SAP NetWeaver Guides

SAP NetWeaver on Amazon

SAP NetWeaver on Amazon: SAP NetWeaver Guides

Services or capabilities described in Amazon Web Services documentation might vary by Region. To see the differences applicable to the China Regions, see Getting Started with Amazon Web Services in China (PDF).

Table of Contents

SAP NetWeaver Guides

This section covers the following guides.

- SAP NetWeaver Environment Setup for Linux on Amazon
- SAP NetWeaver on Amazon Deployment and Operations Guide for Windows
- Microsoft SQL Server for SAP NetWeaver on Amazon Deployment and Operations Guide
- SAP NetWeaver on Amazon: high availability configuration for NetWeaver ASCS
- Migrate SAP NetWeaver applications with Amazon Migration Hub Orchestrator
- Oracle for SAP NetWeaver on Amazon Deployment and Operations Guide for Linux
- SAP ASE for SAP NetWeaver on Amazon Deployment and Operations Guide for Linux
- SAP NetWeaver on Amazon Automation

Additional SAP on Amazon documentation

- General SAP guides
- SAP HANA on Amazon
- Databases for SAP applications on Amazon
- Amazon Launch Wizard for SAP
- Amazon Systems Manager for SAP
- Amazon SDK for SAP ABAP
- SAP BusinessObjects on Amazon
- Amazon Migration Hub Orchestrator

SAP NetWeaver Environment Setup for Linux on Amazon

Planning and Prerequisites

Topics

SAP Landscape Assessment

Before deploying SAP NetWeaver on Amazon, document your existing SAP landscape to inform architecture decisions. This assessment determines landing zone requirements including account and subnet allocation, as well as patterns for infrastructure selection and shared services.

By considering the complete set of requirements, you can optimize resource allocation and plan for deployment automation where applicable. For comprehensive guidance on SAP workload design principles, refer to the SAP Lens of the Amazon Well-Architected Framework.

Review resilience, performance, and connectivity requirements to determine deployment pattern selection (single instance, distributed, or highly available) as well as requirements for web dispatchers and load balancers. Establish non-functional requirements for:

- Maximum tolerable downtime (RTO - Recovery Time Objective)
- Maximum acceptable data loss (RPO - Recovery Point Objective)
- Maintenance window constraints
- Geographic distribution requirements for disaster recovery

Consider sizing and cost implications for infrastructure selection, including Reserved Instances, operating system selection, and requirements for operational consistency and support.

Shared Resources

Before deploying individual EC2 instances, consider resource dependencies and establish reusable patterns.

Shared resources may include:

- Amazon accounts
- Target Amazon Region and Availability Zones
- VPC ID and subnet configurations
- Databases (for example, tenant databases hosting multiple NetWeaver stacks)
- Security groups
- IAM roles
- Shared storage (transport directories, EFS file systems)
- S3 buckets for backups, software distribution, and logging
- Load balancers
- Encryption keys and secrets
- Instance type and Reserved Instance requirements
- AMI selection (SLES for SAP or RHEL for SAP)
- Required Amazon service quotas and limits

For each resource type, establish patterns based on organizational boundaries: business unit, environment criticality (production/non-production), specific environments (development, test, sandbox), application type (BW, ECC), host type (ASCS, web dispatcher, application server), SAP System ID (SID), individual hosts, or Amazon service boundaries.

These design standards directly impact naming conventions, shareable resources, tagging strategies, and automation patterns.

Information gathering

As you work through the deployment process, consider how to populate your design and identify patterns for resource sharing. The following information will help you make consistent decisions across your SAP landscape.

Information	Description	Your Value
Region ID	Region where you want to deploy your Amazon resources	
Availability Zone	Availability Zone within your target region where you want to deploy your resources. For High Availability installations, you need two Availability Zones	
Amazon VPC ID	Amazon VPC where you want to deploy your Amazon EC2 instance for SAP installation	
Subnet ID	Subnet where you want to deploy your Amazon EC2 instance	
Key pair	Key pair generated in your target region with access to the private key	
Security group ID	Security group that you want to assign to your Amazon EC2 instance	
IAM instance profile	IAM instance profile with necessary permissions for SAP operations and Amazon service access	

Use this table to document your decisions and establish consistent patterns that can be reused across similar deployments in your SAP landscape.

Prerequisites

Design Scope

This guide provides technical implementation guidance for SAP NetWeaver infrastructure deployment. It is not a replacement for a comprehensive High Level Design (HLD). Work with your SAP Basis experts or Systems Integration partner to complete a full architectural design that addresses your specific business requirements, integration patterns, and operational procedures.

Specialized Knowledge

This guide assumes familiarity with Amazon services including Amazon VPC, Amazon EC2, Amazon EBS, Amazon EFS, and security groups. SAP NetWeaver architecture knowledge is required, including understanding of ASCS instances, application servers, and database connectivity patterns.

SAP Documentation

SAP Note 1656099 - SAP Applications on Amazon: Supported DB/OS and Amazon EC2 products

SAP Note 1588667 - SAP on Amazon: Overview of related SAP Notes and Web-Links

SAP Note 1588896 - Linux: Support Statement for SLES on Amazon Web Services

SAP Note 1618572 - Linux: Support Statement for RHEL on Amazon Web Services

SAP Note 2369910 - SAP Software on Linux: General information

SAP Note 1827960 - Adjusting operating system limits for SAP instances

Deployment Methods

Amazon Launch Wizard for SAP

Amazon Launch Wizard provides a guided deployment experience for SAP workloads, automatically provisioning and configuring Amazon resources based on SAP best practices. Launch Wizard simplifies the deployment process by:

- Automatically sizing compute and storage resources based on SAP requirements
- Configuring networking and security groups according to SAP communication patterns
- Setting up monitoring and backup solutions
- Providing cost estimates before deployment

For detailed information about Launch Wizard for SAP, see Amazon Launch Wizard for SAP User Guide.

When designing systems deployed through Launch Wizard, understand the underlying architecture and resource relationships. For comprehensive design considerations, see How Amazon Launch Wizard for SAP works.

Infrastructure as Code

For repeatable deployments and standardization across environments, consider Infrastructure as Code approaches such as Amazon CloudFormation.

Manual Deployment using Amazon Console or Amazon CLI

Install and configure the Amazon CLI with appropriate credentials and target region. Ensure IAM permissions include EC2, EBS, EFS, and Systems Manager access as required for the deployment.

Amazon Resource Selection and Configuration

Topics

EC2 Instance Selection

SAP Certified

Amazon provides SAP-certified instance types that meet SAP's performance and reliability requirements. Current generation instances offer improved price-performance ratios and enhanced networking capabilities compared to previous generations.

For SAP NetWeaver deployments, consider these certified instance families:

- Memory-optimized (R-series) - for memory-intensive SAP applications
- Compute-optimized (C-series) - for CPU-intensive workloads
- General purpose (M-series) - balanced compute, memory, and networking

Refer to the current SAP NetWeaver supported instances for the latest certifications and SAPS ratings.

⚠ **Important**

Use only supported operating system versions for SAP NetWeaver deployments. Avoid using OS versions that have reached End-of-Life (EOL), as OS vendors typically do not provide security patches or updates for EOL versions. Using EOL systems increases security risks and may prevent you from applying critical updates. Verify current support status with your OS vendor and SAP before deployment.

AMI Selection

Choose appropriate Amazon Machine Images (AMIs) for your SAP deployment:

- SUSE Linux Enterprise Server (SLES)
- Red Hat Enterprise Linux (RHEL)

Amazon Marketplace provides pre-configured AMIs from SUSE and Red Hat specifically optimized for SAP workloads. These images include:

- SAP-required kernel parameters and system settings
- Pre-installed SAP prerequisites and libraries
- Optimized storage and network configurations
- Regular security updates and patches

Search for:

- SUSE Linux Enterprise Server for SAP Applications
- Red Hat Enterprise Linux for SAP with HA and Update Services

Instance Characteristics

Select instances based on your workload requirements:

- **Enhanced networking**: Verify and enable enhanced networking for improved network performance

```
$ aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
 'Reservations[].Instances[].EnaSupport'
```

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ena-support
```

- **EBS optimization**: Enable EBS optimization for consistent storage performance

```
$ aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
 'Reservations[].Instances[].EbsOptimized'
```

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --ebs-optimized
```

- **Nitro System**: Current generation instances run on the Amazon Nitro System, providing consistent performance and security

Storage Selection

Storage Configuration Matrix

The following table provides a reference configuration for SAP NetWeaver file systems:

Mount Point	Ownership	Local Shared	Type	EFS Ref	Stora Class	Encry	Snap	Device (example)	Suggested Size (GB)
/	root	Local	EBS	-	gp3	Y	Y	/dev/xvda1	> 20 GB
`/tmp` (1)	root	Local	EBS	-	gp3	N	N		
`swap`	root	Local	EBS	-	gp3	N	N	/dev/xvdb	See SAP Guidance
`/usr/sap`	<sid>adm:sapsys (755)	Local	EBS	-	gp3	Y	Y	/dev/xvdc	> 20 GB

Mount Point	Ownership	Local Shared	Type	EFS Ref	Stora Class	Encry	Snap	Device (example)	Suggested Size (GB)
`/usr/sap/<SID>/ASCS<nn>` (2)	<sid>adm:sapsys (755)	Share (SID)	NFS	SHARED_<SID>	-	Y	N/A	-	-
`/usr/sap/<SID>/ERS<nn>` (2)	<sid>adm:sapsys (755)	Share (SID)	NFS	SHARED_<SID>	-	Y	N/A	-	-
`/sapmnt/<SID>`	<sid>adm:sapsys (755)	Share (SID)	NFS	SHARED_<SID>	-	Y	N/A	-	-
`/usr/sap/trans`	<sid>adm:sapsys (755)	Share (Lanc e)	NFS	SHARED_TRANS	-	Y	N/A	-	-
`/software`	root:root (755)	Share (Envi ent)	NFS	SHARED_CC MMON	-	Y	N/A	-	-
`/interfaces`	<sid>adm:sapsys (755)	Share (Envi ent)	NFS	SHARED_CC MMON	-	Y	N/A	-	-

Notes:

1. Consider separating /tmp from your root filesystem

2. Only required if configuring a highly available ASCS Cluster
3. For single systems, or systems entirely in a single AZ, EBS can be used for shared directories and exported from the instance where the ASCS resides.

General:

- Replace `<SID>` with your SAP System ID (e.g., PRD, DEV, QAS)
- Replace `<sid>` with lowercase system ID (e.g., prd, dev, qas)
- Replace <nn> with instance numbers (e.g., 00, 01, 10)
- EFS references represent logical groupings - actual EFS file system names should follow your naming conventions
- Sizes shown are minimum recommendations - adjust based on your specific requirements
- See SAP Note 1597355 - Swap-space recommendation for Linux

Local SAP Storage (EBS)

For SAP ABAP/NetWeaver applications, gp3 is the recommended storage type, providing suitable performance characteristics for most deployments. For critical workloads requiring increased durability, consider using io2 volumes. You can modify IOPS and throughput based on specific workload demands.

We recommend using XFS as your file system - a stable journaling filesystem well-suited for SAP NetWeaver workloads.

Use the following mount options:

For XFS: `noatime,nofail,logbsize=256k` For EXT4: `noatime,nofail,nodiratime`

We suggest encrypting all local EBS volumes and backing up key volumes using snapshots on a regular basis.

Shared SAP Storage

For SAP shared directories requiring concurrent access from multiple instances, Amazon provides several NFS storage options:

- **Amazon EFS**: Serverless, fully elastic NFS storage with automatic scaling

- **Amazon FSx for NetApp ONTAP**: High-performance NFS with advanced data management features
- **Exported EBS**: EBS volumes exported via NFS from a dedicated instance (single AZ deployments)

Amazon EFS

Amazon EFS provides serverless, fully elastic NFS storage that scales automatically without disrupting applications. EFS supports NFSv4.1 and NFSv4.0 protocols, making it suitable for SAP shared directories that require concurrent access from multiple instances.

EFS Configuration for SAP

- **File System Type**: Regional (recommended) - stores data redundantly across multiple Availability Zones for high availability
- **Performance Mode**: General Purpose (default) - provides the lowest latency per operation, suitable for SAP workloads
- **Throughput Mode**: Choose based on workload patterns:
  - `Elastic`: Automatically scales 1 MiB/s to 3 GiB/s based on activity. Recommended if sharing one EFS system across multiple instances.
  - `Bursting`: 100 MiB/s minimum with burst capability. More cost-effective for periodic access patterns like SAP media storage or low usage systems.
- **Storage Classes**: Standard for frequently accessed SAP data, Infrequent Access (IA) for archival content

EFS Security and Encryption

- Encryption at rest: Enable during EFS creation using Amazon managed keys or customer managed KMS keys
- Encryption in transit: Use the amazon-efs-utils package (efs-utils) for TLS encryption during mount operations. See Installing the Amazon EFS client
- Access control: Combine IAM policies, security groups, and POSIX permissions for comprehensive access management

EFS Creation Example

1. **Create an encrypted EFS file system with recommended settings**

```
$ aws efs create-file-system \
--creation-token SAP-TRANS-PROD \
--backup \
--encrypted \
--performance-mode generalPurpose \
--throughput-mode elastic \
--region us-west-2 \
--tags Key=Name,Value="SAP Transport Directory" Key=Environment,Value=Production
```

Parameter Explanations:

- `--backup`: Enables automatic daily backups with 35-day retention using Amazon Backup service (recommended)
- `--encrypted`: Enables encryption at rest using Amazon managed keys (data stored on EFS is encrypted)
- `--throughput-mode` elastic: Automatically scales throughput based on workload (consider bursting for sandbox environments or infrequently accessed filesystems like a locally available media directory)

2. **Create Mount Targets**

   Create mount targets in each subnet where your SAP instances will access EFS:

   ```
   $ aws efs create-mount-target \
   --file-system-id fs-12345678 \
   --subnet-id subnet-12345678 \
   --security-groups sg-12345678 \
   --region us-west-2
   ```

3. **Retrieve File System Information**

   Get the DNS name needed for mounting:

   ```
   $ aws efs describe-file-systems --creation-token SAP-TRANS-PROD --region us-west-2
   ```

4. **Allocate Directories (optional)**

   Based on the storage configuration matrix, organize which directories will be hosted in which Elastic File System according to usage and connectivity, consider whether consolidating EFS file systems may reduce the cost and management overhead - for example using a single mount point for shared administrative file systems. For more critical file systems, consider the resilience and performance scope of impact.

- SID-Specific EFS (EFS_<SID>)
  - `/usr/sap/<SID>/ASCS<nn>` - ASCS instance directory (optional for HA setups)
  - `/usr/sap/<SID>/ERS<nn>` - ERS instance directory (optional for HA setups)
  - `/sapmnt/<SID>` - SAP mount directory for the specific system
- Landscape-Wide EFS (EFS_TRANS)
  - `/usr/sap/trans` - Transport directory shared across all SAP systems in the landscape
- Environment-Wide EFS (EFS_COMMON)
  - `/software` - Software distribution directory (SAP media, installation files)
  - `/interfaces` - Interface files and configurations

For detailed EFS creation and configuration options, refer to the Amazon EFS User Guide.

Amazon FSx for NetApp ONTAP

Amazon FSx for NetApp ONTAP provides high-performance NFS storage with advanced data management capabilities. FSx for ONTAP offers:

FSx for ONTAP is particularly suitable for SAP environments requiring advanced storage features or migrating from on-premises NetApp systems. For detailed configuration guidance, refer to the Amazon FSx for NetApp ONTAP User Guide.

SAP Netweaver IAM Requirements

Policy Area	Purpose	Reference	Custom Policy Required	Managed Policy
Amazon Systems Manager Access	Patch management, parameter store, session manager	Policy Documentation	No	`AmazonSSMManagedInstanceCore`

Policy Area	Purpose	Reference	Custom Policy Required	Managed Policy
Amazon EFS Access	Shared file system mounting	Amazon managed policies for Amazon EFS	Optional (for restricted access)	`AmazonElasticFileSystemClientFullAccess`
Amazon S3 Bucket Access	Installation media, backups, file sync	Amazon managed policies for Amazon S3	Yes	N/A
SAP NetWeaver Pacemaker Cluster Requirements	Instance start/stop, route table updates	SAP NetWeaver on Amazon: high availability configuration for Netweaver (ASCS)	Yes	N/A
SAP Amazon Data Provider	Amazon Data Provider integration	Amazon Data Provider IAM Roles	Yes	N/A
Amazon Systems Manager for SAP	SAP-specific monitoring and management	SSM for SAP Policies	No	`AWSSystemsManagerForSAPReadOnlyAccess` or `AWSSystemsManagerForSAPFullAccess`
Amazon CloudWatch	Monitoring and logging		No	`CloudWatchAgentServerPolicy`

Implementation Notes

- IAM Policy Creation: Creating IAM Policies Guide
- Best Practices: Follow the principle of least privilege when creating custom policies
- Alternative: Use Amazon Launch Wizard for SAP for automated policy configuration

SAP NetWeaver Security Groups

Design security groups following the principle of least privilege, allowing only necessary communication between SAP components.

Implement separate security groups for different functional tiers:

- **SAP Application Security Group**: Communication between SAP application servers and ASCS instances
- **SAP Database Security Group**: Database access restricted to authorized SAP instances only
- **SAP Web Security Group**: External access control for web-based SAP components
- **Management Security Group**: Administrative access for Systems Manager, monitoring, and backup operations

Required ports

For SAP application ports, refer to TCP/IP Ports of All SAP Products and filter on Product Name `Application Server ABAP`.

Administrative ports:

- Port 2049: NFS for Amazon EFS access
- Port 22: SSH access (consider restricting to Amazon Systems Manager Session Manager)

Security group best practices

- Reference other security groups rather than IP ranges where possible
- Use descriptive names and descriptions for operational clarity
- Implement separate security groups for different tiers (application, database, web)
- Regularly review and audit security group rules

Launch and Setup Instance

Topics

- Launch an instance
- Remote Access with Amazon Systems Manager Session Manager
- Configure the hostname
- Install prerequisite packages
- Configure storage

Launch an instance

An instance is a virtual server in the Amazon Cloud. You launch an instance from an Amazon Machine Image (AMI). The AMI provides the operating system, application server, and applications for your instance.

There are multiple console and code driven methods for launching an EC2 Instance, see the full documentation at Launch an Amazon EC2 instance

If you are new to Amazon, we suggest exploring the options using the launch instance wizard, to understand configuration options including location, instance type, local storage allocation, security options and advanced options such as user data which you can specify operating system commands to run when you launch an EC2 instance. A full list of configurable parameters is available at Reference for Amazon EC2 instance configuration parameters

CloudFormation example

If using Amazon CloudFormation for infrastructure as code, the following snippet shows the minimum requirements for a SAP NetWeaver instance. This is not a complete CloudFormation template but demonstrates the key properties:

```
MySAPNetWeaverInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-0f0dcf1e0a0d26ea4  # Choose a marketplace or other AMI
    InstanceType: r7i.xlarge        # Choose a certified instance
    KeyName: my-key-pair
    SubnetId: subnet-1234567        # Distribute your instances in different subnets
```

```
   IamInstanceProfile: !Ref EC2InstanceProfile
   SecurityGroups:
     - !Ref InstanceSecurityGroup  # Define or Select Security Group
   BlockDeviceMappings:
     - DeviceName: /dev/xvda       # Root volume
       Ebs:
         VolumeSize: 50            # Root for OS and NetWeaver
         VolumeType: gp3
         DeleteOnTermination: true
     - DeviceName: /dev/sdb        # SAP application volume
       Ebs:
         VolumeSize: 100
         VolumeType: gp3
         DeleteOnTermination: true
     - DeviceName: /dev/sdc        # Swap volume
       Ebs:
         VolumeSize: 50
         VolumeType: gp3
         DeleteOnTermination: true
   UserData:
     Fn::Base64: | # Sample user data
       #!/bin/bash

       # Set hostname
       hostnamectl set-hostname saphost01

       # Install SSM Agent
       # Provide details depending on OS type and whether it is already part of the
AMI
```

Remote Access with Amazon Systems Manager Session Manager

For secure shell access to your SAP NetWeaver instances, Amazon Systems Manager Session Manager provides browser-based or CLI-based shell access without requiring SSH keys, bastion hosts, or open inbound ports. The SSM Agent comes pre-installed on many AMIs, but ensure it's running and has the necessary IAM permissions during your launch activities. For RHEL-based systems, see Installing SSM Agent on RHEL, and for SLES systems, see Installing SSM Agent on SLES. This allows immediate secure access to your instance for SAP installation and configuration tasks. For complete Session Manager documentation, see Amazon Systems Manager Session Manager.

Configure the hostname

Configure your instance hostname either during launch using user data or after launch through a shell session.

To configure the hostname after launch:

1. Connect to your SAP instance using Amazon Systems Manager Session Manager or SSH with your key pair.
2. Switch to the root user.
3. Update the hostname and domain name according to your requirements.

For detailed steps specific to your operating system, see How do I assign a static hostname to a private Amazon EC2 instance running RHEL or SLES? in the Amazon Knowledge Center.

For SAP-specific hostname requirements, see SAP Note 611361 - Hostnames of SAP ABAP Platform servers and SAP Note 2718300 - Physical and Virtual hostname length limitations.

Install prerequisite packages

Your Amazon EC2 instance requires internet access to download packages from the SUSE or Red Hat repositories.

⚠ **Important**

Ensure your instance has outbound internet connectivity or access to a local package repository before proceeding with package installation.

The following packages are required for SAP NetWeaver installation on Amazon. Depending on your baseline AMI (for example, RHEL for SAP or SLES for SAP), some packages may already be installed.

Package	Description	Manual Download	Required
`nfs-utils`	Network File System utilities for mounting Amazon EFS		Mandatory

Package	Description	Manual Download	Required
`nvme-cli`	NVMe command line interface for viewing Amazon EBS volume mapping		Recommended
`aws-cli`	Amazon Command Line Interface for Amazon service management.	See Installing or updating the latest version of the Amazon CLI for installation instructions.	Recommended
`aws-sap-data-provider`	Amazon Data Provider for SAP - enables SAP systems to retrieve Amazon infrastructure information.	Download the appropriate RPM package first from Amazon Data Provider installation guide.	Mandatory

Use the following commands to install packages on your system:

SLES systems:

```
$ sudo zypper install <package-name>
```

RHEL systems:

```
$ sudo dnf install <package-name>
```

Configure storage

This section explains how to configure the Amazon EBS volumes that were attached during instance launch for SAP NetWeaver installation.

Note

This guide uses NVMe device names (for example, /dev/nvme1n1) which are standard on Nitro-based instances. On non-Nitro instances, devices use different naming (for example, /dev/sdb). Adjust commands according to your device names.

Important

While LVM can be used for volume management, it is not recommended for SAP NetWeaver installations due to potential performance overhead and added complexity. Use direct device formatting as shown in this guide.

Configure local storage

1. **Identify attached volumes**

   Rescan for new or changed block devices, then identify the devices attached to your instance, their sizes, and associated volume IDs.

```
$ sudo partprobe
$ sudo lsblk -o NAME,SIZE,TYPE,FSTYPE,LABEL,PATH,SERIAL | sed 's/vol0/vol-0/g'
```

   Example output:

```
NAME        SIZE TYPE FSTYPE LABEL PATH           SERIAL
nvme0n1      10G disk              /dev/nvme0n1   vol-0abc123def456789a
#nvme0n1p1   10G part xfs    ROOT  /dev/nvme0n1p1
nvme1n1      50G disk              /dev/nvme1n1   vol-0xyz987uvw654321b
nvme2n1      50G disk              /dev/nvme2n1   vol-0pqr456mno789123c
```

2. **Create filesystems**

   Create XFS filesystems on the volumes allocated for SAP NetWeaver. Use labels to ensure consistent mounting across instance restarts.

```
$ sudo mkfs.xfs -f /dev/nvme1n1 -L USR_SAP
```

Tip

Labels provide consistent device identification across instance restarts and instance type changes. Always use labels in /etc/fstab by referencing `/dev/disk/by-label/LABEL_NAME`.

3. **Create mount points and configure fstab**

   Create the required directories and add entries to /etc/fstab for automatic mounting.

```
$ sudo mkdir /usr/sap
$ echo "/dev/disk/by-label/USR_SAP /usr/sap xfs noatime,nodiratime,logbsize=256k 0 0"
 | sudo tee -a /etc/fstab
$ sudo mount -a
$ df -h
```

Configure swap space

Configure swap space for SAP NetWeaver installation. For swap sizing recommendations, see SAP Note 1597355 - Linux swap space requirement.

1. **Create swap filesystem**

```
$ sudo mkswap -f /dev/nvme2n1 -L SWAP
```

2. **Configure swap in fstab and enable**

```
$ echo "/dev/disk/by-label/SWAP none swap sw 0 0" | sudo tee -a /etc/fstab
$ sudo swapon -L SWAP
```

3. **Verify swap is active**

```
$ sudo swapon -s
```

Configure NFS filesystems

Configure NFS filesystems for shared SAP directories such as transport directories or shared installation media.

1. **Test NFS mount and create directory structure**

   Test the NFS mount temporarily and create the required directory structure on the EFS filesystem.

   ```
   $ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,intr,timeo=600
    your-efs-mount-target.efs.region.amazonaws.com:/ /mnt/
   $ sudo mkdir -p /mnt/SHARED_SID
   $ sudo mkdir -p /mnt/SHARED_TRANS
   $ ls -la /mnt
   $ sudo umount /mnt
   ```

2. **Create permanent mount points**

   Create the required local directories for NFS mounts.

   ```
   $ sudo mkdir -p /sapmnt/SID
   $ sudo mkdir -p /usr/sap/trans
   ```

3. **Configure NFS mounts in fstab**

   Add NFS mount entries to /etc/fstab for automatic mounting.

   ```
   $ echo "your-efs-mount-target.efs.region.amazonaws.com:/SHARED_SID /sapmnt/SID nfs4
    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,intr,timeo=600 0 0" | sudo tee -a /etc/
   fstab
   $ echo "your-efs-mount-target.efs.region.amazonaws.com:/SHARED_TRANS /usr/sap/trans
    nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,intr,timeo=600 0 0" | sudo tee -
   a /etc/fstab
   ```

4. **Mount NFS filesystems**

   ```
   $ sudo mount -a
   $ df -h
   ```

> **Note**
>
> Replace `SID` with your actual SAP System ID (for example, PRD or DEV). Replace `your-efs-mount-target.efs.region.amazonaws.com` with your actual Amazon EFS mount target DNS name and region.

SAP NetWeaver on Amazon Deployment and Operations Guide for Windows

SAP specialists, Amazon Web Services

Last updated: November 2022

This guide provides guidance on how to set up Amazon resources and the Microsoft Windows Server operating system to deploy SAP NetWeaver on Amazon EC2 instances.

This guide is intended for SAP architects, SAP engineers, IT architects, and IT administrators who want to deploy SAP NetWeaver on Amazon.

About this Guide

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the Amazon Cloud. For the other guides in the series, ranging from overviews to advanced topics, see the SAP on Amazon Technical Documentation home page.

This guide is for users who are responsible for planning, architecting, and deploying SAP NetWeaver on Amazon. You should have a good understanding of Amazon services, general networking concepts, Windows Server operating systems, and SAP NetWeaver administration. This document guides you through the steps required to successfully launch and configure the resources required for SAP NetWeaver on Windows.

Instructions in this document are based on the recommendations provided by SAP and Microsoft for SAP NetWeaver on Windows as described in the following OSS notes:

SAP NetWeaver on Windows OSS Notes

SAP OSS Note	Description
1656099	SAP Applications on Amazon: Supported DB/ OS and Amazon EC2 products
1409608	Virtualization on Windows
1732161	SAP Systems on Windows Server 2012 (R2)

SAP OSS Note	Description
2384179	SAP Systems on Windows Server 2016
2751450	SAP Systems on Windows Server 2019
1564275	Install SAP Systems Using Virtual Host Names on Windows
3143497	SAP Systems on Windows Server 2022

In addition, this document also follows best practices from Amazon, Microsoft, and SAP for SAP NetWeaver deployments on Windows. See the recommended reading section for more details.

This document doesn't provide guidance on how to set up network and security constructs, such as Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables, ACLs, NAT Gateway, Amazon Identity and Access Management (IAM) roles, and Amazon Security Groups. Instead, it focuses on how to configure and maintain the compute, storage, and operating system constructs for SAP NetWeaver deployment and operation on Windows on Amazon.

SAP NetWeaver is also available to deploy on Linux. If you're considering using Linux, see the SAP NetWeaver Quick Start for Linux.

Prerequisites

Specialized Knowledge

Before you follow the configuration instructions in this guide, we recommend that you become familiar with the following Amazon services. (If you are new to Amazon, start with the Getting Started Resource Center.)

- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Virtual Private Cloud (Amazon VPC)
- Amazon Identity and Access Management (IAM)
- Amazon Elastic Block Store (Amazon EBS)
- Amazon FSx
- Amazon Simple Storage Service (Amazon S3)

- Amazon Systems Manager
- Amazon CloudFormation
- Amazon CloudTrail
- Amazon Control Tower

Recommended Reading

We also recommend reading these overview and best practice guides:

- SAP on Amazon Overview and Planning
- Getting Started with Architecting SAP on the Amazon Cloud
- Best Practices for Windows on Amazon EC2

Technical Requirements

1. Ensure that any service limits are high enough and the current usage low enough to be able to launch the resources that you need. If necessary, request a service limit increase for the Amazon resource that you're planning to use. In particular:
   a. Ensure that your EC2 service limits are sufficient to launch the instances that you need for your SAP NetWeaver system.
   b. Ensure that your VPC service limits are sufficient to launch a new VPC (if necessary) or individual network resources within your VPC, such as Elastic IP addresses.
2. Gather the following information about your existing Amazon resources. You will need this information to create your Amazon EC2 and Amazon EBS resources using the Amazon Command Line Interface (Amazon CLI) commands:

Amazon Resource Information Required

Information Needed	Description
Region ID	Region where you want to deploy your Amazon resources
Availability Zone	Availability Zone within your target Region where you want to deploy your resources

Information Needed	Description
Amazon VPC ID	Amazon VPC where you want to deploy your Amazon EC2 instance for SAP installation
Subnet ID	Subnet where you want to deploy your Amazon EC2 instance
AMI ID	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instance. You can find the latest Linux AMIs in Amazon Marketplace
Key Pair	Make sure that you have generated the key pair in your target Region, and that you have access to the private key
Security Group ID	Name of the security group that you want to assign to your Amazon EC2 instance. See the appendix for detailed information about the security group for SAP instances
Access Key ID	Access key for your Amazon account that will be used with Amazon CLI tools
Secret Access Key	Secret key for your Amazon account that will be used with Amazon CLI tools

a. Ensure that you have a key pair that you can use to launch your Amazon EC2 instances. To import or create a new key pair, see Amazon EC2 Key Pairs and Windows Instances.

b. Ensure that you know the network details, such as VPC-ID and Subnet-ID, of the VPC where you plan to launch your Amazon EC2 instances to host your SAP NetWeaver application.

c. Ensure that you have the required ports open on the security group attached to your Amazon EC2 instance hosting your database, to allow communication between your database and your SAP NetWeaver application. If needed, create new security groups that allow network traffic over both the database ports and the SAP NetWeaver application ports. For a list of SAP ports, see TCP/IP Ports of All SAP Products.

3. If you plan to use the Amazon Command Line Interface (Amazon CLI) to launch your instances, ensure that you have installed and configured the Amazon CLI with the appropriate credentials. See Configuring the Amazon CLI for more details.
4. If you plan to use the Amazon Management Console to launch your instances, ensure that your IAM user has permission to launch and configure Amazon EC2, Amazon EBS, etc. See the IAM User Guide for more details.
5. Ensure that you have the required SAP software available either via an S3 bucket or on a file share accessible from Windows, such as Amazon FSx. For the fastest installation experience, we recommend copying the required software to an EBS volume attached to the relevant EC2 instance before running the install. This is best set up as a separate volume (mapped to a new drive in Windows) that, after completion of the installation, can then be detached and either deleted or re-attached to other EC2 instances for further installations. We recommend using the Amazon CLI for this. Be sure to assign the appropriate IAM role permissions to the EC2 instance to allow S3 access.
6. If the installation type is distributed or high availability (HA), it will need to be a domain-based installation and a domain controller is required. If desired, you can use Amazon Directory Service for this purpose. Amazon Directory Service for Microsoft Active Directory, also known as Amazon Managed Microsoft AD, enables your directory-aware workloads and Amazon resources to use managed Active Directory in Amazon. For details, see Amazon Directory Service and Create Your Amazon Managed Microsoft AD directory.

   When doing a domain-based installation, `sapinst.exe` should be run by a user with domain administration privileges (but not the `<SID>adm` user) or a domain administrator must complete the appropriate preparatory steps. For more details, consult the SAP NetWeaver installation guide for your version of SAP NetWeaver.
7. To create an Amazon FSx file system, you need the following prerequisites:

   a. An Amazon account with the permissions necessary to create an Amazon FSx file system and an Amazon EC2 instance. For more information, see Setting Up.

   b. An Amazon EC2 instance running Microsoft Windows Server in the VPC based on the Amazon VPC service that you want to associate with your Amazon FSx file system. For information on creating an EC2 Windows instance, see Getting Started with Amazon EC2 Windows Instances.

   c. Amazon FSx works with Microsoft Active Directory to perform user authentication. You join your Amazon FSx file system to an Amazon Directory Service for Microsoft Active Directory. For more information, see Create Your File System.

d. This guide assumes that you haven't changed the rules on the default security group for your VPC. If you have changed them, you need to ensure that you add the necessary rules to allow network traffic from your Amazon EC2 instance to your Amazon FSx file system. For more details, see Security.

e. Install and configure the Amazon Command Line Interface (Amazon CLI).

For additional details on these prerequisites, see Prerequisites for Getting Started.

Planning the Deployment

Plan your SAP system landscape according to the SAP Master Guide for your version of SAP NetWeaver and your combination of operating system and database.

Topics

- Select the Region
- Architecture Options
- Security and Compliance
- Sizing
- Operating System
- Compute
- Storage
- Network

Select the Region

In choosing the Region for deployment, you'll need to consider some key factors. For more details, see our Overview and Planning guide.

- Service availability
  - Not all Amazon services or features are available in all Regions. Verify that all services and features that you want to use in your deployment are available in the Region you choose. You can check availability on our website. If certain services or features are not available in your desired Region, there are alternatives that we mention in the guide.
  - For SAP workloads discussed in this guide, this is particularly true for:

    - EC2 instance types
    - Amazon FSx for Windows File Server
    - Amazon Backup
- Proximity and connectivity options
- Data residency
  - You retain complete control and ownership over your data in the Region in which it is physically located, making it easy to meet regional compliance and data residency requirements.

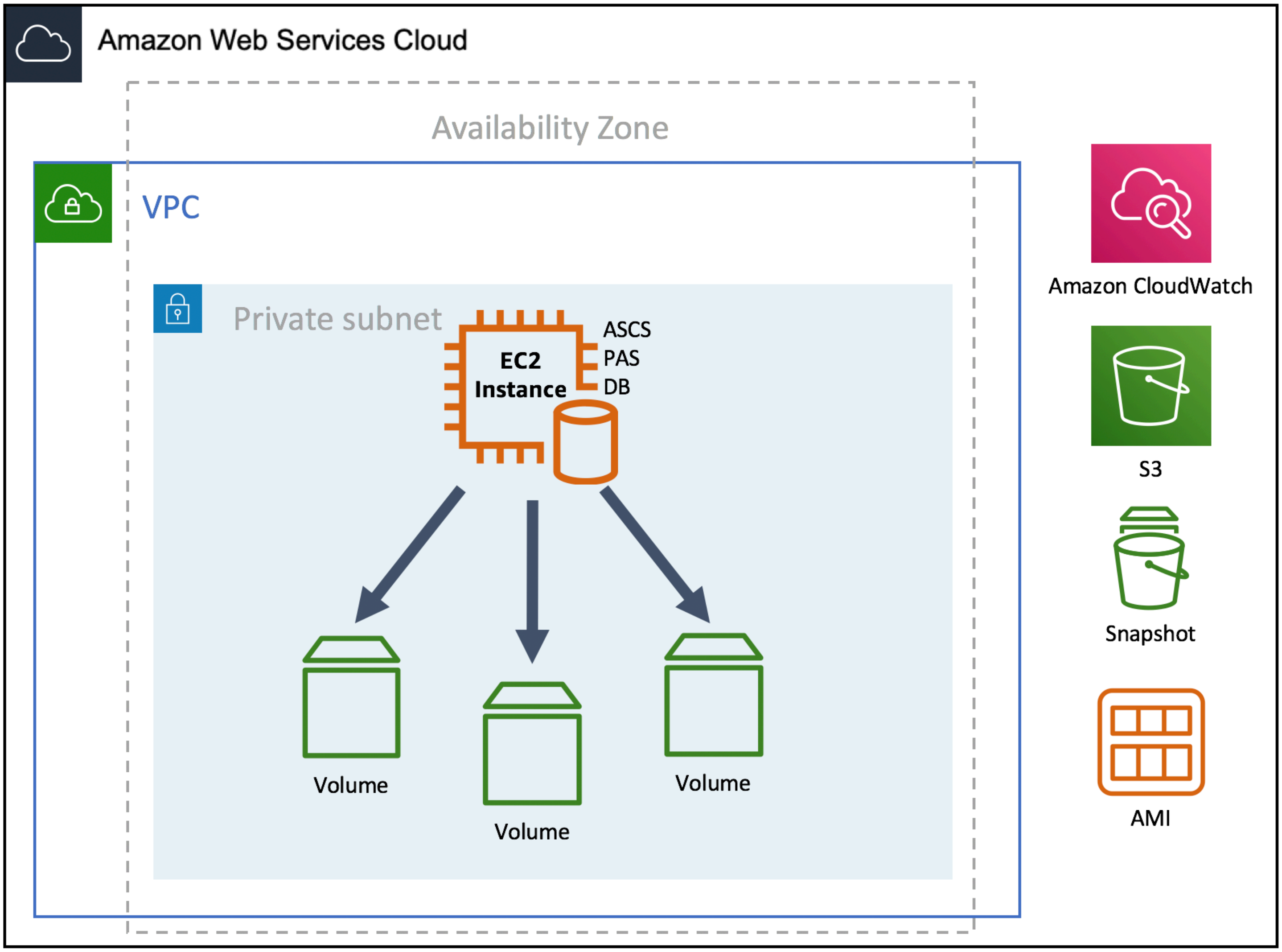
Architecture Options

Topics

- Standard System Deployment
- Distributed System Deployment
- High Availability System Deployment

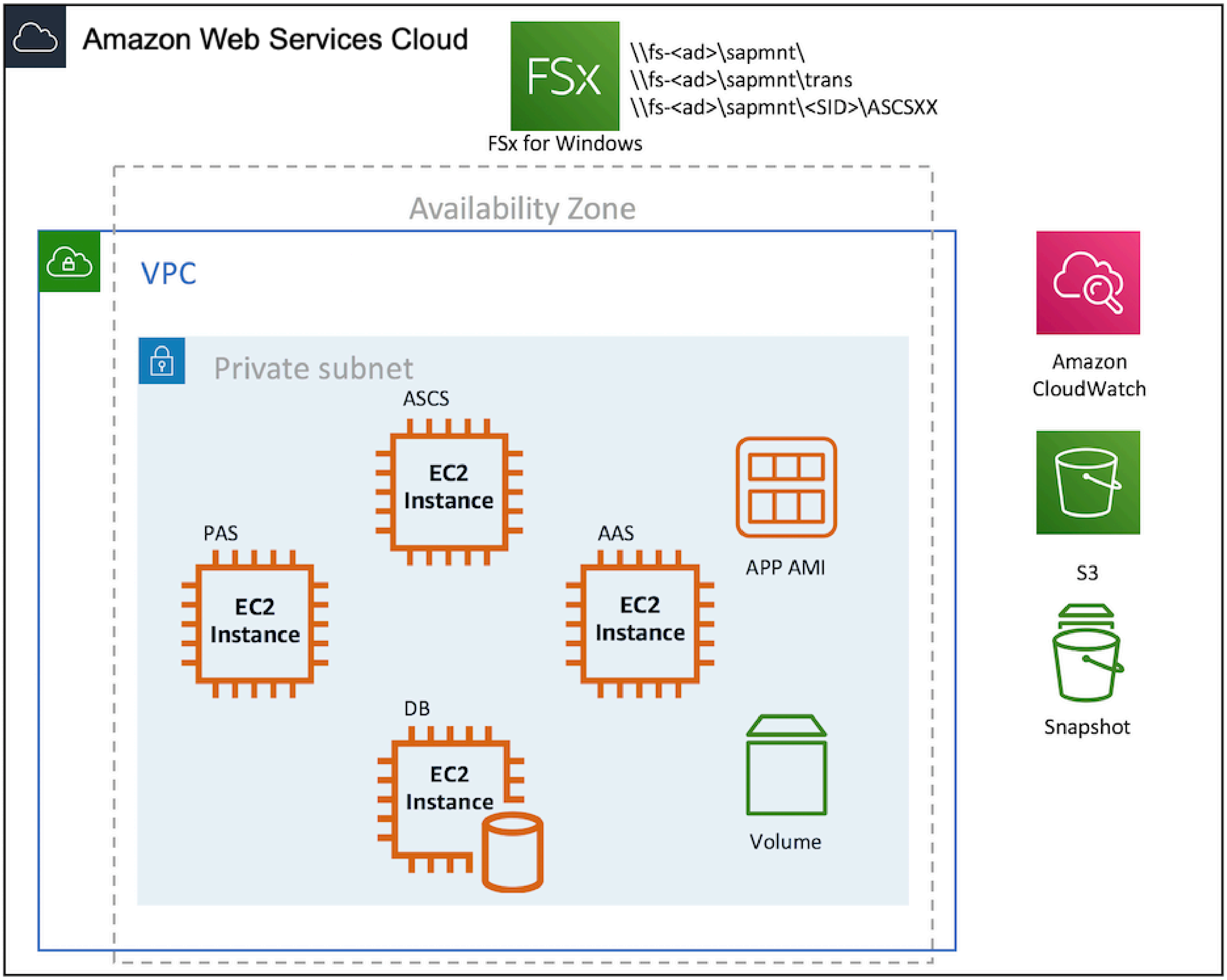
Standard System Deployment

Standard system or single host installation: all main instances of SAP NetWeaver (ASCS/SCS, database, and PAS) run on one Amazon EC2 instance. This option is best suited for non-production workloads.



Distributed System Deployment

Distributed system: every instance of SAP NetWeaver (ASCS/SCS, database, PAS, and optionally AAS) can run on a separate Amazon EC2 instance. This option is suited for both production and non-production workloads.



High Availability System Deployment

High availability (HA) system: used for business-critical applications. With this option, all the services that are single points of failure are deployed across multiple Availability Zones for fault tolerance.

For SAP NetWeaver, the key single points of failure are:

- the central services (ASCS/SCS)
- the global and transport filesystems

To protect against hardware failure of Amazon EC2 within an Availability Zone, you can enable EC2 instance recovery. See Recover Your Instance for more details on this feature. You can use scripts to start the SAP NetWeaver application automatically after instance recovery. You can further configure SAP application work processes to reconnect to your database after recovery. Consult the documentation for further restrictions. This option is not application aware and does not protect the application against Availability Zone failure, which makes it a good option for non-production systems. It also can be used for production systems but you might want to consider a Multi-AZ solution for this situation as well.

For HA solutions, it’s important to be aware of two concepts within a VPC: shared storage and the Overlay IP address.

Shared Storage

EBS volumes are specific to a single Availability Zone and can only be attached to a single EC2 instance at a time. However, in distributed or HA deployments, shared storage is required for the global and transport filesystems. On Amazon, this storage can be provided by building an NFS server or by using Amazon FSx. Amazon FSx provides shared file storage with full support for the SMB protocol, Windows NTFS, Active Directory integration, and Distributed File System (DFS).

If using such a solution in the context of a high availability installation, the shared storage solution you choose could introduce a single point of failure without appropriate protection. This can be protected against by:

- Clustering the NFS server providing the shared filesystem
- Clustering the host that is sharing the filesystems
- Using Amazon FSx. For workloads that require Multi-AZ redundancy to tolerate temporary AZ unavailability, you can create multiple file systems in separate AZs. Amazon FSx supports Microsoft’s Distributed File System (DFS) Replication and Namespaces. DFS Replication allows you to automatically replicate data between two file systems, and DFS Namespaces allows you to configure automatic failover.

High availability

You can use a high availability (HA) clustering solution for autonomous failover of the central services across Availability Zones. There are multiple SAP-certified options for this clustering software on Windows listed on the SAP website, and it’s also possible to build and automate your own solution. HA solutions that have been tested and are known to work on Amazon include:

- Veritas InfoScale:
  - Veritas InfoScale for SAP on Amazon
  - Veritas InfoScale for Windows compatibility list
- SIOS:
  - SIOS DataKeeper with Windows Server Failover Cluster (WSFC)
  - SIOS DataKeeper Cluster Edition on Amazon Quick Start
  - SAP on Amazon Blog: Implementing HA and DR for Microsoft SQL Server
- NEC ExpressCluster
- Windows Server Failover Cluster (WSFC) with native Windows and Amazon services
  - SAP on Amazon Blog: How to setup SAP NetWeaver on Windows MSCS for SAP ASCS/ERS on Amazon

Support and certification

SAP clustering software is supported by the cluster software vendors themselves, not by SAP. SAP only certifies the solution. Any custom-built solution is **not** certified and will need to be supported by the solution builder.

In this guide, we focus on the distributed installation type on Windows in Amazon. More details on how to deploy and operate SIOS, Veritas, and WSFC clusters are available on their respective websites linked above. For effective use of WSFC, Windows Server 2016, or later, is required.

