule was aded to the regional rule group.
- VerifyRuleAddedToRuleGroup.ListActivatedRulesInRuleGroupResponse - Output of the `ListActivatedRulesInRuleGroup` API operation.

AWS-AddWAFRegionalRuleToWebAcl

Description

The AWS-AddWAFRegionalRuleToWebAcl runbook adds an existing Amazon WAF regional rule, rule group or rate-based rule to a Amazon WAF Classic regional web access control list (ACL). This runbook doesn't update existing Amazon WAF Classic regional web ACL's that are managed by Amazon Firewall Manager.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- WebACLId

  Type: String

  Description: (Required) The ID of the web ACL that you want to update.
- ActivatedRulePriority

  Type: Integer

Description: (Required) The priority for the new rule. Rule priority determines the order in which rules in a web ACL are evaluated. Rules with a lower value have higher priority than rules with a higher value. The value must be a unique integer. If you add multiple rules to a regional web ACL, the values don't have to be consecutive.

- ActivatedRuleRuleId

  Type: String

  Description: (Required) The ID for the regular rule, rate-based rule, or group you want to add to the web ACL.

- ActivatedRuleAction

  Type: String

  Valid values: ALLOW | BLOCK | COUNT

  Description: (Optional) Specifies the action that Amazon WAF takes when a web request matches the conditions of the rule.

- ActivatedRuleType

  Type: String

  Valid values: REGULAR | RATE_BASED | GROUP

  Default: REGULAR

  Description: (Optional) The rule type you're adding to the web ACL. Although this field is optional, note that if you try to add a `RATE_BASED` rule to a web ACL without setting the type, the request fails because the request defaults to a `REGULAR` rule.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `waf-regional:GetChangeToken`

- `waf-regional:GetWebACL`
- `waf-regional:UpdateWebACL`

Document Steps

- DetermineWebACLNotInFMSAndRulePriority (aws:executeScript) - Verifies if the Amazon WAF web ACL is in a Firewall Manager security policy and verifies the priority ID doesn't conflict with an existing ACL.
- AddRuleOrRuleGroupToWebACL (aws:executeScript) - Adds the specified rule to the Amazon WAF web ACL.
- VerifyRuleOrRuleGroupAddedToWebAcl (aws:executeScript) - Verifies the specified Amazon WAF rule was added to the target web ACL.

Outputs

- DetermineWebACLNotInFMSAndRulePriority.PrereqResponse: Output from the `DetermineWebACLNotInFMSAndRulePriority` step.
- VerifyRuleOrRuleGroupAddedToWebAcl.VerifyRuleOrRuleGroupAddedToWebACLResponse: Output from the `AddRuleOrRuleGroupToWebACL` step.
- VerifyRuleOrRuleGroupAddedToWebAcl.ListActivatedRulesOrRuleGroupsInWebACLResponse: Output of the `VerifyRuleOrRuleGroupAddedToWebAcl` step.

AWSConfigRemediation-EnableWAFClassicLogging

Description

The `AWSConfigRemediation-EnableWAFClassicLogging` runbook enables logging to Amazon Data Firehose (Firehose) for the Amazon WAF web access control list (web ACL) you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- DeliveryStreamName

  Type: String

  Description: (Required) The name of the Firehose delivery stream that you want to send logs to.
- WebACLId

  Type: String

  Description: (Required) The ID of the Amazon WAF web ACL that you want to enable logging on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:CreateServiceLinkedRole`
- `waf:GetLoggingConfiguration`
- `waf:GetWebAcl`
- `waf:PutLoggingConfiguration`

Document Steps

- `aws:executeAwsApi` - Confirms the delivery stream you specify in the `DeliveryStreamName` exists.
- `aws:executeAwsApi` - Gathers the ARN of the Amazon WAF web ACL specified in the `WebACLId` parameter.
- `aws:executeAwsApi` - Enables logging for the web ACL.
- `aws:assertAwsResourceProperty` - Verifies logging has been enabled on the Amazon WAF web ACL.