The key features to be aware of with the WSFC solution are:

- ASCS and a separate ERS instance set up within Windows Cluster Manager
- Scale-Out File Server is a feature that is designed to provide scale-out file shares that are continuously available for file-based server application storage
- Storage Spaces Direct uses standard servers with local-attached drives to create highly available, highly scalable software-defined storage. **This requires a minimum of Windows Server 2016 and NVMe storage (so nitro-generation EC2 instances are required).**
- Amazon FSx for Windows File Server

Also read the High Availability with Microsoft Failover Clustering section of the SAP NetWeaver installation guide.

Security and Compliance

These additional Amazon security resources can help you achieve the level of security that you require for your SAP NetWeaver environment on Amazon:

- Amazon Cloud Security Center
- CIS Amazon Web Services Foundations whitepaper
- Introduction to Amazon Security
- Amazon Security Best Practices whitepaper
- Amazon Well-Architected Framework Security Pillar whitepaper
- Network and Security topic from the *Amazon EC2 User Guide for Windows Instances*

OS Hardening

You may want to lock down the OS configuration further, for example, to avoid providing a NetWeaver administrator with root credentials when logging into an instance.

We provide guidance on how to best secure your Windows EC2 instances:

- Read our best practices guide for securing Windows on EC2.
- Read our general best practices guide for securing EC2 instances.
- Use Amazon Inspector, an automated security assessment service that helps you test the network accessibility of your EC2 instances and the security state of your applications running on the instances.

You can also refer to the following SAP note:

- 1837765: Security policies for <SID>adm and SapService<SID> on Windows

Encryption

Cloud security at Amazon is the highest priority. A core aspect of securing your workloads is encrypting your data—both at rest and in transit.

When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data at rest, and data in transit between an instance and its attached EBS storage. You can expect the same IOPS performance on encrypted volumes as on unencrypted volumes, with a minimal effect on latency. Encryption and decryption are handled transparently and require no additional action from you or your applications.

Similarly, all Amazon FSx file systems are encrypted at rest with keys that are managed using Amazon Key Management Service (Amazon KMS). Data is automatically encrypted before being written to the file system, and automatically decrypted as it is read. These processes are handled transparently by Amazon FSx, so that you don’t have to modify your applications.

For Amazon S3, you can protect data in transit by using SSL/TLS or client-side encryption, and protect data at rest by using either server-side or client-side encryption.

You can find more information about encryption from the specific service documentation:

- Encrypting Amazon FSx Data at Rest and Data in Transit
- Protecting Amazon S3 Data Using Encryption
- Amazon EBS Encryption

Security Groups / NACLs

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

Customers often separate the SAP system into multiple subnets, with the database in a subnet separate from the application servers, and other components, such as a Web Dispatcher, in another subnet—possibly with external access.

If workloads are scaled horizontally, or high availability is necessary, you might consider including multiple, functionally similar, EC2 instances in the same security group. In this case, you’ll need to add a rule to your security groups.

If Microsoft Windows Server is used, some configuration changes may be necessary in the security groups, route tables, and network access control lists (ACLs). You can refer to the operating system product documentation or other sources, such as the Security Group Rules Reference in the Amazon EC2 documentation, for more information.

A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets (they're stateless firewalls at the subnet level). You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

For further information on network considerations for SAP workloads, see our SAP on Amazon network documentation.

API Call Logging

Amazon CloudTrail is a web service that records Amazon API calls for your account and delivers log files to you. The information recorded includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the Amazon service.

With CloudTrail, you can get a history of Amazon API calls for your account, including API calls made via the Amazon Management Console, Amazon SDKs, command line tools, and higher-level Amazon services, such as Amazon CloudFormation. The Amazon API call history provided by CloudTrailenables security analysis, resource change tracking, and compliance auditing.

Notifications on Access

You can use Amazon Simple Notification Service (Amazon SNS) or third-party applications to send notifications about SSH logins to your email address or mobile phone number.

Sizing

One of the first points to consider is whether this deployment is a completely new project (greenfield) or a migration. Sizing then applies across three key areas: compute, storage, and network.

Compute

Understanding the compute requirement helps you select the best matching EC2 instance type from the available list of SAP-certified instances.

If this is a greenfield deployment, use the SAP Quick Sizer tool to calculate the SAP Application Performance Standard (SAPS) compute requirement and use that value to select the EC2 instance that is the closest match with the best cost. Also check that the EC2 instance you select provides sufficient EBS and overall network throughput to satisfy your application requirements.

For migrations, you can use a number of data sources to help choose the best instance size:

- Source system utilization and workload patterns (EarlyWatch alert reports, etc.)
- Source system specification: CPU, memory, storage size, throughput, IOPS, network
- Source system SAPS rating

Selecting EC2 Instance Type

It’s important to consider storage and network performance as well as compute, to ensure the selection of the best EC2 instance type.

After the workload is running on Amazon, you can use a process called right sizing to refine the size that you actually need. Right sizing is best thought of as an on-going process.

Storage

Deploying SAP NetWeaver on Windows on Amazon requires a minimum amount of storage and storage layout as per the SAP NetWeaver documentation for Windows. See the SAP documentation for further details on minimum and recommended storage sizes and storage layout. The EBS volumes should be created to match these requirements.

Verify that the amount of storage is adequate to provide sufficient I/O performance, as the performance of a General Purpose SSD (gp2) volume is related to the overall volume size. To achieve higher throughput and IOPS performance, the striping of volumes is often considered but this is usually not necessary for the NetWeaver application layer.

Network

Network performance is often not explicitly stated as a requirement in SAP sizing, but you can check the network performance of each EC2 instance type to ensure that you are delivering the required performance.

Operating System

If you plan on using Windows other than via Amazon EC2 for Windows Server, then ensure that you have the appropriate licenses and tenancy type selected. For more details, refer to your licensing terms and conditions, and see our Windows on Amazon webpage.

A base AMI is required to launch an Amazon EC2 instance. For SAP NetWeaver workloads on Windows, you need to run Windows Server 2012 R2, or later, because older versions are no longer supported by SAP. If you are using bring your own license (BYOL) instead of license-included for Windows Server, you will need to create your own AMI. See Microsoft Licensing on Amazon.

Ensure that you have access to the appropriate Windows Server AMIs before proceeding.

As with any operating system, we recommend that you keep the OS up-to-date with the latest patches. You can also refer to the following SAP Notes:

- 2325651: Required Windows Patches for SAP Operations

Compute

Amazon has certified multiple instance families of various sizes for running SAP NetWeaver workloads. For a complete list of the certified EC2 instance types, see Amazon EC2 Instance Types for SAP.

Select the appropriate EC2 instance type based on your CPU, memory, and SAPS requirements. Amazon recommends that, when possible, you use the latest generation of your selected instance family that is SAP certified.

Storage

Refer to the sizing section for resources on SAP’s standard recommendations. If no storage performance requirements are available, Amazon recommends General Purpose SSD (gp2) as the default EBS volume type for SAP workloads.

In practice, application servers will have a minimum of two volumes, mapped to the `C:` and `D:` drives. The `C:` drive is the boot volume containing the OS, and the `D:` drive is used to host the SAP software. We recommend using an additional, temporary volume for SAP software downloads (typically mapped as the `E:` drive).

If the installation type is distributed or HA, fileshares for the global filesystem and transport directories will need to be used across all relevant EC2 instances. In this guide, we use the standard Windows file sharing features to share these directories from the EC2 instance hosting the central services. The `sapinst.exe` installer creates these shares automatically if it is run as a user with appropriate permissions.

Customers can also use NFS-based solutions, such as Amazon FSx, third-party solutions available from the Amazon Marketplace, or custom-built solutions. Choosing the correct NFS solution is beyond the scope of this guide. If you use such a solution as part of a high availability deployment, consider that the NFS solution could itself be a single point of failure without appropriate protection.

Network

Ensure that you have your network constructs set up to deploy resources related to SAP NetWeaver. If you haven't already set up network components, such as Amazon VPC, subnets, and route tables, you can use the Amazon Quick Start for Modular and Scalable VPC Architecture to easily deploy scalable VPC architecture in minutes. See the deployment guide for more details, then set up your EC2 instances for the NetWeaver application server within this VPC.

You also will need to set up a secured network connection between the corporate data center and the VPC, along with the appropriate route table configuration, if this has not already been configured.

Deployment Steps

Topics

Step 1: Prepare your Amazon Account

In this example, we step through setting up a sample environment for the installation, which includes a public subnet for RDP and SSH access via the internet. In this scenario, we are using the Amazon Quick Start for Modular and Scalable VPC Architecture in a Single-AZ deployment to create the VPC, subnets, security groups, and IAM roles. This setup is just an example and you should follow your own network layout and ensure that you comply with your security standards. This could include:

- Using an Amazon Quick Start that suits their requirements such as a Multi-AZ deployment of the Amazon Quick Start for SAP HANA
- Using a landing zone solution, like Amazon Control Tower
- Working with your cloud team (for example, a Cloud Center of Excellence or CCoE) to ensure adherence to existing standards

  1. Check the Region where you want to deploy your Amazon resources:

     a. You'll have picked the Region you want to deploy in during your planning phase.

     b. Display the Amazon CLI configuration data:

     ```
     $ aws configure list
     ```

     In the command output, make sure that the default Region that's listed is the same as the target Region where you want to deploy your Amazon resources and install SAP NetWeaver.

  2. If this is a distributed or HA installation type:

     a. Create a new security group specifically for the EC2 instances running the NetWeaver application servers that allows traffic over the required ports for remote access from the public subnet, for example, RDP.

     b. Edit that security group to allow traffic over ports required for SAP NetWeaver based on your specific use-case. Specify the source as being the security group itself and ensure that this security group is attached to all EC2 instances that will run application servers.

     c. For distributed or HA installations, ensure that the security group attached to each application and central services server allows communication between them over the required ports. You can create a rule that references a security group as its own source, and allow traffic on the required ports for that rule.

  3. Create a JSON file for the Amazon EBS storage volumes (the volume sizes used are indicative only and should be customized based on your sizing requirements):

     ```
     [
        {
            "DeviceName": "xvdb",
     ```

```
            "Ebs": {
                "VolumeSize": 50,
                "VolumeType": "gp2",
                "DeleteOnTermination": true
            }
        },
        {
            "DeviceName": "xvdc",
            "Ebs": {
                "VolumeSize": 50,
                "VolumeType": "gp2",
                "DeleteOnTermination": true
            }
        }
]
```

4. Amazon Windows AMIs provide additional software that prepares an instance when it first boots up. This is either the EC2Config service (Windows AMIs prior to Windows Server 2016) or EC2Launch (Windows Server 2016, or later). After the devices have been mapped to drives, they are initialized and mounted. The root drive is initialized and mounted as `C:\`. By default, when an EBS volume is attached to a Windows instance, it can show up as any drive letter on the instance. You can change the settings to set the drive letters of the volumes per your specifications. For more information, see the device naming section for storage on Windows.

5. Install your selected database product. If this is a distributed or high availability deployment, install your selected database product in a separate EC2 instance dedicated to that purpose. Otherwise, install your database in the existing EC2 instance. For more details, see the Amazon Documentation for your database.

6. Launch EC2 instances for the SAP installation in your target Region by using the information you gathered in the preparation phase. You will also be creating the storage volumes required for the SAP installation and attaching them to the Amazon EC2 instance for the SAP installation.

   Ensure that you enable detailed monitoring on each instance as this is required for SAP support. (The sample commands provided below enable this.)

   Make sure that you choose one of the Amazon EC2 Instance Types for SAP. Sample Amazon CLI syntax is given below.

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--monitoring Enabled=true \
--count <number-of-EC2-instances> \
--instance-type <instance-type> \
--key-name=<name-of-key-pair> \
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings https://<bucket>.s3.amazonaws.com/<file>.json
```

Example

This example enables detailed monitoring (data is available in 1-minute periods for an additional cost) which is a support prerequisite for SAP workloads on Amazon EC2.

```
$ aws ec2 run-instances \
--image-id ami-012345678901234ab \
--monitoring Enabled=true \
--count 1 \
--instance-type m5.2xlarge \
--key-name=my_key \
--security-group-ids sg-01234567890abcdef \
--subnet-id subnet-0123456789abcdefg \
--block-device-mappings https://example.s3.amazonaws.com/file.json
```

Step 2: Prepare Each EC2 Instance for SAP Installation

1. Log into the newly created RDP host in the public subnet. We will call this **jumpbox** for easy reference. Do this by either using Amazon Systems Manager Session Manager (for command line tasks), or by doing the following:

   a. Go to the Amazon Management Console, select the EC2 instance **jumpbox**, and choose **Connect**. Download the RDP file from the pop-up that appears.

   b. Click **Get Password** and provide your private key to decrypt the password. This is the password for the local administrator on **jumpbox**.

   c. Open the RDP file in your preferred RDP program, and connect to **jumpbox**. Log in with user Administrator and the password that you just retrieved in step 1b.

   d. After you are logged in, go back to the Amazon Management Console and repeat step 1a and step 1b, but specify the EC2 instance where you will install NetWeaver. We'll call this **nw-ascs** for reference. Copy the downloaded RDP file to **jumpbox**.

   e. While logged into **jumpbox**, open the RDP file for **nw-ascs** in your preferred RDP program.

2. Log in as a user with administrator privileges but not an existing `<SID>adm` user (as per SAP's requirements).
3. Install the Amazon CLI tools or use the Amazon Tools for PowerShell provided with the Windows AMI.
4. Install the Java Runtime Environment (JRE) version that is compatible with your SAP installation software.
5. Install the Amazon Data Provider, following the instructions for Windows in the Installation and Operations Guide.
6. Install and configure Amazon Systems Management Agent (SSM Agent).

Step 3: Create Amazon FSx Volumes

1. The global fileshare and transport directories need to be available across all your SAP system's EC2 instances. In this guide, we assume that you are using Amazon FSx for this purpose.
2. Be sure that you've satisfied the prerequisites in the Technical Requirements section of this document. You will need to have already deployed your EC2 instances in each of the Availability Zones where you will create Amazon FSx filesystems.
3. Follow the step-by-step instructions in the Getting Started with Amazon FSx documentation
4. For high availability deployments that require Multi-AZ redundancy to tolerate temporary AZ unavailability, follow the instructions to create multiple file systems in separate AZs.

Step 4: Prepare and Run the SAP Installation Prerequisites Check

1. Download the SAP installation media for SWPM (the latest appropriate version for your desired NetWeaver installation), your desired NetWeaver software version for Windows, the latest compatible SAP kernel, and any other required files (such as: the host agent, IGS, database client tools, SAP GUI, the SAPCAR archiving tool, and the SAP download manager) to an attached EBS volume as described in the prerequisites (usually from Amazon S3 using the Amazon CLI tools).
2. Run the SAP prerequisite checker via SWPM on the desired host servers to ensure that you have met SAP's technical prerequisites. When you first run SWPM, you may have to enter the sign-in credentials of the Windows user that you're currently logged in as.
3. Launch SWPM by running the `sapinst.exe` executable. Specify `SAPINST_USE_HOSTNAME=<FQDN>` when launching to override the default DNS name if necessary, for example, with `<hostname>.local`.

4. Complete the recommended prerequisite steps as identified by the SAP prerequisite checker as per your specific requirements. Some common prerequisites for Windows Server operating systems are:

   a. Ensure that the hostname is ⇐ 13 characters in an alphanumeric string (hyphens can also be included). This can be done at the command line using Windows PowerShell by executing the following command:

   ```
   Rename-Computer <new-hostname>
   ```

   b. Optionally add the server to your Active Directory domain (this can be done with Amazon Systems Manager).

   c. Pagefile size will have a minimum recommended value based on services selected.

   d. Continuous Availability feature on Windows Server 2012 R2 can result in long wait times. See SAP note 1823833 for a fix.

Step 5: Install SAP NetWeaver on Amazon EC2

You are now ready to install SAP NetWeaver on this EC2 instance using the downloaded software. Proceed with the instructions in the SAP installation guide for your version of SAP NetWeaver.

You will need to do this for a minimum of:

- the ASCS instance
- the DB instance (on the installed database server)
- the PAS instance

and optionally for:

- other AAS instances
- ERS instance on the second ASCS node (in different AZ)

Operations

Topics

- Tagging Amazon Resources

- Monitoring
- Backup and Restore
- Storage
- Operating System Maintenance
- High Availability
- Disaster Recovery
- Compute
- Cost Optimization
- Automation
- Support

Tagging Amazon Resources

A tag is a label that you assign to an Amazon resource. Each tag consists of a *key* and an optional *value*, both of which you define. Adding tags to the various Amazon resources will not only make managing your SAP environment much easier but can also be used to quickly search for resources. Many Amazon EC2 API calls can be used with a special tag filter. Refer to Amazon Tagging Strategies and use it as a starting point to define the tags you need for your resources. Some examples on how you can use tags for operational needs are:

- You can tag your EBS Volumes to identify their environment (for example Environment= DEV/ QAS/PRD etc.) and use these tags to create backup policies for EBS Volumes
- You can use similar tags as in above example with EC2 instances and use them for patching your operating systems or running scripts to stop/start application or EC2 instances.

Monitoring

Amazon provides multiple native services to monitor and manage your SAP environment. Services like CloudWatch and CloudTrail can be leveraged to monitor your underlying infrastructure and APIs respectively. CloudWatch provides ready-to-use KPIs for CPU, disk utilization and also allows you to create custom metrics if your specific KPIs that you would like to monitor. CloudTrail allows you to log the API calls made to your Amazon infrastructure components.

Backup and Restore

Snapshots and AMIs

A common approach for backing up your SAP NetWeaver application servers is using snapshots and AMIs.

All your data is stored on Amazon EBS volumes attached to the SAP NetWeaver application servers. You can back up the data on these volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups of Amazon EBS volumes, which means that only the blocks on the device that have changed after your most recent snapshot are saved. For more details on this, see Creating an Amazon EBS Snapshot.

An Amazon Machine Image (AMI) provides the information required to launch an instance along with a block device mapping of all EBS volumes attached to it.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can check the No Reboot option.

To take application-consistent snapshots of all EBS volumes attached to your instance using Windows Volume Shadow Copy Service (VSS), see Creating a VSS Application-Consistent Snapshot. This allows you to create a copy of the image without rebooting the instance.

You can use Amazon Backup to centrally configure backup policies and monitor backup activity for these snapshots.

After you have completed the SAP installation and post installation steps, you should create an image of the instance. Amazon provides a very simple and quick way to copy an SAP system. You can use the Amazon Management Console or the Amazon CLI to create a new AMI of an existing SAP system. The new AMI contains a complete copy of the operating system and its configuration, software configurations, and all EBS volumes that are attached to the instance. From the new AMI, you can launch exact copies of the original system. For details on how to create an AMI of an existing EC2 instance, see Creating a Custom Windows AMI.

Example:

```
 $ aws ec2 create-image --instance-id i-1234567890abcdef0
--name "My server" --description "An AMI for my server"
```

Note

When you build an instance using an AMI, make sure that you update the hostname and the `C:\Windows\System32\Drivers\etc\hosts` file with the new metadata. These details usually get copied from the source.

File Backup to Amazon S3

You can perform traditional file-based backups from your EBS volumes to Amazon S3. One way to do this is by using the Amazon CLI and trigger this using Amazon Systems Manager Run Command so that you can centrally manage these.

Third-party Options

There are many third-party backup products for Amazon services, including many solutions that have been certified by SAP. For more information, see Amazon SAP Partner Solutions.

Amazon FSx Backup

With Amazon FSx, backups are file-system-consistent, highly durable, and incremental. To ensure file system consistency, Amazon FSx uses the Volume Shadow Copy Service (VSS) in Microsoft Windows. To ensure high durability, Amazon FSx stores backups in Amazon S3. Amazon FSx backups are incremental, which means that only the changes made after your most recent backup are saved.

Amazon FSx automatically takes backups of your file systems once a day. These daily backups are taken during the daily backup window that you established when you created the file system.

If you want to set up a custom backup schedule, you can deploy our reference solution.

Storage

The storage services we use across this guide are:

- Amazon EBS
  - Provides persistent storage for SAP application and database. The EBS volumes can be resized and even the EBS volume type can be changed without disrupting the applications. For more information, see Requesting Modifications to Your EBS Volumes. You will need to extend the filesystem to match the extended volume size using the Windows operating system tools.

- Amazon FSx for Windows File Server
  - Does not need you to explicitly provision storage at all – you simply pay for what you use.
  - Does need regular maintenance, but you can define your own maintenance window as per Amazon FSx Maintenance Windows.
  - The Amazon FSx Service Level Agreement provides for a service credit if your monthly uptime percentage is below our service commitment in any billing cycle.
- Amazon S3
  - Does not need you to explicitly provision storage at all – you simply pay for what you use.
  - You can use Object Lifecycle Management to set rules that define when objects are transitioned or archived to colder storage, such as S3 Standard-IA, S3 Glacier, or S3 Glacier Deep Archive, and when they expire. These actions happen automatically after being set.

Operating System Maintenance

In general, operating system maintenance across large numbers of EC2 instances can be managed by:

- Tools specific to each operating system, such as Microsoft System Center
- Third-party products, such as those available in Amazon Marketplace
- Using Amazon Systems Manager

Patching

You can follow SAP recommended patching processes to update your landscape on Amazon. For operating system patching, with Amazon Systems Manager Patch Manager you can roll out OS patches as per your corporate policies. There are multiple key features like:

- Scheduling based on tags
- Auto-approving patches with lists of approved and rejected patches
- Defining patch baselines

Amazon Systems Manager Patch Manager integrates with IAM, Amazon CloudTrail, and Amazon CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager Operations Work.

If Amazon Systems Manager Patch Manager does not satisfy your requirements, there are third-party products available as well. Some of these products are available in the Amazon Marketplace.

Maintenance Window

Amazon Systems Manager Maintenance Windows lets you define a schedule for when to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, installing software, or applying patches.

Administrator Access

You can access the backend SAP systems for administration purposes using:

- Amazon Systems Manager Session Manager
- Remote Desktop Protocol (RDP)
- Secure Shell (SSH)

High Availability

After your HA cluster is deployed and configured successfully on Amazon, the operation of the HA software still follows the third-party software interface. This can be best understood by following the operational guides from the respective vendors.

It’s also important to have a test environment available (often called a staging or pre-production environment) that has an identical cluster configuration to your production environment. This environment can be used to test any configuration changes to the cluster before deploying the changes to production.

Two key Amazon features that support the cluster software are:

- Amazon FSx for shared storage: See the storage section for maintenance considerations for Amazon FSx. For Multi-AZ deployments, DFS replication is required across multiple filesystems so ensure that you monitor the replication.
- Overlay IP for IP failover
  - Ensure that IAM authorizations are in place to minimize update access to the route table so that only the cluster agent can edit it.

- Ensure that the route table configuration is coupled with your change management process so that any wider environment updates that might affect this feature are captured and can therefore be tested.

Disaster Recovery

Network Fileshare Copy Out-of-Region

To ensure that you have an independent backup of your data in the secondary Region, you should back up your shared filesystem, as this won't be included in any AMIs or individual EBS snapshots.

To back up your Amazon FSx filesystem, you can rely on the included backup feature. However, this backs up to Amazon S3 in-region. To support out-of-region disaster recovery (DR), you will need to perform a file-level backup of your Amazon FSx filesystem in your secondary Region. You can do this by accessing the filesystem via cross-region VPC peering and then running a file-level copy from an Amazon EC2 instance running in the secondary Region to Amazon S3. This action can be automated and scheduled using Amazon Systems Manager Run Command in combination with Amazon CloudWatch Events.

Fail-back Plan

When your primary Region returns to normal operations, you may consider failing back to it. In the event of a disaster that triggered a recovery to another Region, you copy the AMIs and shared filesystems from the secondary Region back to the primary. In other words, you'll reverse what was the regular process before the disaster. It's important to update your change management or change control processes to reflect this.

AMI Copy

When your primary Amazon Region is affected by an availability event, AMIs allow you to quickly recover your SAP NetWeaver application servers in your DR Region. For recovery across Regions, ensure that the latest AMIs are copied to the disaster recovery Region using the AMI copy feature. New AMIs should be created when there are filesystem level changes to the SAP NetWeaver application servers. This can be caused by:

- SAP kernel changes
- Database client software updates
- Operating system patches

To ensure that you reliably create a new AMI when these events happen, add the AMI creation as a step in your change management process. It’s important that if using a mechanism like this that you integrate the out-of-region AMI copy with this process.

If having the lowest possible recovery time objective (RTO) is a priority, consider keeping at least one application server running in the secondary Region to minimize the recovery time.

Amazon Elastic Disaster Recovery

Amazon Elastic Disaster Recovery (Elastic Disaster Recovery) minimizes downtime and data loss with fast, reliable recovery of on-premises and cloud-based applications using affordable storage, minimal compute, and point-in-time recovery.

You can increase IT resilience when you use Amazon Elastic Disaster Recovery to replicate on-premises or cloud-based applications running on supported operating systems. Use the Amazon Management Console to configure replication and launch settings, monitor data replication, and launch instances for drills or recovery.

For more information, see the following resources.

- What is Elastic Disaster Recovery?
- Disaster recovery for SAP workloads on Amazon using Amazon Elastic Disaster Recovery

Compute

EBS volumes are exposed as NVMe block devices on Nitro-based instances. When changing EC2 instance types from a previous generation to a Nitro generation, and if using a Windows Server 2008 R2, or later, Windows AMI, the Amazon NVMe driver is already included as described in the Amazon EBS and NVMe documentation. If you are not using the latest Windows AMIs provided by Amazon, see Installing or Upgrading Amazon NVMe Drivers.

Besides operating system maintenance, there is also maintenance that you can consider for the EC2 instances themselves. This maintenance can be driven by Amazon Systems Manager Automation documents. Some examples are:

- Use the Amazon-StopEC2InstanceWithApproval document to request that one or more IAM users approve the instance stop action. After the approval is received, automation stops the instance.
- Use the Amazon-StopEC2Instance document to automatically stop instances based on a schedule by using Amazon CloudWatch Events or a Maintenance Window task. For example, you

can configure an automation workflow to stop instances every Friday evening, and then restart them every Monday morning.

- Use the Amazon-UpdateCloudFormationStackWithApproval document to update resources that were deployed using an Amazon CloudFormation template. The update applies a new template. You can configure the automation to request approval by one or more IAM users before the update begins.

We also provide an Amazon Solution called Amazon Instance Scheduler that enables you to easily configure custom start and stop schedules for their Amazon EC2 and Amazon Relational Database Service (Amazon RDS) instances.

Cost Optimization

We recommend that you make cost optimization an on-going process. This is an extensive topic with many Amazon services that help with budgeting, cost control, and proactive cost optimization recommendations.

For more details, see SAP on Amazon Pricing and Optimization guide.

Automation

Automation using Infrastructure as Code with Amazon CloudFormation

We recommend following the infrastructure as code principle in automating and maintaining your workloads on Amazon. Amazon CloudFormation provides a common language to describe and provision all the infrastructure resources in your cloud environment in a repeatable and automated manner.

Automation using Documents

Amazon Systems Manager Automation simplifies common maintenance and deployment tasks associated with Amazon EC2 instances and other Amazon resources. Automation enables you to do the following:

- Build Automation workflows to configure and manage instances and Amazon resources.
- Create custom workflows or use pre-defined workflows maintained by Amazon.
- Receive notifications about Automation tasks and workflows by using Amazon CloudWatch Events.

- Monitor Automation progress and execution details by using the Amazon EC2 or the Amazon Systems Manager console.

There are many Amazon-provided documents specific to Windows already available.

Support

To get help from SAP, SAP requires, at the minimum, a business support agreement with Amazon. Amazon Business Support provides resources and technical support for customers running SAP workloads on Amazon. If you have any Amazon-related technical issues, you can open a case with either SAP or Amazon, and it will be routed to the appropriate teams. Amazon also offers Amazon Enterprise Support for customers running mission critical production workloads on Amazon.

Additional Reading

SAP on Amazon Technical Documentation

- SAP on Amazon Technical Documentation
- SAP on Amazon Whitepapers
- SAP NetWeaver on Amazon Quick Start

  This is for SAP NetWeaver deployments on Linux, but is a useful point of comparison if you are looking to automate a Windows-based deployment, or implement a standard Multi-AZ network layout.
- Amazon for SAP Blog
- Making Application Failover Seamless by Failing Over Your Private Virtual IP Across Availability Zones

SAP Documentation

- SAPS Ratings of Amazon Instance types supported for SAP Note 1656099
- 1588667 - SAP on Amazon: Overview of related SAP Notes and Web-Links
- 1656250 - SAP on Amazon: Support prerequisites
- 2539944 - Windows Server / Microsoft SQL Server on AMI
- 1409604 - Virtualization on Windows: Enhanced monitoring

- 2198693 - Key Monitoring Metrics for SAP on Amazon Web Services

Document Revisions

Date	Change
November 2022	Added SAP Note 3143497 to SAP NetWeaver on Windows OSS Notes list
July 2019	Initial publication

Microsoft SQL Server for SAP NetWeaver on Amazon Deployment and Operations Guide

SAP specialists, Amazon Web Services

Last updated: December 2020

This guide provides guidance on how to set up Amazon resources and the Microsoft Windows Server operating system to deploy Microsoft SQL Server for SAP NetWeaver on Amazon EC2 instances.

This guide is for users who are responsible for planning, architecting, and deploying SQL Server on Amazon for SAP NetWeaver based applications. You should have a good understanding of Amazon services, general networking concepts, Windows Server operating systems, and SQL Server administration.

Overview

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the Amazon Web Services Cloud. For the other guides in the series, ranging from overviews to advanced topics, see SAP on Amazon Technical Documentation home page.

This guide provides guidance on how to set up Amazon resources and the Microsoft Windows Server operating system to deploy Microsoft SQL Server for SAP NetWeaver on Amazon EC2 instances.

Instructions in this document are based on recommendations provided by SAP and Microsoft for SQL Server deployment on Windows via the below SAP notes or KB articles:

Table 1 - SAP NetWeaver on Windows OSS Notes

SAP OSS Note	Description
1656099	SAP Applications on Amazon: Supported DB/ OS and Amazon EC2 products
1409608	Virtualization on Windows

SAP OSS Note	Description
1732161	SAP Systems on Windows Server 2012 (R2)
2384179	SAP Systems on Windows Server 2016
2751450	SAP Systems on Windows Server 2019
1564275	Install SAP Systems Using Virtual Host Names on Windows
1772688	SQL Server AlwaysOn and SAP applications

In addition, this document also follows best practices from Amazon, Microsoft, and SAP for SAP NetWeaver deployments on Windows.

This guide is for users who are responsible for planning, architecting, and deploying SQL Server on Amazon for SAP NetWeaver based applications. You should have a good understanding of Amazon services, general networking concepts, Windows Server operating systems, and SQL Server administration.

This document doesn’t provide guidance on how to set up network and security constructs like Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables, ACLs, NAT Gateway, IAM Roles, Amazon Security Groups, and so on. This document focuses on configuring and maintaining compute, storage, and operating system for Microsoft SQL Server for SAP NetWeaver based applications.

Prerequisites

Specialized Knowledge

Before you follow the instructions in this guide, we recommend that you become familiar with the following Amazon services. (If you are new to Amazon, see Getting Started with Amazon.)

- Amazon EC2
- Amazon EBS
- Amazon FSx

- Amazon VPC
- Amazon CloudFormation
- Amazon Systems Manager
- Amazon Simple Storage Service (Amazon S3)
- Amazon Identity and Access Management (IAM)

Technical Requirements

Before you start to deploy Microsoft SQL Server database for SAP applications on Amazon, ensure that you meet the following requirements:

- Windows Server 2008 R2, 2012 R2, or 2016 operating system
- Microsoft SQL Server 2008 R2 or higher database
- Install Amazon SAP Data provider on Amazon EC2 instances after installing SQL Server database
- If you plan to deploy domain installation, you should have a user ID that is a member of domain admins. Otherwise, the domain admin should create groups and user IDs (such as <sapsid>adm, SAPService<SAPSID>, and so on) as required for SAP in advance. See SAP installation guide for more details.
- Amazon Account with permission to create resources.
- Access to SAP installation media for database and application
- Amazon Business Support or Amazon Enterprise Support plan

Planning

Topics

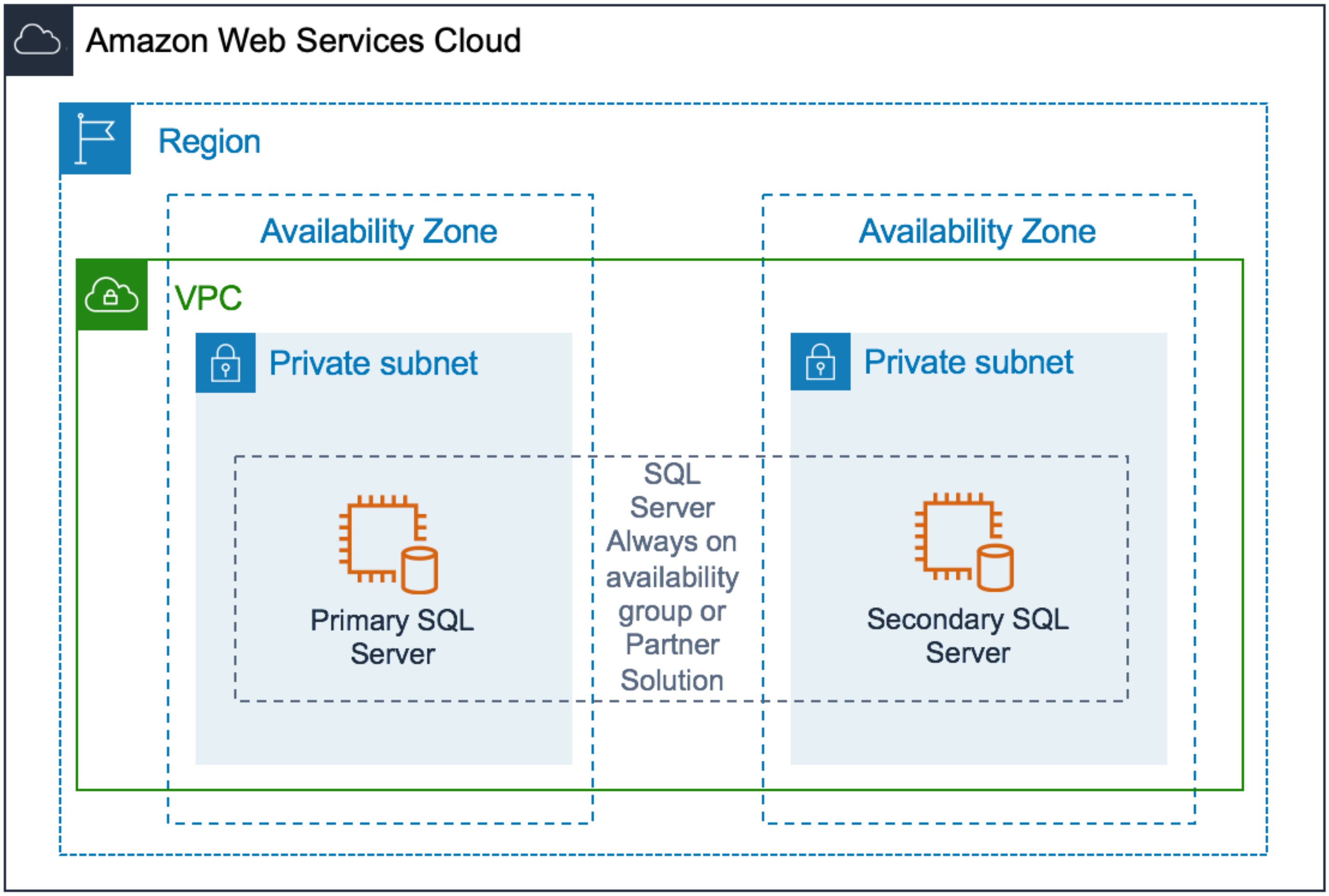
- Storage
- Network
- Business Continuity

Architecture Options

SAP NetWeaver applications based on SQL Server can be installed in three different ways:

- **Standard system or single host installation**: ABAP System Central Services (ASCS)/System Central Services (SCS), Database, and Primary Application Server (PAS) of SAP NetWeaver run in single Amazon EC2 instance. This option is suited for non-critical and non-production workloads.
- **Distributed system**: ASCS/SCS, Database, and PAS of SAP NetWeaver run on separate Amazon EC2 instances. For example, you can choose to run ASCS and PAS on one Amazon EC2 instance and database on another Amazon EC2 instance or other possible combinations. This option is suited for production and non-production workloads.
- **High Availability (HA) system**: For your SAP application to be highly available, you need to protect the single point of failures. Database is one single point of failure in SAP applications. There are two methods you can use to protect SQL Server and make it highly available.
  - Database native solution: SQL Server Always On availability group.
  - Third-party solutions: For example, SIOS Data Keeper, NEC ExpressCluster, Veritas InfoScale.

Regardless of which option you choose to make your SQL Server database highly available, Amazon recommends that you deploy a primary and secondary SQL Server in different Amazon Availability Zones within an Amazon Region. The following diagram provides a high-level architecture for SQL Server high availability on Amazon. This option is suited for business-critical applications.



Deployment Options

Microsoft SQL Server 2008 R2 or later is supported for SAP applications on Amazon. See SAP Note 1656099 - SAP Applications on Amazon: Supported DB/OS and Amazon EC2 products for supported SAP applications and databases on Amazon.

Security

Amazon provides several security capabilities and services to securely run your SAP applications on Amazon platform. In the context of SQL Server for SAP applications, you can use network services and features such as Amazon VPC, Amazon Virtual Private Network, Amazon Direct Connect, and Amazon EC2 security groups, network access controls, route tables, and so on, to restrict the access to your database.

Network Security

Generally, databases for SAP applications do not require direct user access. We recommend that you only allow network traffic to the Amazon EC2 instance running SQL Server from Amazon EC2 instances running SAP application servers (PAS/AAS) and ASCS/SCS.

By default, SQL Server receives communication on TCP port 1433. Depending on your VPC design, you should configure Amazon EC2 security groups, NACLs, and route tables to allow traffic to TCP Port 1433 from SAP application servers (PAS/AAS) and ASCS/SCS.

Encryption

We recommend that you encrypt your data stored in Amazon storage services. See the following documentation for more details:

- Encrypting Data at Rest and in Transit for Amazon FSx
- Protecting S3 objects using encryption
- Amazon EBS Encryption

Sizing

SAP Quick Sizer is generally used to size the SAP environment for new implementations. However, if you are migrating your existing SAP applications based on SQL Server to Amazon, consider using the following additional tools to right-size your SAP environment based on current use.

- **SAP Early Watch Alerts (EWA):** SAP EWA reports are provided by SAP regularly. These reports provide an overview of historical system use. Analyze these reports to see if your existing SAP system is overused or underused. You can use this information to right size your environment.
- **Windows native tools:** Gather and analyze historical use data for CPU/Memory with Performance Monitor/Windows System Resource Manager to right size your environment.
- **Amazon Application Discovery Service:** Amazon Application Discovery Service helps with collecting usage and configuration data about your on-premises servers. You can use this information to analyze and right-size your environment.

Since it is easy to scale up or scale down your Amazon EC2 instances on Amazon, we recommend that you consider the following guidelines when sizing your SAP environment on Amazon.

- Do not add too much capacity to meet future demand.

- Account for the SAP Quick Sizer buffer. SAP Quick Sizer tools provide sizing guidance based on assumptions that for 100% load (as per your inputs to tool) system use will not exceed 65%. Therefore, there is a fair amount of buffer already built into SAP Quick Sizer recommendation. See SAP's Quick Sizer guidance for details.

Operating System

SAP applications based on SQL Server are supported only on Windows operating system. For supported Windows version, see the SAP product availability matrix (PAM) for the SAP application that you plan to deploy on Amazon.

Compute

Amazon provides multiple SAP certified Amazon EC2 instances. See SAP Note 1656099 - SAP Applications on Amazon: Supported DB/OS and Amazon EC2 products for details. Based on results of your sizing exercise, you can deploy your SQL Server on any of the SAP certified Amazon EC2 instances that meets your requirement.

Storage

The following table lists the main directories for SQL Server database.

Table 2 - Main directories for SQL Server database

Usage	Directory	Description

Database data files	`<drive>:\<SAPSID>DATA0`
	`<drive>:\<SAPSID>DATA1`
	.....
	`<drive>:\<SAPSID>DATA<N>`
		Directory for SAP database data files

Database transaction log files	`<drive>:\<SAPSID>log<N>`	Directory for SAP database transaction Log

Usage	Directory	Description
Tempdb data files	`<drive>:\Tempdb`	Directory for temporary database data files
SQL binaries and other data files	`<drive>:\Program Files\Microsoft SQL Server`	Directory for SQL Server program files and master, msdb, and model data files

Amazon Elastic Block Store (Amazon EBS) volumes are designed to be highly available and reliable. Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. Due to this built-in protection, you don't have to configure RAID 1 for volumes containing database transaction log files, tempdb data files, SQL binaries, and other data files.

We also do not recommend RAID 5 for database data files on Amazon due to following reasons.

- Volumes are replicated within Availability Zone by default.
- Parity write operations of RAID 5 consume some of the IOPS available to your volume and will reduce the overall IO available for database operations by 20-30% over RAID 0 configuration.

Network

Ensure that your network constructs are set up to deploy resources related to SAP NetWeaver. If you haven't already set up network components like Amazon VPC, subnets, route tables, and so on, you can use Amazon Quick Start for VPC to easily deploy scalable VPC architecture.

Business Continuity

We recommend that you architect your business-critical applications to be fault tolerant. Depending on your availability requirements, there are different ways in which you can achieve this. This section discusses how you can set up highly available SQL Server for SAP applications.

High Availability

You can configure high availability for SQL Server database on Amazon using Always On availability groups or third-party tools.

SQL Server Always On Availability Groups

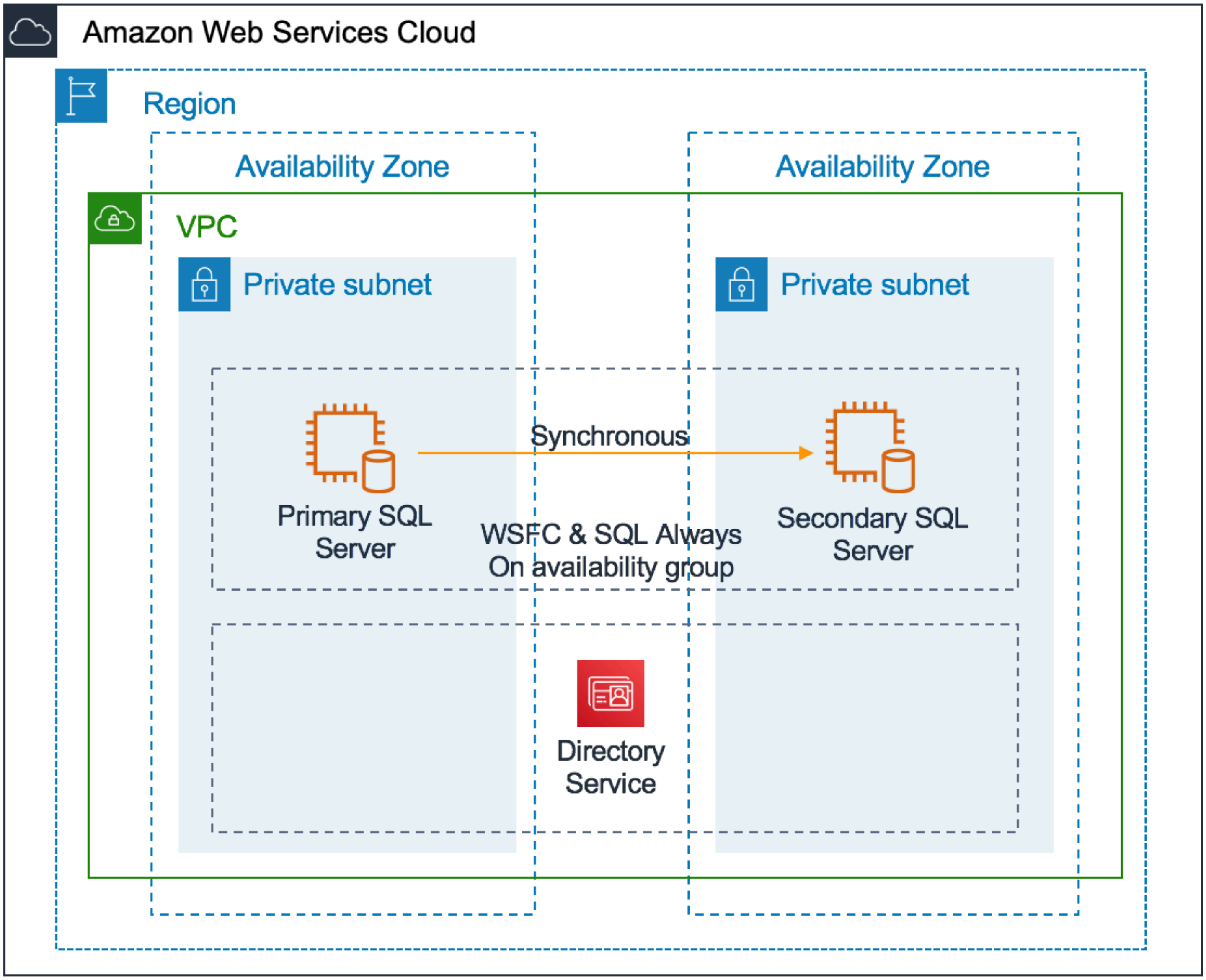
A prerequisite for deploying a SQL Server Always On availability group is Windows Server Failover Clustering (WSFC). SQL Server Always On uses WSFC to increase application availability. WSFC provides infrastructure features that complement the high availability and disaster recovery scenarios supported in the Amazon Cloud. Implementing WSFC cluster on Amazon is very similar to deploying it on-premises provided you meet two key requirements:

- Deploy the cluster nodes inside an Amazon VPC.
- Deploy the cluster nodes in separate subnets that are in different Availability Zones.

See Overview of Always On Availability Groups (SQL Server) for details.

The following figure provides an overview of architecture for SQL Server Always On availability groups on Amazon. This architecture includes following components

- A VPC configured with private subnets across two Availability Zones. This provides the network infrastructure for your SQL Server deployment.
- Amazon Directory Service for Microsoft Active Directory deployed in private subnet. Alternatively, you can also manage your own AD DS deployed on Amazon EC2 instance.
- In a private subnet, Windows Servers configured with WSFC for SQL Server Enterprise edition with SQL Server Always On availability groups.



Third-Party Solutions

You can also use third-party tools like SIOS Data Protection Suite, NEC ExpressCluster, or Veritas InfoScale to provide high-availability for SQL Server. These solutions use WSFC and replicate data from primary to secondary with block level replication of the Amazon EBS volume.

Disaster Recovery

Disaster recovery is about preparing for and recovering from a disaster. Any event that has a negative impact on your business continuity or finances could be termed a disaster. To implement a cost effective Disaster recovery strategy for your SAP applications and databases that meets your business objective you need to consider the following requirements.

Separate DR Strategy from HA Design

First you must evaluate whether a separate DR strategy is required in addition to the HA design offered by Amazon protection.

On Amazon, we recommend that you deploy business critical application in high availability architecture across two Availability Zones in an Amazon Region. Each Availability Zone is designed as an independent failure zone. This means that Availability Zones are physically separated within a typical metropolitan region and are located in lower risk flood plains. Availability Zones include a discrete uninterruptable power supply (UPS) and onsite backup generation facilities, and are each fed via different grids from independent utilities to further reduce single points of failure. The level of protection provided by Availability Zone design is sufficient for most customers and is able to meet their business objectives.

DR in Amazon Regions

If you determine that you need a separate DR strategy, next you must decide if you need a DR plan in a different Amazon Region than your primary Amazon Region or in same Amazon Region as you primary (for example, using third Availability Zone of your primary Amazon Region as DR). Data sovereignty is the primary reason that influences this decision. However, there may be other reasons, such as proximity to users, cost, ease of management, and so on.

DR Architecture

Finally, you must decide on the DR architecture and understand the infrastructure required to implement it. The Recovery Time Objective (RTO) and Recovery Point Objective (RPO) are the primary factors that influence DR architecture. We recommend any of the following three DR architectures:

- **Cold:** This architecture essentially relies on backups. Backups are taken (database – data and log, AMI, Snapshots) on a regular basis and used to rebuild the systems in the target Amazon Region to recover for any disaster. Because this architecture completely depends on backups, the RPO depends on how frequently you take backups, and RTO depends on how large the database is to be recovered.
- **Pilot Light:** This option provides better RTO/RPO over cold option because the SQL server database is synchronously or asynchronously getting replicated to a smaller EC2 instance. If you choose this architecture, you mus resize SQL Server EC2 instances, create application server from AMIs before starting production operations. You can use Amazon CloudFormation to automate these tasks.

- **Hot DR:** SQL Server database for DR EC2 instances are sized the same as production instances which helps to reduce recovery time over Pilot light because you do not need to resize the instances before starting production operations. For application servers, you can choose to replicate the volumes with CloudEndure or other third-party tools, like SIOS, ATAMotion, and so on.

Depending on your specific RTO/RPO, you can implement cold, pilot light, or hot DR architecture. The following table below provides a comparison between cold and pilot light DR for achievable RTO/RPO.

Table 3 - Cold versus Pilot light DR

DR Architecture	Strategy	RTO/RPO
Cold	SQL Server backup/restore	High/High*
Cold	Amazon AMI	Low/High
Cold	Amazon AMI with frequent DB volumes (Data & Log) snapshots	Low/Low*
Pilot Light	Sync Replication (with-in primary region)	Low/Near-Zero
Pilot Light	Async Replication (in different region)	Low/Few Minutes
Hot	Async Replication (in different region)	Few Minutes/Few Minutes

*The exact time it will take to recover database in DR scenario depends on how much you need to catch up to achieve point in time required for Cold architecture. **High** – couple of hours to a day or more. **Low** –less than an hour to couple of hours.

Deployment

Topics

- Windows EC2 Instance Deployment
- SQL Server Deployment
- SQL Server Deployment for High Availability

Windows EC2 Instance Deployment

Deciding the right storage layout is important to ensure you are able to meet required IO. Amazon EBS general purpose volume (gp2) provides 3 IOPS per GB whereas provisioned IOPS (io1) provide a max of 50 IOPS per GB. See EBS features for details. If you decide to separate SQL data, log, and tempdb to different volumes, consider these aspects.

For gp2, with one volume for all (data, log, and tempdb). Create storage config file as below. Replace placeholder `<size>` as per your requirement.

```
[
    {
        "DeviceName": "xvdb",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": "gp2",
            "DeleteOnTermination": true
        }
    }
]
```

For separate volumes, gp2 (data), io1 (log) and io1 (tempdb) create storage configuration file as below. Replace placeholders `<size>` and `<IOPS Required>` with size of the disk and IOPS you need.

```
[
    {
        "DeviceName": "xvdb",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": "gp2",
            "DeleteOnTermination": true
        }
    },
    {
        "DeviceName": "xvdc",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": “io1",
            "Iops": <IOPS Required>,
            "DeleteOnTermination": true
        }
    },
    {
        "DeviceName": "xvdd",
        "Ebs": {
            "VolumeSize": <size>,
            "VolumeType": “io1",
            "Iops": <IOPS Required>,
            "DeleteOnTermination": true
        }
    }
]
```

SQL Server Deployment

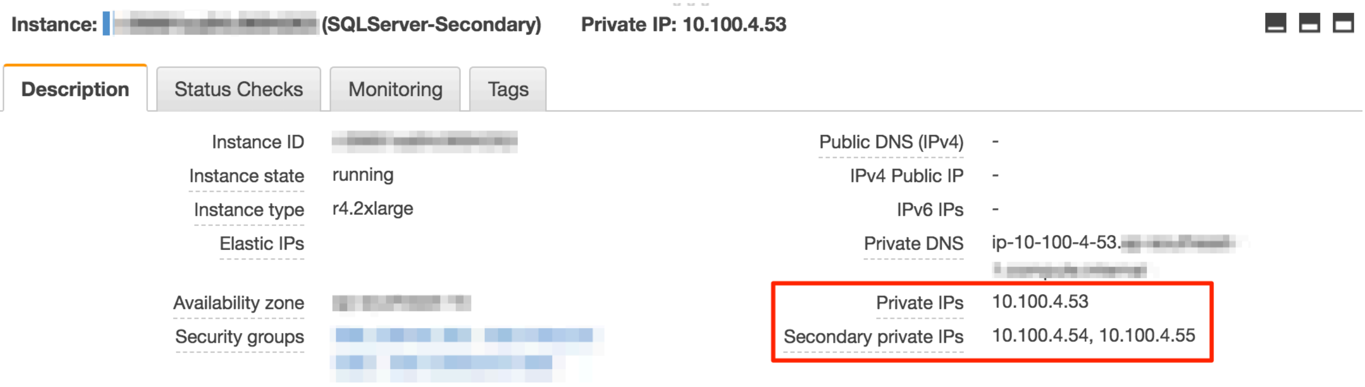
Follow the instructions in the appropriate SAP installation guide for your version of SAP NetWeaver and your combination of operating system and database. See SAP installation guides.

SQL Server Deployment for High Availability

1. Deploy the SAP NetWeaver ASCS instance. For instructions, see the SAP NetWeaver on Amazon Deployment and Operations Guide for Windows .
2. Create two EC2 instances for Microsoft SQL server, one in each Availability Zone. See the Windows EC2 instances deployment section for steps.
3. Assign two secondary IP addresses to each instance from the same subnet CIDR in which they are installed:
    a. Use one address for Windows Server Failover Cluster (WSFC).
    b. Use the second address for the Availability Group listener.

       You can assign IP addresses through the Amazon Management Console, Amazon Command Line Interface (Amazon CLI), or Amazon Tools for Windows PowerShell. For detailed working instructions, see Multiple IP Addresses.

For example, in the screenshot that follows, 10.100.4.53 is the primary private IP address of the EC2 instance. It has been allocated two secondary private addresses: 10.100.4.54 and 10.100.4.55.



Instance: [redacted] (SQLServer-Secondary) Private IP: 10.100.4.53

Instance ID [redacted]
Instance state running
Instance type r4.2xlarge
Elastic IPs
Availability zone [redacted]
Security groups [redacted]

Public DNS (IPv4) -
IPv4 Public IP -
IPv6 IPs -
Private DNS ip-10-100-4-53.[redacted]
Private IPs 10.100.4.53
Secondary private IPs 10.100.4.54, 10.100.4.55

4. Domain join EC2 instances created in Step 1. If you are using Amazon Managed Microsoft AD, see Amazon Directory Service documentation for detailed steps.

5. Log in to the EC2 instance as admin, open PowerShell, and execute the following command to install the Windows Failover Clustering feature.

```
Install-WindowsFeature -Name Failover-Clustering -restart -IncludeAllSubFeature
```

Note

This command may force your EC2 instance to restart. Make sure you execute the command on both EC2 instances.

6. Log in as domain admin into one of the EC2 instance and execute the following command to create the Windows Server Failover Cluster. Make sure to replace the placeholders before executing the command.

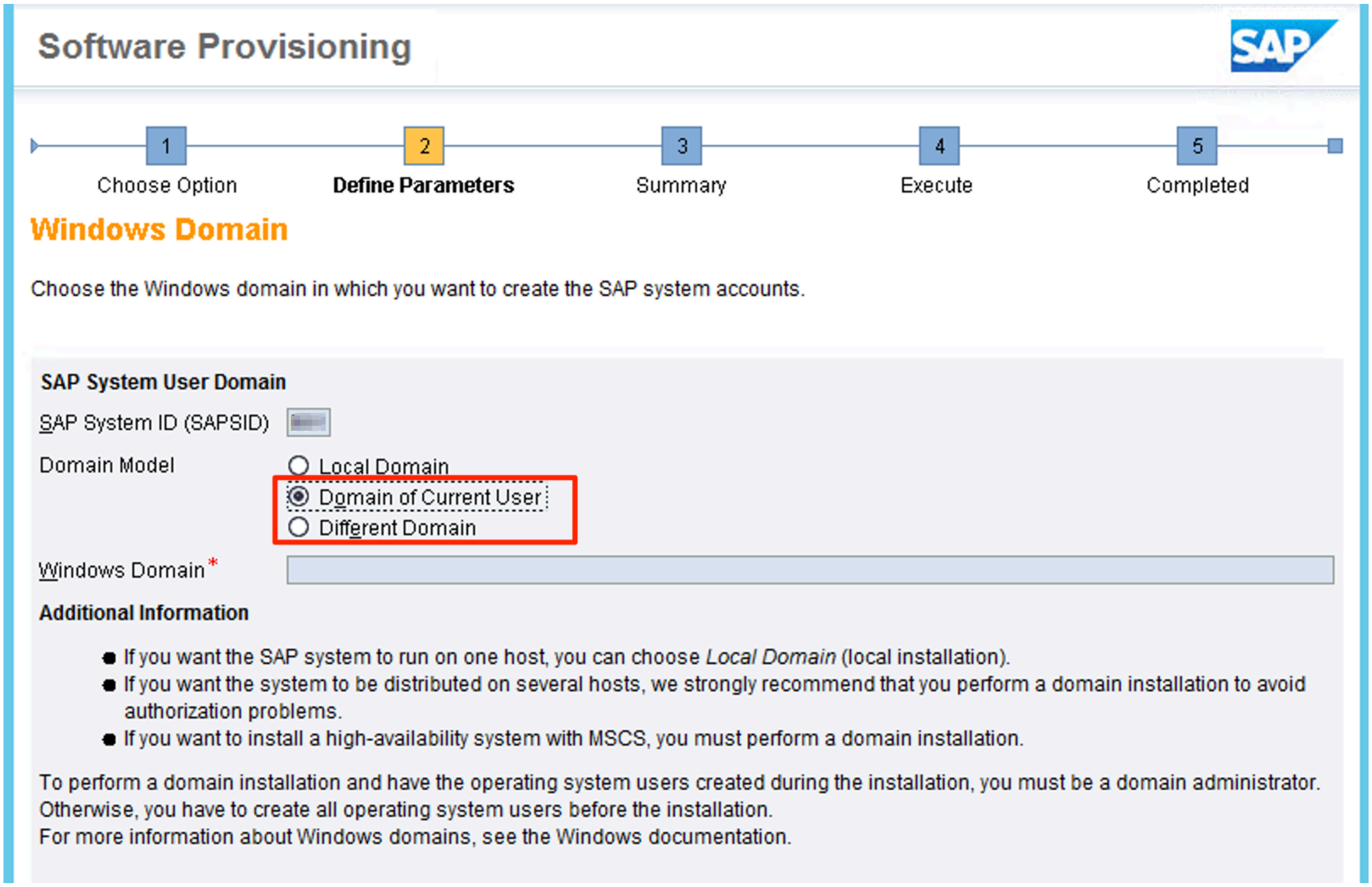
```
New-Cluster -Name <ClusterName> -Node <Node1>,<Node2> -NoStorage
```

For example:

```
New-Cluster -Name SAPSQLCluster -Node primarysql,secondarysql -NoStorage
```

7. Install SQL Server on both EC2 instances. For instructions, see the SAP installation guide.

Install the database instance on the primary node. Follow SAP installation guide to install Database instance on primary node. Make sure you perform domain installations and choose **Domain of Current User** or **Different Domain** as appropriate during parameter selection.



Windows Domain

Choose the Windows domain in which you want to create the SAP system accounts.

SAP System User Domain

SAP System ID (SAPSID)

Domain Model
○ Local Domain
● Domain of Current User
○ Different Domain

Windows Domain*

Additional Information

- If you want the SAP system to run on one host, you can choose *Local Domain* (local installation).
- If you want the system to be distributed on several hosts, we strongly recommend that you perform a domain installation to avoid authorization problems.
- If you want to install a high-availability system with MSCS, you must perform a domain installation.

To perform a domain installation and have the operating system users created during the installation, you must be a domain administrator. Otherwise, you have to create all operating system users before the installation.
For more information about Windows domains, see the Windows documentation.

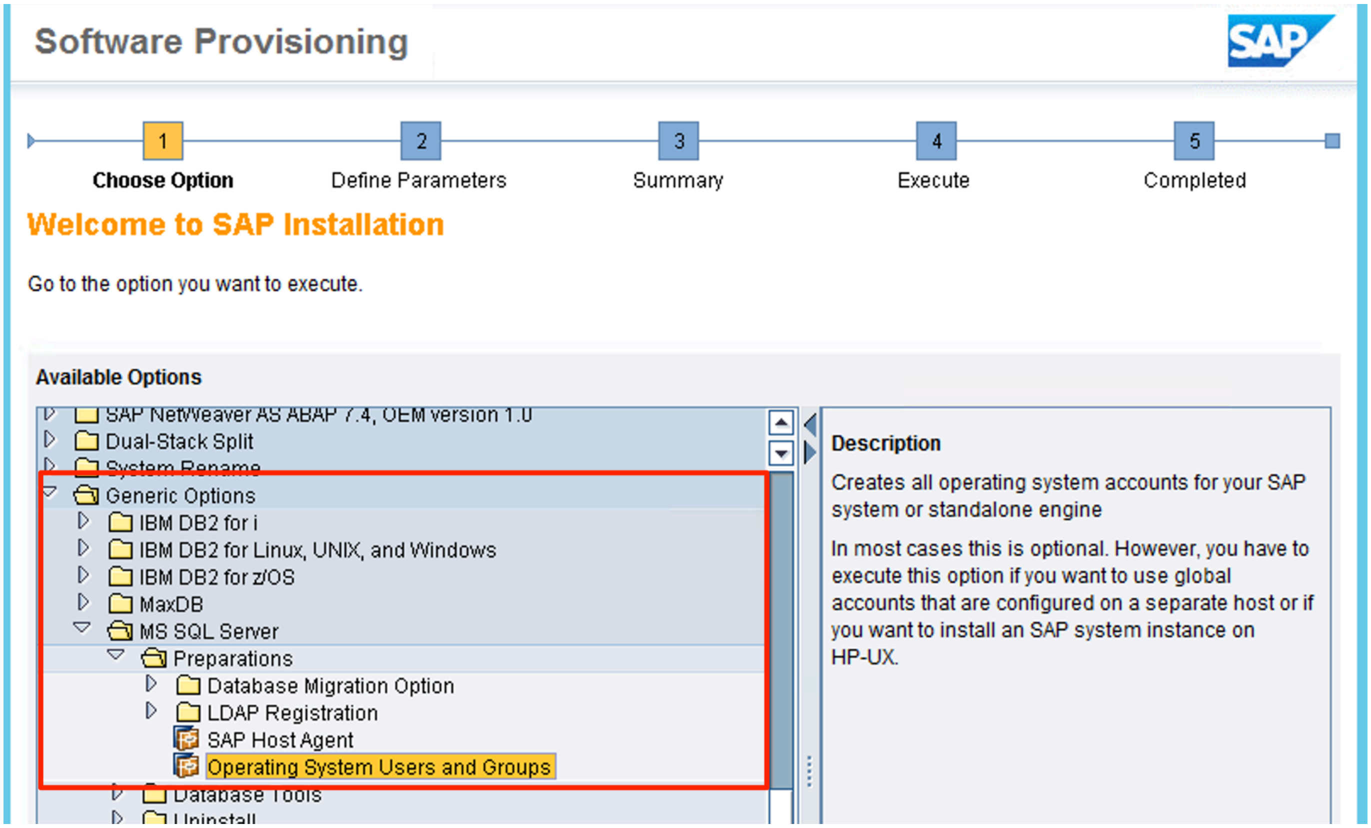
8. Create operating system users on secondary instance.

   a. Start **sapinst**, and in the **Available Options** pane, navigate to **Generic Options > MS SQL Server > Preparations > Operating System Users and Groups**.

   > **Note**
   >
   > The navigation path can vary depending on the version of SWPM you are using.

   b. Create users and groups for this instance, as appropriate.

> **Note**
>
> You do not need to create users on the primary instance because the database instance was installed on the primary node operating system.



Software Provisioning

Welcome to SAP Installation

Go to the option you want to execute.

Available Options

Description

Creates all operating system accounts for your SAP system or standalone engine

In most cases this is optional. However, you have to execute this option if you want to use global accounts that are configured on a separate host or if you want to install an SAP system instance on HP-UX.

9. Install SAP Host agent on secondary instance with SWPM.

10. Create a SQL Server Always On availability group. See the [Microsoft documentation]() for SQL Always On availability group installation instructions.

11. Adjust the SAP profile files for parameters per the following example. Make sure to replace the `<availabilitygroup listener>` placeholder with appropriate the value for your setup. For details, refer to [SAP Note 1772688 - SQL Server Always On and SAP applications]().

```
dbs/mss/server = <availabilitygroup listener>;MultiSubnetFailover=yes
```

    SAPDBHOST = `<availabilitygroup listener>`

12. Perform failover and failback of SQL Server to validate it is working correctly.

13. Continue with installation of primary application server (PAS) and additional application server (AAS) following the instructions in SAP installation guides.

Operations

This section provides information on Amazon services that help you with day-to-day operations of your SQL Server database for SAP applications.

Topics

- Monitoring
- Backup and Recovery
- Storage
- Operating System Maintenance
- Business Continuity
- Support
- Cost Optimization

Monitoring

Amazon provides multiple services to monitor and manage your infrastructure and applications on Amazon. You can use services like Amazon CloudWatch and Amazon CloudTrail to monitor your underlying infrastructure and APIs, respectively.

CloudWatch provides ready-to-use key performance indicators (KPIs) that you can use to monitor both CPU and disk utilization.

You can also create custom metrics for monitoring SQL server.

With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your Amazon infrastructure. Amazon CloudTrail is enabled on all Amazon accounts and records your account activity upon account creation. You can view and download the last 90 days of your account activity for create, modify, and delete operations of supported services without the need to manually set up CloudTrail.

Backup and Recovery

You need to regularly back up your operating system and database to recover them in case of any failure. Amazon provides various services and tools that you can use to back up your SQL Server database of SAP applications.

Amazon Machine Images (AMIs)

You can use the Amazon Management Console or the Amazon CLI to create a new AMI of your existing SAP system. This AMI can be used to recover your existing SAP system or to create a clone.

The Amazon CLI `create image` command creates a new AMI based on an existing Amazon EC2 instance. The new AMI contains a complete copy of the operating system and its configuration, software configurations, and optionally all Amazon EBS volumes that are attached to the instance. For details on how to create an AMI of an existing Amazon EC2 instance, see Creating an Amazon EBS Backed Windows AMI. AMI creation and lifecycle can be centrally managed in Amazon Backup Amazon Backup.

Amazon EBS Snapshots

You can back up your Amazon EBS volumes to Amazon Simple Storage Service by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved.

Snapshots are suited to back up SAP file systems like `/usr/sap/` , `/sapmnt/`. If you decide to take snapshots of your EBS volumes containing data and log files, make sure to use Volume Shadow Copy Service or shut down your database before a snapshot is triggered for consistency. You can use Amazon Backup to create backups using VSS functionalities.

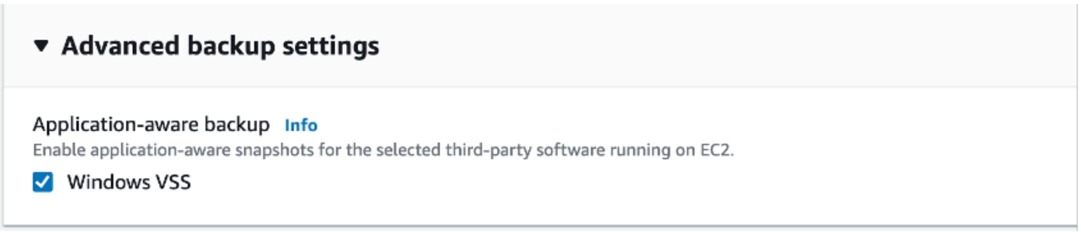
The following command creates a snapshot of volume (with example `volume id vol-1234567890abcdef0`). You can use this command in Amazon CLI to create your own volume snapshot.

```
aws ec2 create-snapshot --volume-id <vol-1234567890abcdef0> --description "This is my
 volume snapshot."
```

Database Backups

For SQL Server database backup, you can use one of the following methods:

- **SQL native tools to take backup on disk:** Backup requires high throughput compared to IOPS. We recommend using Throughput Optimized HDD (st1) which provides maximum throughput of 500 MB/s per volume. Once the backup completes on disk, you can use scripts to move it to an Amazon S3 bucket.
- **Amazon Backup** for application-consistent backups via Microsoft's Volume Shadow Copy Services (VSS). Ensure that the flag in the advanced backup settings is enabled:



▼ Advanced backup settings

Application-aware backup

Enable application-aware snapshots for the selected third-party software running on EC2.

☑ Windows VSS

- **Third-party backint tools:** Partners like Commvault, Veritas, and so on use SAP backint interface and store backups directly in Amazon S3 buckets.

Storage

The following list includes Amazon storage services included in this guide.

Amazon EBS

Amazon EBS provides persistent storage for SAP application and database. You can increase EBS volume size or change the type of volume (for example, gp2 to io1) without requiring downtime. For more information, see Modifying Amazon EBS volume.

Amazon FSx for Windows File Server

Amazon FSx does not require you to explicitly provision storage at all – you simply pay for what you use.

Amazon FSx requires regular maintenance for patching, but you can define the maintenance windows as per your business requirements. For details, see FSx Maintenance Windows.

Amazon S3

Amazon S3 does not require you to explicitly provision storage at all – you simply pay for what you use.

Operating System Maintenance

In general, operating system maintenance across large estates of EC2 instances can be managed by:

- Tools specific to each operating system, such as Microsoft System Center 2019
- Third-party products, such as those available on Amazon Marketplace
- Amazon Systems Manager

Amazon Systems Manager can help with the following key operating system maintenance tasks.

Patching

You can follow SAP recommended patching processes to update your landscape on Amazon. For operating system patching, use Amazon Systems Manager Patch Manager to roll out OS patches as per your corporate policies. Patch manager includes features like:

- Scheduling based on tags
- Auto-approving patches with lists of approved and rejected patches
- Defining patch baselines

Amazon Systems Manager Patch Manager integrates with Amazon Identity and Access Management (IAM), Amazon CloudTrail, and Amazon CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager Operations Work. If Amazon Systems Manager Patch Manager does not fulfil your requirements, there are third-party products available on the Amazon Marketplace.

Maintenance Window

Amazon Systems Manager Maintenance Windows let you define a schedule for when to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, or installing software or patches.

Automation using Documents

Amazon Systems Manager Automation simplifies common maintenance and deployment tasks of Amazon EC2 instances and other Amazon resources. Automation enables you to do the following:

- Build Automation workflows to configure and manage instances and Amazon resources.
- Create custom workflows or use pre-defined workflows maintained by Amazon.
- Receive notifications about Automation tasks and workflows by using Amazon CloudWatch Events.
- Monitor Automation progress and execution details by using the Amazon EC2 or the Amazon Systems Manager console.

There are many Amazon provided documents specific to Windows already available.

Business Continuity

Amazon recommends that you periodically schedule business continuity process validations by executing disaster recovery (DR) tests. This planned activity will help to flush out any potential unknowns and help the organization to deal with any real disaster in a streamlined manner. Depending on your disaster recovery architecture, business continuity may include:

- Backup/recovery of database from AmazonS3
- Creation of systems from AMI and point-in-time recovery via snapshots
- Changing the EC2 instance size of pilot light system
- Validation of integration (AD/DNS, email, third party, and so on.)

Support

SAP requires customers to have a minimum of an Amazon Business Support plan with Amazon. This ensures that any critical issues raised with SAP are also handled by Amazon on priority. Amazon Business Support provides less than one hour response time for production down scenarios. For a response time of less than 15 minute for business critical systems along with other benefits, you can choose Amazon Enterprise Support.

For any SAP application issues, Amazon suggests that you raise an incident with SAP via the SAP Support portal. After the first level of investigation, SAP can redirect the incident to Amazon Support if the issue is infrastructure-related. However, if you choose to raise support issues for SAP applications with Amazon Support, we cannot redirect the tickets to SAP. For any infrastructure-related issues, you can raise the issue directly with Amazon Support.

Cost Optimization

Resources (CPU, Memory, additional application servers, system copies for different tests/ validations, and so on) require SAP landscape changes over time. Amazon recommends that you monitor system utilization and the need for existing systems on a regular basis and take actions to reduce cost. In case of a database like SQL Server, the only opportunity to right-size the database server is by scaling up/down or shutting it down, if not required. Here are few suggestions that you can consider for cost optimization:

- Consider Reserved instances over On-Demand instances if the requirement is to run your instances 24x7 365 days per year. Reserved instances provide up to a 75% discount over On-Demand instances. See EC2 pricing for details.
- Consider running occasionally required systems like training, sandbox, and so on, on-demand for the duration required.
- Monitor CPU and memory utilization over time for other non-production systems like Dev/QA and right-size them when possible.

FAQ

Q. Can I use Amazon RDS for SQL Server as a database to deploy SAP NetWeaver based applications?

A. No, Amazon RDS for SQL Server is not certified by SAP for SAP NetWeaver based applications. However, it is certified to be used as database for SAP Business Objects BI (BObj BI)

Q. Can I purchase and use a Microsoft SQL Server license from Amazon, such as Microsoft SQL Server 2019 Enterprise on Windows Server 2022, Amazon Machine Image (AMI), to host my SAP NetWeaver based workloads, and other SAP workloads?

A. Yes, Amazon provides a variety of options for Microsoft SQL Server license-included AMIs, as a pre-installed package with different combinations of Microsoft Windows Server and Microsoft SQL Server versions and editions available. For more information, see Licensing options and Find a SQL Server license-included AMI.

There are some differences in how SAP manages technical support, when the support ticket is raised with SAP support, and if the issue raised is found to be with Microsoft SQL Server, when those licenses are from Amazon. In that situation, you need to raise a separate ticket with Support for SQL Server technical support, following the terms of your Support plan.

Document Revisions

Date	Change
December 2020	Minor updates to text in Backup & Recovery section
July 2019	Initial publication

SAP NetWeaver on Amazon: high availability configuration for Netweaver (ASCS)

This topic applies to SAP with high availability and update services operating system for SAP NetWeaver applications on Amazon cloud. It covers the instructions for configuration of a pacemaker cluster for the ABAP SAP Central Service (ASCS) and the Enqueue Replication Server (ERS) when deployed on Amazon EC2 instances in two different Availability Zones within an Amazon Region.

Topics

- SAP NetWeaver on Amazon: high availability configuration for SUSE Linux Enterprise Server (SLES) for SAP applications
- SAP NetWeaver on Amazon: high availability configuration for Red Hat Enterprise Linux (RHEL) for SAP applications

SAP NetWeaver on Amazon: high availability configuration for SUSE Linux Enterprise Server (SLES) for SAP applications

This topic applies to SUSE Linux Enterprise Server (SLES) operating system for SAP NetWeaver applications on Amazon cloud. It covers the instructions for configuration of a pacemaker cluster for the ABAP SAP Central Service (ASCS) and the Enqueue Replication Server (ERS) when deployed on Amazon EC2 instances in two different Availability Zones within an Amazon Region.

This topic covers instructions for the following configuration options.

Topics

- Planning
- Prerequisites
- SAP ASCS and Cluster Setup
- Operations
- Testing

Planning

This section covers the following topics.

Topics

- Setup Overview
- Deployment Guidance
- Concepts
- Parameter Reference
- Architecture diagrams

Setup Overview

You must meet the following prerequisites before commencing setup.

Topics

- Deployed Cluster Infrastructure
- Supported Operating System
- SAP and SUSE references
- Required Access for Setup
- Reliability Requirements Defined

Deployed Cluster Infrastructure

Ensure that your Amazon networking requirements and Amazon EC2 instances where SAP workloads are installed, are correctly configured for SAP. For more information, see SAP NetWeaver Environment Setup for Linux on Amazon.

See the following ASCS cluster specific requirements.

- Two cluster nodes created in private subnets in separate Availability Zones within the same Amazon VPC and Amazon Region
- Access to the route table(s) that are associated with the chosen subnets

  For more information, see Amazon – Overlay IP.

- Amazon EC2 instances must have connectivity to the Amazon EC2 endpoint via either internet or an Amazon VPC endpoint.

Supported Operating System

Protecting the ABAP SAP Central Services (ASCS) with a pacemaker cluster requires packages from SUSE, including targeted cluster resource agents for SAP and Amazon that may not be available in standard repositories.

For deploying SAP applications on SUSE, SAP and SUSE recommend using SUSE Linux Enterprise Server for SAP applications (SLES for SAP). SLES for SAP provides additional benefits, including Extended Service Pack Overlap Support (ESPOS), configuration and tuning packages for SAP applications, and High Availability Extensions (HAE). For more details, see SUSE website at SUSE Linux Enterprise Server for SAP Applications.

SLES for SAP is available at Amazon Marketplace with an hourly or annual subscription. You can also use the bring your own subscription (BYOS) model.

SAP and SUSE references

In addition to this guide, see the following references for more details.

- SUSE documentation – SAP S/4 HANA - Enqueue Replication 2 High Availability Cluster With Simple Mount
- SUSE documentation – SAP S/4 HANA - Enqueue Replication 2 High Availability Cluster
- SAP Note: 1656099 - SAP Applications on Amazon: Supported DB/OS and Amazon EC2 products
- SAP Note: 1984787 - SUSE Linux Enterprise Server 12: Installation Notes
- SAP Note: 2578899 - SUSE Linux Enterprise Server 15: Installation Notes
- SAP Note: 1275776 - Linux: Preparing SLES for SAP environments

You must have SAP portal access for reading all SAP Notes.

Required Access for Setup

The following access is required for setting up the cluster.

- An IAM user with the following privileges.
  - modify Amazon VPC route tables

  - modify Amazon EC2 instance properties
  - create IAM policies and roles
  - create Amazon EFS file systems
- Root access to the operating system of both cluster nodes
- SAP administrative user access – `<sid>adm`

  In case of a new install, this user is created by the install process.

Reliability Requirements Defined

The SAP Lens of the Well-Architected framework, in particular the Reliability pillar, can be used to understand the reliability requirements for your SAP workload.

The ASCS is a single point of failure in a highly available SAP architecture. The impact of an outage of this component must be evaluated against factors, such as, recovery point objective (RPO), recovery time objective (RTO), cost and operation complexity. For more information, see Reliability in SAP Lens - Amazon Well-Architected Framework.

Deployment Guidance

The following section details the documentation and deployment guidance from SUSE.

Topics

- Deployment Patterns
- Automated Deployment
- Pacemaker - simple-mount and classic architecture

Deployment Patterns

The following table outlines the supported SAP deployment types and their corresponding Amazon configuration patterns for high availability clustering.

SAP Deployment Type	Support Status	Amazon Configuration Patterns	Notes
SAP NetWeaver ASCS/ ERS (ENSA1)	Amazon Documented & Supported	SAPNetweaver-Classic, SAPNetweaver-Simple-mount	
SAP NetWeaver ASCS/ ERS (ENSA2)	Amazon Documented & Supported	SAPNetweaver-Classic, SAPNetweaver-Simple-mount	
SAP S/4HANA ASCS/ERS	Amazon Documented & Supported	SAPNetweaver-Classic, SAPNetweaver-Simple-mount	S/4HANA only supports ERS2
SAP SCS (Java)	Vendor Documented & Supported		Follows SAP Documentation

Automated Deployment

You can set up a cluster manually using the instructions provided here. You can also automate parts of this process to ensure consistency and repeatability.

Use Amazon Launch Wizard for SAP for automated deployments of SAP NetWeaver, SAP S/4 HANA, SAP B/4HANA, and Solution Manager. Launch Wizard uses Amazon CloudFormation scripts to quickly provision the resources needed to deploy SAP NetWeaver and S/4 HANA. The automation performs SAP enqueue replication and pacemaker setup so that only validation and testing are required. For more information, see Amazon Launch Wizard for SAP.

To ensure that the behavior and operation of your cluster is well understood regardless of how your system is set up, we recommend a thorough test cycle. See Testing for more details.

Pacemaker - simple-mount and classic architecture

This guide covers two architectures for SAP cluster solutions on SLES for SAP – simple-mount and classic (previous standard). Simple-mount was certified as the SLES for SAP Applications cluster solution in late 2021. It is now the recommended architecture for both ENSA1 and ENSA2 deployments running on SLES for SAP 15 and above. For more details, see SUSE blog Simple Mount Structure for SAP Application Platform.

If you are configuring a new SAP installation, we recommend the simple-mount architecture. If you already have the classic architecture, and wish to migrate to the simple-mount architecture, see Switching architecture to simple-mount.

The following are the differences between the classic and simple-mount architectures.

- Removing file system resources from cluster – a file system is required but it is not mounted and unmounted by the cluster. The executable directory for the ASCS and ERS can be permanently mounted on both nodes.
- Addition of SAPStartSrv – SAPStartSrv controls the matching SAPStartSrv framework process.
- Sapping and sappong services – these services manage the start of SAPStartSrv services with sapinit.

See the Architecture diagrams for more details.

Switching architecture to simple-mount

Follow along these steps if you want to switch an existing cluster with classic architecture to use the recommended configuration of simple-mount architecture.

These steps must be performed in an outage window, allowing stop/start of services and basic testing.

1. Put the cluster in maintenance mode. See the section called "Maintenance mode"
2. Stop SAP services, including application servers connected to the cluster as well as ASCS and ERS.
3. Install any missing operating system packages. See the section called "Install Missing Operating System Packages".

   It might be necessary to install `sapstartsrv-resource-agents`. However, all operating system prerequisites must be checked and updated to ensure that versions are compatible.

4. Add entries for ASCS and ERS mount point on both nodes (if not already added). See the section called “Update /etc/fstab”
5. Enable `sapping/sappong` services. See the section called “Enable sapping and sappong Services (Simple-Mount Only)”
6. Align and disable `systemd` services. See the section called “Ensure ASCS and ERS SAP Services can run on either node (systemd)”
7. Backup the configuration with the following command.

   ```
   # crm config show >> /tmp/classic_ha_setup.txt
   ```

   See the section called “Prepare for Resource Creation”
8. *Optional* – delete the configuration. You can edit in place but we recommend starting with a blank configuration. This ensures that latest timeout and priority parameters are in place. See the section called “Reset Configuration – Optional”

   ```
   # crm config erase
   # crm config show
   ```
9. Configure cluster resources again.

10.Check the cluster and perform some tests.

11Resume standard operations by starting any additional services, including application servers.

Concepts

This section covers Amazon, SAP, and SUSE concepts.

Topics

- SAP – ABAP SAP Central Services (ASCS)
- SAP – Enqueue Replication Server (ERS)
- Amazon – Availability Zones
- Amazon – Overlay IP
- Amazon – Shared VPC
- Pacemaker - STONITH fencing agent

SAP – ABAP SAP Central Services (ASCS)

The ABAP SAP Central Services (ASCS) is an SAP instance consisting of the following two services. It is considered a single point of failure (SPOF) in a resilient SAP architecture.

- **Message server** – Responsible for application load distribution (GUI and RFC), communication between application servers, and centralised configuration information for web dispatchers and application servers.
- **Enqueue server (standalone)** – Maintains a lock table in main memory (shared memory). Unlike a database lock, an enqueue lock can exist across multiple logical units of work (LUW), and is set by a SAP Dialog work process. The lock mechanism prevents two transactions from changing the same data in the database simultaneously.

> **Note**
>
> With ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. It replaces the previous version (ENSA1) but can be configured for the previous versions. See SAP Note 2630416 - Support for Standalone Enqueue Server 2 (SAP portal access required) for more information.
> This document includes modifications to align with the correct ENSA version.

SAP – Enqueue Replication Server (ERS)

The Enqueue Replication Server (ERS) is an SAP instance containing a replica of the lock table (replication table).

In a resilient setup, if the standalone enqueue server (EN/ENQ) fails, it can be restarted either by restart parameters or by high availability software, such as Pacemaker. The enqueue server retrieves the replication table remotely or by failing over to the host where the ERS is running.

Amazon – Availability Zones

Availability Zone is one or more discreet data centers with redundant power, networking, and connectivity in an Amazon Region. For more information, see Regions and Availability Zones.

For mission critical deployments of SAP on Amazon where the goal is to minimise the recovery time objective (RTO), we suggest distributing single points of failure across Availability Zones.

Compared with single instance or single Availability Zone deployments, this increases resilience and isolation against a broad range of failure scenarios and issues, including natural disasters.

Each Availability Zone is physically separated by a meaningful distance (many kilometers) from another Availability Zone. All Availability Zones in an Amazon Region are interconnected with high-bandwidth, low-latency network, over fully redundant, dedicated metro fiber. This enables synchronous replication. All traffic between Availability Zones is encrypted.

Amazon – Overlay IP

Overlay IP enables a connection to the application, regardless of which Availability Zone (and subnet) contains the active primary node.

When deploying an Amazon EC2 instance in Amazon, IP addresses are allocated from the CIDR range of the allocated subnet. The subnet cannot span across multiple Availability Zones, and therefore the subnet IP addresses may be unavailable after faults, including network connectivity or hardware issues which require a failover to the replication target in a different Availability Zone.

To address this, we suggest that you configure an overlay IP, and use this in the connection parameters for the application. This IP address is a non-overlapping RFC1918 private IP address from outside of VPC CIDR block and is configured as an entry in the route table or tables. The route directs the connection to the active node and is updated during a failover by the cluster software.

You can select any one of the following RFC1918 private IP addresses for your overlay IP address.

- 10.0.0.0 – 10.255.255.255 (10/8 prefix)
- 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

If, for example, you use the 10/8 prefix in your SAP VPC, selecting a 172 or a 192 IP address may help to differentiate the overlay IP. Consider the use of an IP Address Management (IPAM) tool such as Amazon VPC IP Address Manager to plan, track, and monitor IP addresses for your Amazon workloads. For more information, see What is IPAM?

The overlay IP agent in the cluster can also be configured to update multiple route tables which contain the Overlay IP entry if your subnet association or connectivity requires it.

Access to overlay IP

The overlay IP is outside of the range of the VPC, and therefore cannot be reached from locations that are not associated with the route table, including on-premises and other VPCs.

Use Amazon Transit Gateway as a central hub to facilitate the network connection to an overlay IP address from multiple locations, including Amazon VPCs, other Amazon Regions, and on-premises using Amazon Direct Connect or Amazon Client VPN.

If you do not have Amazon Transit Gateway set up as a network transit hub or if it is not available in your preferred Amazon Region, you can use a Network Load Balancer to enable network access to an overlay IP.

For more information, see SAP on Amazon High Availability with Overlay IP Address Routing.

Amazon – Shared VPC

An enterprise landing zone setup or security requirements may require the use of a separate cluster account to restrict the route table access required for the Overlay IP to an isolated account. For more information, see Share your VPC with other accounts.

Evaluate the operational impact against your security posture before setting up shared VPC. To set up, see Shared VPC – optional.

Pacemaker - STONITH fencing agent

In a two-node cluster setup for a primary resource and its replication pair, it is important that there is only one node in the primary role with the ability to modify your data. In the event of a failure scenario where a node is unresponsive or incommunicable, ensuring data consistency requires that the faulty node is isolated by powering it down before the cluster commences other actions, such as promoting a new primary. This arbitration is the role of the fencing agent.

Since a two-node cluster introduces the possibility of a fence race in which a dual shoot out can occur with communication failures resulting in both nodes simultaneously claiming, "I can’t see you, so I am going to power you off". The fencing agent is designed to minimise this risk by providing an external witness.

SLES supports several fencing agents, including the one recommended for use with Amazon EC2 Instances (external/ec2). This resource uses API commands to check its own instance status - "Is my instance state anything other than running?" before proceeding to power off its pair. If it is already in a stopping or stopped state it will admit defeat and leave the surviving node untouched.

Parameter Reference

The cluster setup relies on the following parameters. Gather this information prior to configuring Pacemaker to ensure a smooth setup process.

Topics

- Global Amazon parameters
- Amazon EC2 instance parameters
- SAP Instance Parameters
- Pacemaker Parameters

Global Amazon parameters

Name	Parameter	Example
Amazon account ID	`<account_id>`	`123456789100`
Amazon Region	`<region_id>`	`us-east-1`

- Amazon account – For more details, see Your Amazon account ID and its alias.
- Amazon Region – For more details, see Describe your Regions.

Amazon EC2 instance parameters

Name	Parameter	Host 1	Host 2
Amazon EC2 instance ID	`<instance_id>`	`i-xxxxinstidforhost1`	`i-xxxxinstidforhost2`
Hostname	`<hostname>`	`slxhost01`	`slxhost02`
Host IP	`<host_ip>`	`10.1.10.1`	`10.1.20.1`
Host additional IP	`<host_additional_ip>`	`10.1.10.2`	`10.1.20.2`

Name	Parameter	Host 1	Host 2
Configured subnet	`<subnet_id>`	`subnet-xxxxxxxxxxsubnet1`	`subnet-xxxxxxxxxxsubnet2`
Associated VPC Route Table(s)	`<routetable_id>`	`rtb-xxxxxroutetable1 [,rtb-xxxxxroutetable2]`	
Sapmnt NFS ID or CNAME	`<sapmnt_nfs_id>`	`fs-xxxxxxxxxxxxxefs1`	

- **Hostname** – Hostnames must comply with SAP requirements outlined in SAP Note 611361 - Hostnames of SAP ABAP Platform servers (requires SAP portal access).

  Run the following command on your instances to retrieve the hostname.

  ```
  # hostname
  ```

- **Amazon EC2 instance ID** – run the following command (IMDSv2 compatible) on your instances to retrieve instance metadata.

  ```
  # /usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $(curl --
  noproxy '*' -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
  metadata-token-ttl-seconds: 21600")" http://169.254.169.254/latest/meta-data/
  instance-id
  ```

  For more details, see Retrieve instance metadata and Instance identity documents.

- **Amazon EC2 subnet ID** – run the following command to retrieve the subnet ID for each of your instances.

  ```
  # INSTANCE_ID=i-xxxxinstidforhost1
  # aws ec2 describe-instances --instance-ids $INSTANCE_ID --query
   'Reservations[0].Instances[0].SubnetId' --output text
  ```

  For more details, see describe-instances and VPC subnets.

- **Route table(s) for subnets** – run the following Amazon CLI commands to retrieve the route table(s) associated with both cluster node subnets.