AWSConfigRemediation-EnableWAFClassicRegionalLogging

Description

The `AWSConfigRemediation-EnableWAFClassicRegionalLogging` runbook enables logging to Amazon Data Firehose (Firehose) for the Amazon WAF web access control list (ACL) you specify.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LogDestinationConfigs

  Type: String

Description: (Required) The Amazon Resource Name (ARN) of the Firehose delivery stream that you want to send logs to.

- WebACLId

  Type: String

  Description: (Required) The ID of the Amazon WAF web ACL that you want to enable logging on.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `iam:CreateServiceLinkedRole`
- `waf-regional:GetLoggingConfiguration`
- `waf-regional:GetWebAcl`
- `waf-regional:PutLoggingConfiguration`

Document Steps

- `aws:executeAwsApi` - Gathers the ARN of the Amazon WAF web ACL specified in the `WebACLId` parameter.
- `aws:executeAwsApi` - Enables logging for the web ACL.
- `aws:assertAwsResourceProperty` - Verifies logging has been enabled on the Amazon WAF web ACL.

AWSConfigRemediation-EnableWAFV2Logging

Description

The `AWSConfigRemediation-EnableWAFV2Logging` runbook enables logging for an Amazon WAF (Amazon WAFV2) web access control list (web ACL) with the specified Amazon Data Firehose (Firehose) delivery stream.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.
- LogDestinationConfigs

  Type: String

  Description: (Required) The Firehose delivery stream ARN that you want to associate with the web ACL.

  > **Note**
  >
  > The Firehose delivery stream ARN must begin with the prefix `aws-waf-logs-` . For example, `aws-waf-logs-us-east-2-analytics` . For more information, see Amazon Data Firehose .

- WebAclArn

  Type: String

  Description: (Required) ARN of the web ACL for which logging will be enabled.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `firehose:DescribeDeliveryStream`
- `wafv2:PutLoggingConfiguration`
- `wafv2:GetLoggingConfiguration`

Document Steps

- `aws:executeScript` - Enables logging for the Amazon WAFV2 web ACL and verifies that the logging has the specified configuration.

Amazon WorkSpaces

Amazon Systems Manager Automation provides predefined runbooks for Amazon WorkSpaces. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

- AWS-CreateWorkSpace
- AWSSupport-RecoverWorkSpace

`AWS-CreateWorkSpace`

Description

The `AWS-CreateWorkSpace` runbook creates a new Amazon WorkSpaces virtual desktop, known as a WorkSpace, based on the values that you specify for the input parameters. For information about WorkSpaces, see What is Amazon WorkSpaces? in the *Amazon WorkSpaces Administration Guide*.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.
- BundleId

  Type: String

  Description: (Required) The ID of the bundle to use for the WorkSpace.
- ComputeTypeName

  Type: String

  Valid values: VALUE | STANDARD | PERFORMANCE | POWER | GRAPHICS | POWERPRO | GRAPHICSPRO

  Description: (Optional) The compute type for your WorkSpace.
- DirectoryId

  Type: String

  Description: (Required) The ID of the directory to add your WorkSpace to.
- RootVolumeEncryptionEnabled

Type: Boolean

Valid values: true | false

Default: false

Description: (Optional) Determines whether the root volume of the WorkSpace is encrypted.

- RootVolumeSizeGib

  Type: Integer

  Description: (Required) The size of the root volume for the WorkSpace.

- RunningMode

  Type: String

  Valid values: ALWAYS_ON | AUTO_STOP

  Description: (Required) The running mode of the WorkSpace.

- RunningModeAutoStopTimeoutInMinutes

  Type: Integer

  Description: (Optional) The time after a user logs off when the WorkSpaces stops. Specify a value in 60-minute intervals.

- Tags

  Type: String

  Description: (Optional) Tags that you want to apply to the WorkSpace.

- UserName

  Type: String

  Description: (Required) The user name to associate with the WorkSpace.