```
# SUBNET_ID_1=subnet-xxxxxxxxxxsubnet1
# SUBNET_ID_2=subnet-xxxxxxxxxxsubnet2
# aws ec2 describe-route-tables --filters "Name=association.subnet-id,Values=
$SUBNET_ID_1,$SUBNET_ID_2" --query 'RouteTables[].RouteTableId' --output text
```

  If both cluster nodes are in subnets associated with the same route table, only one route table ID will be returned. If they are associated with different route tables, both route table IDs will be returned.

  For more details, see describe-route-tables and Route tables.

SAP Instance Parameters

Name	Parameter	Example
SID	<SID> or <sid>	SLX
ASCS Alias	<ascs_virt_hostname>	slxascs
ASCS System Number	<ascs_sys_nr>	00
ASCS Overlay IP	<ascs_overlayip>	172.16.1.50
ASCS NFS Mount Point	<ascs_nfs_mount_point>	/SLX_ASCS00
ERS Alias	<ers_virt_hostname>	slxers
ERS System Number	<ers_sys_nr>	10
ERS Overlay IP	<ers_overlayip>	172.16.1.51
ERS NFS Mount Point	<ers_nfs_mount_point>	/SLX_ERS10
ENSA Type	<ensa_type>	ENSA2

Pacemaker Parameters

Name	Parameter	Example
Cluster user	`cluster_user`	`hacluster`
Cluster password	`cluster_password`	
Cluster tag	`cluster_tag`	`pacemaker`
Amazon CLI cluster profile	`aws_cli_cluster_profile`	`cluster`
Cluster connector	`cluster_connector`	`sap-suse-cluster-connector`

Architecture diagrams

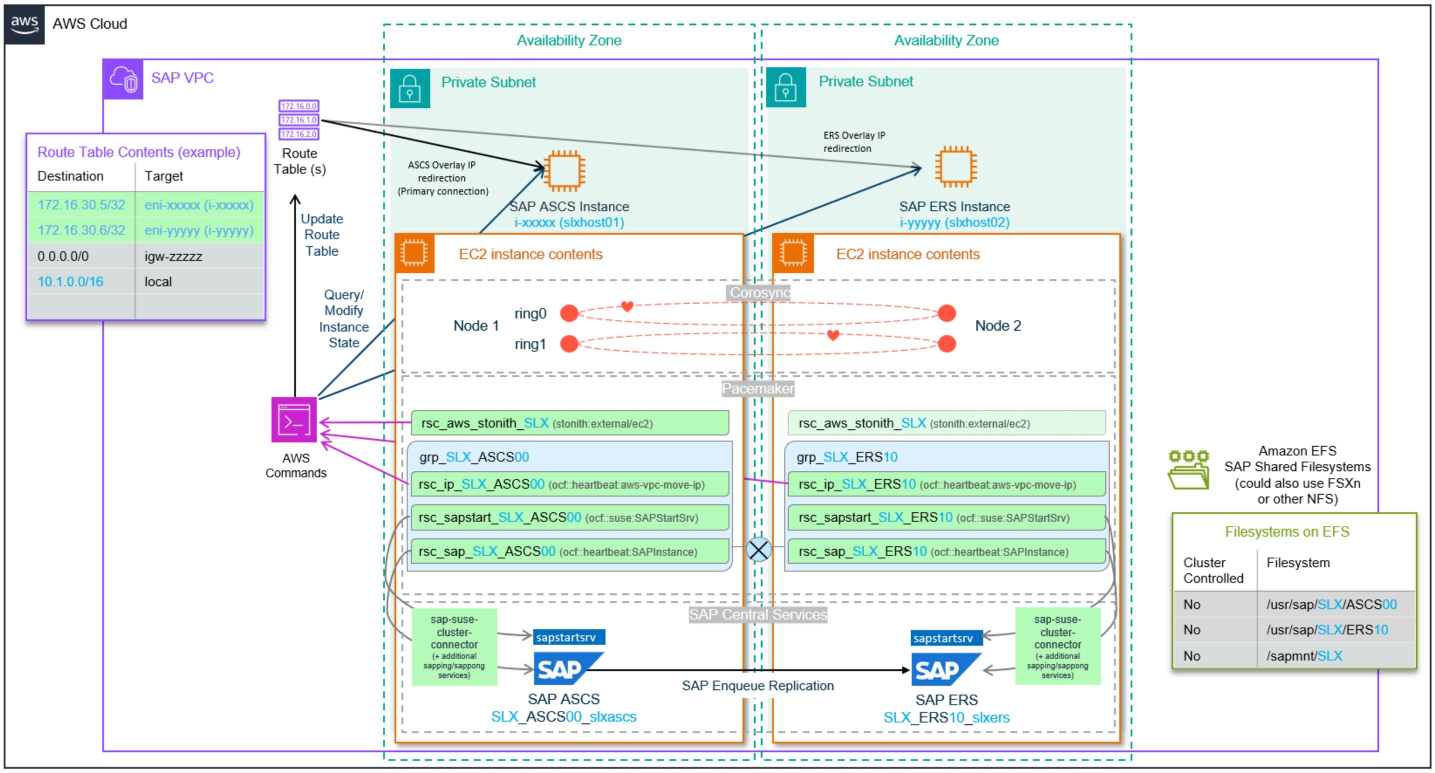
This guide covers two architectures for SAP cluster solutions on SLES for SAP – simple-mount and classic (previous standard). See the following images to learn more.

Topics

- Pacemaker - simple-mount architecture
- Pacemaker - classic architecture

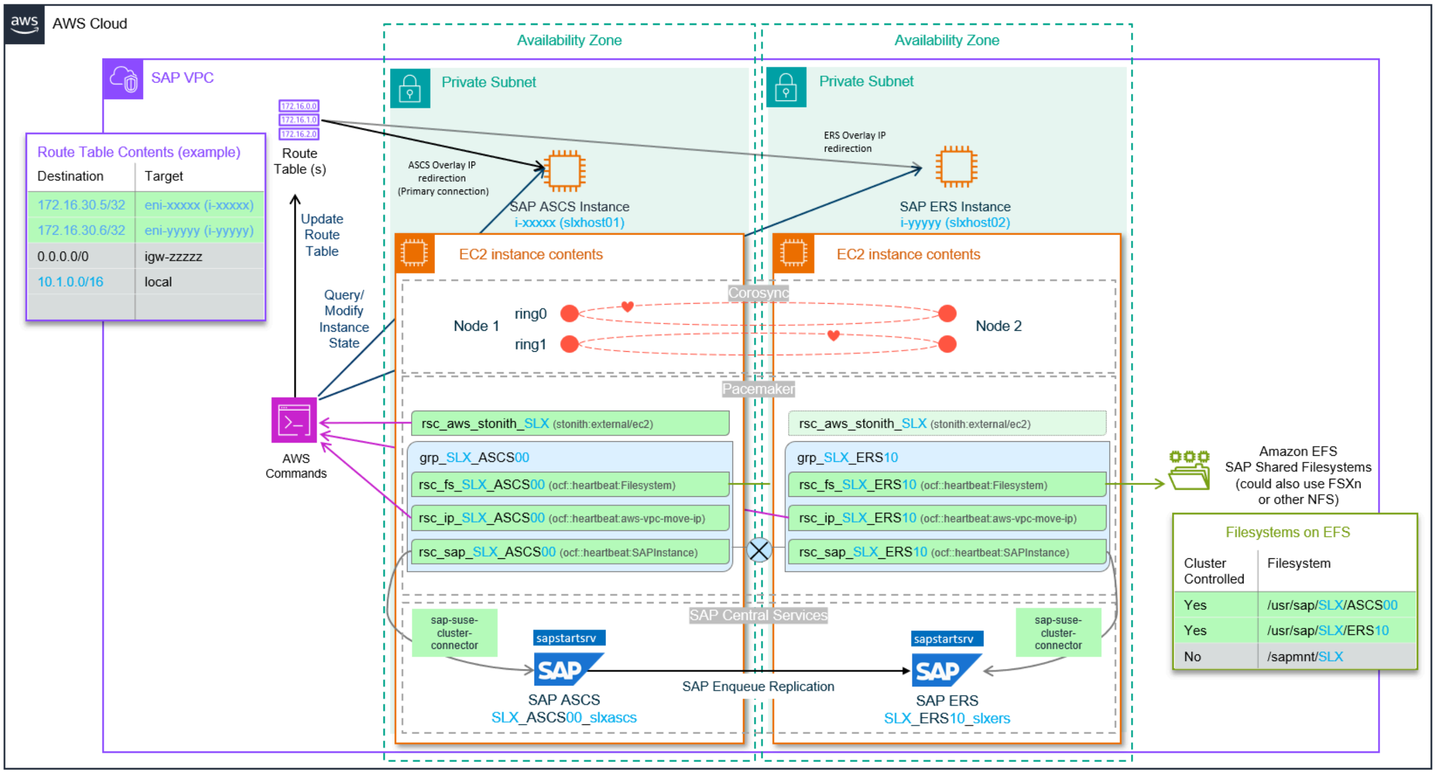
Pacemaker - simple-mount architecture

See the following image for more details.


aws
AWS Cloud
Availability Zone
Availability Zone
SAP VPC
Private Subnet
Private Subnet
172.16.0.0
172.16.1.0
172.16.2.0
Route Table (s)
Route Table Contents (example)
Destination
Target
172.16.30.5/32
eni-xxxxx (i-xxxxx)
172.16.30.6/32
eni-yyyyy (i-yyyyy)
0.0.0.0/0
igw-zzzzz
10.1.0.0/16
local
Update Route Table
Query/ Modify Instance State
AWS Commands
ASCS Overlay IP redirection (Primary connection)
SAP ASCS Instance
i-xxxxx (slxhost01)
ERS Overlay IP redirection
SAP ERS Instance
i-yyyyy (slxhost02)
EC2 instance contents
EC2 instance contents
Corosync
Node 1
ring0
ring1
Node 2
Pacemaker
rsc_aws_stonith_SLX (stonith:external/ec2)
grp_SLX_ASCS00
rsc_ip_SLX_ASCS00 (ocf::heartbeat:aws-vpc-move-ip)
rsc_sapstart_SLX_ASCS00 (ocf::suse:SAPStartSrv)
rsc_sap_SLX_ASCS00 (ocf::heartbeat:SAPInstance)
rsc_aws_stonith_SLX (stonith:external/ec2)
grp_SLX_ERS10
rsc_ip_SLX_ERS10 (ocf::heartbeat:aws-vpc-move-ip)
rsc_sapstart_SLX_ERS10 (ocf::suse:SAPStartSrv)
rsc_sap_SLX_ERS10 (ocf::heartbeat:SAPInstance)
SAP Central Services
sap-suse-cluster-connector (+ additional sapping/sappong services)
sapstartsrv
SAP
SAP ASCS
SLX_ASCS00_slxascs
SAP Enqueue Replication
sapstartsrv
SAP
SAP ERS
SLX_ERS10_slxers
sap-suse-cluster-connector (+ additional sapping/sappong services)
Amazon EFS
SAP Shared Filesystems (could also use FSXn or other NFS)
Filesystems on EFS
Cluster Controlled
Filesystem
No
/usr/sap/SLX/ASCS00
No
/usr/sap/SLX/ERS10
No
/sapmnt/SLX

Pacemaker - classic architecture

See the following image for more details.



Prerequisites

Topics

- Amazon Infrastructure Setup
- EC2 Instance Configuration
- Operating System Requirements

Amazon Infrastructure Setup

This section covers the one-time setup tasks required to prepare your Amazon environment for the cluster deployment:

> **Note**
>
> We recommend using administrative privileges from an administrative workstation or Amazon Console for the initial infrastructure setup instead of granting instance-based privileges, as this maintains the principle of least privilege. Infrastructure setup APIs (such as CreateRoute, ModifyInstanceAttribute, and CreateTags) are only required during initial configuration and are not needed for ongoing cluster operations.

Topics

- Create IAM Roles and Policies for Pacemaker
- Modify Security Groups for Cluster Communication
- Add VPC Route Table Entries for Overlay IPs

Create IAM Roles and Policies for Pacemaker

In addition to the permissions required for standard SAP operations, two IAM policies are required for the cluster to control Amazon resources. These policies must be assigned to your Amazon EC2 instance using an IAM role. This enables Amazon EC2 instance, and therefore the cluster to call Amazon services.

Note

Create policies with least-privilege permissions, granting access to only the specific resources that are required within the cluster. For multiple clusters, you may need to create multiple policies.

For more information, see IAM roles for Amazon EC2.

STONITH Policy

The SLES STONITH resource agent (external/ec2) requires permission to start and stop both the nodes of the cluster. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0"
      ]
    }
  ]
}
```

Amazon Overlay IP Policy

The SLES Overlay IP resource agent (aws-vpc-move-ip) requires permission to modify a routing entry in route tables. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:ReplaceRoute",
            "Resource": [
                 "arn:aws:ec2:us-east-1:123456789012:route-table/
rtb-0123456789abcdef0",
                 "arn:aws:ec2:us-east-1:123456789012:route-table/rtb-0123456789abcdef0"
                        ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeRouteTables",
            "Resource": "*"
        }
    ]
}
```

Shared VPC (optional)

> **Note**
>
> The following directions are only required for setups which include a Shared VPC.

Amazon VPC sharing enables you to share subnets with other Amazon accounts within the same Amazon Organizations. Amazon EC2 instances can be deployed using the subnets of the shared Amazon VPC.

In the pacemaker cluster, the aws-vpc-move-ip resource agent has been enhanced to support a shared VPC setup while maintaining backward compatibility with previous existing features.

The following checks and changes are required. We refer to the Amazon account that owns Amazon VPC as the sharing VPC account, and to the consumer account where the cluster nodes are going to be deployed as the cluster account.

Minimum Version Requirements

The latest version of the aws-vpc-move-ip agent shipped with SLES15 SP3 supports the shared VPC setup by default. The following are the minimum version required to support a shared VPC Setup:

- SLES 12 SP5 - resource-agents-4.3.018.a7fb5035-3.79.1.x86_64
- SLES 15 SP2 - resource-agents-4.4.0+git57.70549516-3.30.1.x86_64
- SLES 15 SP3 - resource-agents-4.8.0+git30.d0077df0-8.5.1

IAM Roles and Policies

Using the Overlay IP agent with a shared Amazon VPC requires a different set of IAM permissions to be granted on both Amazon accounts (sharing VPC account and cluster account).

Sharing VPC Account

In sharing VPC account, create an IAM role to delegate permissions to the EC2 instances that will be part of the cluster. During the IAM Role creation, select "Another Amazon account" as the type of trusted entity, and enter the Amazon account ID where the EC2 instances will be deployed/running from.

After the IAM role has been created, create the following IAM policy on the sharing VPC account, and attach it to an IAM role. Add or remove route table entries as needed.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "ec2:ReplaceRoute",
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:route-table/rtb-0123456789abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:route-table/rtb-0123456789abcdef0"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "ec2:DescribeRouteTables",
      "Resource": "*"
    }
  ]
}
```

Next, edit move to the "Trust relationships" tab in the IAM role, and ensure that the Amazon account you entered while creating the role has been correctly added.

In cluster account, create the following IAM policy, and attach it to an IAM role. This is the IAM Role that is going to be attached to the EC2 instances.

STS Policy

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/sharing-vpc-account-cluster-role"
    }
  ]
}
```

STONITH Policy

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    }
  ]
}
```

Modify Security Groups for Cluster Communication

A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For more information, see Control traffic to your Amazon resources using security groups.

In addition to the standard ports required to access SAP and administrative functions, the following rules must be applied to the security groups assigned to all Amazon EC2 instances in the cluster.

Source	Protocol	Port range	Description
The security group ID (its own resource ID)	UDP	5405	Allows UDP traffic between cluster resources for corosync communication
Bastion host security group or CIDR range for administration	TCP	7630	(optional) Used for SLES Hawk2 Interface for monitoring and administration using a Web Interface. For more details, see SUSE documentation Configuring and Managing Cluster Resources with Hawk2.

- Note the use of the UDP protocol.

- If you are running a local firewall, such as iptables, ensure that communication on the preceding ports is allowed between two Amazon EC2 instances.

Add VPC Route Table Entries for Overlay IPs

You need to add initial route table entries for the Overlay IP. For more information on Overlay IP, see Amazon – Overlay IP.

Add entries to the VPC route table or tables associated with the subnets of your Amazon EC2 instance for the cluster. The entries for destination (Overlay IP CIDR) and target (Amazon EC2 instance or ENI) must be added manually for the ASCS and the ERS. This ensures that the cluster resource has a route to modify. It also supports the install of SAP using the virtual names associated with the Overlay IP before the configuration of the cluster.

Using either the Amazon VPC console, or an Amazon CLI command add a route to the table or tables for the Overlay IP.

Amazon Console

1. Identify the EC2 instance IDs for both cluster nodes and determine which route tables are associated with their subnets. For details, see Parameter Reference
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc
3. In the navigation pane, choose **Route Tables**, select the first route table.
4. Choose **Actions** → **Edit routes**.
5. Choose **Add route** and configure the ASCS route:

Destination	Target
`<ascs_overlayip>/32`	`i-xxxxinstidforhost1`

6. Choose **Add route** and configure the ERS route:

Destination	Target
`<ers_overlayip>/32`	`i-xxxxinstidforhost2`

7. Choose **Save changes**.

8. Repeat for any additional associated route tables or route tables from the VPC which require connectivity to the ASCS.

   Your route table now includes entries for required Overlay IPs, in addition to the standard routes.

Amazon CLI

Identify the EC2 instance IDs for both cluster nodes and determine which route tables are associated with their subnets. For details, see. Parameter Reference.

For the ASCS:

```
$ aws ec2 create-route --route-table-id <routetable_id> --destination-cidr-block
 <ascs_overlayip>/32 --instance-id <instance_id_1>
```

For the ERS:

```
$ aws ec2 create-route --route-table-id <routetable_id> --destination-cidr-block
 <ers_overlayip>/32 --instance-id <instance_id_2>
```

EC2 Instance Configuration

Amazon EC2 instance settings can be applied using Infrastructure as Code or manually using Amazon Command Line Interface or Amazon Console. We recommend Infrastructure as Code automation to reduce manual steps, and ensure consistency.

Topics

- Assign or Review Pacemaker IAM Role
- Assign or Review Security Groups
- Assign Secondary IP Addresses
- Disable Source/Destination Check
- Review Stop Protection
- Review Automatic Recovery
- Create Amazon EC2 Resource Tags Used by Amazon EC2 STONITH Agent

Important

The following configurations must be performed on all cluster nodes. Ensure consistency across nodes to prevent cluster issues.

Assign or Review Pacemaker IAM Role

The two cluster resource IAM policies must be assigned to an IAM role associated with your Amazon EC2 instance. If an IAM role is not associated to your instance, create a new IAM role for cluster operations.

The following Amazon Console or Amazon CLI commands can be used to modify the IAM role assignment.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the navigation pane, choose **Actions** → **Security** → **Modify IAM role**.
4. Choose the IAM role that contains the policies created in Create IAM Roles and Policies for Pacemaker.
5. Choose **Update IAM role**.
6. Repeat these steps for all nodes in the cluster.

Amazon CLI

To assign an IAM role using the Amazon CLI:

```
$ aws ec2 associate-iam-instance-profile --instance-id <instance_id> --iam-instance-
profile Name=<iam_instance_profile_name>
```

Repeat for all nodes in the cluster.

You can verify the IAM role assignment on your instances using the Amazon CLI:

```
$ aws ec2 describe-instances --instance-ids <instance_id> --query
 'Reservations[0].Instances[0].IamInstanceProfile' --output table
```

You can check the specific permissions of the roles created for pacemaker in the section called “Create IAM Roles and Policies for Pacemaker” by running the following on both your instances.

When --dry-run is used, the Amazon CLI or SDK sends the request to the EC2 service with this flag. EC2 then performs all necessary permission checks and validates the request parameters. If the user has the required permissions and the request is well-formed, the service returns a DryRunOperation error response, indicating that the operation would have succeeded.

Check that the tags are correctly set and can be queried from both instances if using the ec2/stonith fencing agent:

```
$ aws ec2 describe-tags --filters "Name=resource-id,Values=<instance_id_1>"
 "Name=key,Values=
<cluster_tag>" --region=<region> --output=text | cut -f5
```

Check that the fencing resource has the permission to shut down both instances:

```
$ aws ec2 stop-instances --instance-ids <instance_id_1> --dry-run
$ aws ec2 stop-instances --instance-ids <instance_id_2> --dry-run
```

Check that the overlay IP resource has the pemissions to update the route tables:

```
$ aws ec2 replace-route --route-table-id <routetable_id> --destination-cidr-block
 <ascs_overlayip>/32 --instance-id <instance_id_1> --dry-run
```

Assign or Review Security Groups

The security group rules created in the Amazon Modify Security Groups for Cluster Communication section must be assigned to your Amazon EC2 instances. If a security group is not associated with your instance, or if the required rules are not present in the assigned security group, add the security group or update the rules.

The following Amazon Console or Amazon CLI commands can be used to modify security group assignments.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the **Security** tab, review the security groups, ports, and source of traffic.
4. If required, choose **Actions** → **Security** → **Change security groups**.
5. Under **Associated security groups**, search for and select the required groups.
6. Choose **Save**.
7. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify security groups using the Amazon CLI:

```
$ aws ec2 modify-instance-attribute --instance-id <instance_id> --groups
 <security_group_id1> <security_group_id2>
```

Repeat for all nodes in the cluster.

You can verify the security group rules on your instances using the Amazon CLI:

```
$ aws ec2 describe-instance-attribute --instance-id <instance_id> --attribute groupSet
```

Assign Secondary IP Addresses

Secondary IP addresses are used to create a redundant communication channel (secondary ring) in corosync for clusters. The cluster nodes can use the secondary ring to communicate in case of underlying network disruptions.

These IPs are only used in cluster configurations. The secondary IPs provide the same fault tolerance as a secondary Elastic Network Interface (ENI). For more information, see Secondary IP addresses for your EC2 Instance.

The following Amazon Console or Amazon CLI commands can be used to assign secondary IP addresses.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the **Networking** tab, choose the network interface ID.
4. Choose **Actions** → **Manage IP addresses**.
5. Choose **Assign new IP address**.
6. Select **Auto-assign** or specify an IP from the subnet range.
7. Choose **Yes, Update**.
8. Repeat these steps for all nodes in the cluster.

Amazon CLI

To assign secondary IP addresses using the Amazon CLI:

```
$ ENI_ID=$(aws ec2 describe-instances --instance-id <instance_id> \
    --query 'Reservations[0].Instances[0].NetworkInterfaces[0].NetworkInterfaceId' \
    --output text)
$ aws ec2 assign-private-ip-addresses --network-interface-id $ENI_ID --secondary-
private-ip-address-count 1
```

Repeat for all nodes in the cluster.

You can verify the secondary IP configuration on your instances using the Amazon CLI:

```
$ aws ec2 describe-instances --instance-id <instance_id> \
    --query
 'Reservations[*].Instances[*].NetworkInterfaces[*].PrivateIpAddresses[*].PrivateIpAddress'
 \
    --output text
```

Verify that:

- Each instance returns two IP addresses from the same subnet
- The primary network interface (eth0) has both IPs assigned
- The secondary IPs will be used later for ring0_addr and ring1_addr in corosync.conf

Disable Source/Destination Check

Amazon EC2 instances perform source/destination checks by default, requiring that an instance is either the source or the destination of any traffic it sends or receives. In the pacemaker cluster, source/destination check must be disabled on both instances receiving traffic from the Overlay IP.

The following Amazon Console or Amazon CLI commands can be used to modify the attribute.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the navigation pane, choose **Actions** → **Networking** → **Change source/destination check**.
4. For Source/Destination Checking, choose **Stop** to allow traffic when the source or destination is not the instance itself.
5. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify using the Amazon CLI (requires appropriate configuration permissions):

```
$ aws ec2 modify-instance-attribute --instance-id <instance_id> --no-source-dest-
check
```

Repeat for all nodes in the cluster.

To confirm the value of an attribute for a particular instance, use the following command. The value `false` means source/destination checking is disabled

```
$ aws ec2 describe-instance-attribute --instance-id <instance_id> --attribute
 sourceDestCheck
```

The output

```
{
    "InstanceId": "i-xxxxinstidforhost1",
    "SourceDestCheck": {
        "Value": false
    }
}
```

Review Stop Protection

To ensure that STONITH actions can be executed, you must ensure that stop protection is disabled for Amazon EC2 instances that are part of a pacemaker cluster. If the default settings have been modified, use the following commands for both instances to disable stop protection via Amazon CLI.

The following Amazon Console or CLI commands can be used to modify the attribute.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. Choose **Actions** → **Instance settings** → **Change stop protection**.
4. Ensure **Stop protection** is not enabled.
5. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify using the Amazon CLI (requires appropriate configuration permissions):

```
$ aws ec2 modify-instance-attribute --instance-id <instance_id> --no-disable-api-
stop
```

Repeat this command for all nodes in the cluster.

To confirm the value of an attribute for a particular instance, use the following command. The value `false` means it is possible to stop the instance using an Amazon CLI.

```
$ aws ec2 describe-instance-attribute --instance-id <instance_id> --attribute
 disableApiStop
```

The output

```
{
    "InstanceId": "i-xxxxinstidforhost1",
    "DisableApiStop": {
        "Value": false
    }
}
```

Review Automatic Recovery

After a failure, cluster-controlled operations must be resumed in a coordinated way. This helps ensure that the cause of failure is known and addressed, and the status of the cluster is as expected. For example, verifying that there are no pending fencing actions.

The following Amazon Console or CLI commands can be used to modify the attribute.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. Choose **Actions** → **Instance settings** → **Change auto-recovery behavior**.
4. Select **Off** to disable auto-recovery for system status check failures.
5. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify auto-recovery settings (requires appropriate configuration permissions):

```
$ aws ec2 modify-instance-maintenance-options --instance-id <instance_id> --auto-recovery disabled
```

Repeat this command for all nodes in the cluster.

To confirm the value of an attribute for a particular instance, use the following command. The value `disabled` means autorecovery will not be attempted.

```
$ aws ec2 describe-instances --instance-ids <instance_id> --query
 'Reservations[*].Instances[*].MaintenanceOptions.AutoRecovery'
```

The output:

```
[
    [
        "disabled"
    ]
]
```

Create Amazon EC2 Resource Tags Used by Amazon EC2 STONITH Agent

Amazon EC2 STONITH agent uses Amazon resource tags to identify Amazon EC2 instances. Create tag for the primary and secondary Amazon EC2 instances via Amazon Console or Amazon CLI. For more information, see Using Tags.

Use the same tag key and the local hostname returned using the command hostname across instances. For example, a configuration with the values defined in Global Amazon parameters would require the tags shown in the following table.

Amazon EC2	Key example	Value example
`<instance_id>`	`<cluster_tag>`	`<hostname>`
`i-xxxxinstidforhost1`	`pacemaker`	`slxhost01`
`i-xxxxinstidforhost2`	`pacemaker`	`slxhost02`

The following Amazon Console or Amazon CLI commands can be used to create resource tags.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the **Tags** tab, choose **Manage tags**.
4. Choose **Add tag**.
5. For **Key**, enter the cluster tag (for example, `pacemaker`).
6. For **Value**, enter the hostname of the instance.
7. Choose **Save**.

8. Repeat these steps for all nodes in the cluster.

Amazon CLI

To create tags using the Amazon CLI:

```
$ aws ec2 create-tags --resources <instance_id> --tags
 Key=<cluster_tag>,Value=<hostname>
```

Repeat for all nodes in the cluster with their respective hostnames.

You can run the following command locally to validate the tag values and IAM permissions to describe the tags. Run this command on all instances in the cluster, for all instances in the cluster.

```
$ aws ec2 describe-tags --filters "Name=resource-id,Values=<instance_id>"
 "Name=key,Values=<cluster_tag>" --region=<region> --output=text | cut -f5
```

Operating System Requirements

This section outlines the required operating system configurations for SUSE Linux Enterprise Server for SAP (SLES for SAP) cluster nodes. Note that this is not a comprehensive list of configuration requirements for running SAP on Amazon, but rather focuses specifically on cluster management prerequisites.

Consider using configuration management tools or automated deployment scripts to ensure accurate and repeatable setup across your cluster infrastructure.

Topics

Important

The following configurations must be performed on all cluster nodes. Ensure consistency across nodes to prevent cluster issues.

Root Access

Verify root access on both cluster nodes. The majority of the setup commands in this document are performed with the root user. Assume that commands should be run as root unless there is an explicit call out to choose otherwise.

Install Missing Operating System Packages

This is applicable to all cluster nodes. You must install any missing operating system packages.

The following packages and their dependencies are required for the pacemaker setup. Depending on your baseline image, for example, SLES for SAP, these packages may already be installed.

Package	Description	Category	Required	Configuration Pattern
chrony	Time Synchronization	System Support	Mandatory	All
rsyslog	System Logging	System Support	Mandatory	All
pacemaker	Cluster Resource Manager	Core Cluster	Mandatory	All
corosync	Cluster Communication Engine	Core Cluster	Mandatory	All
resource-agents	Resource Agents including SAPInstance	Core Cluster	Mandatory	All
fence-agents	Fencing Capabilities	Core Cluster	Mandatory	All

Package	Description	Category	Required	Configuration Pattern
sap-suse-cluster-connector	SAP HA-Script Connector (≥3.1.1 for SimpleMount)	SAP Integration	Mandatory	All
sapstartsrv-resource-agents	SAP Start Service Resource Agents	SAP Integration	Mandatory*	SimpleMount
supportutils	System Information Gathering	Support Tools	Recommended	All
sysstat	Performance Monitoring Tools	Support Tools	Recommended	All
zypper-lifecycle-plugin	Software Lifecycle Management	Support Tools	Recommended	All
supportutils-plugin-ha-sap	HA/SAP Support Data Collection	Support Tools	Recommended	All
supportutils-plugin-suse-public-cloud	Cloud Support Data Collection	Support Tools	Recommended	All
dstat	System Resource Statistics	Monitoring	Recommended	All
iotop	I/O Monitoring	Monitoring	Recommended	All

Note

Refer to Vendor Support of Deployment Types for more information on Configuration Patterns. `Mandatory*` indicates that this package is mandatory based on the Configuration Pattern.

```
#!/bin/bash
# Mandatory core packages for SAP NetWeaver HA on AWS
mandatory_packages="corosync pacemaker resource-agents fence-agents rsyslog chrony sap-suse-cluster-connector"

# SimpleMount specific packages
simplemount_packages="sapstartsrv-resource-agents"

# Recommended monitoring and support packages
support_packages="supportutils supportutils-plugin-ha-sap supportutils-plugin-suse-public-cloud sysstat dstat iotop zypper-lifecycle-plugin"

# Default to checking all packages
packages="${mandatory_packages} ${simplemount_packages} ${support_packages}"

missingpackages=""

echo "Checking SAP NetWeaver HA package requirements..."
echo "Note: sapstartsrv-resource-agents is only required for SimpleMount architecture"

for package in ${packages}; do
    echo "Checking if ${package} is installed..."
    if ! rpm -q ${package} --quiet; then
        echo " ${package} is missing and needs to be installed"
        missingpackages="${missingpackages} ${package}"
    fi
done

if [ -z "$missingpackages" ]; then
    echo "All packages are installed."
else
    echo "Missing mandatory packages: $(echo ${missingpackages} | tr ' ' '\n' | grep -E "^($(echo ${mandatory_packages} | tr ' ' '|'))$")"
    echo "Missing SimpleMount packages: $(echo ${missingpackages} | tr ' ' '\n' | grep -E "^($(echo ${simplemount_packages} | tr ' ' '|'))$")"
    echo "Missing support packages: $(echo ${missingpackages} | tr ' ' '\n' | grep -E "^($(echo ${support_packages} | tr ' ' '|'))$")"

    echo -n "Do you want to install the missing packages (y/n)? "
    read response
    if [ "$response" = "y" ]; then
        zypper install -y $missingpackages
    fi
```

```
fi

# Check sap-suse-cluster-connector version if installed
if rpm -q sap-suse-cluster-connector --quiet; then
    version=$(rpm -q sap-suse-cluster-connector --qf '%{VERSION}')
    echo "sap-suse-cluster-connector version: $version"
    if [[ $(echo "$version" | cut -d. -f1) -ge 3 ]] && [[ $(echo "$version" | cut -d. -
f2) -ge 1 ]] && [[ $(echo "$version" | cut -d. -f3) -ge 1 ]]; then
        echo "sap-suse-cluster-connector version is suitable for SimpleMount
 architecture"
    else
        echo "WARNING: SimpleMount architecture requires sap-suse-cluster-connector
 version 3.1.1 or higher"
    fi
fi
```

If a package is not installed, and you are unable to install it using zypper, it may be because SUSE Linux Enterprise High Availability extension is not available as a repository in your chosen image. You can verify the availability of the extension using the following command:

```
$ sudo zypper repos
```

To install or update a package or packages with confirmation, use the following command:

```
$ sudo zypper install <package_name(s)>
```

Update and Check Operating System Versions

You must update and confirm versions across nodes. Apply all the latest patches to your operating system versions. This ensures that bugs are addressed and new features are available.

You can update the patches individually or update all system patches using the `zypper update` command. A clean reboot is recommended prior to setting up a cluster.

```
$ sudo zypper update
$ sudo reboot
```

Compare the operating system package versions on the two cluster nodes and ensure that the versions match on both nodes.

System Logging

Both systemd-journald and rsyslog are suggested for comprehensive logging. Systemd-journald (enabled by default) provides structured, indexed logging with immediate access to events, while rsyslog is maintained for backward compatibility and traditional file-based logging. This dual approach ensures both modern logging capabilities and compatibility with existing log management tools and practices.

1. Enable and start rsyslog:

```
# systemctl enable --now rsyslog
```

2. (Optional) Configure persistent logging for systemd-journald:

If you are not using a logging agent (like the Amazon CloudWatch Unified Agent or Vector) to ship logs to a centralized location, you may want to configure persistent logging to retain logs after system reboots.

```
# mkdir -p /etc/systemd/journald.conf.d
```

Create `/etc/systemd/journald.conf.d/99-logstorage.conf` with:

```
[Journal]
Storage=persistent
```

Persistent logging requires careful storage management. Configure appropriate retention and rotation settings in `journald.conf` to prevent logs from consuming excessive disk space. Review `man  journald.conf` for available options such as SystemMaxUse, RuntimeMaxUse, and MaxRetentionSec.

To apply the changes, restart journald:

```
# systemctl restart systemd-journald
```

After enabling persistent storage, only new logs will be stored persistently. Existing logs from the current boot session will remain in volatile storage until the next reboot.

3. Verify services are running:

```
# systemctl status systemd-journald
# systemctl status rsyslog
```

Time Synchronization Services

Time synchronization is important for cluster operation. Ensure that chrony rpm is installed, and configure appropriate time servers in the configuration file.

You can use Amazon Time Sync Service that is available on any instance running in a VPC. It does not require internet access. To ensure consistency in the handling of leap seconds, don’t mix Amazon Time Sync Service with any other ntp time sync servers or pools.

Create or check the `/etc/chrony.d/ec2.conf` file to define the server:

```
# Amazon EC2 time source config
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

Start the chronyd.service, using the following command:

```
# systemctl enable --now chronyd.service
# systemctl status chronyd
```

Verify time synchronization is working:

```
# chronyc tracking
```

Ensure the output shows `Reference ID : A9FEA97B (169.254.169.123)` confirming synchronization with Amazon Time Sync Service.

For more information, see Set the time for your Linux instance.

Install Amazon CLI and Configure Profiles

The Amazon cluster resource agents require Amazon Command Line Interface (Amazon CLI). Check if Amazon CLI is already installed, and install it if necessary.

Check if Amazon CLI is installed:

```
# aws --version
```

If the command is not found, install Amazon CLI v2 using the following commands:

```
# cd /tmp
# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
# dnf install -y unzip
# unzip awscliv2.zip
# sudo ./aws/install --update
```

Create symlinks to ensure Amazon CLI is in the system PATH:

```
# sudo ln -sf /usr/local/bin/aws /usr/bin/aws
```

Verify the installation:

```
# aws --version
```

The installation creates a symbolic link at `/usr/local/bin/aws` which is typically in the system PATH by default.

For more information, see [Installing or updating to the latest version of the Amazon CLI]().

After installing Amazon CLI, you need to create an Amazon CLI profile for the root account.

You can either edit the config file at `/root/.aws` manually or by using the `aws configure` Amazon CLI command.

You should skip providing the information for the access and secret access keys. The permissions are provided through IAM roles attached to Amazon EC2 instances.

```
# aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: <region>
Default output format [None]:
```

The profile name is `default` unless configured. If you choose to use a different name you can specify `--profile`. The name chosen in this example is cluster. It is used in the Amazon resource agent definition for pacemaker. The Amazon Region must be the default Amazon Region of the instance.

```
# aws configure --profile cluster
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: <region>
Default output format [None]:
```

On the hosts, you can verify the available profiles using the following command:

```
# aws configure list-profiles
```

And review that an assumed role is associated by querying the caller identity:

```
# aws sts get-caller-identity --profile=<profile_name>
```

Pacemaker Proxy Settings (Optional)

If your Amazon EC2 instance has been configured to access the internet and/or Amazon Cloud through proxy servers, then you need to replicate the settings in the pacemaker configuration. For more information, see Using an HTTP Proxy.

Add the following lines to `/etc/sysconfig/pacemaker`:

```
http_proxy=http://<proxyhost>:<proxyport>
https_proxy=http://<proxyhost>:<proxyport>
no_proxy=127.0.0.1,localhost,169.254.169.254,fd00:ec2::254
```

- Modify proxyhost and proxyport to match your settings.
- Ensure that you exempt the address used to access the instance metadata.
- Configure no_proxy to include the IP address of the instance metadata service – 169.254.169.254 (IPV4) and fd00:ec2::254 (IPV6). This address does not vary.

SAP ASCS and Cluster Setup

This section covers the following topics.

Topics

- SAP Shared File Systems

- Check IP availability and resolution
- Install SAP
- Configure SAP for Cluster Control
- Cluster Node Setup
- Cluster Configuration

SAP Shared File Systems

Topics

Select Shared Storage

SAP NetWeaver high availability deployments require shared file systems. On Linux, you can use either Amazon Elastic File System or Amazon FSx for NetApp ONTAP. Choose between these options based on your requirements for resilience, performance, and cost. For detailed setup information, see Getting started with Amazon Elastic File System or Getting started with Amazon FSx for NetApp ONTAP.

We recommend sharing a single Amazon EFS or FSx for ONTAP file system across multiple SIDs within an account.

The file system's DNS name is the simplest mounting option. When connecting from an Amazon EC2 instance, the DNS automatically resolves to the mount target's IP address in that instance's Availability Zone. You can also create an alias (CNAME) to help identify the shared file system's purpose. Throughout this document, we use `<nfs.fqdn>`.

Examples:

- `file-system-id.efs.aws-region.amazonaws.com`
- `svm-id.fs-id.fsx.aws-region.amazonaws.com`
- `qas_sapmnt_share.example.com`

Note

Review the `enableDnsHostnames` and `enableDnsSupport` DNS attributes for your VPC. For more information, see View and update DNS attributes for your VPC.

Create file systems

The following shared file systems are covered in this document:

NFS Location Structure	NFS Location Example	File System Location Structure	File System Location Example
<SID>_sapmnt	`SLX_sapmnt`	/sapmnt/<SID>	`/sapmnt/SLX`
<SID>_ASCS<ascs_sys_nr>	`SLX_ASCS00`	/usr/sap/<SID>/ASCS<ascs_sys_nr>	`/usr/sap/SLX/ASCS00`
<SID>_ERS<ers_sys_nr>	`SLX_ERS10`	/usr/sap/<SID>/ERS<ers_sys_nr>	`/usr/sap/SLX/ERS10`

The following options can differ depending on how you architect and operate your systems:

- ASCS and ERS mount points - In simple-mount architecture, you can share the entire `/usr/sap/<SID>` directory. This document uses separate mount points to simplify migration and follow SAP's recommendation for local application server executables when co-hosting ASCS/ERS.
- Transport directory - `/usr/sap/trans` is optional for ASCS installations. Add this shared directory if your change management processes require it.
- Home directory - This document uses local home directories to ensure `<sid>adm` access during NFS issues. Consider a shared home directory if you need consistent user environments across nodes.
- NFS location naming - The "NFS Location" names are arbitrary and can be chosen based on your naming conventions (e.g., `myEFSMount1`, `prod_sapmnt`, etc.). The "File system location" follows the standard SAP directory structure and should use the parameter references shown.

For more information, see SAP System Directories on UNIX.

Using the NFS ID created in the previous step, temporarily mount the root directory of the NFS. `/mnt` is available by default; it can also be substituted with another temporary location.

Note

The following commands use the NFS location names from the table above. Replace `<SID>_sapmnt`, `<SID>_ASCS<ascs_sys_nr>`, and `<SID>_ERS<ers_sys_nr>` with your chosen NFS location names and parameter values.

```
# mount <nfs.fqdn>:/ /mnt
# mkdir -p /mnt/<SID>_sapmnt
# mkdir -p /mnt/<SID>_ASCS<ascs_sys_nr>
# mkdir -p /mnt/<SID>_ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/ /mnt
# mkdir -p /mnt/SLX_sapmnt
# mkdir -p /mnt/SLX_ASCS00
# mkdir -p /mnt/SLX_ERS10
```

During SAP installation, the `<sid>adm` user and proper directory ownership will be created. Until then, we need to ensure the installation process has sufficient access. Set temporary permissions on the directories:

```
# chmod 777 /mnt/<SID>_sapmnt /mnt/<SID>_ASCS<ascs_sys_nr> /mnt/<SID>_ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# chmod 777 /mnt/SLX_sapmnt /mnt/SLX_ASCS00 /mnt/SLX_ERS10
```

The SAP installation process will automatically set the correct ownership and permissions for operational use.

Unmount the temporary mount:

```
# umount /mnt
```

Create mount point directories

This is applicable to both cluster nodes. Create the directories for the required mount points (permanent or cluster controlled):

```
# mkdir /sapmnt
# mkdir /usr/sap/<SID>/ASCS<ascs_sys_nr>
# mkdir /usr/sap/<SID>/ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

  ```
  # mkdir /sapmnt
  # mkdir /usr/sap/SLX/ASCS00
  # mkdir /usr/sap/SLX/ERS10
  ```

Update /etc/fstab

This is applicable to both cluster nodes. `/etc/fstab` is a configuration table containing the details required for mounting and unmounting file systems to a host.

Add the file systems not managed by the cluster to `/etc/fstab`.

For both **simple-mount** and **classic** architectures, prepare and append an entry for the `sapmnt` file system to `/etc/fstab`:

```
<nfs.fqdn>/<SID>_sapmnt    /sapmnt    nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport    0    0
```

Simple-mount only – prepare and append entries for the ASCS and ERS file systems to `/etc/fstab`:

```
<nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr>   /usr/sap/<SID>/ASCS<ascs_sys_nr>  nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport    0    0
<nfs.fqdn>:/<SID>_ERS<ers_sys_nr>     /usr/sap/<SID>/ERS<ers_sys_nr>    nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport    0    0
```

- *Example using values from Parameter Reference* :

```
fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_sapmnt    /sapmnt
 nfs    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
  0    0
fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_ASCS00    /usr/sap/SLX/ASCS00
 nfs    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
  0    0
fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_ERS10     /usr/sap/SLX/ERS10
 nfs    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
  0    0
```

Verify that your mount options are:

- Compatible with your operating system version
- Supported by your chosen NFS file system type (EFS or FSx for ONTAP)
- Aligned with current SAP recommendations

Consult SAP and Amazon documentation for the latest mount option recommendations.

Use the following command to mount the file systems defined in `/etc/fstab`:

```
# mount -a
```

Use the following command to check that the required file systems are available:

```
# df -h
```

Temporarily mount ASCS and ERS directories for installation (classic only)

This is only applicable to the classic architecture. Simple-mount architecture has these directories permanently available in `/etc/fstab`.

Mount ASCS and ERS directories for installation.

Use the following command on the instance where you plan to install ASCS:

```
# mount <nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr>  /usr/sap/<SID>/ASCS<ascs_sys_nr>
```

Use the following command on the instance where you plan to install ERS:

```
# mount <nfs.fqdn>:/<SID>_ERS<ers_sys_nr>  /usr/sap/<SID>/ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_ASCS00  /usr/sap/SLX/
ASCS00
# mount fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_ERS10   /usr/sap/SLX/
ERS10
```

Check IP availability and resolution

Add Overlay IP for SAP Installation

SAP Installation should be done using the virtual names assigned to the overlay IP. Before adding the overlay IPs to the instances, ensure that the VPC route table entries have been created as described in the section called "Add VPC Route Table Entries for Overlay IPs".

To facilitate SAP installation, manually add the Overlay IPs to the instances:

1. To the instance where you intend to install the **ASCS**

```
# ip addr add <ascs_overlayip>/32 dev eth0
```

2. To the instance where you intend to install the **ERS**

```
# ip addr add <ers_overlayip>/32 dev eth0
```

Note the following:

- Route table entries for the overlay IPs must be created first (see the section called "Add VPC Route Table Entries for Overlay IPs")
- This IP configuration is temporary and will be lost after instance reboot
- The cluster will take over management of these IPs once configured

Hostname Resolution

You must ensure that all instances can resolve all hostnames in use. Add the hostnames for cluster nodes to `/etc/hosts` file on all cluster nodes. This ensures that hostnames for cluster nodes can be resolved even in case of DNS issues. Configure the `/etc/hosts` file for a two-node cluster:

```
# cat /etc/hosts
<primary_ip_1> <hostname_1>.example.com <hostname_1>
<primary_ip_2> <hostname_2>.example.com <hostname_2>
<ascs_overlayip> <ascs_virt_hostname>.example.com <ascs_virt_hostname>
<ers_overlayip> <ers_virt_hostname>.example.com <ers_virt_hostname>
```

- *Example using values from [Parameter Reference](#)* :

  ```
  # cat /etc/hosts
  10.1.10.1 slxhost01.example.com slxhost01
  10.1.20.1 slxhost02.example.com slxhost02
  172.16.30.5 slxascs.example.com slxascs
  172.16.30.6 slxers.example.com slxers
  ```

In this configuration, the secondary IPs used for the second cluster ring are not mentioned. They are only used in the cluster configuration. You can allocate virtual hostnames for administration and identification purposes.

> **Important**
>
> The Overlay IP is out of VPC range, and cannot be reached from locations not associated with the route table, including on-premises.

Install SAP

The following topics provide information about installing SAP on Amazon Cloud in a highly available cluster. Review SAP Documentation for more details.

Topics

- [Final checks for software provisioning](#)
- [Install SAP ASCS and ERS instances](#)

- Kernel upgrade and ENSA2 – optional
- Check SAP host agent version

Final checks for software provisioning

Before running SAP Software Provisioning Manager (SWPM), ensure that the following prerequisites are consistent across both cluster nodes:

- Collect any missing details and populate the Parameter Reference section to ensure clarity on the specific values used in installation commands.
- **User and Group Configuration** - If operating system groups are pre-defined, ensure matching UID and GID values for `<sid>adm` and `sapsys` across both cluster nodes.
- **Installation Software** - Download the latest version of Software Provisioning Manager (SWPM) and SAP installation media for your SAP release from Software Provisioning Manager.
- **Network Configuration** - Verify both cluster nodes have identical configuration with all routes, overlay IPs, and virtual hostnames accessible. This ensures that either node can run ASCS or ERS roles.
- **File Systems** - Verify all shared file systems are mounted and accessible from both nodes with consistent mount points and permissions.

Install SAP ASCS and ERS instances

Install the SAP ASCS and ERS instances using their virtual hostnames to ensure installation against the overlay IP addresses. This approach is required for proper cluster integration.

Install the ASCS instance on `<instance_id_1>` using virtual hostname `<ascs_virt_hostname>` with the `SAPINST_USE_HOSTNAME` parameter. This ensures the installation uses the overlay IP rather than the physical hostname:

Example using values from Parameter Reference :

```
# <swpm location>/sapinst SAPINST_USE_HOSTNAME=<ascs_virt_hostname>
```

Install the ERS instance on `<instance_id_2>` using virtual hostname `<ers_virt_hostname>` with the `SAPINST_USE_HOSTNAME` parameter. This ensures the installation uses the overlay IP rather than the physical hostname:

```
# <swpm location>/sapinst SAPINST_USE_HOSTNAME=<ers_virt_hostname>
```

Once the ASCS and ERS installations are complete, you will need to install and configure the database and SAP Primary Application Server (PAS) - these components are not covered in this cluster setup documentation. Optionally, you can also install and configure Additional Application Server (AAS). For more details on installing these SAP NetWeaver components, refer to SAP Help Portal.

For additional information on unattended installation options, see SAP Note 2230669 – System Provisioning Using an Input Parameter File (requires SAP portal access).

Kernel upgrade and ENSA2 – optional

As of AS ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. ENSA2 replaces the previous version – ENSA1.

If you have an older version of SAP NetWeaver, consider following the SAP guidance to upgrade the kernel and update the Enqueue Server configuration. An upgrade will allow you to take advantage of the features available in the latest version. For more information, see the following SAP Notes (require SAP portal access):

- SAP Note 2630416 – Support for Standalone Enqueue Server 2
- SAP Note 2711036 – Usage of the Standalone Enqueue Server 2 in an HA Environment

Check SAP host agent version

This is applicable to both cluster nodes. The SAP host agent is used for system instance control and monitoring. This agent is used by SAP cluster resource agents and hooks. It is recommended that you have the latest version installed on both instances. For more details, see SAP Note 2219592 – Upgrade Strategy of SAP Host Agent.

Use the following command to check the version of the host agent:

```
# /usr/sap/hostctrl/exe/saphostexec -version
```

Configure SAP for Cluster Control

Modify SAP service configurations, user permissions, and system integration settings to enable proper cluster control of ASCS and ERS instances.

Topics

- Add <sid>adm to haclient group
- Modify SAP profiles for start operations and cluster hook
- Enable sapping and sappong Services (Simple-Mount Only)
- Ensure ASCS and ERS SAP Services can run on either node (systemd)
- Configure dependencies for Pacemaker and SAP services (systemd)
- (Alternative) Ensure ASCS and ERS SAP Services can run on either node (sysV)

Add <sid>adm to haclient group

This is applicable to both cluster nodes. An `haclient` operating system group is created when the cluster connector package is installed. Adding the `<sid>adm` user to this group ensures that your cluster has necessary access. Run the following command as root:

```
# usermod -a -G haclient <sid>adm
```

- *Example using values from Parameter Reference* :

  ```
  # usermod -a -G haclient slxadm
  ```

Modify SAP profiles for start operations and cluster hook

This action ensures that there is compatibility between the SAP start framework and cluster actions. Modify SAP profiles to change the start behavior of the SAP instance and processes. Ensure that `sapcontrol` is aware that the system is being managed by a pacemaker cluster.

- ASCS profile – `/usr/sap/<SID>/SYS/profile/<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>`
- ERS profile – `/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>`

The profile directory /usr/sap/<SID>/SYS/profile/ is typically a symbolic link to /sapmnt/<SID>/profile/ on the shared NFS filesystem. This means profile modifications made on one node are immediately visible on all cluster nodes. You can modify the profiles from either node.

- *Example using values from [Parameter Reference](#)* :
  - ASCS profile example – `/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs`
  - ERS profile example – `/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers`

Follow the procedure outlined below to make the necessary changes:

1. **Program or process start behavior** – In case of failure, processes must be restarted. Determining where the process starts and in what order needs to be controlled by the cluster, and not SAP start framework behavior defined in the profiles. Your locks can be lost if this parameter is not changed. In newer SAP installations, the profiles may already contain `Start_Program_XX` instead of `Restart_Program_XX`. If `Start_Program_XX` is already present, no changes are needed for this step.

   Example

   ENSA1

   ASCS

   ```
   #For ENSA1 (_EN)
   #Changing Restart to Start for Cluster compatibility
   #Old value: Restart_Program_XX = local $(_EN) pf=$(_PF)

   Start_Program_XX = local $(_EN) pf=$(_PF)
   ```

   ERS

   ```
   #For ENSA1 (_ER)
   #Changing Restart to Start for Cluster compatibility
   #Old value: Restart_Program_XX = local $(_ER) pf=$(_PFL)NR=$(SCSID)

   Start_Program_XX = local $(_ER) pf=$(_PFL) NR=$(SCSID)
   ```

   `XX` indicates the start-up order. This value may be different in your install; retain the unchanged value.

   ENSA2

   ASCS

```
#For ENSA2 (_ENQ)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQ) pf=$(_PF)

Start_Program_XX = local $(_ENQ) pf=$(_PF)
```

ERS

```
#For ENSA2 (_ENQR)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQR) pf=$(_PFL)NR=$(SCSID)

Start_Program_XX = local $(_ENQR) pf=$(_PFL) NR=$(SCSID)
```

`XX` indicates the start order. This value may be different in your install; retain the unchanged value.

2. **Disable instance auto start in both profiles** – When an instance restarts, SAP start framework should not start ASCS and ERS automatically. Add the following parameter on both profiles to prevent an auto start:

```
# Disable instance auto start
Autostart = 0
```

3. **Add cluster connector details in both profiles** – The connector integrates the SAP start and control frameworks of SAP NetWeaver with SUSE cluster to assist with maintenance and awareness of state. Add the following parameters on both profiles:

```
# Added for Cluster Connectivity
service/halib = $(DIR_EXECUTABLE)/saphascriptco.so
service/halib_cluster_connector = /usr/bin/sap_suse_cluster_connector
```

> **Important**
>
> RPM package `sap-suse-cluster-connector` has *dashes*. The executable `/usr/bin/sap_suse_cluster_connector` available after installation has *underscores*. Ensure that the correct name, that is executable `/usr/bin/sap_suse_cluster_connector`, is used in both profiles.

4. **Restart services** – Restart SAP services for ASCS and ERS to ensure that the preceding settings take effect. Adjust the system number to match the service.

   ASCS

   ```
   # /usr/sap/hostctrl/exe/sapcontrol -nr <ascs_sys_nr> -function RestartService
   ```

   ERS

   ```
   # /usr/sap/hostctrl/exe/sapcontrol -nr <ers_sys_nr> -function RestartService
   ```

   - *Example using values from Parameter Reference* :

     ASCS

     ```
     # /usr/sap/hostctrl/exe/sapcontrol -nr 00 -function RestartService
     ```

     ERS

     ```
     # /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function RestartService
     ```

5. **Check integration using `sapcontrol`** – `sapcontrol` includes functions: `HACheckConfig` and `HACheckFailoverConfig`. These functions can be used to check configuration, including awareness of the cluster connector. These checks have limited value before the cluster is configured, but you can run HACheckFailoverConfig to ensure the base configuration is in place.

   ASCS

   ```
   # /usr/sap/hostctrl/exe/sapcontrol -nr <ascs_sys_nr> -function HACheckFailoverConfig
   ```

   - *Example using values from Parameter Reference* :

     ASCS

     ```
     # /usr/sap/hostctrl/exe/sapcontrol -nr 00 -function HACheckFailoverConfig

     10.10.2025 01:23:55
     HACheckFailoverConfig
     OK
     state, category, description, comment
     ```

```
SUCCESS, SAP CONFIGURATION, SAPInstance RA sufficient version, SAPInstance includes
 is-ers patch
```

Enable sapping and sappong Services (Simple-Mount Only)

For simple-mount architecture, enable the sapping and sappong systemd services on both cluster nodes. These services ensure proper SAP instance startup coordination between systemd and the cluster.

The sapping service runs before sapinit during boot and temporarily hides the `/usr/sap/sapservices` file to prevent automatic SAP instance startup. The sappong service runs after sapinit and restores the sapservices file, making it available for cluster management while maintaining compatibility with SAP management tools.

```
# systemctl enable sapping
# systemctl enable sappong
```

Verify the services are enabled:

```
# systemctl status sapping
# systemctl status sappong
```

> **Note**
>
> Both services will show "inactive (dead)" status, which is normal for one-shot services that only run during system boot.

Ensure ASCS and ERS SAP Services can run on either node (systemd)

This is applicable to both cluster nodes.

To ensure that the cluster can orchestrate availability by starting and stopping instances on either cluster node, the SAP Services must be registerd on both nodes and auto-start should be disabled.

In recent Operating System and SAP kernel versions, SAP offers systemd integration for sapstartsrv which controls how SAP instances are stopped and started. This is the recommended configuration and a requirement for Simple Mount Configuration.

For more details, see the following SAP Notes (require SAP portal access):

- SAP Note 3139184 – Linux: systemd integration for sapstartsrv and SAP Host Agent
- SAP Note 3115048 – sapstartsrv with native Linux systemd support

You can confirm whether systemd is in place by running the following command. Systemd is in place if SAP Services (e.g., SAPSLX_00.service, SAPSLX_10.service) are listed.

```
# systemctl list-unit-files SAP*
```

If you have installed an ASCS or ERS on this host but no SAP Services are returned, the classic SysV init may be in use. In that case you can skip to section the section called “(Alternative) Ensure ASCS and ERS SAP Services can run on either node (sysV)”

1. **On the instance where the ASCS was installed**

   Register the missing ERS service on the node where you have installed ASCS.

   a. Temporarily mount the ERS directory (classic only):

   ```
   # mount <nfs.fqdn>:/<SID>_ERS<ers_sys_nr>  /usr/sap/<SID>/ERS<ers_sys_nr>
   ```

   b. Register the ERS service:

   ```
   # export LD_LIBRARY_PATH=/usr/sap/<SID>/ERS<ers_sys_nr>/exe
   # /usr/sap/<SID>/ERS<ers_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/SYS/profile/
   <SID>_ERS<ers_sys_nr>_<ers_virt_hostname> -reg
   # systemctl start SAP<SID>_<ers_sys_nr>
   ```

   c. Check the existence and state of SAP services (example):

   ```
   # systemctl list-unit-files SAP*
   UNIT FILE                    STATE   VENDOR PRESET
   SAPSLX.service            disabled disabled
   SAPSLX.service            disabled disabled
   SAP.slice                   static   -
   3 unit files listed.
   ```

   d. If the state is not disabled, run the following command to disable `sapservices` integration for SAP<SID>_<ascs_sys_nr> and SAP<SID>_<ers_sys_nr> on both nodes:

> ⚠ **Important**
>
> Stopping these services also stops the associated SAP instances.

```
# systemctl stop SAP<SID>_<ascs_sys_nr>.service
# systemctl disable SAP<SID>_<ascs_sys_nr>.service
# systemctl stop SAP<SID>_<ers_sys_nr>.service
# systemctl disable SAP<SID>_<ers_sys_nr>.service
```

e. Unmount the ERS directory (classic only):

```
# umount /usr/sap/<SID>/ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount <nfs.fqdn>:/SLX_ERS10  /usr/sap/SLX/ERS10
# export LD_LIBRARY_PATH=/usr/sap/SLX/ERS10/exe
# /usr/sap/SLX/ERS10/exe/sapstartsrv pf=/usr/sap/SLX/SYS/profile/
SLX_ERS10_slxers -reg
# systemctl start SAPSLX_10
# systemctl stop SAPSLX_00.service
# systemctl disable SAPSLX_00.service
# systemctl stop SAPSLX_10.service
# systemctl disable SAPSLX_10.service
# umount /usr/sap/SLX/ERS10
```

2. **On the instance where the ERS was installed**

Register the missing ASCS service on the node where you have installed ERS.

a. Temporarily mount the ASCS directory (classic only):

```
# mount <nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr> /usr/sap/<SID>/ASCS<ascs_sys_nr>
```

b. Register the ASCS service:

```
# export LD_LIBRARY_PATH=/usr/sap/<SID>/ASCS<ascs_sys_nr>/exe
# /usr/sap/<SID>/ASCS<ascs_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname> -reg
```

```
# systemctl start SAP<SID>_<ascs_sys_nr>
```

c. Check the existence and state of SAP services (example):

```
# systemctl list-unit-files SAP*
UNIT FILE                    STATE   VENDOR PRESET
SAPSLX00.service           disabled disabled
SAPSLX00.service           disabled disabled
SAP.slice                   static   -
3 unit files listed.
```

d. If the state is not disabled, run the following command to disable `sapservices` integration for `SAP<SID>_<ascs_sys_nr>` and `SAP<SID>_<ers_sys_nr>` on both nodes:

> **Important**
>
> Stopping these services also stops the associated SAP instances.

```
# systemctl stop SAP<SID>_<ascs_sys_nr>.service
# systemctl disable SAP<SID>_<ascs_sys_nr>.service
# systemctl stop SAP<SID>_<ers_sys_nr>.service
# systemctl disable SAP<SID>_<ers_sys_nr>.service
```

e. Unmount the ASCS directory (classic only):

```
# umount /usr/sap/<SID>/ASCS<ascs_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount <nfs.fqdn>:/SLX_ASCS00 /usr/sap/SLX/ASCS00
# export LD_LIBRARY_PATH=/usr/sap/SLX/ASCS00/exe
# /usr/sap/SLX/ASCS00/exe/sapstartsrv pf=/usr/sap/SLX/SYS/profile/
SLX_ASCS00_slxascs -reg
# systemctl start SAPSLX_00
# systemctl stop SAPSLX_00.service
# systemctl disable SAPSLX_00.service
# systemctl stop SAPSLX_10.service
# systemctl disable SAPSLX_10.service
# umount /usr/sap/SLX/ASCS00
```

Configure dependencies for Pacemaker and SAP services (systemd)

This step is required on both cluster nodes when using systemd integration.

When an EC2 instance shuts down unexpectedly, Pacemaker (the cluster resource manager) may trigger unnecessary fencing actions because it cannot distinguish between planned SAP service shutdowns and system failures. To prevent this, configure systemd dependencies that inform Pacemaker about the relationship between SAP services and cluster operations.

Create a systemd drop-in configuration for the `resource-agents-deps.target`, which is a systemd target that Pacemaker uses to understand external service dependencies:

```
# mkdir -p /etc/systemd/system/resource-agents-deps.target.d/
# cd /etc/systemd/system/resource-agents-deps.target.d/

# cat > sap_systemd_<sid>.conf <<_EOF
[Unit]
Requires=sapinit.service
After=sapinit.service
After=SAP<SID>_<ascs_sys_nr>.service
After=SAP<SID>_<ers_sys_nr>.service
_EOF

# systemctl daemon-reload
```

- *Example using values from Parameter Reference* :

  ```
  # cat > sap_systemd_slx.conf <<_EOF
  [Unit]
  Requires=sapinit.service
  After=sapinit.service
  After=SAPSLX_00.service
  After=SAPSLX_10.service
  _EOF

  # systemctl daemon-reload
  ```

(Alternative) Ensure ASCS and ERS SAP Services can run on either node (sysV)

This is only applicable for if systemd integration is not in place.

To ensure that SAP instance can be managed by the cluster and also manually during planned maintenance activities, add the missing entries for ASCS and ERS `sapstartsrv` service in `/usr/sap/sapservices` file on both cluster nodes (ASCS and ERS host). Copy the missing entry from both hosts. Post-modifications, the `/usr/sap/sapservices` file looks as follows on both hosts:

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/<SID>/ASCS<ascs_sys_nr>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<SID>/ASCS<ascs_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/
SYS/profile/<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname> -D -u <sid>adm
LD_LIBRARY_PATH=/usr/sap/<SID>/ERS<ers_sys_nr>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<SID>/ERS<ers_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/SYS/
profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname> -D -u <sid>adm
```

- *Example using values from [Parameter Reference](#)* :

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/SLX/ASCS00/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH; /
usr/sap/SLX/ASCS00/exe/sapstartsrv pf=/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs -D
 -u slxadm
LD_LIBRARY_PATH=/usr/sap/SLX/ERS10/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH; /
usr/sap/SLX/ERS10/exe/sapstartsrv pf=/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers -D -u
 slxadm
```

Cluster Node Setup

Establish cluster communication between nodes using Corosync and configure required authentication.

Topics

Change the hacluster Password

On all cluster nodes, change the password of the operating system user hacluster:

```
# passwd hacluster
```

Setup Passwordless Authentication

SUSE cluster tools provide comprehensive reporting and troubleshooting capabilities for cluster activity. Many of these tools require passwordless SSH access between nodes to collect cluster-wide information effectively. SUSE recommends configuring passwordless SSH for the root user to enable seamless cluster diagnostics and reporting.

EC2 instances typically have no root password set. Use the shared `/sapmnt` filesystem to exchange SSH keys:

On the primary node (<hostname1>):

```
# ssh-keygen -t rsa -b 4096 -f /root/.ssh/id_rsa -N ''
# cp /root/.ssh/id_rsa.pub /sapmnt/node1_key.pub
```

On the secondary node (<hostname2>):

```
# ssh-keygen -t rsa -b 4096 -f /root/.ssh/id_rsa -N ''
# cp /root/.ssh/id_rsa.pub /sapmnt/node2_key.pub
# cat /sapmnt/node1_key.pub >> /root/.ssh/authorized_keys
# chmod 600 /root/.ssh/authorized_keys
```

Back on the primary node (<hostname1>):

```
# cat /sapmnt/node2_key.pub >> /root/.ssh/authorized_keys
# chmod 600 /root/.ssh/authorized_keys
```

Test connectivity from both nodes:

```
# ssh root@<opposite_hostname> 'hostname'
```

Clean up temporary files (from either node):

```
# rm /sapmnt/node1_key.pub /sapmnt/node2_key.pub
```

An alternative is to review the SUSE Dcoumentation for Running cluster reports without root access

> **Warning**
>
> Review the security implications for your organization, including root access controls and network segmentation, before implementing this configuration.

Configure the Cluster Nodes

Initialize the cluster framework on the first node to recognise both cluster nodes.

On the primary node as root, run:

```
# crm cluster init -u -n <cluster_name> -N <hostname_1> <hostname_2>
```

Example using values from Parameter Reference :

```
# crm cluster init -u -y -n slx-sap-cluster -N slxhost01 -N slxhost02
INFO: Detected "amazon-web-services" platform
INFO: Loading "default" profile from /etc/crm/profiles.yml
INFO: "amazon-web-services" profile does not exist in /etc/crm/profiles.yml

INFO: Configuring csync2
INFO: Starting csync2.socket service on slxhost01
INFO: BEGIN csync2 checking files
INFO: END csync2 checking files
INFO: Configuring corosync (unicast)
WARNING: Not configuring SBD - STONITH will be disabled.
INFO: Hawk cluster interface is now running. To see cluster status, open:
INFO:   https://10.2.10.1:7630/
INFO: Log in with username 'hacluster'
INFO: Starting pacemaker.service on slxhost01
INFO: BEGIN Waiting for cluster
...........
INFO: END Waiting for cluster
INFO: Loading initial cluster configuration
INFO: Done (log saved to /var/log/crmsh/crmsh.log on slxhost01)
```

```
INFO: Adding node slxhost02 to cluster
INFO: Running command on slxhost02: crm cluster join -y  -c root@slxhost01
INFO: Configuring csync2
INFO: Starting csync2.socket service
INFO: BEGIN csync2 syncing files in cluster
INFO: END csync2 syncing files in cluster
INFO: Merging known_hosts
INFO: BEGIN Probing for new partitions
INFO: END Probing for new partitions
INFO: Hawk cluster interface is now running. To see cluster status, open:
INFO:   https://10.1.20.7:7630/
INFO: Log in with username 'hacluster'
INFO: Starting pacemaker.service on slxhost02
INFO: BEGIN Waiting for cluster
INFO: END Waiting for cluster
INFO: Set property "priority" in rsc_defaults to 1
INFO: BEGIN Reloading cluster configuration
INFO: END Reloading cluster configuration
INFO: Done (log saved to /var/log/crmsh/crmsh.log on slxhost02)
```

This command:

- Initializes a two-node cluster named `myCluster`
- Configures unicast communication (-u)
- Sets up the basic corosync configuration
- Automatically joins the second node to the cluster
- We do not configure SBD as an Amazon Fencing Agent will be used for STONITH in Amazon environments.
- QDevice configuration is possible but not covered in this document. Refer to SUSE Linux Enterprise High Availability Documentation - QDevice and QNetD.

Modify Generated Corosync Configuration

After initializing the cluster, the generated corosync configuration requires some modification to be optimised for cloud envrironments.

1. Edit the corosync configuration:

```
# vi /etc/corosync/corosync.conf
```

The generated file typically looks like this:

```
# Please read the corosync.conf.5 manual page
totem {
        version: 2
        cluster_name: myCluster
        clear_node_high_bit: yes
        interface {
                ringnumber: 0
                mcastport: 5405
                ttl: 1
        }

        transport: udpu
        crypto_hash: sha1
        crypto_cipher: aes256
        token: 5000     # This needs to be changed
        join: 60
        max_messages: 20
        token_retransmits_before_loss_const: 10
}

logging {
        fileline: off
        to_stderr: no
        to_logfile: yes
        logfile: /var/log/cluster/corosync.log
        to_syslog: yes
        debug: off
        timestamp: on
        logger_subsys {
                subsys: QUORUM
                debug: off
        }

}

nodelist {
    node {
        ring0_addr: <node1_primary_ip>    # Only single ring configured
        nodeid: 1
    }
    node {
        ring0_addr: <node2_primary_ip>    # Only single ring configured
        nodeid: 2
    }
}

quorum {

        # Enable and configure quorum subsystem (default: off)
        # see also corosync.conf.5 and votequorum.5
        provider: corosync_votequorum
        expected_votes: 2
        two_node: 1
}

totem {
    version: 2
    token: 5000             # This needs to be changed
    transport: udpu
    interface {
        ringnumber: 0
        mcastport: 5405
    }
}
```

2. Modify the configuration to add the second ring and optimize settings:

```
totem {
    token: 15000           # Changed from 5000 to 15000
    rrp_mode: passive      # Added for dual ring support
}

nodelist {
    node {
        ring0_addr: <node1_primary_ip>     # Primary network
        ring1_addr: <node1_secondary_ip>   # Added secondary network
        nodeid: 1
    }
    node {
        ring0_addr: <node2_primary_ip>     # Primary network
        ring1_addr: <node2_secondary_ip>   # Added secondary network
        nodeid: 2
    }
}
```

Example IP configuration:

Network Interface	Node 1	Node 2
ring0_addr	10.2.10.1	10.2.20.1
ring1_addr	10.2.10.2	10.2.20.2

3. Synchronize the modified configuration to all nodes:

```
# csync2 -xvF /etc/corosync/corosync.conf
```

4. Restart the cluster

```
# crm cluster restart
# ssh root@<hostname2> 'crm cluster restart'
```

Verify Corosync Configuration

Verify network rings are active:

```
# corosync-cfgtool -s
```

Example output:

```
Printing ring status.
Local node ID 1
RING ID 0
        id      = 10.2.10.1
        status  = ring 0 active with no faults
RING ID 1
        id      = 10.2.10.2
        status  = ring 1 active with no faults
```

Both network rings should report "active with no faults". If either ring is missing, review the corosync configuration and check that `/etc/corosync/corosync.conf` changes have been synced to the secondary node. You may need to do this manually. Restart the cluster if needed.

Configure Cluster Services

Enable pacemaker to start automatically after reboot:

```
# systemctl enable pacemaker
```

Enabling pacemaker also handles corosync through service dependencies. The cluster will start automatically after reboot. For troubleshooting scenarios, you can choose to manually start services after boot instead.

Verify Cluster Status

1. Check pacemaker service status:

```
# systemctl status pacemaker
```

2. Verify cluster status:

```
# crm_mon -1
```

Example output:

```
Cluster Summary:
  * Stack: corosync
  * Current DC: slxhost01 (version 2.1.5+20221208.a3f44794f) - partition with quorum
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ slxhost01 slxhost02 ]

Active Resources:
  * No active resources
```

Cluster Configuration

The following sections provide details on the resources, groups and constraints necessary to ensure high availability of SAP Central Services.

Topics

- Prepare for Resource Creation

Prepare for Resource Creation

To ensure that the cluster does not perform unexpected actions during setup of resources and configuration, set the maintenance mode to true.

Run the following command to put the cluster in maintenance mode:

```
# crm maintenance on
```

To verify the current maintenance state:

```
# crm status
```

Note

There are two types of maintenance mode:

- Cluster-wide maintenance (set with `crm maintenance on`)
- Node-specific maintenance (set with `crm node maintenance nodename`)

Always use cluster-wide maintenance mode when making configuration changes. For node-specific operations like hardware maintenance, refer to the Operations for proper procedures.

To disable maintenance mode after configuration is complete:

```
# crm maintenance off
```

Cluster Bootstrap

Configure Cluster Properties

Configure cluster properties to establish fencing behavior and resource failover settings:

```
# crm configure property stonith-enabled="true"
# crm configure property stonith-timeout="600"
# crm configure property priority-fencing-delay="20"
```

- The **priority-fencing-delay** is recommended for protecting the SAP ASCS nodes during network partitioning events. When a cluster partition occurs, this delay gives preference to nodes hosting higher priority resources, with the ASCS receiving additional priority weighting over the ERS . This helps ensure the ASCS node survives in split-brain scenarios. The recommended 20 second priority-fencing-delay works in conjunction with the pcmk_delay_max (10 seconds) configured in the stonith resource, providing a total potential delay of up to 30 seconds before fencing occurs

To verify your cluster property settings:

```
# crm configure show property
```

Configure Resource Defaults

Configure resource default behaviors:

```
# crm configure rsc_defaults resource-stickiness="1"
# crm configure rsc_defaults migration-threshold="3"
# crm configure rsc_defaults failure-timeout="600s"
```

- The **resource-stickiness** value of 1 encourages the ASCS resource to stay on its current node, avoiding unnecessary resource movement.
- The **migration-threshold** of causes a resource to move to a different node after 3 consecutive failures, ensuring timely failover when issues persist.

- The **failure-timeout** automatically removes a failure count after 10 minutes, preventing individual historical failures from accumulating and affecting long-term resource behavior. If testing failover scenarios in quick succession, it may be necessary to manually query and clear accumulated failure counts between tests. Use `crm resource failcount <resource_name> show <hostname>` and `crm resource refresh`.

Individual resources may override these defaults with their own defined values.

To verify your resource default settings:

```
# crm configure show rsc_defaults
```

Configure Operation Defaults

Configure operation timeout defaults:

```
# crm configure op_defaults timeout="600"
```

- The **op_defaults timeout** ensures all cluster operations have a reasonable default timeout of 600 seconds. Individual resources may override this with their own timeout values.

To verify your operation default settings:

```
# crm configure show op_defaults
```

Create STONITH (external/ec2) resource

Create the STONITH or Fencing resource using resource agent **`external/ec2`** :

```
# crm configure primitive <stonith_resource_name> stonith:external/ec2 \
params tag="<cluster_tag>" profile="<cli_cluster_profile>"
 pcmk_delay_max="<delay_value>" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="300" timeout="60"
```

Details:

- **tag** - EC2 instance tag key name that associates instances with this cluster configuration. This tag key must be unique within the Amazon account and have a value which matches the instance hostname. See the section called “Create Amazon EC2 Resource Tags Used by Amazon EC2 STONITH Agent” for EC2 instance tagging configuration.
- **profile** - (optional) Amazon CLI profile name for API authentication. Verify profile exists with `aws configure list-profiles`. If a profile is not explicitly configured the default profile will be used.
- **pcmk_delay_max** - Random delay before fencing operations. Works in conjunction with cluster property `priority-fencing-delay` to prevent simultaneous fencing. For ENSA1 use 30 seconds, for ENSA2 use 10 seconds (lower value sufficient as `priority-fencing-delay` handles primary node protection).

Example

ENSA1

Example using values from Parameter Reference :

```
# crm configure primitive res_stonith_ec2 stonith:external/ec2 \
params tag="pacemaker" profile="cluster" \
pcmk_delay_max="30" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="300" timeout="60"
```

ENSA2

Example using values from Parameter Reference :

```
# crm configure primitive res_stonith_ec2 stonith:external/ec2 \
params tag="pacemaker" profile="cluster" \
pcmk_delay_max="10" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="300" timeout="60"
```

Create Filesystem resources (classic only)

In classic configuration, the mounting and unmounting of file system resources to align with the location of the SAP services is done using cluster resources.

Create **ASCS** file system resources:

```
# crm configure primitive rsc_fs_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:Filesystem \
params \
device="<nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr>" \
directory="/usr/sap/<SID>/ASCS<ascs_sys_nr>" \
fstype="nfs4" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40"
```

Create **ERS** file system resources:

```
# crm configure primitive rsc_fs_<SID>_ERS<ers_sys_nr> ocf:heartbeat:Filesystem \
params \
device="<nfs.fqdn>:/<SID>_ERS<ers_sys_nr>" \
directory="/usr/sap/<SID>/ERS<ers_sys_nr>" \
fstype="nfs4" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40"
```

- *Example using values from Parameter Reference* :

```
# crm configure primitive rsc_fs_SLX_ASCS00 ocf:heartbeat:Filesystem \
params \
device="fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_ASCS00" \
directory="/usr/sap/SLX/ASCS00" \
fstype="nfs4" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40"

# crm configure primitive rsc_fs_SLX_ERS10 ocf:heartbeat:Filesystem \
params \
device="fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/SLX_ERS10" \
directory="/usr/sap/SLX/ERS10" \
fstype="nfs4" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40"
```

Notes

- Review the mount options to ensure that they match with your operating system, NFS file system type, and the latest recommendations from SAP.
- <nfs.fqdn> can either be an alias or the default file system resource name of the NFS or FSx for ONTAP resource. For example, `fs-xxxxxx.efs.xxxxxx.amazonaws.com`.

Create Overlay IP (aws-vpc-move-ip) resources

The IP resource provides the details necessary to update the route table entry for overlay IP.

Create **ASCS** IP Resource:

```
# crm configure primitive rsc_ip_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:aws-vpc-move-ip
 \
params \
ip="<ascs_overlayip>" \
routing_table="<routetable_id>" \
interface="eth0" \
profile="<cli_cluster_profile>" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"
```

Create **ERS** IP Resource:

```
# crm configure primitive rsc_ip_<SID>_ERS<ers_sys_nr> ocf:heartbeat:aws-vpc-move-ip \
params \
ip="<ers_overlayip>" \
routing_table="<routetable_id>" \
interface="eth0" \
profile="<cli_cluster_profile>" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"
```

- *Example using values from [Parameter Reference](#)* :

```
# crm configure primitive rsc_ip_SLX_ASCS00 ocf:heartbeat:aws-vpc-move-ip \
params \
ip="172.16.30.5" \
routing_table="rtb-xxxxxroutetable1" \
interface="eth0" \
profile="cluster" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"

# crm configure primitive rsc_ip_SLX_ERS10 ocf:heartbeat:aws-vpc-move-ip \
params \
ip="172.16.30.6" \
routing_table="rtb-xxxxxroutetable1" \
interface="eth0" \
profile="cluster" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"
```

Notes

- If more than one route table is required for connectivity or because of subnet associations, the `routing_table` parameter can have multiple values separated by a comma. For example, `routing_table=rtb-xxxxxroutetable1,rtb-xxxxxroutetable2`.
- Additional parameters – `lookup_type` and `routing_table_role` are required for shared VPC. For more information, see {https---docs-aws-amazon-com-sap-latest-sap-netweaver-sles-netweaver-ha-settings-html-sles-netweaver-ha-shared-vpc}[Shared VPC – optional].

Create SAPStartSrv resources (simple-mount only)

In simple-mount architecture, the `sapstartsrv` process that is used to control start/stop and monitoring of an SAP instance, is controlled by a cluster resource. This new resource adds additional control that removes the requirement for file system resources to be restricted to a single node.

Modify and run the commands in the table to create `sapstartsrv` resource.

Create **ASCS** SAPStartSrv Resource

Use the following command to create an ASCS SAPStartSrv resource.

```
# crm configure primitive rsc_sapstart_<SID>_ASCS<ascs_sys_nr> ocf:suse:SAPStartSrv \
params \
InstanceName=<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>
```

Create **ERS** SAPStartSrv Resource

Use the following command to create an ERS SAPStartSrv resource.

```
# crm configure primitive rsc_sapstart_<SID>_ERS<ers_sys_nr> ocf:suse:SAPStartSrv \
params  \
InstanceName=<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>
```

- *Example using values from Parameter Reference* :

  ```
  #crm configure primitive rsc_sapstart_SLX_ASCS00 ocf:suse:SAPStartSrv \
  params \
  InstanceName=SLX_ASCS00_slxascs

  #crm configure primitive rsc_sapstart_SLX_ERS10 ocf:suse:SAPStartSrv \
  params \
  InstanceName=SLX_ERS10_slxers
  ```

Create SAPInstance resources (simple-mount only)

The minor difference in creating SAP instance resources between classic and simple-mount configurations is the addition of `MINIMAL_PROBE=true` parameters.

The SAP instance is started and stopped using cluster resources.

Example

ENSA1

Create an **ASCS** SAP instance resource:

```
# crm configure primitive rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance
 \
params \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
operations \$id="rsc_sap_<SID>_ASCS<ascs_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
resource-stickiness="5000" \
failure-timeout="60" \
migration-threshold="1" \
priority="10"
```

Create an **ERS** SAP instance resource:

```
# crm configure primitive rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
params \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
IS_ERS="true" \
operations \$id="rsc_sap_<SID>_ERS<ers_sys_nr>-operations" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
priority="1000"
```

- *Example using values from Parameter Reference* :

```
# crm configure primitive rsc_sap_SLX_ASCS00 ocf:heartbeat:SAPInstance \
params \
InstanceName="SLX_ASCS00_slxascs" \
START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
operations \$id="rsc_sap_SLX_ASCS00-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
resource-stickiness="5000" \
failure-timeout="60" \
migration-threshold="1" \
priority="10"

# crm configure primitive rsc_sap_SLX_ERS10 ocf:heartbeat:SAPInstance \
params \
InstanceName="SLX_ERS10_slxers" \
START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
IS_ERS="true" \
operations \$id="rsc_sap_SLX_ERS10-operations" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
priority="1000"
```

ENSA2

Create an **ASCS** SAP instance resource:

```
# crm configure primitive rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance
 \
params \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
operations \$id="rsc_sap_<SID>_ASCS<ascs_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
resource-stickiness="5000" \
priority="1000"
```

Create an **ERS** SAP instance resource:

```
# crm configure primitive rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
params \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
IS_ERS="true" \
operations \$id="rsc_sap_<SID>_ERS<ers_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart"
```

- *Example using values from [Parameter Reference](#) :*

  ```
  # crm configure primitive rsc_sap_SLX_ASCS00 ocf:heartbeat:SAPInstance \
  params \
  InstanceName="SLX_ASCS00_slxascs" \
  START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs" \
  AUTOMATIC_RECOVER="false" \
  MINIMAL_PROBE="true" \
  operations \$id="rsc_sap_SLX_ASCS00-operations" \
  op start interval="0" timeout="600" \
  op stop interval="0" timeout="240" \
  op monitor interval="11" timeout="60" on-fail="restart" \
  meta \
  resource-stickiness="5000" \
  priority="1000"

  # crm configure primitive rsc_sap_SLX_ERS10 ocf:heartbeat:SAPInstance \
  params \
  InstanceName="SLX_ERS10_slxers" \
  ```

```
START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
IS_ERS="true" \
operations \$id="rsc_sap_SLX_ERS10-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart"
```

The difference between ENSA1 and ENSA2 is that ENSA2 allows the lock table to be consumed remotely, which means that for ENSA2, ASCS can restart in its current location (assuming the node is still available). This change impacts stickiness, migration and priority parameters. Ensure that you use the right command for your enqueue version.

Create SAPInstance resources (classic only)

The SAP instance is started and stopped using cluster resources.

Example

ENSA1

Create an **ASCS** SAPInstance resource:

```
# crm configure primitive rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance
 \
params \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
operations \$id="rsc_sap_<SID>_ASCS<ascs_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
resource-stickiness="5000" \
failure-timeout="60" \
migration-threshold="1" \
priority="10"
```

Create an **ERS** SAPInstance resource:

```
# crm configure primitive rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
params \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
operations \$id="rsc_sap_<SID>_ERS<ers_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
priority="1000"
```

- *Example using values from Parameter Reference* :

```
# crm configure primitive rsc_sap_SLX_ASCS00 ocf:heartbeat:SAPInstance \
params \
InstanceName="SLX_ASCS00_slxascs" \
START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs" \
AUTOMATIC_RECOVER="false" \
operations \$id="rsc_sap_SLX_ASCS00-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
resource-stickiness="5000" \
failure-timeout="60" \
migration-threshold="1" \
priority="10"

# crm configure primitive rsc_sap_SLX_ERS10 ocf:heartbeat:SAPInstance \
params \
InstanceName="SLX_ERS10_slxers" \
START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers" \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
operations \$id="rsc_sap_SLX_ERS10-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
priority="1000"
```

ENSA2

Create an **ASCS** SAPInstance resource:

```
# crm configure primitive rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance
 \
params \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
operations \$id="rsc_sap_<SID>_ASCS<ascs_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
resource-stickiness="5000" \
priority="1000"
```

Create an **ERS** SAPInstance resource:

```
# crm configure primitive rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
params \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
operations \$id="rsc_sap_<SID>_ERS<ers_sys_nr>-operations" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart"
```

- *Example using values from Parameter Reference* :

```
# crm configure primitive rsc_sap_SLX_ASCS00 ocf:heartbeat:SAPInstance \
params \
InstanceName="SLX_ASCS00_slxascs" \
START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ASCS00_slxascs" \
  AUTOMATIC_RECOVER="false" \
  operations \$id="rsc_sap_SLX_ASCS00-operations" \
  op start interval="0" timeout="600" \
  op stop interval="0" timeout="240" \
  op monitor interval="11" timeout="60" on-fail="restart" \
  meta \
  resource-stickiness="5000" \
  priority="1000"

  # crm configure primitive rsc_sap_SLX_ERS10 ocf:heartbeat:SAPInstance \
  params \
  InstanceName="SLX_ERS10_slxers" \
  START_PROFILE="/usr/sap/SLX/SYS/profile/SLX_ERS10_slxers" \
  AUTOMATIC_RECOVER="false" \
  IS_ERS="true" \
  operations \$id="rsc_sap_SLX_ERS10-operations" \
  op start interval="0" timeout="600" \
  op stop interval="0" timeout="240" \
  op monitor interval="11" timeout="60" on-fail="restart"
```

The change between ENSA1 and ENSA2 allows the lock table to be consumed remotely. If the node is still available, ASCS can restart in its current location for ENSA2. This impacts stickiness, migration, and priority parameters. Make sure to use the right command, depending on your enqueue server.

Create resource groups for aws-vpc-move-ip / SAPStartSrv / SAPInstance (simple-mount only)

A cluster resource group is a set of resources that need to be located together, start sequentially, and stopped in the reverse order.

In simple-mount architecture, the overlay IP must be available first, then the SAP start services are started before the SAP instance can start. The order of the group must be as defined here.

Create an **ASCS** cluster resource group:

```
# crm configure group grp_<SID>_ASCS<ascs_sys_nr> \
rsc_ip_<SID>_ASCS<ascs_sys_nr> \
rsc_sapstart_<SID>_ASCS<ascs_sys_nr> \
rsc_sap_<SID>_ASCS<ascs_sys_nr> \
meta resource-stickiness="3000"
```

Create an **ERS** cluster resource group:

```
# crm configure group grp_<SID>_ERS<ers_sys_nr> \
rsc_ip_<SID>_ERS<ers_sys_nr> \
rsc_sapstart_<SID>_ERS<ers_sys_nr> \
rsc_sap_<SID>_ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

  ```
  # crm configure group grp_SLX_ASCS00 \
  rsc_ip_SLX_ASCS00 \
  rsc_sapstart_SLX_ASCS00 \
  rsc_sap_SLX_ASCS00 \
  meta resource-stickiness="3000"

  # crm configure group grp_SLX_ERS10 \
  rsc_ip_SLX_ERS10 \
  rsc_sapstart_SLX_ERS10 \
  rsc_sap_SLX_ERS10
  ```

Create resource groups for Filesystem / aws-vpc-move-ip / SAPInstance (classic only)

A cluster resource group is a set of resources that need to be located together, start sequentially, and stopped in the reverse order.

In classic architecture, the file system is mounted first, then the overlay IP must be available before the SAP instance can start.

Create an **ASCS** cluster resource group:

```
# crm configure group grp_<SID>_ASCS<ascs_sys_nr> \
rsc_fs_<SID>_ASCS<ascs_sys_nr> \
rsc_ip_<SID>_ASCS<ascs_sys_nr> \
rsc_sap_<SID>_ASCS<ascs_sys_nr> \
meta resource-stickiness="3000"
```

Create an **ERS** cluster resource group:

```
# crm configure group grp_<SID>_ERS<ers_sys_nr> \
rsc_fs_<SID>_ERS<ers_sys_nr> \
rsc_ip_<SID>_ERS<ers_sys_nr> \
rsc_sap_<SID>_ERS<ers_sys_nr>
```

- *Example using values from [Parameter Reference](#)* :

```
# crm configure group grp_SLX_ASCS00 \
rsc_fs_SLX_ASCS00 \
rsc_ip_SLX_ASCS00 \
rsc_sap_SLX_ASCS00 \
meta resource-stickiness="3000"

# crm configure group grp_SLX_ERS10 \
rsc_fs_SLX_ERS10 \
rsc_ip_SLX_ERS10 \
rsc_sap_SLX_ERS10
```

Create resource constraints

Resource constraints are used to determine where resources run per the conditions. Constraints for SAP NetWeaver ensure that ASCS and ERS are started on separate nodes and locks are preserved in case of failures. The following are the different types of constraints.

Colocation constraint

The negative score ensures that ASCS and ERS are run on separate nodes, wherever possible.

```
# crm configure colocation col_sap_<SID>_ascs_ers_separate_nodes \
-5000: grp_<SID>_ERS<ers_sys_nr> grp_<SID>_ASCS<ascs_sys_nr>
```

- *Example using values from [Parameter Reference](#)* :

```
# crm configure colocation col_sap_SLX_ascs_ers_separate_nodes \
-5000: grp_SLX_ERS10 grp_SLX_ASCS00
```

Order constraint

This constraint ensures the ASCS instance is started prior to stopping the ERS instance. This is necessary to consume the lock table.

```
# crm configure order ord_sap_<SID>_ascs_start_before_ers_stop \
Optional: rsc_sap_<SID>_ASCS<ascs_sys_nr>:start rsc_sap_<SID>_ERS<ers_sys_nr>:stop \
symmetrical="false"
```

- *Example using values from [Parameter Reference](#)* :

```
# crm configure order ord_sap_SLX_ascs_start_before_ers_stop \
Optional: rsc_sap_SLX_ASCS00:start rsc_sap_SLX_ERS10:stop \
symmetrical="false"
```

Location constraint (ENSA1 only)

This constraint is only required for ENSA1. The lock table can be retrieved remotely for ENSA2, and as a result ASCS doesn't failover to where ERS is running.

```
# crm configure location loc_sap_<SID>_ascs_follows_ers \
rsc_sap_<SID>_ASCS<ascs_sys_nr> rule 2000: runs_ers_<SID> eq 1
```

- *Example using values from [Parameter Reference](#)* :

```
# crm configure location loc_sap_SLX_ascs_follows_ers \
rsc_sap_SLX_ASCS00 rule 2000: runs_ers_SLX eq 1
```

Reset Configuration – Optional

> ⚠ **Important**
>
> The following instructions help you reset the complete configuration. Run these commands only if you want to start setup from the beginning. You can make minor changes with the crm edit command.

Run the following command to back up the current configuration for reference:

```
# crm config show > /tmp/crmconfig_backup.txt
```

Run the following command to clear the current configuration:

```
# crm configure erase
```

Once the preceding erase command is executed, it removes all of the cluster resources from Cluster Information Base (CIB), and disconnects the communication from corosync to the cluster.

Before starting the resource configuration run crm cluster restart, so that cluster reestablishes communication with corosync, and retrieves the configuration. The restart of cluster removes maintenance mode. Reapply before commencing additional configuration and resource setup.

Operations

This section covers the following topics.

Topics

- Viewing the cluster state
- Performing planned maintenance
- Post-failure analysis and reset
- Alerting and monitoring

Viewing the cluster state

You can view the state of the cluster in two ways - based on your operating system or with a web based console provided by SUSE.

Topics

- Operating system based
- SUSE Hawk2

Operating system based

There are multiple operating system commands that can be run as root or as a user with appropriate permissions. The commands enable you to get an overview of the status of the cluster and its services. See the following commands for more details.