- UserVolumeEncryptionEnabled

  Type: Boolean

  Valid values: true | false

Default: false

Description: (Optional) Determines whether the user volume of the WorkSpace is encrypted.

- UserVolumeSizeGib

  Type: Integer

  Description: (Required) The size of the user volume for the WorkSpace.

- VolumeEncryptionKey

  Type: String

  Description: (Optional) The symmetric Amazon Key Management Service key that you want to use to encrypt data stored on your WorkSpace.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `workspaces:CreateWorkspaces`
- `workspaces:DescribeWorkspaces`

Document Steps

- `aws:executeScript` - Creates a WorkSpace based on the values that you specify for the input parameters.
- `aws:waitForAwsResourceProperty` - Verifies the state of the WorkSpace is `AVAILABLE`.

Outputs

`CreateWorkspace.WorkspaceId`

AWSSupport-RecoverWorkSpace

Description

The `AWSSupport-RecoverWorkSpace` runbook performs recovery steps on the Amazon WorkSpaces virtual desktop, known as a WorkSpace, you specify. The runbook reboots the WorkSpace, and if the state is still `UNHEALTHY` , restores or rebuilds the WorkSpace based on the values you specify for the input parameters. Before using this runbook we recommend reviewing Troubleshooting WorkSpaces Issues in the *Amazon WorkSpaces Administration Guide* .

> **Important**
>
> Restoring or rebuilding a WorkSpace is a potentially destructive action that can result in the loss of data. This is because the WorkSpace is restored from the last available snapshot and data recovered from snapshots can be as old as 12 hours.
> The restore option recreates both the root volume and user volume based on the most recent snapshots. The rebuild option recreates the user volume from the most recent snapshot and recreates the WorkSpace from the image associated with the bundle the WorkSpace was created from. Applications that were installed or system settings that were changed after the WorkSpace was created are lost. For more information about restoring and rebuilding WorkSpaces, see Restore a WorkSpace and Rebuild a WorkSpace in the *Amazon WorkSpaces Administration Guide* .

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

Description: (Optional) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf. If no role is specified, Systems Manager Automation uses the permissions of the user that starts this runbook.

- Acknowledge

  Type: String

  Valid values: Yes

  Description: (Required) Entering yes means that you understand the restore and rebuild actions will try to recover the WorkSpace from the most recent snapshot, and that data restored from these snapshots can be as old as 12 hours.

- Reboot

  Type: String

  Valid values: Yes | No

  Default: Yes

  Description: (Required) Determines whether the WorkSpace is rebooted.

- Rebuild

  Type: String

  Valid values: Yes | No

  Default: No

  Description: (Required) Determines whether the WorkSpace is rebuilt.

- Restore

  Type: String

  Valid values: Yes | No

  Default: No

  Description: (Required) Determines whether the WorkSpace is restored.

- WorkspaceId

  Type: String

  Description: (Required) The ID of the WorkSpace you want to recover.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `workspaces:DescribeWorkspaces`
- `workspaces:DescribeWorkspaceSnapshots`
- `workspaces:RebootWorkspaces`
- `workspaces:RebuildWorkspaces`
- `workspaces:RestoreWorkspace`
- `workspaces:StartWorkspaces`

Document Steps

- `aws:executeAwsApi` - Gathers the state of the WorkSpace you specify in the `WorkspaceId` parameter.
- `aws:assertAwsResourceProperty` - Verifies the state of the WorkSpace is `AVAILABLE` , `ERROR` , `IMPAIRED` , `STOPPED` , or `UNHEALTHY` .
- `aws:branch` - Branches based on the state of the WorkSpace.
- `aws:executeAwsApi` - Starts the WorkSpace.
- `aws:branch` - Branches based on the value you specify for the `Action` parameter.
- `aws:waitForAwsResourceProperty` - Waits for the WorkSpace status after being started.
- `aws:waitForAwsResourceProperty` - Waits for the WorkSpace state to change to `AVAILABLE` , `ERROR` , `IMPAIRED` , or `UNHEALTHY` after being started.
- `aws:executeAwsApi` - Gathers the state of the WorkSpace after being started.
- `aws:branch` - Branches based on the state of the WorkSpace after being started.