```
# crm status
```

Sample output:

```
slxhost01:~ # crm status
Cluster Summary:
  * Stack: corosync
  * Current DC: slxhost01 (version
 2.0.5+20201202.ba59be712-150300.4.24.1-2.0.5+20201202.ba59be712) - partition with
 quorum
  * Last updated: Tue Nov  1 13:41:58 2022
  * Last change:  Fri Oct 28 08:55:43 2022 by root via crm_attribute on slxhost02
  * 2 nodes configured
  * 7 resource instances configured

Node List:
  * Online: [ slxhost01 slxhost02 ]

Full List of Resources:
  * Resource Group: grp_SLX_ASCS00:
    * rsc_ip_SLX_ASCS00 (ocf::heartbeat:aws-vpc-move-ip):        Started slxhost01
    * rsc_sapstart_SLX_ASCS00     (ocf::suse:SAPStartSrv):         Started slxhost01
    * rsc_sap_SLX_ASCS00            (ocf::heartbeat:SAPInstance):    Started slxhost01
  * res_AWS_STONITH     (stonith:external/ec2):  Started slxhost02
  * Resource Group: grp_SLX_ERS10:
    * rsc_ip_SLX_ERS10  (ocf::heartbeat:aws-vpc-move-ip):        Started slxhost02
    * rsc_sapstart_SLX_ERS10      (ocf::suse:SAPStartSrv):         Started slxhost02
    * rsc_sap_SLX_ERS10 (ocf::heartbeat:SAPInstance):    Started slxhost02
```

The following table provides a list of useful commands.

Command	Description
`crm_mon`	Display cluster status on the console with updates as they occur
`crm_mon -1`	Display cluster status on the console just once, and exit
`crm_mon -Arnf`	-A Display node attributes -n Group resources by node -r Display inactive resources -f Display resource fail counts
`crm help`	View more options

Command	Description
`crm_mon --help-all`	View more options

SUSE Hawk2

Hawk2 is a web-based graphical user interface for managing and monitoring pacemaker highly availability clusters. It must be enabled on every node in the cluster, to point your web browser on any node for accessing it. Use the following command to enable Hawk2.

```
# systemctl enable --now hawk
# systemctl status hawk
```

Use the following URL to check security groups for access on port 7630 from your administrative host.

```
https://your-server:7630/

e.g https://slxhost01:7630
```

For more information, see Configuring and Managing Cluster Resources with Hawk2 in the SUSE Documentation.

Performing planned maintenance

The cluster connector is designed to integrate the cluster with SAP start framework (`sapstartsrv`), including the rolling kernel switch (RKS) awareness. Stopping and starting the SAP system using `sapcontrol` should not result in any cluster remediation activities as these actions are not interpreted as failures. Validate this scenario when testing your cluster.

There are different options to perform planned maintenance on nodes, resources, and the cluster.

Topics

Maintenance mode

Use maintenance mode if you want to make any changes to the configuration or take control of the resources and nodes in the cluster. In most cases, this is the safest option for administrative tasks.

Example

On

Use one of the following commands to turn on maintenance mode.

```
# crm maintenance on
```

```
# crm configure property maintenance-mode="true"
```

Off

Use one of the following commands to turn off maintenance mode.

```
# crm maintenance off
```

```
# crm configure property maintenance-mode="false"
```

Placing a node in standby mode

To perform maintenance on the cluster without system outage, the recommended method for moving active resources is to place the node you want to remove from the cluster in standby mode.

```
# crm node standby <hostname>
```

The cluster will cleanly relocate resources, and you can perform activities, including reboots on the node in standby mode. When maintenance activities are complete, you can re-introduce the node with the following command.

```
# crm node online <hostname>
```

Moving a resource

Moving individual resources is not recommended because of the migration or move constraints that are created to lock the resource in its new location. These can be cleared as described in the info messages, but this introduces an additional setup.

```
<slxhost01>:~ crm resource move grp_<SLX>_ASCS<00> <slxhost02>
INFO: Move constraint created for grp_<SLX>_ASCS<00> to <slxhost02>
INFO: Use `crm resource clear grp_<SLX>_ASCS<00>` to remove this constraint
```

Use the following command once the resources have relocated to their target location.

```
# crm resource clear grp_SLX_ASCS00
```

Post-failure analysis and reset

A review must be conducted after each failure to understand the source of failure as well the reaction of the cluster. In most scenarios, the cluster prevents an application outage. However, a manual action is often required to reset the cluster to a protective state for any subsequent failures.

Topics

- Checking the logs
- Cleanup crm status
- Restart failed nodes or pacemaker
- Further Analysis

Checking the logs

- For troubleshooting cluster issues, use journalctl to examine both pacemaker and corosync logs:

  ```
  # journalctl -u pacemaker -u corosync --since "1 hour ago"
  ```

  - Use `--since` to specify time periods (e.g., "2 hours ago", "today")
  - Add `-f` to follow logs in real-time
  - Combine with grep for specific searches
- System messages and resource agent activity can be found in `/var/log/messages`.

Application based failures can be investigated in the SAP work directory.

Cleanup crm status

If failed actions are reported using the `crm status` command, and if they have already been investigated, then you can clear the reports with the following command.

```
# crm resource cleanup <resource> <hostname>
```

Restart failed nodes or pacemaker

It is recommended that failed (or fenced) nodes are not automatically restarted. It gives operators a chance to investigate the failure, and ensure that the cluster doesn't make assumptions about the state of resources.

You need to restart the instance or the pacemaker service based on your approach.

Further Analysis

For cluster-specific issues, use `hb_report` to generate a targeted analysis of cluster components across all nodes:

```
# hb_report -f "YYYY-MM-DD HH:MM:SS" -t "YYYY-MM-DD HH:MM:SS" /tmp/hb_report
```

For quick analysis of recent events, you can use:

```
# crm history events
# crm history log
```

- Both `hb_report` and `crm history` commands require passwordless SSH between nodes
- For more information, see SUSE Documentation - Usage of hb_report for SLES HAE

Alerting and monitoring

This section covers the following topics.

Topics

- Using Amazon CloudWatch Application Insights
- Using the cluster alert agents

Using Amazon CloudWatch Application Insights

For monitoring and visibility of cluster state and actions, Application Insights includes metrics for monitoring enqueue replication state, cluster metrics, and SAP and high availability checks. Additional metrics, such as EFS and CPU monitoring can also help with root cause analysis.

For more information, see Get started with Amazon CloudWatch Application Insights and SAP NetWeaver High Availability on Amazon EC2.

Using the cluster alert agents

Within the cluster configuration, you can call an external program (an alert agent) to handle alerts. This is a *push* notification. It passes information about the event via environment variables.

The agents can then be configured to send emails, log to a file, update a monitoring system, etc. For example, the following script can be used to access Amazon SNS.

```
#!/bin/sh

# alert_sns.sh
# modified from /usr/share/pacemaker/alerts/alert_smtp.sh.sample

##############################################################################
# SETUP
# * Create an SNS Topic and subscribe email or chatbot
# * Note down the ARN for the SNS topic
# * Give the IAM Role attached to both Instances permission to publish to the SNS Topic
# * Ensure the aws cli is installed
# * Copy this file to /usr/share/pacemaker/alerts/alert_sns.sh or other location on
 BOTH nodes
# * Ensure the permissions allow for hacluster and root to execute the script
# * Run the following as root (modify file location if necessary and replace SNS ARN):
#
# SLES:
# crm configure alert aws_sns_alert /usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S" to <{ arn:aws:sns:region:account-
id:myPacemakerAlerts  }>
#
# RHEL:
# pcs alert create id=aws_sns_alert path=/usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S"
# pcs alert recipient add aws_sns_alert value=arn:aws:sns:region:account-
id:myPacemakerAlerts
```

```
################################################################################

# Additional information to send with the alerts
node_name=`uname -n`
sns_body=`env | grep CRM_alert_`

# Required for SNS
TOKEN=$(/usr/bin/curl --noproxy '*' -s -X PUT "http://169.254.169.254/latest/api/token"
 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Get metadata
REGION=$(/usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $TOKEN"
 http://169.254.169.254/latest/dynamic/instance-identity/document | grep region | awk -
F\" '{print $4}')

sns_subscription_arn=${CRM_alert_recipient}

# Format depending on alert type
case ${CRM_alert_kind} in
   node)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Node '${CRM_alert_node}' is
 now '${CRM_alert_desc}'"
   ;;
   fencing)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Fencing ${CRM_alert_desc}"
   ;;
   resource)
     if [ ${CRM_alert_interval} = "0" ]; then
         CRM_alert_interval=""
     else
         CRM_alert_interval=" (${CRM_alert_interval})"
     fi
     if [ ${CRM_alert_target_rc} = "0" ]; then
         CRM_alert_target_rc=""
     else
         CRM_alert_target_rc=" (target: ${CRM_alert_target_rc})"
     fi
     case ${CRM_alert_desc} in
         Cancelled)
           ;;
         *)
           sns_subject="${CRM_alert_timestamp}: Resource operation
 '${CRM_alert_task}${CRM_alert_interval}' for '${CRM_alert_rsc}' on
 '${CRM_alert_node}': ${CRM_alert_desc}${CRM_alert_target_rc}"
```

```
           ;;
     esac
     ;;
   attribute)
     sns_subject="${CRM_alert_timestamp}: The '${CRM_alert_attribute_name}' attribute
 of the '${CRM_alert_node}' node was updated in '${CRM_alert_attribute_value}'"
     ;;
   *)
     sns_subject="${CRM_alert_timestamp}: Unhandled $CRM_alert_kind alert"
     ;;
esac

# Use this information to send the email.
aws sns publish --topic-arn "${sns_subscription_arn}" --subject "${sns_subject}" --
message "${sns_body}" --region ${REGION}
```

Testing

We recommend scheduling regular fault scenario recovery testing at least annually, and as part of the operating system or SAP kernel updates that may impact operations. For more details on best practices for regular testing, see SAP Lens – Best Practice 4.3 – Regularly test business continuity plans and fault recovery.

The tests described here simulate failures. These can help you understand the behavior and operational requirements of your cluster.

In addition to checking the state of cluster resources, ensure that the service you are trying to protect is in the required state. Can you still connect to SAP? Are locks still available in SM12?

Define the recovery time to ensure that it aligns with your business objectives. Record recovery actions in runbooks.

Topics

- Test 1: Stop ASCS on the primary node using sapcontrol
- Test 2: Stop ERS on the secondary node using sapcontrol
- Test 3: Kill the message server process on the primary node
- Test 4: Kill the enqueue server process on the primary node
- Test 5: Kill the ER process
- Test 6: Simulate hardware failure of an individual node, and repeat for other node

- Test 7: Simulate a network failure
- Test 8: Simulate an NFS failure
- Test 9: Accidental shutdown

Test 1: Stop ASCS on the primary node using `sapcontrol`

Notes – Ensure that the connector has been installed and the parameters have been updated.

Simulate failure – On `slxhost01` as `slxadm`:

```
sapcontrol -nr <00> -function Stop
```

Expected behavior – ASCS should be stopped on `slxhost01`, and the cluster should not perform any activity.

Recovery action – Start ASCS manually.

Test 2: Stop ERS on the secondary node using `sapcontrol`

Notes – Ensure that the connector has been installed, and the parameters are updated.

Simulate failure – On `slxhost02` as `slxadm`:

```
sapcontrol -nr <10> -function Stop
```

Expected behavior – ERS should be stopped on `slxhost02`, and the cluster should not perform any activity.

Recovery action – Start ERS manually.

Test 3: Kill the message server process on the primary node

Simulate failure – On `slxhost01` as `slxadm`:

```
kill -9 $(pgrep -f "ms.sap<SLX>_ASCS<00>")
```

Expected behavior – The message server should immediately respawn based on the Restart parameter.

Recovery action – No action required.

Test 4: Kill the enqueue server process on the primary node

Notes – Check that locks have persisted, and review the location constraints that only exist for ENSA1.

Simulate failure – On `slxhost01` as `slxadm`:

```
kill -9 $(pgrep -f "[en|enq].sap<SLX>_ASCS<00>")
```

Expected behavior – ENSA2: Cluster will restart the ENQ process and retrieve the locks remotely. ENSA1: Cluster will failover the ASCS resource to the node where the ERS is running.

Recovery action – No action required.

Test 5: Kill the ER process

Simulate failure – On `slxhost02` as `slxadm`:

```
kill -9 $(pgrep -f "[er|enqr].sap<SLX>_ERS<10>")
```

Expected behavior – Cluster will restart the ERS on the same node.

Recovery action – No action required.

Test 6: Simulate hardware failure of an individual node, and repeat for other node

Notes – To simulate a system crash, you must first ensure that `/proc/sys/kernel/sysrq` is set to 1.

Simulate failure – On both nodes as root:

```
echo 'b' > /proc/sysrq-trigger
```

Expected behavior – The node which has been killed fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 7: Simulate a network failure

Notes – See the following list.

- Iptables must be installed.
- Use a subnet in this command because of the secondary ring.
- Check for any existing iptables rules as iptables -F will flush all rules.
- Review pcmk_delay and priority parameters if you see neither node survives the fence race.

Simulate failure – On either node as root:

```
iptables -A INPUT -s <CIDR_of_other_subnet> -j DROP; iptables -A OUTPUT -d
 <CIDR_of_other_subnet> -j DROP
```

Expected behavior – The cluster detects the network failure, and fences one of the nodes to avoid a split-brain situation.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 8: Simulate an NFS failure

Notes – See the following list.

- Iptables must be installed.
- Check for any existing iptables rules as iptables -F will flush all rules.
- Although rare, this is an important scenario to test. Depending on the activity it may take some time (10 min +) to notice that I/O to EFS is not occurring and fail either the Filesystem or SAP resources.

Simulate failure – On either node as root:

```
iptables -A OUTPUT -p tcp --dport 2049 -m state --state NEW,ESTABLISHED,RELATED -j
 DROP; iptables -A INPUT -p tcp --sport 2049 -m state --state ESTABLISHED -j DROP
```

Expected behavior – The cluster detects that NFS is not available, and the SAP Instance resource agent will fail and move to the FAILED state. Because of the option "on-fail=restart" configuration, the cluster will try a local restart before eventually fencing the node and failing over.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 9: Accidental shutdown

Notes – See the following list.

- Avoid shutdowns without cluster awareness.
- We recommend the use of systemd to ensure predictable behaviour.
- Ensure the resource dependencies are in place.

Simulate failure – Login to Amazon Management Console, and stop the instance or issue a shutdown command.

Expected behavior – The node which has been shut down fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node. If systemd and resource dependencies are not configured, you may notice that while the EC2 instance is shutting down gracefully, the cluster will detect an unclean stop of cluster services on the node and will fence the EC2 instance being shut down. For more information, see SUSE documentation – Stopping the Cluster Services on a Node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online, and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

SAP NetWeaver on Amazon: high availability configuration for Red Hat Enterprise Linux (RHEL) for SAP applications

This topic applies to Red Hat Enterprise Linux (RHEL) operating system for SAP NetWeaver applications on Amazon cloud. It covers the instructions for configuration of a pacemaker cluster for the ABAP SAP Central Service (ASCS) and the Enqueue Replication Server (ERS) when deployed on Amazon EC2 instances in two different Availability Zones within an Amazon Region.

This topic covers instructions for the following configuration options.

Topics

- Planning
- Prerequisites
- SAP ASCS and Cluster Setup
- Operations
- Testing

Planning

This section covers the following topics.

Topics

- Setup Overview
- Vendor Support
- Concepts
- Parameter Reference
- Architecture diagrams

Setup Overview

You must meet the following prerequisites before commencing setup.

Topics

- Deployed Cluster Infrastructure
- Supported Operating System
- SAP and Red Hat references
- Required Access for Setup
- Reliability Requirements Defined

Deployed Cluster Infrastructure

Ensure that your Amazon networking requirements and Amazon EC2 instances where SAP workloads are installed, are correctly configured for SAP. For more information, see SAP NetWeaver Environment Setup for Linux on Amazon.

See the following ASCS cluster specific requirements.

- Two cluster nodes created in private subnets in separate Availability Zones within the same Amazon VPC and Amazon Region
- Access to the route table(s) that are associated with the chosen subnets

  For more information, see Amazon – Overlay IP.
- Amazon EC2 instances must have connectivity to the Amazon EC2 endpoint via either internet or an Amazon VPC endpoint.

Supported Operating System

Protecting the ABAP SAP Central Services (ASCS) with a pacemaker cluster requires packages from Red Hat, including targeted cluster resource agents for SAP and Amazon that may not be available in standard repositories.

For deploying SAP applications on Red Hat, SAP and Red Hat recommend using Red Hat Enterprise Linux for SAP Solutions (RHEL for SAP). RHEL for SAP provides additional benefits, including Extended Update Support (EUS), configuration and tuning packages for SAP applications, and High Availability Add-On. For more details, see Red Hat website at Red Hat Enterprise Linux for SAP Solutions.

RHEL for SAP is available at Amazon Marketplace with an hourly or annual subscription. You can also use the bring your own subscription (BYOS) model.

SAP and Red Hat references

In addition to this guide, see the following references for more details.

RHEL 9 Documentation (Recommended):

- Red Hat documentation – Deploying SAP NetWeaver or S/4HANA Application Server High Availability with Simple Mount (RHEL 9)

- Red Hat documentation – Configuring HA clusters to manage SAP NetWeaver or SAP S/4HANA Application server instances using the RHEL HA Add-On (RHEL 9)
- SAP Note: 3108316 - Red Hat Enterprise Linux 9.x: Installation and Configuration

RHEL 8 Documentation:

- Red Hat documentation – Configuring HA clusters to manage SAP NetWeaver or SAP S/4HANA Application server instances using the RHEL HA Add-On (RHEL 8)
- SAP Note: 2772999 - Red Hat Enterprise Linux 8.x: Installation and Configuration
- SAP Note: 2777782 - SAP HANA DB: Recommended OS settings for RHEL 8

RHEL 7 Documentation (Extended Life Phase - Not recommended for new installations):

- Red Hat documentation – RHEL Guidelines for Configuring SAP S/4HANA ASCS/ERS with Standalone Enqueue Server 2 (ENSA2) in Pacemaker (RHEL 7)
- SAP Note: 2002167 - Red Hat Enterprise Linux 7.x: Installation and Upgrade

General SAP Notes:

- SAP Note: 1656099 - SAP Applications on Amazon: Supported DB/OS and Amazon EC2 products

You must have SAP portal access for reading all SAP Notes.

Required Access for Setup

The following access is required for setting up the cluster.

- An IAM user with the following privileges.
  - modify Amazon VPC route tables
  - modify Amazon EC2 instance properties
  - create IAM policies and roles
  - create Amazon EFS file systems
- Root access to the operating system of both cluster nodes
- SAP administrative user access – `<sid>adm`

  In case of a new install, this user is created by the install process.

Reliability Requirements Defined

The SAP Lens of the Well-Architected framework, in particular the Reliability pillar, can be used to understand the reliability requirements for your SAP workload.

The ASCS is a single point of failure in a highly available SAP architecture. The impact of an outage of this component must be evaluated against factors, such as, recovery point objective (RPO), recovery time objective (RTO), cost and operation complexity. For more information, see Reliability in SAP Lens - Amazon Well-Architected Framework.

Vendor Support

The following section details the documentation and deployment guidance from Red Hat.

Topics

- Deployment Patterns
- Automated Deployment
- Pacemaker - simple-mount and classic architecture

Deployment Patterns

The following table outlines the supported SAP deployment types and their corresponding Amazon configuration patterns for high availability clustering.

SAP Deployment Type	Support Status	Amazon Configuration Patterns	Notes
SAP NetWeaver ASCS/ ERS (ENSA1)	Amazon Documented & Supported	SAPNetweaver-Classic, SAPNetweaver-Simple-mount	
SAP NetWeaver ASCS/ ERS (ENSA2)	Amazon Documented & Supported	SAPNetweaver-Classic, SAPNetweaver-Simple-mount	

SAP Deployment Type	Support Status	Amazon Configuration Patterns	Notes
SAP S/4HANA ASCS/ERS	Amazon Documented & Supported	SAPNetweaver-Classic, SAPNetweaver-Simple-mount	S/4HANA only supports ERS2
SAP SCS (Java)	Vendor Documented & Supported		Follows SAP Documentation

Automated Deployment

You can set up a cluster manually using the instructions provided here. You can also automate parts of this process to ensure consistency and repeatability.

Use Amazon Launch Wizard for SAP for automated deployments of SAP NetWeaver, SAP S/4 HANA, SAP B/4HANA, and Solution Manager. Launch Wizard uses Amazon CloudFormation scripts to quickly provision the resources needed to deploy SAP NetWeaver and S/4 HANA. The automation performs SAP enqueue replication and pacemaker setup so that only validation and testing are required. For more information, see Amazon Launch Wizard for SAP.

To ensure that the behavior and operation of your cluster is well understood regardless of how your system is set up, we recommend a thorough test cycle. See Testing for more details.

Pacemaker - simple-mount and classic architecture

This guide covers two architectures for SAP cluster solutions on RHEL for SAP – simple-mount and classic (previous standard). Simple-mount was certified as the RHEL for SAP Applications cluster solution in 2025. It is now the recommended architecture for both ENSA1 and ENSA2 deployments running on RHEL for SAP 9 and above. For more details, see Red Hat documentation – Deploying SAP NetWeaver or S/4HANA Application Server High Availability with Simple Mount.

If you are configuring a new SAP installation, we recommend the simple-mount architecture. If you already have the classic architecture, and wish to migrate to the simple-mount architecture, see Switching architecture to simple-mount.

The following are the differences between the classic and simple-mount architectures.

- Removing file system resources from cluster – a file system is required but it is not mounted and unmounted by the cluster. The executable directory for the ASCS and ERS can be permanently mounted on both nodes.
- Addition of SAPStartSrv – SAPStartSrv controls the matching SAPStartSrv framework process.
- Sapping and sappong services – these services manage the start of SAPStartSrv services with sapinit.

See the Architecture diagrams for more details.

Switching architecture to simple-mount

TODO - Review and update internal links.

Follow along these steps if you want to switch an existing cluster with classic architecture to use the recommended configuration of simple-mount architecture.

These steps must be performed in an outage window, allowing stop/start of services and basic testing.

1. Put the cluster in maintenance mode. See the section called “Maintenance mode”
2. Stop SAP services, including application servers connected to the cluster as well as ASCS and ERS.
3. Install any missing operating system packages. See the section called “Install Missing Operating System Packages”.

   It might be necessary to install `sapstartsrv-resource-agents`. However, all operating system prerequisites must be checked and updated to ensure that versions are compatible.
4. Add entries for ASCS and ERS mount point on both nodes (if not already added). See the section called “Update /etc/fstab”
5. Enable `sapping`/`sappong` services. See the section called “Enable sapping and sappong Services (Simple-Mount Only)”
6. Align and disable `systemd` services. See the section called “Ensure ASCS and ERS SAP Services can run on either node (systemd)”
7. Backup the configuration with the following command.

```
# pcs config show >> /tmp/classic_ha_setup.txt
```

See the section called “Prepare for Resource Creation”

8. *Optional* – delete the configuration. You can edit in place but we recommend starting with a blank configuration. This ensures that latest timeout and priority parameters are in place. See Reset Configuration

```
# pcs resource cleanup
# pcs config show
```

9. Configure cluster resources again.
10. Check the cluster and perform some tests.
11. Resume standard operations by starting any additional services, including application servers.

Concepts

This section covers Amazon, SAP, and Red Hat concepts.

Topics

- SAP – ABAP SAP Central Services (ASCS)
- SAP – Enqueue Replication Server (ERS)
- Amazon – Availability Zones
- Amazon – Overlay IP
- Amazon – Shared VPC
- Pacemaker - STONITH fencing agent

SAP – ABAP SAP Central Services (ASCS)

The ABAP SAP Central Services (ASCS) is an SAP instance consisting of the following two services. It is considered a single point of failure (SPOF) in a resilient SAP architecture.

- **Message server** – Responsible for application load distribution (GUI and RFC), communication between application servers, and centralised configuration information for web dispatchers and application servers.

- **Enqueue server (standalone)** – Maintains a lock table in main memory (shared memory). Unlike a database lock, an enqueue lock can exist across multiple logical units of work (LUW), and is set by a SAP Dialog work process. The lock mechanism prevents two transactions from changing the same data in the database simultaneously.

> **Note**
>
> With ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. It replaces the previous version (ENSA1) but can be configured for the previous versions. See SAP Note 2630416 - Support for Standalone Enqueue Server 2 (SAP portal access required) for more information.
> This document includes modifications to align with the correct ENSA version.

SAP – Enqueue Replication Server (ERS)

The Enqueue Replication Server (ERS) is an SAP instance containing a replica of the lock table (replication table).

In a resilient setup, if the standalone enqueue server (EN/ENQ) fails, it can be restarted either by restart parameters or by high availability software, such as Pacemaker. The enqueue server retrieves the replication table remotely or by failing over to the host where the ERS is running.

Amazon – Availability Zones

Availability Zone is one or more discreet data centers with redundant power, networking, and connectivity in an Amazon Region. For more information, see Regions and Availability Zones.

For mission critical deployments of SAP on Amazon where the goal is to minimise the recovery time objective (RTO), we suggest distributing single points of failure across Availability Zones. Compared with single instance or single Availability Zone deployments, this increases resilience and isolation against a broad range of failure scenarios and issues, including natural disasters.

Each Availability Zone is physically separated by a meaningful distance (many kilometers) from another Availability Zone. All Availability Zones in an Amazon Region are interconnected with high-bandwidth, low-latency network, over fully redundant, dedicated metro fiber. This enables synchronous replication. All traffic between Availability Zones is encrypted.

Amazon – Overlay IP

Overlay IP enables a connection to the application, regardless of which Availability Zone (and subnet) contains the active primary node.

When deploying an Amazon EC2 instance in Amazon, IP addresses are allocated from the CIDR range of the allocated subnet. The subnet cannot span across multiple Availability Zones, and therefore the subnet IP addresses may be unavailable after faults, including network connectivity or hardware issues which require a failover to the replication target in a different Availability Zone.

To address this, we suggest that you configure an overlay IP, and use this in the connection parameters for the application. This IP address is a non-overlapping RFC1918 private IP address from outside of VPC CIDR block and is configured as an entry in the route table or tables. The route directs the connection to the active node and is updated during a failover by the cluster software.

You can select any one of the following RFC1918 private IP addresses for your overlay IP address.

- 10.0.0.0 – 10.255.255.255 (10/8 prefix)
- 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

If, for example, you use the 10/8 prefix in your SAP VPC, selecting a 172 or a 192 IP address may help to differentiate the overlay IP. Consider the use of an IP Address Management (IPAM) tool such as Amazon VPC IP Address Manager to plan, track, and monitor IP addresses for your Amazon workloads. For more information, see What is IPAM?

The overlay IP agent in the cluster can also be configured to update multiple route tables which contain the Overlay IP entry if your subnet association or connectivity requires it.

Access to overlay IP

The overlay IP is outside of the range of the VPC, and therefore cannot be reached from locations that are not associated with the route table, including on-premises and other VPCs.

Use Amazon Transit Gateway as a central hub to facilitate the network connection to an overlay IP address from multiple locations, including Amazon VPCs, other Amazon Regions, and on-premises using Amazon Direct Connect or Amazon Client VPN.

If you do not have Amazon Transit Gateway set up as a network transit hub or if it is not available in your preferred Amazon Region, you can use a Network Load Balancer to enable network access to an overlay IP.

For more information, see SAP on Amazon High Availability with Overlay IP Address Routing.

Amazon – Shared VPC

An enterprise landing zone setup or security requirements may require the use of a separate cluster account to restrict the route table access required for the Overlay IP to an isolated account. For more information, see Share your VPC with other accounts.

Evaluate the operational impact against your security posture before setting up shared VPC. To set up, see Shared VPC – optional.

Pacemaker - STONITH fencing agent

In a two-node cluster setup for a primary resource and its replication pair, it is important that there is only one node in the primary role with the ability to modify your data. In the event of a failure scenario where a node is unresponsive or incommunicable, ensuring data consistency requires that the faulty node is isolated by powering it down before the cluster commences other actions, such as promoting a new primary. This arbitration is the role of the fencing agent.

Since a two-node cluster introduces the possibility of a fence race in which a dual shoot out can occur with communication failures resulting in both nodes simultaneously claiming, "I can’t see you, so I am going to power you off". The fencing agent is designed to minimise this risk by providing an external witness.

RHEL supports several fencing agents, including the one recommended for use with Amazon EC2 Instances (fence_aws). This resource uses API commands to check its own instance status - "Is my instance state anything other than running?" before proceeding to power off its pair. If it is already in a stopping or stopped state it will admit defeat and leave the surviving node untouched.

Parameter Reference

The cluster setup relies on the following parameters.

Topics

- Global Amazon parameters
- Amazon EC2 instance parameters

- SAP Instance Parameters
- Pacemaker Parameters

Global Amazon parameters

Name	Parameter	Example
Amazon account ID	`<account_id>`	`123456789100`
Amazon Region	`<region_id>`	`us-east-1`

- Amazon account – For more details, see Your Amazon account ID and its alias.
- Amazon Region – For more details, see Describe your Regions.

Amazon EC2 instance parameters

Name	Parameter	Host 1	Host 2
Amazon EC2 instance ID	`<instance_id>`	`i-xxxxinstidforhost1`	`i-xxxxinstidforhost2`
Hostname	`<hostname>`	`rhxhost01`	`rhxhost02`
Host IP	`<host_ip>`	`10.1.10.1`	`10.1.20.1`
Host additional IP	`<host_additional_ip>`	`10.1.10.2`	`10.1.20.2`
Configured subnet	`<subnet_id>`	`subnet-xxxxxxxxxxsubnet1`	`subnet-xxxxxxxxxxsubnet2`
Associated VPC Route Table(s)	`<routetable_id>`	`rtb-xxxxxroutetable1 [,rtb-xxxxxroutetable2]`	

Name	Parameter	Host 1	Host 2
Sapmnt NFS ID or CNAME	<sapmnt_nfs_id>	fs-xxxxxx xxxxxxxefs1	

- Hostname – Hostnames must comply with SAP requirements outlined in SAP Note 611361 - Hostnames of SAP ABAP Platform servers (requires SAP portal access).

  Run the following command on your instances to retrieve the hostname.

  ```
  # hostname
  ```

- **Amazon EC2 instance ID** – run the following command (IMDSv2 compatible) on your instances to retrieve instance metadata.

  ```
  # /usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $(curl --
  noproxy '*' -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
  metadata-token-ttl-seconds: 21600")" http://169.254.169.254/latest/meta-data/
  instance-id
  ```

  For more details, see Retrieve instance metadata and Instance identity documents.

- **Amazon EC2 subnet ID** – run the following command to retrieve the subnet ID for each of your instances.

  ```
  # INSTANCE_ID=i-xxxxinstidforhost1
  # aws ec2 describe-instances --instance-ids $INSTANCE_ID --query
   'Reservations[0].Instances[0].SubnetId' --output text
  ```

  For more details, see describe-instances and VPC subnets.

- **Route table(s) for subnets** – run the following Amazon CLI commands to retrieve the route table(s) associated with both cluster node subnets.

  ```
  # SUBNET_ID_1=subnet-xxxxxxxxxxsubnet1
  # SUBNET_ID_2=subnet-xxxxxxxxxxsubnet2
  # aws ec2 describe-route-tables --filters "Name=association.subnet-id,Values=
  $SUBNET_ID_1,$SUBNET_ID_2" --query 'RouteTables[].RouteTableId' --output text
  ```

If both cluster nodes are in subnets associated with the same route table, only one route table ID will be returned. If they are associated with different route tables, both route table IDs will be returned.

For more details, see describe-route-tables and Route tables.

SAP Instance Parameters

Name	Parameter	Example
SID	`<SID>` or `<sid>`	RHX
ASCS Alias	`<ascs_virt_hostname>`	`rhxascs`
ASCS System Number	`<ascs_sys_nr>`	`00`
ASCS Overlay IP	`<ascs_oip>`	`172.16.1.50`
ASCS NFS Mount Point	`<ascs_nfs_mount_point>`	`/RHX_ASCS00`
ERS Alias	`<ers_virt_hostname>`	`rhxers`
ERS System Number	`<ers_sys_nr>`	`10`
ERS Overlay IP	`<ers_oip>`	`172.16.1.51`
ERS NFS Mount Point	`<ers_nfs_mount_point>`	`/RHX_ERS10`
ENSA Type	`<ensa_type>`	ENSA2

Pacemaker Parameters

Name	Parameter	Example
Cluster user	`cluster_user`	`hacluster`

Name	Parameter	Example
Cluster password	`cluster_password`	
Cluster tag	`cluster_tag`	`pacemaker`
Amazon CLI cluster profile	`aws_cli_cluster_profile`	`cluster`
Cluster connector	`cluster_connector`	`sap-redhat-cluster-connector`

Architecture diagrams

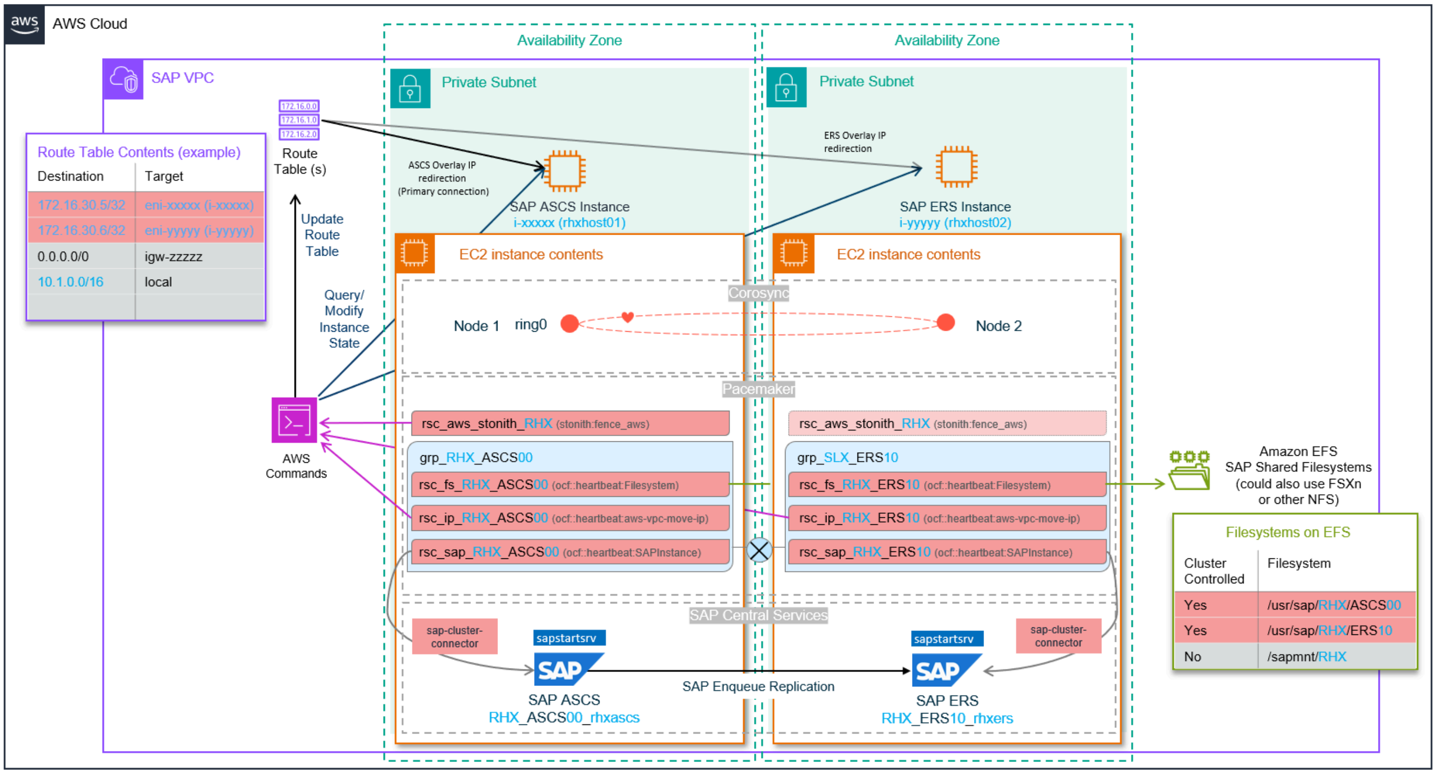
This guide covers two architectures for SAP cluster solutions on RHEL for SAP – simple-mount and classic (previous standard). See the following images to learn more.

Topics

- Pacemaker - simple-mount architecture
- Pacemaker - classic architecture

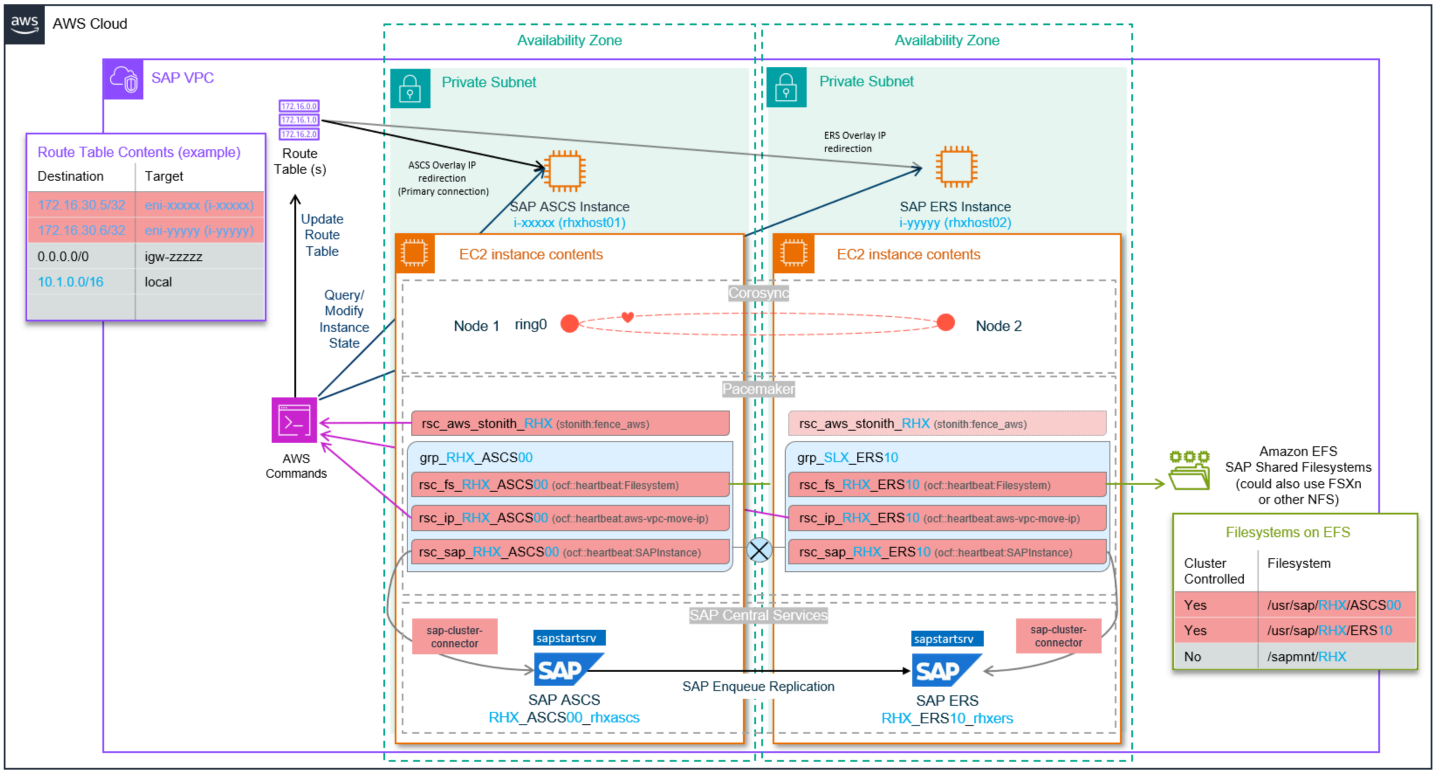
Pacemaker - simple-mount architecture

See the following image for more details.



Pacemaker - classic architecture

See the following image for more details.



Prerequisites

Topics

- Amazon Infrastructure Setup
- EC2 Instance Configuration
- Operating System Requirements

Amazon Infrastructure Setup

This section covers the one-time setup tasks required to prepare your Amazon environment for the cluster deployment:

> **Note**
>
> We recommend using administrative privileges from an administrative workstation or Amazon Console for the initial infrastructure setup instead of granting instance-based privileges, as this maintains the principle of least privilege. Infrastructure setup APIs (such as CreateRoute, ModifyInstanceAttribute, and CreateTags) are only required during initial configuration and are not needed for ongoing cluster operations.

Topics

- Create IAM Roles and Policies for Pacemaker
- Modify Security Groups for Cluster Communication
- Add VPC Route Table Entries for Overlay IPs

Create IAM Roles and Policies for Pacemaker

In addition to the permissions required for standard SAP operations, two IAM policies are required for the cluster to control Amazon resources. These policies must be assigned to your Amazon EC2 instance using an IAM role. This enables Amazon EC2 instance, and therefore the cluster to call Amazon services.

Note

Create policies with least-privilege permissions, granting access to only the specific resources that are required within the cluster. For multiple clusters, you may need to create multiple policies.

For more information, see IAM roles for Amazon EC2.

STONITH Policy

The RHEL STONITH resource agent (fence_aws) requires permission to start and stop both the nodes of the cluster. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeTags"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0"
      ]
    }
  ]
}
```

Amazon Overlay IP Policy

The RHEL Overlay IP resource agent (aws-vpc-move-ip) requires permission to modify a routing entry in route tables. Create a policy as shown in the following example. Attach this policy to the IAM role assigned to both Amazon EC2 instances in the cluster.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:ReplaceRoute",
            "Resource": [
                 "arn:aws:ec2:us-east-1:123456789012:route-table/
rtb-0123456789abcdef0",
                 "arn:aws:ec2:us-east-1:123456789012:route-table/rtb-0123456789abcdef0"
                        ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeRouteTables",
            "Resource": "*"
        }
    ]
}
```

Shared VPC (optional)

Note

The following directions are only required for setups which include a Shared VPC.

Amazon VPC sharing enables you to share subnets with other Amazon accounts within the same Amazon Organizations. Amazon EC2 instances can be deployed using the subnets of the shared Amazon VPC.

In the pacemaker cluster, the aws-vpc-move-ip resource agent has been enhanced to support a shared VPC setup while maintaining backward compatibility with previous existing features.

The following checks and changes are required. We refer to the Amazon account that owns Amazon VPC as the sharing VPC account, and to the consumer account where the cluster nodes are going to be deployed as the cluster account.

Minimum Version Requirements

The latest version of the aws-vpc-move-ip agent shipped with RHEL8 and RHEL9 supports the shared VPC setup by default. The following are the minimum version required to support a shared VPC Setup:

- RHEL 8 - resource-agents-4.1.1-90.el8_4.7.x86_64
- RHEL 9 - resource-agents-4.9.0-16.el9_0.6.x86_64

IAM Roles and Policies

Using the Overlay IP agent with a shared Amazon VPC requires a different set of IAM permissions to be granted on both Amazon accounts (sharing VPC account and cluster account).

Sharing VPC Account

In sharing VPC account, create an IAM role to delegate permissions to the EC2 instances that will be part of the cluster. During the IAM Role creation, select "Another Amazon account" as the type of trusted entity, and enter the Amazon account ID where the EC2 instances will be deployed/running from.

After the IAM role has been created, create the following IAM policy on the sharing VPC account, and attach it to an IAM role. Add or remove route table entries as needed.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "ec2:ReplaceRoute",
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:route-table/rtb-0123456789abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:route-table/rtb-0123456789abcdef0"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "ec2:DescribeRouteTables",
      "Resource": "*"
    }
  ]
}
```

Next, edit move to the "Trust relationships" tab in the IAM role, and ensure that the Amazon account you entered while creating the role has been correctly added.

In cluster account, create the following IAM policy, and attach it to an IAM role. This is the IAM Role that is going to be attached to the EC2 instances.

STS Policy

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/sharing-vpc-account-cluster-role"
    }
  ]
}
```

STONITH Policy

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0",
        "arn:aws:ec2:us-east-1:123456789012:instance/arn:aws:ec2:us-
east-1:123456789012:instance/i-1234567890abcdef0"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    }
  ]
}
```

Modify Security Groups for Cluster Communication

A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For more information, see Control traffic to your Amazon resources using security groups.

In addition to the standard ports required to access SAP and administrative functions, the following rules must be applied to the security groups assigned to all Amazon EC2 instances in the cluster.

Source	Protocol	Port range	Description
The security group ID (its own resource ID)	UDP	5405	Allows UDP traffic between cluster resources for corosync communica tion

- Note the use of the UDP protocol.
- If you are running a local firewall, such as iptables, ensure that communication on the preceding ports is allowed between two Amazon EC2 instances.

Add VPC Route Table Entries for Overlay IPs

You need to add initial route table entries for the Overlay IP. For more information on Overlay IP, see Amazon – Overlay IP.

Add entries to the VPC route table or tables associated with the subnets of your Amazon EC2 instance for the cluster. The entries for destination (Overlay IP CIDR) and target (Amazon EC2 instance or ENI) must be added manually for the ASCS and the ERS. This ensures that the cluster resource has a route to modify. It also supports the install of SAP using the virtual names associated with the Overlay IP before the configuration of the cluster.

Using either the Amazon VPC console, or an Amazon CLI command add a route to the table or tables for the Overlay IP.

Amazon Console

1. Identify the EC2 instance IDs for both cluster nodes and determine which route tables are associated with their subnets. For details, see Parameter Reference
2. Open the Amazon VPC console at https://console.aws.amazon.com/vpc
3. In the navigation pane, choose **Route Tables**, select the first route table.
4. Choose **Actions** → **Edit routes**.
5. Choose **Add route** and configure the ASCS route:

Destination	Target
`<ascs_overlayip>/32`	`i-xxxxinstidforhost1`

6. Choose **Add route** and configure the ERS route:

Destination	Target
`<ers_overlayip>/32`	`i-xxxxinstidforhost2`

7. Choose **Save changes**.
8. Repeat for any additional associated route tables or route tables from the VPC which require connectivity to the ASCS.

   Your route table now includes entries for required Overlay IPs, in addition to the standard routes.

Amazon CLI

Identify the EC2 instance IDs for both cluster nodes and determine which route tables are associated with their subnets. For details, see. Parameter Reference.

For the ASCS:

```
$ aws ec2 create-route --route-table-id <routetable_id> --destination-cidr-block
 <ascs_overlayip>/32 --instance-id <instance_id_1>
```

For the ERS:

```
$ aws ec2 create-route --route-table-id <routetable_id> --destination-cidr-block
 <ers_overlayip>/32 --instance-id <instance_id_2>
```

EC2 Instance Configuration

Amazon EC2 instance settings can be applied using Infrastructure as Code or manually using Amazon Command Line Interface or Amazon Console. We recommend Infrastructure as Code automation to reduce manual steps, and ensure consistency.

Topics

- Assign or Review Pacemaker IAM Role
- Assign or Review Security Groups
- Assign Secondary IP Addresses
- Disable Source/Destination Check
- Review Stop Protection
- Review Automatic Recovery

> ⚠ **Important**
>
> The following configurations must be performed on all cluster nodes. Ensure consistency across nodes to prevent cluster issues.

Assign or Review Pacemaker IAM Role

The two cluster resource IAM policies must be assigned to an IAM role associated with your Amazon EC2 instance. If an IAM role is not associated to your instance, create a new IAM role for cluster operations.

The following Amazon Console or Amazon CLI commands can be used to modify the IAM role assignment.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the navigation pane, choose **Actions** → **Security** → **Modify IAM role**.
4. Choose the IAM role that contains the policies created in Create IAM Roles and Policies for Pacemaker.
5. Choose **Update IAM role**.
6. Repeat these steps for all nodes in the cluster.

Amazon CLI

To assign an IAM role using the Amazon CLI:

```
$ aws ec2 associate-iam-instance-profile --instance-id <instance_id> --iam-instance-profile Name=<iam_instance_profile_name>
```

Repeat for all nodes in the cluster.

You can verify the IAM role assignment on your instances using the Amazon CLI:

```
$ aws ec2 describe-instances --instance-ids <instance_id> --query
 'Reservations[0].Instances[0].IamInstanceProfile' --output table
```

You can check the specific permissions of the roles created for pacemaker in the section called "Create IAM Roles and Policies for Pacemaker" by running the following on both your instances.

When --dry-run is used, the Amazon CLI or SDK sends the request to the EC2 service with this flag. EC2 then performs all necessary permission checks and validates the request parameters.

If the user has the required permissions and the request is well-formed, the service returns a DryRunOperation error response, indicating that the operation would have succeeded.

Check that the fencing resource has the permission to shut down both instances:

```
$ aws ec2 stop-instances --instance-ids <instance_id_1> --dry-run
$ aws ec2 stop-instances --instance-ids <instance_id_2> --dry-run
```

Check that the overlay IP resource has the pemissions to update the route tables:

```
$ aws ec2 replace-route --route-table-id <routetable_id> --destination-cidr-block
 <ascs_overlayip>/32 --instance-id <instance_id_1> --dry-run
```

Assign or Review Security Groups

The security group rules created in the Amazon Modify Security Groups for Cluster Communication section must be assigned to your Amazon EC2 instances. If a security group is not associated with your instance, or if the required rules are not present in the assigned security group, add the security group or update the rules.

The following Amazon Console or Amazon CLI commands can be used to modify security group assignments.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the **Security** tab, review the security groups, ports, and source of traffic.
4. If required, choose **Actions** → **Security** → **Change security groups**.
5. Under **Associated security groups**, search for and select the required groups.
6. Choose **Save**.
7. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify security groups using the Amazon CLI:

```
$ aws ec2 modify-instance-attribute --instance-id <instance_id> --groups
 <security_group_id1> <security_group_id2>
```

Repeat for all nodes in the cluster.

You can verify the security group rules on your instances using the Amazon CLI:

```
$ aws ec2 describe-instance-attribute --instance-id <instance_id> --attribute groupSet
```

Assign Secondary IP Addresses

Secondary IP addresses are used to create a redundant communication channel (secondary ring) in corosync for clusters. The cluster nodes can use the secondary ring to communicate in case of underlying network disruptions.

These IPs are only used in cluster configurations. The secondary IPs provide the same fault tolerance as a secondary Elastic Network Interface (ENI). For more information, see Secondary IP addresses for your EC2 Instance.

The following Amazon Console or Amazon CLI commands can be used to assign secondary IP addresses.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the **Networking** tab, choose the network interface ID.
4. Choose **Actions** → **Manage IP addresses**.
5. Choose **Assign new IP address**.
6. Select **Auto-assign** or specify an IP from the subnet range.
7. Choose **Yes, Update**.
8. Repeat these steps for all nodes in the cluster.

Amazon CLI

To assign secondary IP addresses using the Amazon CLI:

```
$ ENI_ID=$(aws ec2 describe-instances --instance-id <instance_id> \
    --query 'Reservations[0].Instances[0].NetworkInterfaces[0].NetworkInterfaceId' \
    --output text)
$ aws ec2 assign-private-ip-addresses --network-interface-id $ENI_ID --secondary-
private-ip-address-count 1
```

Repeat for all nodes in the cluster.

You can verify the secondary IP configuration on your instances using the Amazon CLI:

```
$ aws ec2 describe-instances --instance-id <instance_id> \
    --query
 'Reservations[*].Instances[*].NetworkInterfaces[*].PrivateIpAddresses[*].PrivateIpAddress'
 \
    --output text
```

Verify that:

- Each instance returns two IP addresses from the same subnet
- The primary network interface (eth0) has both IPs assigned
- The secondary IPs will be used later for ring0_addr and ring1_addr in corosync.conf

Disable Source/Destination Check

Amazon EC2 instances perform source/destination checks by default, requiring that an instance is either the source or the destination of any traffic it sends or receives. In the pacemaker cluster, source/destination check must be disabled on both instances receiving traffic from the Overlay IP.

The following Amazon Console or Amazon CLI commands can be used to modify the attribute.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. In the navigation pane, choose **Actions** → **Networking** → **Change source/destination check**.
4. For Source/Destination Checking, choose **Stop** to allow traffic when the source or destination is not the instance itself.
5. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify using the Amazon CLI (requires appropriate configuration permissions):

```
$ aws ec2 modify-instance-attribute --instance-id <instance_id> --no-source-dest-
check
```

Repeat for all nodes in the cluster.

To confirm the value of an attribute for a particular instance, use the following command. The value `false` means source/destination checking is disabled

```
$ aws ec2 describe-instance-attribute --instance-id <instance_id> --attribute
 sourceDestCheck
```

The output

```
{
    "InstanceId": "i-xxxxinstidforhost1",
    "SourceDestCheck": {
        "Value": false
    }
}
```

Review Stop Protection

To ensure that STONITH actions can be executed, you must ensure that stop protection is disabled for Amazon EC2 instances that are part of a pacemaker cluster. If the default settings have been modified, use the following commands for both instances to disable stop protection via Amazon CLI.

The following Amazon Console or CLI commands can be used to modify the attribute.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. Choose **Actions** → **Instance settings** → **Change stop protection**.
4. Ensure **Stop protection** is not enabled.

5. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify using the Amazon CLI (requires appropriate configuration permissions):

```
$ aws ec2 modify-instance-attribute --instance-id <instance_id> --no-disable-api-
stop
```

Repeat this command for all nodes in the cluster.

To confirm the value of an attribute for a particular instance, use the following command. The value `false` means it is possible to stop the instance using an Amazon CLI.

```
$ aws ec2 describe-instance-attribute --instance-id <instance_id> --attribute
 disableApiStop
```

The output

```
{
    "InstanceId": "i-xxxxinstidforhost1",
    "DisableApiStop": {
        "Value": false
    }
}
```

Review Automatic Recovery

After a failure, cluster-controlled operations must be resumed in a coordinated way. This helps ensure that the cause of failure is known and addressed, and the status of the cluster is as expected. For example, verifying that there are no pending fencing actions.

The following Amazon Console or CLI commands can be used to modify the attribute.

Amazon Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2.
2. Select one of your cluster nodes.
3. Choose **Actions** → **Instance settings** → **Change auto-recovery behavior**.

4. Select **Off** to disable auto-recovery for system status check failures.
5. Repeat these steps for all nodes in the cluster.

Amazon CLI

To modify auto-recovery settings (requires appropriate configuration permissions):

```
$ aws ec2 modify-instance-maintenance-options --instance-id <instance_id> --auto-
recovery disabled
```

Repeat this command for all nodes in the cluster.

To confirm the value of an attribute for a particular instance, use the following command. The value `disabled` means autorecovery will not be attempted.

```
$ aws ec2 describe-instances --instance-ids <instance_id> --query
 'Reservations[*].Instances[*].MaintenanceOptions.AutoRecovery'
```

The output:

```
[
    [
        "disabled"
    ]
]
```

Operating System Requirements

This section outlines the required operating system configurations for Red Hat Enterprise Linux for SAP (RHEL for SAP) cluster nodes. Note that this is not a comprehensive list of configuration requirements for running SAP on Amazon, but rather focuses specifically on cluster management prerequisites.

Consider using configuration management tools or automated deployment scripts to ensure accurate and repeatable setup across your cluster infrastructure.

Topics

- Install Missing Operating System Packages
- Update and Check Operating System Versions
- System Logging
- Disable NetworkManager Cloud Services
- Disable kdump
- Time Synchronization Services
- Install Amazon CLI and Configure Profiles
- Pacemaker Proxy Settings (Optional)

> **Important**
>
> The following configurations must be performed on all cluster nodes. Ensure consistency across nodes to prevent cluster issues.

Root Access

Verify root access on both cluster nodes. The majority of the setup commands in this document are performed with the root user. Assume that commands should be run as root unless there is an explicit call out to choose otherwise.

Install Missing Operating System Packages

This is applicable to all cluster nodes. You must install any missing operating system packages.

The following packages and their dependencies are required for the pacemaker setup. Depending on your baseline image, for example, RHEL for SAP, these packages may already be installed.

Package	Description	Category	Required	Configuration Pattern
chrony	Time Synchronization	System Support	Mandatory	All
rsyslog	System Logging	System Support	Mandatory	All

Package	Description	Category	Required	Configuration Pattern
pacemaker	Cluster Resource Manager	Core Cluster	Mandatory	All
corosync	Cluster Communication Engine	Core Cluster	Mandatory	All
resource-agents	Resource Agents including SAPInstance	Core Cluster	Mandatory	All
resource-agents-cloud	Cloud Resource agents including aws-vpc-move-ip	Core Cluster	Mandatory	RHEL 9 and above
fence-agents-aws	Amazon Fencing Capabilities	Core Cluster	Mandatory	All
resource-agents-sap	SAP Resource Agents	SAP Integration	Mandatory	resource-agents-sap-4.15.1 required for SimpleMount
sap-cluster-connector	SAP HA-Script Connector	SAP Integration	Mandatory	All
pcs	Pacemaker Configuration System	Core Cluster	Mandatory	All
sysstat	Performance Monitoring Tools	Support Tools	Recommended	All
dstat	System Resource Statistics	Monitoring	Recommended	All
iotop	I/O Monitoring	Monitoring	Recommended	All

> **Note**
>
> Refer to Vendor Support of Deployment Types for more information on Configuration Patterns. `Mandatory*` indicates that this package is mandatory based on the Configuration Pattern.

You can use the following script to check for missing packages and optionally install them:

```
#!/bin/bash
# Mandatory core packages for SAP NetWeaver HA on AWS
mandatory_packages="corosync pacemaker resource-agents resource-agents-cloud fence-
agents-aws rsyslog chrony sap-cluster-connector pcs resource-agents-sap"

# Recommended monitoring and support packages
support_packages="sysstat dstat iotop"

# Default to checking all packages
packages="${mandatory_packages} ${support_packages}"

missingpackages=""

echo "Checking SAP NetWeaver HA package requirements..."

for package in ${packages}; do
    echo "Checking if ${package} is installed..."
    if ! rpm -q ${package} --quiet; then
        echo " ${package} is missing and needs to be installed"
        missingpackages="${missingpackages} ${package}"
    fi
done

if [ -z "$missingpackages" ]; then
    echo "All packages are installed."
else
    echo "Missing mandatory packages: $(echo ${missingpackages} | tr ' ' '\n' | grep -E
 "^($(echo ${mandatory_packages} | tr ' ' '|'))$")"
    echo "Missing support packages: $(echo ${missingpackages} | tr ' ' '\n' | grep -E
 "^($(echo ${support_packages} | tr ' ' '|'))$")"

    echo -n "Do you want to install the missing packages (y/n)? "
    read response
    if [ "$response" = "y" ]; then
        dnf install -y $missingpackages
    fi
fi
```

If a package is not installed, and you are unable to install it using dnf, it may be because Red Hat Enterprise Linux High Availability Add-On is not available as a repository in your chosen image. You can verify the availability of the add-on using the following command:

```
$ sudo dnf repolist
```

To install or update a package or packages with confirmation, use the following command:

```
$ sudo dnf install <package_name(s)>
```

Update and Check Operating System Versions

You must update and confirm versions across nodes. Apply all the latest patches to your operating system versions. This ensures that bugs are addressed and new features are available.

You can update the patches individually or update all system patches using the `dnf update` command. A clean reboot is recommended prior to setting up a cluster.

```
$ sudo dnf update
$ sudo reboot
```

Compare the operating system package versions on the two cluster nodes and ensure that the versions match on both nodes.

System Logging

Both systemd-journald and rsyslog are suggested for comprehensive logging. Systemd-journald (enabled by default) provides structured, indexed logging with immediate access to events, while rsyslog is maintained for backward compatibility and traditional file-based logging. This dual approach ensures both modern logging capabilities and compatibility with existing log management tools and practices.

1. Enable and start rsyslog:

```
# systemctl enable --now rsyslog
```

2. (Optional) Configure persistent logging for systemd-journald:

If you are not using a logging agent (like the Amazon CloudWatch Unified Agent or Vector) to ship logs to a centralized location, you may want to configure persistent logging to retain logs after system reboots.

```
# mkdir -p /etc/systemd/journald.conf.d
```

Create `/etc/systemd/journald.conf.d/99-logstorage.conf` with:

```
[Journal]
Storage=persistent
```

Persistent logging requires careful storage management. Configure appropriate retention and rotation settings in `journald.conf` to prevent logs from consuming excessive disk space. Review `man journald.conf` for available options such as SystemMaxUse, RuntimeMaxUse, and MaxRetentionSec.

To apply the changes, restart journald:

```
# systemctl restart systemd-journald
```

After enabling persistent storage, only new logs will be stored persistently. Existing logs from the current boot session will remain in volatile storage until the next reboot.

3. Verify services are running:

```
# systemctl status systemd-journald
# systemctl status rsyslog
```

Disable NetworkManager Cloud Services

When using Red Hat Enterprise Linux 8.6 or later, the NetworkManager cloud setup services must be disabled to maintain cluster stability. These services can interfere with cluster operations by automatically removing the overlay IP address from network interfaces.

Run these commands on each cluster node:

```
# systemctl disable --now nm-cloud-setup.timer
# systemctl disable --now nm-cloud-setup
```

Verify the services are disabled and stopped:

```
# systemctl status nm-cloud-setup.timer
# systemctl status nm-cloud-setup
```

The status commands should show both services as "disabled" and "inactive (dead)".

Disable kdump

The kernel crash dump facility (kdump) should be disabled with the following commands on each cluster node:

```
# systemctl stop kdump
# systemctl disable kdump
```

When kdump triggers an immediate system reboot during a kernel panic, it bypasses Pacemaker’s controlled failover process, potentially leaving cluster resources in an inconsistent state.

Time Synchronization Services

Time synchronization is important for cluster operation. Ensure that chrony rpm is installed, and configure appropriate time servers in the configuration file.

You can use Amazon Time Sync Service that is available on any instance running in a VPC. It does not require internet access. To ensure consistency in the handling of leap seconds, don’t mix Amazon Time Sync Service with any other ntp time sync servers or pools.

Create or check the `/etc/chrony.d/ec2.conf` file to define the server:

```
# Amazon EC2 time source config
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

Start the chronyd.service, using the following command:

```
# systemctl enable --now chronyd.service
# systemctl status chronyd
```

Verify time synchronization is working:

```
# chronyc tracking
```

Ensure the output shows `Reference ID : A9FEA97B (169.254.169.123)` confirming synchronization with Amazon Time Sync Service.

For more information, see Set the time for your Linux instance.

Install Amazon CLI and Configure Profiles

The Amazon cluster resource agents require Amazon Command Line Interface (Amazon CLI). Check if Amazon CLI is already installed, and install it if necessary.

Check if Amazon CLI is installed:

```
# aws --version
```

If the command is not found, install Amazon CLI v2 using the following commands:

```
# cd /tmp
# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
# dnf install -y unzip
# unzip awscliv2.zip
# sudo ./aws/install --update
```

Create symlinks to ensure Amazon CLI is in the system PATH:

```
# sudo ln -sf /usr/local/bin/aws /usr/bin/aws
```

Verify the installation:

```
# aws --version
```

The installation creates a symbolic link at `/usr/local/bin/aws` which is typically in the system PATH by default.

For more information, see Installing or updating to the latest version of the Amazon CLI.

After installing Amazon CLI, you need to create an Amazon CLI profile for the root account.

You can either edit the config file at `/root/.aws` manually or by using the `aws configure` Amazon CLI command.

You should skip providing the information for the access and secret access keys. The permissions are provided through IAM roles attached to Amazon EC2 instances.

```
# aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: <region>
Default output format [None]:
```

The profile name is `default` unless configured. If you choose to use a different name you can specify `--profile`. The name chosen in this example is cluster. It is used in the Amazon resource agent definition for pacemaker. The Amazon Region must be the default Amazon Region of the instance.

```
# aws configure --profile cluster
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: <region>
Default output format [None]:
```

On the hosts, you can verify the available profiles using the following command:

```
# aws configure list-profiles
```

And review that an assumed role is associated by querying the caller identity:

```
# aws sts get-caller-identity --profile=<profile_name>
```

Pacemaker Proxy Settings (Optional)

If your Amazon EC2 instance has been configured to access the internet and/or Amazon Cloud through proxy servers, then you need to replicate the settings in the pacemaker configuration. For more information, see Using an HTTP Proxy.

Add the following lines to `/etc/sysconfig/pacemaker`:

```
http_proxy=http://<proxyhost>:<proxyport>
```

```
https_proxy=http://<proxyhost>:<proxyport>
no_proxy=127.0.0.1,localhost,169.254.169.254,fd00:ec2::254
```

- Modify proxyhost and proxyport to match your settings.
- Ensure that you exempt the address used to access the instance metadata.
- Configure no_proxy to include the IP address of the instance metadata service – 169.254.169.254 (IPV4) and fd00:ec2::254 (IPV6). This address does not vary.

SAP ASCS and Cluster Setup

This section covers the following topics.

Topics

- SAP Shared File Systems
- Check IP availability and resolution
- Install SAP
- Configure SAP for Cluster Control
- Cluster Node Setup
- Cluster Configuration

SAP Shared File Systems

Topics

- Select Shared Storage
- Create file systems
- Create mount point directories
- Update /etc/fstab
- Temporarily mount ASCS and ERS directories for installation (classic only)

Select Shared Storage

SAP NetWeaver high availability deployments require shared file systems. On Linux, you can use either Amazon Elastic File System or Amazon FSx for NetApp ONTAP. Choose between these options based on your requirements for resilience, performance, and cost. For detailed setup information, see Getting started with Amazon Elastic File System or Getting started with Amazon FSx for NetApp ONTAP.

We recommend sharing a single Amazon EFS or FSx for ONTAP file system across multiple SIDs within an account.

The file system's DNS name is the simplest mounting option. When connecting from an Amazon EC2 instance, the DNS automatically resolves to the mount target's IP address in that instance's Availability Zone. You can also create an alias (CNAME) to help identify the shared file system's purpose. Throughout this document, we use `<nfs.fqdn>`.

Examples:

- `file-system-id.efs.aws-region.amazonaws.com`
- `svm-id.fs-id.fsx.aws-region.amazonaws.com`
- `qas_sapmnt_share.example.com`

Note

Review the `enableDnsHostnames` and `enableDnsSupport` DNS attributes for your VPC. For more information, see View and update DNS attributes for your VPC.

Create file systems

The following shared file systems are covered in this document:

NFS Location Structure	NFS Location Example	File System Location Structure	File System Location Example
<SID>_sapmnt	`RHX_sapmnt`	/sapmnt/<SID>	`/sapmnt/RHX`
<SID>_ASCS<ascs_sys_nr>	`RHX_ASCS00`	/usr/sap/<SID>/ASCS<ascs_sys_nr>	`/usr/sap/RHX/ASCS00`
<SID>_ERS<ers_sys_nr>	`RHX_ERS10`	/usr/sap/<SID>/ERS<ers_sys_nr>	`/usr/sap/RHX/ERS10`

The following options can differ depending on how you architect and operate your systems:

- ASCS and ERS mount points - In simple-mount architecture, you can share the entire `/usr/sap/<SID>` directory. This document uses separate mount points to simplify migration and follow SAP's recommendation for local application server executables when co-hosting ASCS/ERS.
- Transport directory - `/usr/sap/trans` is optional for ASCS installations. Add this shared directory if your change management processes require it.
- Home directory - This document uses local home directories to ensure `<sid>adm` access during NFS issues. Consider a shared home directory if you need consistent user environments across nodes.
- NFS location naming - The "NFS Location" names are arbitrary and can be chosen based on your naming conventions (e.g., `myEFSMount1`, `prod_sapmnt`, etc.). The "File system location" follows the standard SAP directory structure and should use the parameter references shown.

For more information, see SAP System Directories on UNIX.

Using the NFS ID created in the previous step, temporarily mount the root directory of the NFS. `/mnt` is available by default; it can also be substituted with another temporary location.

Note

The following commands use the NFS location names from the table above. Replace `<SID>_sapmnt`, `<SID>_ASCS<ascs_sys_nr>`, and `<SID>_ERS<ers_sys_nr>` with your chosen NFS location names and parameter values.

```
# mount <nfs.fqdn>:/ /mnt
# mkdir -p /mnt/<SID>_sapmnt
# mkdir -p /mnt/<SID>_ASCS<ascs_sys_nr>
# mkdir -p /mnt/<SID>_ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/ /mnt
# mkdir -p /mnt/RHX_sapmnt
# mkdir -p /mnt/RHX_ASCS00
# mkdir -p /mnt/RHX_ERS10
```

During SAP installation, the `<sid>adm` user and proper directory ownership will be created. Until then, we need to ensure the installation process has sufficient access. Set temporary permissions on the directories:

```
# chmod 777 /mnt/<SID>_sapmnt /mnt/<SID>_ASCS<ascs_sys_nr> /mnt/<SID>_ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

  ```
  # chmod 777 /mnt/RHX_sapmnt /mnt/RHX_ASCS00 /mnt/RHX_ERS10
  ```

The SAP installation process will automatically set the correct ownership and permissions for operational use.

Unmount the temporary mount:

```
# umount /mnt
```

Create mount point directories

This is applicable to both cluster nodes. Create the directories for the required mount points (permanent or cluster controlled):

```
# mkdir /sapmnt
# mkdir /usr/sap/<SID>/ASCS<ascs_sys_nr>
# mkdir /usr/sap/<SID>/ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

  ```
  # mkdir /sapmnt
  # mkdir /usr/sap/RHX/ASCS00
  # mkdir /usr/sap/RHX/ERS10
  ```

Update /etc/fstab

This is applicable to both cluster nodes. `/etc/fstab` is a configuration table containing the details required for mounting and unmounting file systems to a host.

Add the file systems not managed by the cluster to `/etc/fstab`.

For both **simple-mount** and **classic** architectures, prepare and append an entry for the `sapmnt` file system to `/etc/fstab`:

```
<nfs.fqdn>/<SID>_sapmnt    /sapmnt    nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport    0    0
```

Simple-mount only – prepare and append entries for the ASCS and ERS file systems to `/etc/fstab`:

```
<nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr>   /usr/sap/<SID>/ASCS<ascs_sys_nr>  nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport    0    0
<nfs.fqdn>:/<SID>_ERS<ers_sys_nr>     /usr/sap/<SID>/ERS<ers_sys_nr>    nfs
 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport    0    0
```

- *Example using values from [Parameter Reference](#)* :

```
fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_sapmnt    /sapmnt
 nfs    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
  0    0
fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_ASCS00    /usr/sap/RHX/ASCS00
 nfs    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
  0    0
fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_ERS10     /usr/sap/RHX/ERS10
 nfs    nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
  0    0
```

Verify that your mount options are:

- Compatible with your operating system version
- Supported by your chosen NFS file system type (EFS or FSx for ONTAP)
- Aligned with current SAP recommendations

Consult SAP and Amazon documentation for the latest mount option recommendations.

Use the following command to mount the file systems defined in `/etc/fstab`:

```
# mount -a
```

Use the following command to check that the required file systems are available:

```
# df -h
```

Temporarily mount ASCS and ERS directories for installation (classic only)

This is only applicable to the classic architecture. Simple-mount architecture has these directories permanently available in `/etc/fstab`.

Mount ASCS and ERS directories for installation.

Use the following command on the instance where you plan to install ASCS:

```
# mount <nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr>  /usr/sap/<SID>/ASCS<ascs_sys_nr>
```

Use the following command on the instance where you plan to install ERS:

```
# mount <nfs.fqdn>:/<SID>_ERS<ers_sys_nr>  /usr/sap/<SID>/ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

  ```
  # mount fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_ASCS00  /usr/sap/RHX/
  ASCS00
  # mount fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_ERS10   /usr/sap/RHX/
  ERS10
  ```

Check IP availability and resolution

Add Overlay IP for SAP Installation

SAP Installation should be done using the virtual names assigned to the overlay IP. Before adding the overlay IPs to the instances, ensure that the VPC route table entries have been created as described in the section called “Add VPC Route Table Entries for Overlay IPs”.

To facilitate SAP installation, manually add the Overlay IPs to the instances:

1. To the instance where you intend to install the **ASCS**

   ```
   # ip addr add <ascs_overlayip>/32 dev eth0
   ```

2. To the instance where you intend to install the **ERS**

```
# ip addr add <ers_overlayip>/32 dev eth0
```

Note the following:

- Route table entries for the overlay IPs must be created first (see the section called “Add VPC Route Table Entries for Overlay IPs”)
- This IP configuration is temporary and will be lost after instance reboot
- The cluster will take over management of these IPs once configured

Hostname Resolution

You must ensure that all instances can resolve all hostnames in use. Add the hostnames for cluster nodes to `/etc/hosts` file on all cluster nodes. This ensures that hostnames for cluster nodes can be resolved even in case of DNS issues. Configure the `/etc/hosts` file for a two-node cluster:

```
# cat /etc/hosts
<primary_ip_1> <hostname_1>.example.com <hostname_1>
<primary_ip_2> <hostname_2>.example.com <hostname_2>
<ascs_overlayip> <ascs_virt_hostname>.example.com <ascs_virt_hostname>
<ers_overlayip> <ers_virt_hostname>.example.com <ers_virt_hostname>
```

- *Example using values from Parameter Reference* :

  ```
  # cat /etc/hosts
  10.1.10.1 rhxhost01.example.com rhxhost01
  10.1.20.1 rhxhost02.example.com rhxhost02
  172.16.30.5 rhxascs.example.com rhxascs
  172.16.30.6 rhxers.example.com rhxers
  ```

In this configuration, the secondary IPs used for the second cluster ring are not mentioned. They are only used in the cluster configuration. You can allocate virtual hostnames for administration and identification purposes.

⚠ **Important**

The Overlay IP is out of VPC range, and cannot be reached from locations not associated with the route table, including on-premises.

Install SAP

The following topics provide information about installing SAP on Amazon Cloud in a highly available cluster. Review SAP Documentation for more details.

Topics

Final checks for software provisioning

Before running SAP Software Provisioning Manager (SWPM), ensure that the following prerequisites are consistent across both cluster nodes:

- Collect any missing details and populate the Parameter Reference section to ensure clarity on the specific values used in installation commands.
- **User and Group Configuration** - If operating system groups are pre-defined, ensure matching UID and GID values for `<sid>adm` and `sapsys` across both cluster nodes.
- **Installation Software** - Download the latest version of Software Provisioning Manager (SWPM) and SAP installation media for your SAP release from Software Provisioning Manager.
- **Network Configuration** - Verify both cluster nodes have identical configuration with all routes, overlay IPs, and virtual hostnames accessible. This ensures that either node can run ASCS or ERS roles.
- **File Systems** - Verify all shared file systems are mounted and accessible from both nodes with consistent mount points and permissions.

Install SAP ASCS and ERS instances

Install the SAP ASCS and ERS instances using their virtual hostnames to ensure installation against the overlay IP addresses. This approach is required for proper cluster integration.

Install the ASCS instance on `<instance_id_1>` using virtual hostname `<ascs_virt_hostname>` with the `SAPINST_USE_HOSTNAME` parameter. This ensures the installation uses the overlay IP rather than the physical hostname:

Example using values from Parameter Reference :

```
# <swpm location>/sapinst SAPINST_USE_HOSTNAME=<ascs_virt_hostname>
```

Install the ERS instance on `<instance_id_2>` using virtual hostname `<ers_virt_hostname>` with the `SAPINST_USE_HOSTNAME` parameter. This ensures the installation uses the overlay IP rather than the physical hostname:

```
# <swpm location>/sapinst SAPINST_USE_HOSTNAME=<ers_virt_hostname>
```

Once the ASCS and ERS installations are complete, you will need to install and configure the database and SAP Primary Application Server (PAS) - these components are not covered in this cluster setup documentation. Optionally, you can also install and configure Additional Application Server (AAS). For more details on installing these SAP NetWeaver components, refer to SAP Help Portal.

For additional information on unattended installation options, see SAP Note 2230669 – System Provisioning Using an Input Parameter File (requires SAP portal access).

Kernel upgrade and ENSA2 – optional

As of AS ABAP Release 7.53 (ABAP Platform 1809), the new Standalone Enqueue Server 2 (ENSA2) is installed by default. ENSA2 replaces the previous version – ENSA1.

If you have an older version of SAP NetWeaver, consider following the SAP guidance to upgrade the kernel and update the Enqueue Server configuration. An upgrade will allow you to take advantage of the features available in the latest version. For more information, see the following SAP Notes (require SAP portal access):

- SAP Note 2630416 – Support for Standalone Enqueue Server 2

- SAP Note 2711036 – Usage of the Standalone Enqueue Server 2 in an HA Environment

Check SAP host agent version

This is applicable to both cluster nodes. The SAP host agent is used for system instance control and monitoring. This agent is used by SAP cluster resource agents and hooks. It is recommended that you have the latest version installed on both instances. For more details, see SAP Note 2219592 – Upgrade Strategy of SAP Host Agent.

Use the following command to check the version of the host agent:

```
# /usr/sap/hostctrl/exe/saphostexec -version
```

Configure SAP for Cluster Control

Modify SAP service configurations, user permissions, and system integration settings to enable proper cluster control of ASCS and ERS instances.

Topics

Add <sid>adm to haclient group

This is applicable to both cluster nodes. An `haclient` operating system group is created when the cluster connector package is installed. Adding the `<sid>adm` user to this group ensures that your cluster has necessary access. Run the following command as root:

```
# usermod -a -G haclient <sid>adm
```

- *Example using values from Parameter Reference* :

  ```
  # usermod -a -G haclient rhxadm
  ```

Modify SAP profiles for start operations and cluster hook

This action ensures that there is compatibility between the SAP start framework and cluster actions. Modify SAP profiles to change the start behavior of the SAP instance and processes. Ensure that `sapcontrol` is aware that the system is being managed by a pacemaker cluster.

- ASCS profile – `/usr/sap/<SID>/SYS/profile/<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>`
- ERS profile – `/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>`

The profile directory /usr/sap/<SID>/SYS/profile/ is typically a symbolic link to /sapmnt/<SID>/profile/ on the shared NFS filesystem. This means profile modifications made on one node are immediately visible on all cluster nodes. You can modify the profiles from either node.

- *Example using values from Parameter Reference* :
  - ASCS profile example – `/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs`
  - ERS profile example – `/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers`

Follow the procedure outlined below to make the necessary changes:

1. **Program or process start behavior** – In case of failure, processes must be restarted. Determining where the process starts and in what order needs to be controlled by the cluster, and not SAP start framework behavior defined in the profiles. Your locks can be lost if this parameter is not changed. In newer SAP installations, the profiles may already contain `Start_Program_XX` instead of `Restart_Program_XX`. If `Start_Program_XX` is already present, no changes are needed for this step.

   Example

   ENSA1

   ASCS

   ```
   #For ENSA1 (_EN)
   #Changing Restart to Start for Cluster compatibility
   #Old value: Restart_Program_XX = local $(_EN) pf=$(_PF)
   ```

```
Start_Program_XX = local $(_EN) pf=$(_PF)
```

ERS

```
#For ENSA1 (_ER)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ER) pf=$(_PFL)NR=$(SCSID)

Start_Program_XX = local $(_ER) pf=$(_PFL) NR=$(SCSID)
```

`XX` indicates the start-up order. This value may be different in your install; retain the unchanged value.

ENSA2

ASCS

```
#For ENSA2 (_ENQ)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQ) pf=$(_PF)

Start_Program_XX = local $(_ENQ) pf=$(_PF)
```

ERS

```
#For ENSA2 (_ENQR)
#Changing Restart to Start for Cluster compatibility
#Old value: Restart_Program_XX = local $(_ENQR) pf=$(_PFL)NR=$(SCSID)

Start_Program_XX = local $(_ENQR) pf=$(_PFL) NR=$(SCSID)
```

`XX` indicates the start order. This value may be different in your install; retain the unchanged value.

2. **Disable instance auto start in both profiles –** When an instance restarts, SAP start framework should not start ASCS and ERS automatically. Add the following parameter on both profiles to prevent an auto start:

```
# Disable instance auto start
Autostart = 0
```

3. **Add cluster connector details in both profiles –** The connector integrates the SAP start and control frameworks of SAP NetWeaver with Red Hat cluster to assist with maintenance and awareness of state. Add the following parameters on both profiles:

```
# Added for Cluster Connectivity
service/halib = $(DIR_EXECUTABLE)/saphascriptco.so
service/halib_cluster_connector = /usr/bin/sap_cluster_connector
```

> **Important**
>
> RPM package `sap-cluster-connector` has *dashes*. The executable `/usr/bin/sap_cluster_connector` available after installation has *underscores*. Ensure that the correct name, that is executable `/usr/bin/sap_cluster_connector`, is used in both profiles.

4. **Restart services –** Restart SAP services for ASCS and ERS to ensure that the preceding settings take effect. Adjust the system number to match the service.

   ASCS

```
# /usr/sap/hostctrl/exe/sapcontrol -nr 00 -function RestartService
```

   ERS

```
# /usr/sap/hostctrl/exe/sapcontrol -nr <ers_sys_nr> -function RestartService
```

   - *Example using values from Parameter Reference* :

     ASCS

```
# /usr/sap/hostctrl/exe/sapcontrol -nr 00 -function RestartService
```

     ERS

```
# /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function RestartService
```

5. **Check integration using `sapcontrol`** – `sapcontrol` includes functions: `HACheckConfig` and `HACheckFailoverConfig`. These functions can be used to check configuration, including

awareness of the cluster connector. These checks have limited value before the cluster is configured, but you can run HACheckFailoverConfig to ensure the base configuration is in place.

ASCS

```
# /usr/sap/hostctrl/exe/sapcontrol -nr <ascs_sys_nr> -function HACheckFailoverConfig
```

- *Example using values from [Parameter Reference](#) :*

  ASCS

  ```
  # /usr/sap/hostctrl/exe/sapcontrol -nr 00 -function HACheckFailoverConfig

  10.10.2025 01:23:55
  HACheckFailoverConfig
  OK
  state, category, description, comment
  SUCCESS, SAP CONFIGURATION, SAPInstance RA sufficient version, SAPInstance includes
   is-ers patch
  ```

Enable sapping and sappong Services (Simple-Mount Only)

For simple-mount architecture, enable the sapping and sappong systemd services on both cluster nodes. These services ensure proper SAP instance startup coordination between systemd and the cluster.

The sapping service runs before sapinit during boot and temporarily hides the `/usr/sap/sapservices` file to prevent automatic SAP instance startup. The sappong service runs after sapinit and restores the sapservices file, making it available for cluster management while maintaining compatibility with SAP management tools.

```
# systemctl enable sapping
# systemctl enable sappong
```

Verify the services are enabled:

```
# systemctl status sapping
# systemctl status sappong
```

Note

Both services will show "inactive (dead)" status, which is normal for one-shot services that only run during system boot.

Ensure ASCS and ERS SAP Services can run on either node (systemd)

This is applicable to both cluster nodes.

To ensure that the cluster can orchestrate availability by starting and stopping instances on either cluster node, the SAP Services must be registerd on both nodes and auto-start should be disabled.

In recent Operating System and SAP kernel versions, SAP offers systemd integration for sapstartsrv which controls how SAP instances are stopped and started. This is the recommended configuration and a requirement for Simple Mount Configuration.

For more details, see the following SAP Notes (require SAP portal access):

- SAP Note 3139184 – Linux: systemd integration for sapstartsrv and SAP Host Agent
- SAP Note 3115048 – sapstartsrv with native Linux systemd support

You can confirm whether systemd is in place by running the following command. Systemd is in place if SAP Services (e.g., SAPRHX_00.service, SAPRHX_10.service) are listed.

```
# systemctl list-unit-files SAP*
```

If you have installed an ASCS or ERS on this host but no SAP Services are returned, the classic SysV init may be in use. In that case you can skip to section the section called “(Alternative) Ensure ASCS and ERS SAP Services can run on either node (sysV)”

1. **On the instance where the ASCS was installed**

   Register the missing ERS service on the node where you have installed ASCS.

   a. Temporarily mount the ERS directory (classic only):

   ```
   # mount <nfs.fqdn>:/<SID>_ERS<ers_sys_nr>  /usr/sap/<SID>/ERS<ers_sys_nr>
   ```

   b. Register the ERS service:

```
# export LD_LIBRARY_PATH=/usr/sap/<SID>/ERS<ers_sys_nr>/exe
# /usr/sap/<SID>/ERS<ers_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/SYS/profile/
<SID>_ERS<ers_sys_nr>_<ers_virt_hostname> -reg
# systemctl start SAP<SID>_<ers_sys_nr>
```

c. Check the existence and state of SAP services (example):

```
# systemctl list-unit-files SAP*
UNIT FILE                     STATE   VENDOR PRESET
SAPRHX_00.service            disabled disabled
SAPRHX_10.service            disabled disabled
SAP.slice                    static  -
3 unit files listed.
```

d. If the state is not disabled, run the following command to disable `sapservices` integration for SAP<SID>_<ascs_sys_nr> and SAP<SID>_<ers_sys_nr> on both nodes:

> **Important**
>
> Stopping these services also stops the associated SAP instances.

```
# systemctl stop SAP<SID>_<ascs_sys_nr>.service
# systemctl disable SAP<SID>_<ascs_sys_nr>.service
# systemctl stop SAP<SID>_<ers_sys_nr>.service
# systemctl disable SAP<SID>_<ers_sys_nr>.service
```

e. Unmount the ERS directory (classic only):

```
# umount /usr/sap/<SID>/ERS<ers_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount <nfs.fqdn>:/RHX_ERS10  /usr/sap/RHX/ERS10
# export LD_LIBRARY_PATH=/usr/sap/RHX/ERS10/exe
# /usr/sap/RHX/ERS10/exe/sapstartsrv pf=/usr/sap/RHX/SYS/profile/
RHX_ERS10_rhxers -reg
# systemctl start SAPRHX_10
# systemctl stop SAPRHX_00.service
# systemctl disable SAPRHX_00.service
# systemctl stop SAPRHX_10.service
# systemctl disable SAPRHX_10.service
# umount /usr/sap/RHX/ERS10
```

2. **On the instance where the ERS was installed**

   Register the missing ASCS service on the node where you have installed ERS.

   a. Temporarily mount the ASCS directory (classic only):

   ```
   # mount <nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr> /usr/sap/<SID>/ASCS<ascs_sys_nr>
   ```

   b. Register the ASCS service:

   ```
   # export LD_LIBRARY_PATH=/usr/sap/<SID>/ASCS<ascs_sys_nr>/exe
   # /usr/sap/<SID>/ASCS<ascs_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/SYS/profile/
   <SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname> -reg
   # systemctl start SAP<SID>_<ascs_sys_nr>
   ```

   c. Check the existence and state of SAP services (example):

   ```
   # systemctl list-unit-files SAP*
   UNIT FILE                    STATE    VENDOR PRESET
   SAPRHX_00.service           disabled disabled
   SAPRHX_10.service           disabled disabled
   SAP.slice                   static   -
   3 unit files listed.
   ```

   d. If the state is not disabled, run the following command to disable `sapservices` integration for `SAP<SID>_<ascs_sys_nr>` and `SAP<SID>_<ers_sys_nr>` on both nodes:

   > **Important**
   >
   > Stopping these services also stops the associated SAP instances.

   ```
   # systemctl stop SAP<SID>_<ascs_sys_nr>.service
   # systemctl disable SAP<SID>_<ascs_sys_nr>.service
   # systemctl stop SAP<SID>_<ers_sys_nr>.service
   # systemctl disable SAP<SID>_<ers_sys_nr>.service
   ```

   e. Unmount the ASCS directory (classic only):

```
# umount /usr/sap/<SID>/ASCS<ascs_sys_nr>
```

- *Example using values from Parameter Reference* :

```
# mount <nfs.fqdn>:/RHX_ASCS00 /usr/sap/RHX/ASCS00
# export LD_LIBRARY_PATH=/usr/sap/RHX/ASCS00/exe
# /usr/sap/RHX/ASCS00/exe/sapstartsrv pf=/usr/sap/RHX/SYS/profile/
RHX_ASCS00_rhxascs -reg
# systemctl start SAPRHX_00
# systemctl stop SAPRHX_00.service
# systemctl disable SAPRHX_00.service
# systemctl stop SAPRHX_10.service
# systemctl disable SAPRHX_10.service
# umount /usr/sap/RHX/ASCS00
```

Configure dependencies for Pacemaker and SAP services (systemd)

This step is required on both cluster nodes when using systemd integration.

When an EC2 instance shuts down unexpectedly, Pacemaker (the cluster resource manager) may trigger unnecessary fencing actions because it cannot distinguish between planned SAP service shutdowns and system failures. To prevent this, configure systemd dependencies that inform Pacemaker about the relationship between SAP services and cluster operations.

Create a systemd drop-in configuration for the `resource-agents-deps.target`, which is a systemd target that Pacemaker uses to understand external service dependencies:

```
# mkdir -p /etc/systemd/system/resource-agents-deps.target.d/
# cd /etc/systemd/system/resource-agents-deps.target.d/

# cat > sap_systemd_<sid>.conf <<_EOF
[Unit]
Requires=sapinit.service
After=sapinit.service
After=SAP<SID>_<ascs_sys_nr>.service
After=SAP<SID>_<ers_sys_nr>.service
_EOF

# systemctl daemon-reload
```

- *Example using values from [Parameter Reference](#) :*

```
# cat > sap_systemd_rhx.conf <<_EOF
[Unit]
Requires=sapinit.service
After=sapinit.service
After=SAPRHX_00.service
After=SAPRHX_10.service
_EOF

# systemctl daemon-reload
```

(Alternative) Ensure ASCS and ERS SAP Services can run on either node (sysV)

This is only applicable for if systemd integration is not in place.

To ensure that SAP instance can be managed by the cluster and also manually during planned maintenance activities, add the missing entries for ASCS and ERS `sapstartsrv` service in `/usr/sap/sapservices` file on both cluster nodes (ASCS and ERS host). Copy the missing entry from both hosts. Post-modifications, the `/usr/sap/sapservices` file looks as follows on both hosts:

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/<SID>/ASCS<ascs_sys_nr>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<SID>/ASCS<ascs_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/
SYS/profile/<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname> -D -u <sid>adm
LD_LIBRARY_PATH=/usr/sap/<SID>/ERS<ers_sys_nr>/exe:$LD_LIBRARY_PATH; export
 LD_LIBRARY_PATH; /usr/sap/<SID>/ERS<ers_sys_nr>/exe/sapstartsrv pf=/usr/sap/<SID>/SYS/
profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname> -D -u <sid>adm
```

- *Example using values from [Parameter Reference](#) :*

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/RHX/ASCS00/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH; /
usr/sap/RHX/ASCS00/exe/sapstartsrv pf=/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs -D
 -u rhxadm
LD_LIBRARY_PATH=/usr/sap/RHX/ERS10/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH; /
usr/sap/RHX/ERS10/exe/sapstartsrv pf=/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers -D -u
 rhxadm
```

Cluster Node Setup

Establish cluster communication between nodes using Corosync and configure required authentication.