- aws:executeAwsApi - Gathers the available snapshots for restoring or rebuilding the WorkSpace.
- aws:branch - Branches based on the value you specify for the Reboot parameter.
- aws:executeAwsApi - Reboots the WorkSpace.
- aws:executeAwsApi - Gathers the state of the WorkSpace after being started.
- aws:waitForAwsResourceProperty - Waits for the state of the WorkSpace to change to REBOOTING .
- aws:waitForAwsResourceProperty - Waits for the WorkSpace state to change to AVAILABLE , ERROR , or UNHEALTHY after being rebooted.
- aws:executeAwsApi - Gathers the state of the WorkSpace after being rebooted.
- aws:branch - Branches based on the state of the WorkSpace after rebooting.
- aws:branch - Branches based on the value you specify for the Restore parameter.
- aws:executeAwsApi - Restores the WorkSpace. If the restore fails, the runbook tries to rebuild the WorkSpace.
- aws:waitForAwsResourceProperty - Waits for the state of the WorkSpace to change to RESTORING .
- aws:waitForAwsResourceProperty - Waits for the WorkSpace state to change to AVAILABLE , ERROR , or UNHEALTHY after being restored.
- aws:executeAwsApi - Gathers the state of the WorkSpace after being restored.
- aws:branch - Branches based on the state of the WorkSpace after restoring.
- aws:branch - Branches based on the value you specify for the Rebuild parameter.
- aws:executeAwsApi - Rebuilds the WorkSpace.
- aws:waitForAwsResourceProperty - Waits for the state of the WorkSpace to change to REBUILDING .
- aws:waitForAwsResourceProperty - Waits for the WorkSpace state to change to AVAILABLE , ERROR , or UNHEALTHY after being rebuilt.
- aws:executeAwsApi - Gathers the state of the WorkSpace after being rebuilt.
- aws:assertAwsResourceProperty - Confirms the state of the WorkSpace is AVAILABLE .

X-Ray

Amazon Systems Manager Automation provides predefined runbooks for Amazon X-Ray. For more information about runbooks, see Working with runbooks. For information about how to view runbook content, see View runbook content.

Topics

AWSConfigRemediation-UpdateXRayKMSKey

Description

The `AWSConfigRemediation-UpdateXRayKMSKey` runbook enables encryption on your Amazon X-Ray data using an Amazon Key Management Service (Amazon KMS) key. This runbook should only be used as a baseline to ensure that your Amazon X-Ray data is encrypted according to minimum recommended security best practices. We recommend encrypting multiple sets of data with different KMS keys.

Run this Automation (console)

Document type

Automation

Owner

Amazon

Platforms

Linux, macOS, Windows

Parameters

- AutomationAssumeRole

  Type: String

  Description: (Required) The Amazon Resource Name (ARN) of the Amazon Identity and Access Management (IAM) role that allows Systems Manager Automation to perform the actions on your behalf.

- KeyId

  Type: String

  Description: (Required) The Amazon Resource Name (ARN), key ID, or the key alias of the KMS key you want Amazon X-Ray to use to encrypt data.

Required IAM permissions

The `AutomationAssumeRole` parameter requires the following actions to use the runbook successfully.

- `ssm:StartAutomationExecution`
- `ssm:GetAutomationExecution`
- `kms:DescribeKey`
- `xray:GetEncryptionConfig`
- `xray:PutEncryptionConfig`

Document Steps

- `aws:executeAwsApi` - Enables encryption on your X-Ray data using the KMS key you specify in the `KeyId` parameter.
- `aws:waitForAwsResourceProperty` - Waits for the encryption configuration status of your X-Ray to be `ACTIVE` .
- `aws:executeAwsApi` - Gathers the ARN of the key you specify in the `KeyId` parameter.
- `aws:assertAwsResourceProperty` - Verifies encryption has been enabled on your X-Ray.