Topics

- Change the hacluster Password
- Setup Passwordless Authentication
- Start and Enable the pcsd Service
- Authorize the Cluster
- Generate Corosync Configuration
- Start and Verify the Cluster
- Configure Cluster Services
- Verify Cluster Status

Change the hacluster Password

On all cluster nodes, change the password of the operating system user hacluster:

```
# passwd hacluster
```

Setup Passwordless Authentication

Red Hat cluster tools provide comprehensive reporting and troubleshooting capabilities for cluster activity. Many of these tools require passwordless SSH access between nodes to collect cluster-wide information effectively. Red Hat recommends configuring passwordless SSH for the root user to enable seamless cluster diagnostics and reporting.

For more details, see Red Hat Documentation How to setup SSH Key passwordless login in Red Hat Enterprise Linux.

> ⚠ **Warning**
>
> Review the security implications for your organization, including root access controls and network segmentation, before implementing this configuration.

Start and Enable the pcsd Service

On all cluster nodes, enable and start the pcsd service:

```
# systemctl enable pcsd --now
```

Authorize the Cluster

Run the following command to authenticate the cluster nodes. You will be prompted for the hacluster password you set earlier:

```
# pcs host auth <hostname_1> <hostname_2> -u hacluster -p <password>
```

- *Example using values from [Parameter Reference](#)* :

  ```
  # pcs host auth rhxhost01 rhxhost02 -u hacluster -p <password>
  ```

Generate Corosync Configuration

Corosync provides membership and member-communication needs for high availability clusters. Initial setup can be performed using the following command with dual network rings for redundant communication:

```
# pcs cluster setup <cluster_name> \
<hostname_1> addr=<host_ip_1> addr=<host_additional_ip_1> \
<hostname_2> addr=<host_ip_2> addr=<host_additional_ip_2>
```

- *Example using values from [Parameter Reference](#)* :

  ```
  # pcs cluster setup myCluster rhxhost01 addr=10.1.10.1 addr=10.1.10.2 rhxhost02
   addr=10.1.20.1 addr=10.1.20.2
  Destroying cluster on hosts: 'rhxhost01', 'rhxhost02'...
  rhxhost01: Successfully destroyed cluster
  rhxhost02: Successfully destroyed cluster
  Requesting remove 'pcsd settings' from 'rhxhost01', 'rhxhost02'
  rhxhost01: successful removal of the file 'pcsd settings'
  rhxhost02: successful removal of the file 'pcsd settings'
  Sending 'corosync authkey', 'pacemaker authkey' to 'rhxhost01', 'rhxhost02'
  rhxhost01: successful distribution of the file 'corosync authkey'
  rhxhost01: successful distribution of the file 'pacemaker authkey'
  ```

```
rhxhost02: successful distribution of the file 'corosync authkey'
rhxhost02: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to 'rhxhost01', 'rhxhost02'
rhxhost01: successful distribution of the file 'corosync.conf'
rhxhost02: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
```

The timing parameters are optimized for Amazon cloud environments. Update the token timeout to provide reliable cluster operation while accommodating normal cloud network characteristics:

```
# pcs cluster config update totem token=15000
```

Start and Verify the Cluster

Start the cluster on all nodes:

```
# pcs cluster start --all
```

Note

By enabling the pacemaker service, the server automatically joins the cluster after a reboot. This ensures that your system is protected. Alternatively, you can start the pacemaker service manually on boot to investigate the cause of any failure.

Run the following command to check the cluster status:

```
# pcs status
```

Example output:

```
Cluster name: myCluster

WARNINGS:
No stonith devices and stonith-enabled is not false

Cluster Summary:
  * Stack: corosync
  * Current DC: rhxhost01 (version 2.1.2-4.el9_0.5-ada5c3b36e2) - partition with quorum
  * Last updated: Fri Oct 24 06:35:46 2025
  * Last change:  Fri Oct 24 06:26:38 2025 by hacluster via crmd on rhxhost01
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ rhxhost01 rhxhost02 ]

Full List of Resources:
  * No resources

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

Both cluster nodes must show up as online. You can find the ring status and the associated IP addresses of the cluster with the corosync-cfgtool command:

```
# corosync-cfgtool -s
```

Example output:

```
Local node ID 1, transport knet
LINK ID 0 udp
        addr    = 10.1.10.114
        status:
                nodeid:          1:     localhost
                nodeid:          2:     connected
LINK ID 1 udp
        addr    = 10.1.10.215
        status:
                nodeid:          1:     localhost
                nodeid:          2:     connected
```

Both network rings should report "active with no faults". If either ring is missing, review the corosync configuration and check that `/etc/corosync/corosync.conf` changes have been synced to the secondary node. You may need to do this manually. Restart the cluster if needed.

Configure Cluster Services

Enable pacemaker to start automatically after reboot:

```
# pcs cluster enable --all
```

Enabling pacemaker also handles corosync through service dependencies. The cluster will start automatically after reboot. For troubleshooting scenarios, you can choose to manually start services after boot instead.

Verify Cluster Status

1. Check pacemaker service status:

```
# systemctl status pacemaker
```

2. Verify cluster status:

```
# pcs status
```

Example output:

```
Cluster name: myCluster
Cluster Summary:
  * Stack: corosync
  * Current DC: rhxhost01 (version 2.1.5+20221208.a3f44794f) - partition with quorum
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ rhxhost01 rhxhost02 ]

Full List of Resources:
  * No resources
```

Cluster Configuration

The following sections provide details on the resources, groups and constraints necessary to ensure high availability of SAP Central Services.

Topics

- Create STONITH Fencing Resource
- SAP Resource Groups and Ordering
- Create Filesystem resources (classic only)
- Create overlay IP resources
- Create SAPStartSrv resources (simple-mount only)
- Create SAPInstance resources (simple-mount only)
- Create SAPInstance resources (classic only)
- Review ASCS Resource group and modify stickiness.
- Create resource constraints
- Reset Configuration – Optional

Prepare for Resource Creation

To ensure that the cluster does not perform any unexpected actions during setup of resources and configuration, set the maintenance mode to true.

Run the following command to put the cluster in maintenance mode:

```
# pcs property set maintenance-mode=true
```

To verify the current maintenance state:

```
$ pcs status
```

Note

There are two types of maintenance mode:

- Cluster-wide maintenance (set with `pcs property set maintenance-mode=true`)
- Node-specific maintenance (set with `pcs node maintenance nodename`)

Always use cluster-wide maintenance mode when making configuration changes. For node-specific operations like hardware maintenance, refer to the Operations section for proper procedures.
To disable maintenance mode after configuration is complete:

```
# pcs property set maintenance-mode=false
```

Cluster Bootstrap

Configure Cluster Properties

Configure cluster properties to establish fencing behavior and resource failover settings:

```
# pcs property set stonith-enabled="true"
# pcs property set stonith-timeout="600"
# pcs property set priority-fencing-delay="20"
```

- The **priority-fencing-delay** is recommended for protecting the SAP ASCS nodes during network partitioning events. When a cluster partition occurs, this delay gives preference to nodes hosting higher priority resources, with the ASCS receiving additional priority weighting over the ERS . This helps ensure the ASCS node survives in split-brain scenarios. The recommended 20 second priority-fencing-delay works in conjunction with the pcmk_delay_max (10 seconds) configured in the stonith resource, providing a total potential delay of up to 30 seconds before fencing occurs

To verify your cluster property settings:

```
# pcs property config
# pcs property config <property_name>
```

Configure Resource Defaults

Configure resource default behaviors:

RHEL 8.4 and above

```
# pcs resource defaults update resource-stickiness="1"
# pcs resource defaults update migration-threshold="3"
# pcs resource defaults update failure-timeout="600s"
```

RHEL 7.x and RHEL 8.0 to 8.3

```
# pcs resource defaults resource-stickiness="1"
# pcs resource defaults migration-threshold="3"
# pcs resource defaults failure-timeout="600s"
```

- The **resource-stickiness** value of 1 encourages the ASCS resource to stay on its current node, avoiding unnecessary resource movement.
- The **migration-threshold** causes a resource to move to a different node after 3 consecutive failures, ensuring timely failover when issues persist.
- The **failure-timeout** automatically removes a failure count after 10 minutes, preventing individual historical failures from accumulating and affecting long-term resource behavior. If testing failover scenarios in quick succession, it may be necessary to manually query and clear accumulated failure counts between tests. Use `pcs resource failcount` and `pcs resource refresh`.

Individual resources may override these defaults with their own defined values.

To verify your resource default settings:

```
# pcs resource defaults
```

Configure Operation Defaults

```
# pcs resource op defaults update timeout="600"
```

- The **op_defaults timeout** ensures all cluster operations have a reasonable default timeout of 600 seconds. Individual resources may override this with their own timeout values.

To verify your operation default settings:

```
# pcs resource op defaults
```

Create STONITH Fencing Resource

An Amazon STONITH resource is required for proper cluster fencing operations. The `fence_aws` resource is recommended for Amazon deployments as it leverages the Amazon API to safely fence failed or incommunicable nodes by stopping their EC2 instances.

Create the STONITH resource using resource agent **`fence_aws`** :

```
# pcs stonith create <stonith_resource_name> fence_aws \
pcmk_host_map="<hostname_1>:<instance_id_1>;<hostname_2>:<instance_id_2>" \
region="<aws_region>" \
skip_os_shutdown="true" \
pcmk_delay_max="10" \
pcmk_reboot_timeout="300" \
pcmk_reboot_retries="2" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="180" timeout="60"
```

Details:

- **pcmk_host_map** - Maps cluster node hostnames to their EC2 instance IDs. This mapping must be unique within the Amazon account and follow the format hostname:instance-id, with multiple entries separated by semicolons.
- **region** - Amazon region where the EC2 instances are deployed
- **pcmk_delay_max** - Random delay before fencing operations. Works in conjunction with cluster property `priority-fencing-delay` to prevent simultaneous fencing. Historically set to higher values, but with `priority-fencing-delay` now handling primary node protection, a lower value (10s) is sufficient.
- **pcmk_reboot_timeout** - Maximum time in seconds allowed for a reboot operation
- **pcmk_reboot_retries** - Number of times to retry a failed reboot operation
- **skip_os_shutdown** (NEW) - Leverages a new ec2 stop-instance API flag to forcefully stop an EC2 Instance by skipping the shutdown of the Operating System.
  - Red Hat Solution 4963741 - fence_aws fence action fails with "Timed out waiting to power OFF" (requires Red Hat Customer Portal access)

ENSA1

Example using values from Parameter Reference :

```
# pcs stonith create rsc_fence_aws fence_aws \
pcmk_host_map="rhxhost01:i-xxxxinstidforhost1;rhxhost02:i-xxxxinstidforhost2" \
region="us-east-1" \
skip_os_shutdown="true" \
pcmk_delay_max="30" \
pcmk_reboot_timeout="120" \
pcmk_reboot_retries="4" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="180" timeout="60"
```

ENSA2

Example using values from Parameter Reference :

```
# pcs stonith create rsc_fence_aws fence_aws \
pcmk_host_map="rhxhost01:i-xxxxinstidforhost1;rhxhost02:i-xxxxinstidforhost2" \
region="us-east-1" \
skip_os_shutdown="true" \
pcmk_delay_max="10" \
pcmk_reboot_timeout="120" \
pcmk_reboot_retries="4" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="180" timeout="60"
```

SAP Resource Groups and Ordering

When creating the resources for the SAP ASCS and ERS, it is necessary to specify a group.

A cluster resource group is a set of resources that need to be located together, start sequentially, and stopped in the reverse order.

Depending on the configuration pattern the following groups will be created for the ASCS and ERS

- **Classic**: Filesystem, IP, SAPInstance
- **SimpleMount**: IP, SAPStartSrv, SAPInstance

Since RHEL 9.4 a new syntax for creating a resource in a group has been introduced in addition to the --group parameter. You receive the following deprecation warning now:

```
Deprecation Warning: Using '--group' is deprecated and will be replaced with 'group' in
 a future release. Specify --future to switch to the future behavior.
```

Create Filesystem resources (classic only)

In classic configuration, the mounting and unmounting of file system resources to align with the location of the SAP services is done using cluster resources.

Create **ASCS** file system resources:

```
# pcs resource create rsc_fs_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:Filesystem \
device="<nfs.fqdn>:/<SID>_ASCS<ascs_sys_nr>" \
directory="/usr/sap/<SID>/ASCS<ascs_sys_nr>" \
fstype="nfs4" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
force_unmount="safe" \
fast_stop="no" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40" \
--group "grp_<SID>_ASCS<ascs_sys_nr>"
```

Create **ERS** file system resources:

```
# pcs resource create rsc_fs_<SID>_ERS<ers_sys_nr> ocf:heartbeat:Filesystem \
device="<nfs.fqdn>:/<SID>_ERS<ers_sys_nr>" \
directory="/usr/sap/<SID>/ERS<ers_sys_nr>" \
fstype="nfs4" \
force_unmount="safe" \
fast_stop="no" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40" \
--group "grp_<SID>_ERS<ers_sys_nr>"
```

- *Example using values from Parameter Reference* :

```
# pcs resource create rsc_fs_RHX_ASCS00 ocf:heartbeat:Filesystem \
device="fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_ASCS00" \
directory="/usr/sap/RHX/ASCS00" \
fstype="nfs4" \
force_unmount="safe" \
fast_stop="no" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40"

# pcs resource create rsc_fs_RHX_ERS10 ocf:heartbeat:Filesystem \
device="fs-xxxxxxxxxxxxxefs1.efs.us-east-1.amazonaws.com:/RHX_ERS10" \
directory="/usr/sap/RHX/ERS10" \
fstype="nfs4" \
force_unmount="safe" \
fast_stop="no" \
options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
op start timeout="60" interval="0" \
op stop timeout="60" interval="0" \
op monitor interval="20" timeout="40"
```

Notes

- Review the mount options to ensure that they match with your operating system, NFS file system type, and the latest recommendations from SAP.
- <nfs.fqdn> can either be an alias or the default file system resource name of the NFS or FSx for ONTAP resource. For example, `fs-xxxxxx.efs.xxxxxx.amazonaws.com`.
- `force_unmount` and `fast_stop` are recommendations for ensuring the filesystem can be quickly unmounted. See Red Hat solutions:
  - Red Hat Solution 3357961 - During failover of a pacemaker resources, a Filesystem resource kills processes not using the filesystem (requires Red Hat customer portal login)
  - Red Hat Solution 4801371 - What is the fast_stop option for a Filesystem resource in a Pacemaker cluster? (requires Red Hat customer portal login)

Create overlay IP resources

The IP resource provides the details necessary to update the route table entry for overlay IP.

Create **ASCS** IP Resource:

```
# pcs resource create rsc_ip_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:aws-vpc-move-ip \
ip="<ascs_overlayip>" \
routing_table="<routetable_id>" \
interface="eth0" \
profile="<cli_cluster_profile>" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"
--group "grp_<SID>_ASCS<ascs_sys_nr>"
```

Create **ERS** IP Resource:

```
# pcs resource create rsc_ip_<SID>_ERS<ers_sys_nr> ocf:heartbeat:aws-vpc-move-ip \
ip="<ers_overlayip>" \
routing_table="<routetable_id>" \
interface="eth0" \
profile="<cli_cluster_profile>" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40" \
--group "grp_<SID>_ERS<ers_sys_nr>"
```

- *Example using values from [Parameter Reference](#)* :

```
# pcs resource create rsc_ip_RHX_ASCS00 ocf:heartbeat:aws-vpc-move-ip \
ip="172.16.30.5" \
routing_table="rtb-xxxxxroutetable1" \
interface="eth0" \
profile="cluster" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40" \
--group grp_RHX_ASCS00

# pcs resource create rsc_ip_RHX_ERS10 ocf:heartbeat:aws-vpc-move-ip \
ip="172.16.30.6" \
routing_table="rtb-xxxxxroutetable1" \
interface="eth0" \
profile="cluster" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"
```

- *Example using values from [Parameter Reference](#)* :

```
# pcs resource create rsc_ip_RHX_ASCS00 ocf:heartbeat:aws-vpc-move-ip \
ip="172.16.30.5" \
routing_table="rtb-xxxxxroutetable1" \
interface="eth0" \
profile="cluster" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40" \
--group grp_RHX_ASCS00

# pcs resource create rsc_ip_RHX_ERS10 ocf:heartbeat:aws-vpc-move-ip \
ip="172.16.30.6" \
routing_table="rtb-xxxxxroutetable1" \
interface="eth0" \
profile="cluster" \
op start interval="0" timeout="180" \
op stop interval="0" timeout="180" \
op monitor interval="20" timeout="40"
```

Notes

- If more than one route table is required for connectivity or because of subnet associations, the `routing_table` parameter can have multiple values separated by a comma. For example, `routing_table=rtb-xxxxxroutetable1,rtb-xxxxxroutetable2`.
- Additional parameters – `lookup_type` and `routing_table_role` are required for shared VPC. For more information, see {https---docs-aws-amazon-com-sap-latest-sap-netweaver-rhel-netweaver-ha-settings-html-rhel-netweaver-ha-shared-vpc}[Shared VPC – optional].

Create SAPStartSrv resources (simple-mount only)

In simple-mount architecture, the `sapstartsrv` process that is used to control start/stop and monitoring of an SAP instance, is controlled by a cluster resource. This new resource adds additional control that removes the requirement for file system resources to be restricted to a single node.

Modify and run the commands in the table to create `sapstartsrv` resource.

Create **ASCS** SAPStartSrv Resource

Use the following command to create an ASCS SAPStartSrv resource.

```
# pcs resource create rsc_sapstart_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPStartSrv \
InstanceName=<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>
op monitor interval=0 timeout=20 enabled=0
--group grp_<SID>_ASCS<instance>
```

Create **ERS** SAPStartSrv Resource

Use the following command to create an ERS SAPStartSrv resource.

```
# pcs resource create rsc_sapstart_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPStartSrv \
InstanceName=<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>
op monitor interval=0 timeout=20 enabled=0
--group grp_<SID>_ERS<ers_sys_nr>
```

- *Example using values from [Parameter Reference](#)* :

```
#crm configure primitive rsc_sapstart_RHX_ASCS00 ocf:heartbeat:SAPStartSrv \
params \
InstanceName=RHX_ASCS00_rhxascs \
op monitor interval=0 timeout=20 enabled=0 \
--group grp_RHX_ASCS00

#crm configure primitive rsc_sapstart_RHX_ERS10 ocf:heartbeat:SAPStartSrv \
params \
InstanceName=RHX_ERS10_rhxers \
op monitor interval=0 timeout=20 enabled=0 \
--group grp_RHX_ERS10
```

Create SAPInstance resources (simple-mount only)

The minor difference in creating SAP instance resources between classic and simple-mount configurations is the addition of `MINIMAL_PROBE=true` parameters.

The SAP instance is started and stopped using cluster resources.

Example

ENSA1

Create an **ASCS** SAP instance resource:

```
# pcs resource create rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta failure-timeout="60" \
meta migration-threshold="1" \
meta priority="10"
```

Create an **ERS** SAP instance resource:

```
# pcs resource create rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
IS_ERS="true" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta priority="1000"
```

- *Example using values from Parameter Reference* :

```
# pcs resource create rsc_sap_RHX_ASCS00 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ASCS00_rhxascs" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta failure-timeout="60" \
meta migration-threshold="1" \
meta priority="10"

# pcs resource create rsc_sap_RHX_ERS10 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ERS10_rhxers" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
IS_ERS="true" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta priority="1000"
```

ENSA2

Create an **ASCS** SAP instance resource:

```
# pcs resource create rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="20" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta priority="1000"
```

Create an **ERS** SAP instance resource:

```
# pcs resource create rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="20" timeout="60" on-fail="restart"
```

- *Example using values from Parameter Reference* :

```
# pcs resource create rsc_sap_RHX_ASCS00 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ASCS00_rhxascs" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs" \
AUTOMATIC_RECOVER="false" \
MINIMAL_PROBE="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="20" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta priority="1000"

# pcs resource create rsc_sap_RHX_ERS10 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ERS10_rhxers" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers" \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="20" timeout="60" on-fail="restart"
```

The difference between ENSA1 and ENSA2 is that ENSA2 allows the lock table to be consumed remotely, which means that for ENSA2, ASCS can restart in its current location (assuming the node is still available). This change impacts stickiness, migration and priority parameters. Ensure that you use the right command for your enqueue version.

Create SAPInstance resources (classic only)

The SAP instance is started and stopped using cluster resources.

Example

ENSA1

Create an **ASCS** SAPInstance resource:

```
# pcs resource create rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta failure-timeout="60" \
meta migration-threshold="1" \
meta priority="10" \
--group "grp_<SID>_ASCS<ascs_sys_nr>"
```

Create an **ERS** SAPInstance resource:

```
# pcs resource create rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta \
priority="1000"
--group "grp_<SID>_ERS<ers_sys_nr>"
```

- *Example using values from Parameter Reference* :

```
# pcs resource create rsc_sap_RHX_ASCS00 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ASCS00_rhxascs" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs" \
AUTOMATIC_RECOVER="false" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta failure-timeout="60" \
meta migration-threshold="1" \
meta priority="10"

# pcs resource create rsc_sap_RHX_ERS10 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ERS10_rhxers" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers" \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta priority="1000"
```

ENSA2

Create an **ASCS** SAPInstance resource:

```
# pcs resource create rsc_sap_<SID>_ASCS<ascs_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/
<SID>_ASCS<ascs_sys_nr>_<ascs_virt_hostname>" \
AUTOMATIC_RECOVER="false" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta priority="1000" \
--group "grp_<SID>_ASCS<ascs_sys_nr>"
```

Create an **ERS** SAP instance resource:

```
# pcs resource create rsc_sap_<SID>_ERS<ers_sys_nr> ocf:heartbeat:SAPInstance \
InstanceName="<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>" \
START_PROFILE="/usr/sap/<SID>/SYS/profile/<SID>_ERS<ers_sys_nr>_<ers_virt_hostname>"
 \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
--group "grp_<SID>_ERS<ers_sys_nr>"
```

- *Example using values from Parameter Reference* :

```
# pcs resource create rsc_sap_RHX_ASCS00 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ASCS00_rhxascs" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ASCS00_rhxascs" \
AUTOMATIC_RECOVER="false" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart" \
meta resource-stickiness="5000" \
meta priority="1000"

# pcs resource create rsc_sap_RHX_ERS10 ocf:heartbeat:SAPInstance \
InstanceName="RHX_ERS10_rhxers" \
START_PROFILE="/usr/sap/RHX/SYS/profile/RHX_ERS10_rhxers" \
AUTOMATIC_RECOVER="false" \
IS_ERS="true" \
op start interval="0" timeout="600" \
op stop interval="0" timeout="240" \
op monitor interval="11" timeout="60" on-fail="restart"
```

The change between ENSA1 and ENSA2 allows the lock table to be consumed remotely. If the node is still available, ASCS can restart in its current location for ENSA2. This impacts stickiness, migration, and priority parameters. Make sure to use the right command, depending on your enqueue server.

Review ASCS Resource group and modify stickiness.

A cluster resource group is a set of resources that need to be located together, start sequentially, and stopped in the reverse order.

```
# pcs resource meta grp_<SID>_ASCS<ascs_sys_nr> resource-stickiness=3000
```

In simple-mount architecture, the overlay IP must be available first, then the SAP services are started before the SAP instance can start.

Create resource constraints

Resource constraints are used to determine where resources run per the conditions. Constraints for SAP NetWeaver ensure that ASCS and ERS are started on separate nodes and locks are preserved in case of failures. The following are the different types of constraints.

Colocation constraint

The negative score ensures that ASCS and ERS are run on separate nodes, wherever possible.

```
# pcs constraint colocation add grp_<SID>_ERS<ers_sys_nr> with
 grp_<SID>_ASCS<ascs_sys_nr> score=-5000
```

- *Example using values from Parameter Reference* :

  ```
  # pcs constraint colocation add grp_RHX_ERS10 with grp_RHX_ASCS00 score=-5000
  ```

Order constraint

This constraint ensures the ASCS instance is started prior to stopping the ERS instance. This is necessary to consume the lock table.

```
# pcs constraint order start rsc_sap_<SID>_ASCS<ascs_sys_nr> then stop
 rsc_sap_<SID>_ERS<ers_sys_nr> kind=Optional symmetrical=false
```

- *Example using values from Parameter Reference* :

  ```
  # pcs constraint order start rsc_sap_RHX_ASCS00 then stop rsc_sap_RHX_ERS10
   kind=Optional symmetrical=false
  ```

Location constraint (ENSA1 only)

This constraint is only required for ENSA1. The lock table can be retrieved remotely for ENSA2, and as a result ASCS doesn’t failover to where ERS is running.

```
# pcs constraint location rsc_sap_<SID>_ASCS<ascs_sys_nr> rule score=2000
 runs_ers_<SID> eq 1
```

- *Example using values from Parameter Reference* :

  ```
  # pcs constraint location rsc_sap_RHX_ASCS00 rule score=2000 runs_ers_RHX eq 1
  ```

Reset Configuration – Optional

> **Important**
>
> The following instructions help you reset the complete configuration. Run these commands only if you want to start setup from the beginning. You can make minor changes with the crm edit command.

Run the following command to back up the current configuration for reference:

```
# pcs config > /tmp/pcsconfig_backup.txt
```

Run the following command to clear the current configuration:

```
# pcs cluster cib-push --config /dev/null
```

Once the preceding command is executed, it removes all of the cluster resources from Cluster Information Base (CIB). Before starting the resource configuration, run pcs cluster start --all to ensure the cluster is running properly. The restart removes maintenance mode. Reapply maintenance mode before commencing additional configuration and resource setup.

Operations

This section covers the following topics.

Topics

- Viewing the cluster state
- Performing planned maintenance
- Post-failure analysis and reset
- Alerting and monitoring

Viewing the cluster state

You can view the state of the cluster in two ways - based on your operating system or with a web based console provided by Red Hat.

Topics

- Operating system based
- Red Hat Cockpit

Operating system based

There are multiple operating system commands that can be run as root or as a user with appropriate permissions. The commands enable you to get an overview of the status of the cluster and its services. See the following commands for more details.

```
# pcs status
```

Sample output:

```
rhxhost01:~ # pcs status
Cluster name: rhx-cluster
Cluster Summary:
  * Stack: corosync
  * Current DC: rhxhost01 (version 2.1.0-8.el8-7c3f660707) - partition with quorum
  * Last updated: Tue Nov  1 13:41:58 2022
  * Last change:  Fri Oct 28 08:55:43 2022 by root via crm_attribute on rhxhost02
  * 2 nodes configured
  * 7 resource instances configured

Node List:
  * Online: [ rhxhost01 rhxhost02 ]

Full List of Resources:
  * Resource Group: grp_RHX_ASCS00:
    * rsc_ip_RHX_ASCS00 (ocf::heartbeat:aws-vpc-move-ip):        Started rhxhost01
    * rsc_sapstart_RHX_ASCS00   (ocf::heartbeat:SAPStartSrv):         Started rhxhost01
    * rsc_sap_RHX_ASCS00        (ocf::heartbeat:SAPInstance):    Started rhxhost01
  * res_AWS_STONITH     (stonith:fence_aws):  Started rhxhost02
  * Resource Group: grp_RHX_ERS10:
    * rsc_ip_RHX_ERS10  (ocf::heartbeat:aws-vpc-move-ip):        Started rhxhost02
    * rsc_sapstart_RHX_ERS10    (ocf::heartbeat:SAPStartSrv):         Started rhxhost02
    * rsc_sap_RHX_ERS10 (ocf::heartbeat:SAPInstance):    Started rhxhost02
```

The following table provides a list of useful commands.

Command	Description
`pcs status`	Display cluster status on the console
`pcs status --full`	Display detailed cluster status including inactive resources
`pcs status nodes`	Display node status and attributes
`pcs status resources`	Display resource status and fail counts
`pcs cluster status`	Display cluster daemon status
`pcs help`	View more options

Command	Description
`pcs status --help`	View more options

Red Hat Cockpit

Cockpit is a web-based graphical user interface for managing and monitoring Red Hat Enterprise Linux systems, including pacemaker highly availability clusters. It must be enabled on every node in the cluster, to point your web browser on any node for accessing it. Use the following command to enable Cockpit.

```
# systemctl enable --now cockpit.socket
# systemctl status cockpit.socket
```

Use the following URL to check security groups for access on port 9090 from your administrative host.

```
https://your-server:9090/

e.g https://rhxhost01:9090
```

For more information, see Configuring and Managing High Availability Clusters in the Red Hat Documentation.

Performing planned maintenance

The cluster connector is designed to integrate the cluster with SAP start framework (`sapstartsrv`), including the rolling kernel switch (RKS) awareness. Stopping and starting the SAP system using `sapcontrol` should not result in any cluster remediation activities as these actions are not interpreted as failures. Validate this scenario when testing your cluster.

There are different options to perform planned maintenance on nodes, resources, and the cluster.

Topics

Maintenance mode

Use maintenance mode if you want to make any changes to the configuration or take control of the resources and nodes in the cluster. In most cases, this is the safest option for administrative tasks.

Example

On

Use one of the following commands to turn on maintenance mode.

```
# pcs property set maintenance-mode=true
```

```
# pcs cluster maintenance --all
```

Off

Use one of the following commands to turn off maintenance mode.

```
# pcs property set maintenance-mode=false
```

```
# pcs cluster maintenance --all --wait=60
```

Placing a node in standby mode

To perform maintenance on the cluster without system outage, the recommended method for moving active resources is to place the node you want to remove from the cluster in standby mode.

```
# pcs node standby <hostname>
```

The cluster will cleanly relocate resources, and you can perform activities, including reboots on the node in standby mode. When maintenance activities are complete, you can re-introduce the node with the following command.

```
# pcs node unstandby <hostname>
```

Moving a resource

Moving individual resources is not recommended because of the migration or move constraints that are created to lock the resource in its new location. These can be cleared as described in the info messages, but this introduces an additional setup.

```
<rhxhost01>:~ pcs resource move grp_<RHX>_ASCS<00> <rhxhost02>
Location constraint to move resource 'grp_<RHX>_ASCS<00>' has been created
Run 'pcs resource clear grp_<RHX>_ASCS<00>' to remove this constraint
```

Use the following command once the resources have relocated to their target location.

```
# pcs resource clear grp_RHX_ASCS00
```

Post-failure analysis and reset

A review must be conducted after each failure to understand the source of failure as well the reaction of the cluster. In most scenarios, the cluster prevents an application outage. However, a manual action is often required to reset the cluster to a protective state for any subsequent failures.

Topics

- Checking the logs
- Cleanup pcs status
- Restart failed nodes or pacemaker
- Further Analysis

Checking the logs

- For troubleshooting cluster issues, use journalctl to examine both pacemaker and corosync logs:

  ```
  # journalctl -u pacemaker -u corosync --since "1 hour ago"
  ```

  - Use `--since` to specify time periods (e.g., "2 hours ago", "today")
  - Add `-f` to follow logs in real-time
  - Combine with grep for specific searches
- System messages and resource agent activity can be found in `/var/log/messages`.

Application based failures can be investigated in the SAP work directory.

Cleanup pcs status

If failed actions are reported using the `pcs status` command, and if they have already been investigated, then you can clear the reports with the following command.

```
# pcs resource cleanup <resource> <hostname>
```

Restart failed nodes or pacemaker

It is recommended that failed (or fenced) nodes are not automatically restarted. It gives operators a chance to investigate the failure, and ensure that the cluster doesn't make assumptions about the state of resources.

You need to restart the instance or the pacemaker service based on your approach.

Further Analysis

For cluster-specific issues, use `sosreport` to generate a targeted analysis of cluster components:

```
# sosreport --batch --tmp-dir /tmp
```

For quick analysis of recent events, you can use:

```
# pcs status --full
# journalctl -u pacemaker --since "1 hour ago"
```

- `sosreport` collects system configuration and diagnostic information
- For more information, see Red Hat Documentation - What is sosreport and how to create and retrieve one

Alerting and monitoring

This section covers the following topics.

Topics

- Using Amazon CloudWatch Application Insights
- Using the cluster alert agents

Using Amazon CloudWatch Application Insights

For monitoring and visibility of cluster state and actions, Application Insights includes metrics for monitoring enqueue replication state, cluster metrics, and SAP and high availability checks. Additional metrics, such as EFS and CPU monitoring can also help with root cause analysis.

For more information, see Get started with Amazon CloudWatch Application Insights and SAP NetWeaver High Availability on Amazon EC2.

Using the cluster alert agents

Within the cluster configuration, you can call an external program (an alert agent) to handle alerts. This is a *push* notification. It passes information about the event via environment variables.

The agents can then be configured to send emails, log to a file, update a monitoring system, etc. For example, the following script can be used to access Amazon SNS.

```
#!/bin/sh

# alert_sns.sh
# modified from /usr/share/pacemaker/alerts/alert_smtp.sh.sample

##############################################################################
# SETUP
# * Create an SNS Topic and subscribe email or chatbot
# * Note down the ARN for the SNS topic
# * Give the IAM Role attached to both Instances permission to publish to the SNS Topic
# * Ensure the aws cli is installed
# * Copy this file to /usr/share/pacemaker/alerts/alert_sns.sh or other location on
 BOTH nodes
# * Ensure the permissions allow for hacluster and root to execute the script
# * Run the following as root (modify file location if necessary and replace SNS ARN):
#
# SLES:
# crm configure alert aws_sns_alert /usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S" to <{ arn:aws:sns:region:account-
id:myPacemakerAlerts  }>
#
# RHEL:
# pcs alert create id=aws_sns_alert path=/usr/share/pacemaker/alerts/alert_sns.sh meta
 timeout=30s timestamp-format="%Y-%m-%d_%H:%M:%S"
# pcs alert recipient add aws_sns_alert value=arn:aws:sns:region:account-
id:myPacemakerAlerts
```

```
##############################################################################

# Additional information to send with the alerts
node_name=`uname -n`
sns_body=`env | grep CRM_alert_`

# Required for SNS
TOKEN=$(/usr/bin/curl --noproxy '*' -s -X PUT "http://169.254.169.254/latest/api/token"
 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Get metadata
REGION=$(/usr/bin/curl --noproxy '*' -w "\n" -s -H "X-aws-ec2-metadata-token: $TOKEN"
 http://169.254.169.254/latest/dynamic/instance-identity/document | grep region | awk -
F\" '{print $4}')

sns_subscription_arn=${CRM_alert_recipient}

# Format depending on alert type
case ${CRM_alert_kind} in
   node)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Node '${CRM_alert_node}' is
 now '${CRM_alert_desc}'"
   ;;
   fencing)
     sns_subject="${CRM_alert_timestamp} ${cluster_name}: Fencing ${CRM_alert_desc}"
   ;;
   resource)
     if [ ${CRM_alert_interval} = "0" ]; then
         CRM_alert_interval=""
     else
         CRM_alert_interval=" (${CRM_alert_interval})"
     fi
     if [ ${CRM_alert_target_rc} = "0" ]; then
         CRM_alert_target_rc=""
     else
         CRM_alert_target_rc=" (target: ${CRM_alert_target_rc})"
     fi
     case ${CRM_alert_desc} in
         Cancelled)
           ;;
         *)
           sns_subject="${CRM_alert_timestamp}: Resource operation
 '${CRM_alert_task}${CRM_alert_interval}' for '${CRM_alert_rsc}' on
 '${CRM_alert_node}': ${CRM_alert_desc}${CRM_alert_target_rc}"
```

```
           ;;
     esac
     ;;
   attribute)
     sns_subject="${CRM_alert_timestamp}: The '${CRM_alert_attribute_name}' attribute
 of the '${CRM_alert_node}' node was updated in '${CRM_alert_attribute_value}'"
     ;;
   *)
     sns_subject="${CRM_alert_timestamp}: Unhandled $CRM_alert_kind alert"
     ;;
esac

# Use this information to send the email.
aws sns publish --topic-arn "${sns_subscription_arn}" --subject "${sns_subject}" --
message "${sns_body}" --region ${REGION}
```

Testing

We recommend scheduling regular fault scenario recovery testing at least annually, and as part of the operating system or SAP kernel updates that may impact operations. For more details on best practices for regular testing, see SAP Lens – Best Practice 4.3 – Regularly test business continuity plans and fault recovery.

The tests described here simulate failures. These can help you understand the behavior and operational requirements of your cluster.

In addition to checking the state of cluster resources, ensure that the service you are trying to protect is in the required state. Can you still connect to SAP? Are locks still available in SM12?

Define the recovery time to ensure that it aligns with your business objectives. Record recovery actions in runbooks.

Topics

- Test 1: Stop ASCS on the primary node using sapcontrol
- Test 2: Stop ERS on the secondary node using sapcontrol
- Test 3: Kill the message server process on the primary node
- Test 4: Kill the enqueue server process on the primary node
- Test 5: Kill the ER process
- Test 6: Simulate hardware failure of an individual node, and repeat for other node

- Test 7: Simulate a network failure
- Test 8: Simulate an NFS failure
- Test 9: Accidental shutdown

Test 1: Stop ASCS on the primary node using `sapcontrol`

Notes – Ensure that the connector has been installed and the parameters have been updated.

Simulate failure – On `rhxhost01` as `rhxadm`:

```
sapcontrol -nr <00> -function Stop
```

Expected behavior – ASCS should be stopped on `rhxhost01`, and the cluster should not perform any activity.

Recovery action – Start ASCS manually.

Test 2: Stop ERS on the secondary node using `sapcontrol`

Notes – Ensure that the connector has been installed, and the parameters are updated.

Simulate failure – On `rhxhost02` as `rhxadm`:

```
sapcontrol -nr <10> -function Stop
```

Expected behavior – ERS should be stopped on `rhxhost02`, and the cluster should not perform any activity.

Recovery action – Start ERS manually.

Test 3: Kill the message server process on the primary node

Simulate failure – On `rhxhost01` as `rhxadm`:

```
kill -9 $(pgrep -f "ms.sap<RHX>_ASCS<00>")
```

Expected behavior – The message server should immediately respawn based on the Restart parameter.

Recovery action – No action required.

Test 4: Kill the enqueue server process on the primary node

Notes – Check that locks have persisted, and review the location constraints that only exist for ENSA1.

Simulate failure – On `rhxhost01` as `rhxadm`:

```
kill -9 $(pgrep -f "[en|enq].sap<RHX>_ASCS<00>")
```

Expected behavior – ENSA2: Cluster will restart the ENQ process and retrieve the locks remotely. ENSA1: Cluster will failover the ASCS resource to the node where the ERS is running.

Recovery action – No action required.

Test 5: Kill the ER process

Simulate failure – On `rhxhost02` as `rhxadm`:

```
kill -9 $(pgrep -f "[er|enqr].sap<RHX>_ERS<10>")
```

Expected behavior – Cluster will restart the ERS on the same node.

Recovery action – No action required.

Test 6: Simulate hardware failure of an individual node, and repeat for other node

Notes – To simulate a system crash, you must first ensure that `/proc/sys/kernel/sysrq` is set to 1.

Simulate failure – On both nodes as root:

```
echo 'b' > /proc/sysrq-trigger
```

Expected behavior – The node which has been killed fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 7: Simulate a network failure

Notes – See the following list.

- Iptables must be installed.
- Use a subnet in this command because of the secondary ring.
- Check for any existing iptables rules as iptables -F will flush all rules.
- Review pcmk_delay and priority parameters if you see neither node survives the fence race.

Simulate failure – On either node as root:

```
iptables -A INPUT -s <CIDR_of_other_subnet> -j DROP; iptables -A OUTPUT -d
 <CIDR_of_other_subnet> -j DROP
```

Expected behavior – The cluster detects the network failure, and fences one of the nodes to avoid a split-brain situation.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 8: Simulate an NFS failure

Notes – See the following list.

- Iptables must be installed.
- Check for any existing iptables rules as iptables -F will flush all rules.
- Although rare, this is an important scenario to test. Depending on the activity it may take some time (10 min +) to notice that I/O to EFS is not occurring and fail either the Filesystem or SAP resources.

Simulate failure – On either node as root:

```
iptables -A OUTPUT -p tcp --dport 2049 -m state --state NEW,ESTABLISHED,RELATED -j
 DROP; iptables -A INPUT -p tcp --sport 2049 -m state --state ESTABLISHED -j DROP
```

Expected behavior – The cluster detects that NFS is not available, and the SAP Instance resource agent will fail and move to the FAILED state. Because of the option "on-fail=restart" configuration, the cluster will try a local restart before eventually fencing the node and failing over.

Recovery action – If the node where the command was run survives, execute iptables -F to clear the network failure. Start the EC2 node and pacemaker service. The cluster will detect that the node is online and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Test 9: Accidental shutdown

Notes – See the following list.

- Avoid shutdowns without cluster awareness.
- We recommend the use of systemd to ensure predictable behaviour.
- Ensure the resource dependencies are in place.

Simulate failure – Login to Amazon Management Console, and stop the instance or issue a shutdown command.

Expected behavior – The node which has been shut down fails. The cluster will move the resources (ASCS/ERS) which were running on the failed node to the surviving node. If systemd and resource dependencies are not configured, you may notice that while the EC2 instance is shutting down gracefully, the cluster will detect an unclean stop of cluster services on the node and will fence the EC2 instance being shut down. For more information, see Red Hat documentation – Stopping the Cluster Services on a Node.

Recovery action – Start the EC2 node and pacemaker service. The cluster will detect that the node is online, and move the ERS resource so that the ASCS and ERS are not running on the same node (colocation constraint).

Migrate SAP NetWeaver applications with Amazon Migration Hub Orchestrator

Amazon Migration Hub Orchestrator simplifies and automates the migration of servers and enterprise applications to Amazon. It provides a single location to run and track your migrations. It helps reduce migration costs and time by automating many migration tasks. Migration Hub Orchestrator offers templates to create a migration workflow that can be customized to fit your unique migration requirements.

With Migration Hub Orchestrator, you can migrate SAP NetWeaver based applications running on SAP HANA or any other database, such as Oracle, MSSQL, SAP ASE, etc., to Amazon. For more information, see What is Amazon Migration Hub Orchestrator?

You can access Amazon Migration Hub Orchestrator from link: https://console.aws.amazon.com/migrationhub/orchestrator/ or from the Amazon Command Line Interface.

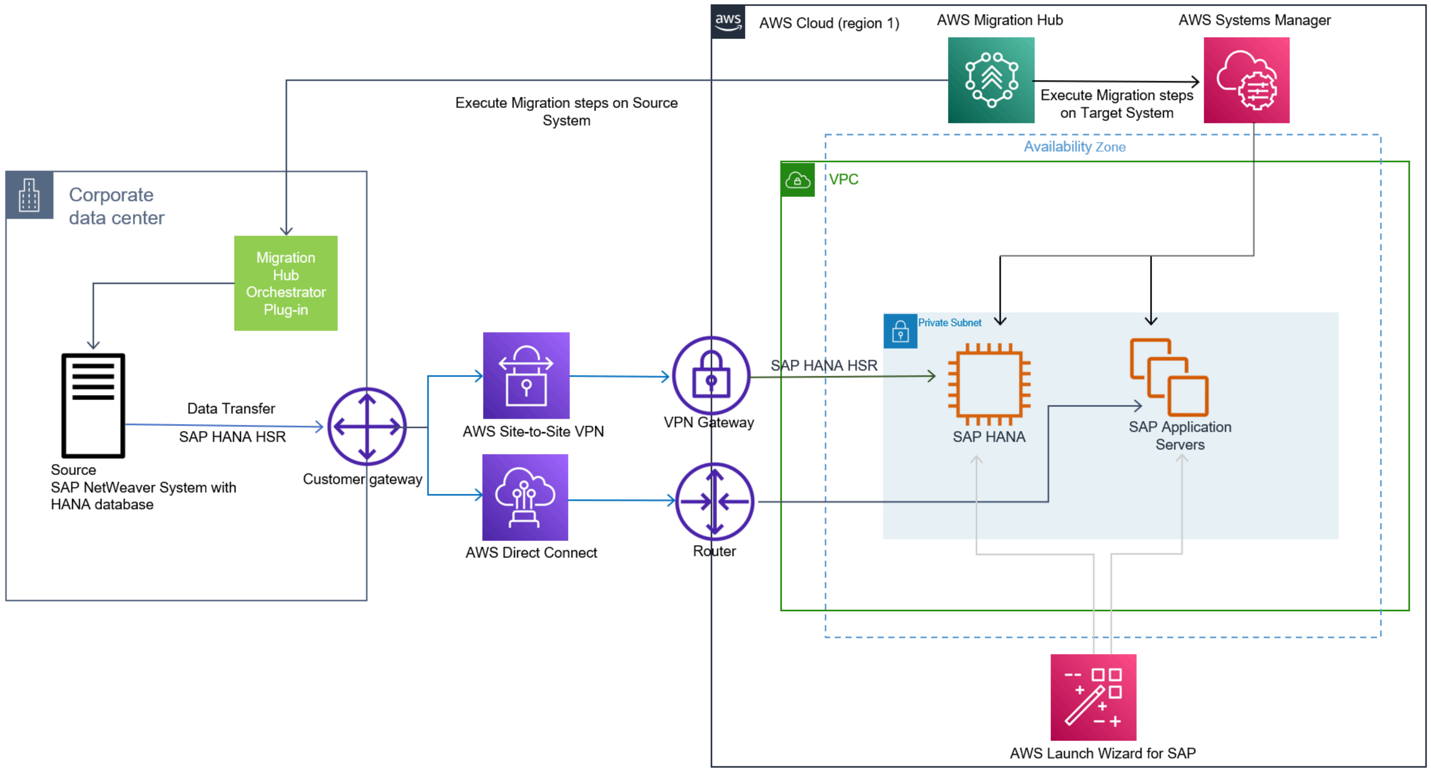
Topics

- Migrate applications with SAP HANA
- Migrate applications with any database

Migrate applications with SAP HANA

To migrate SAP NetWeaver based applications running on SAP HANA database, use the Migrate SAP NetWeaver based applications and SAP HANA databases to Amazon template.

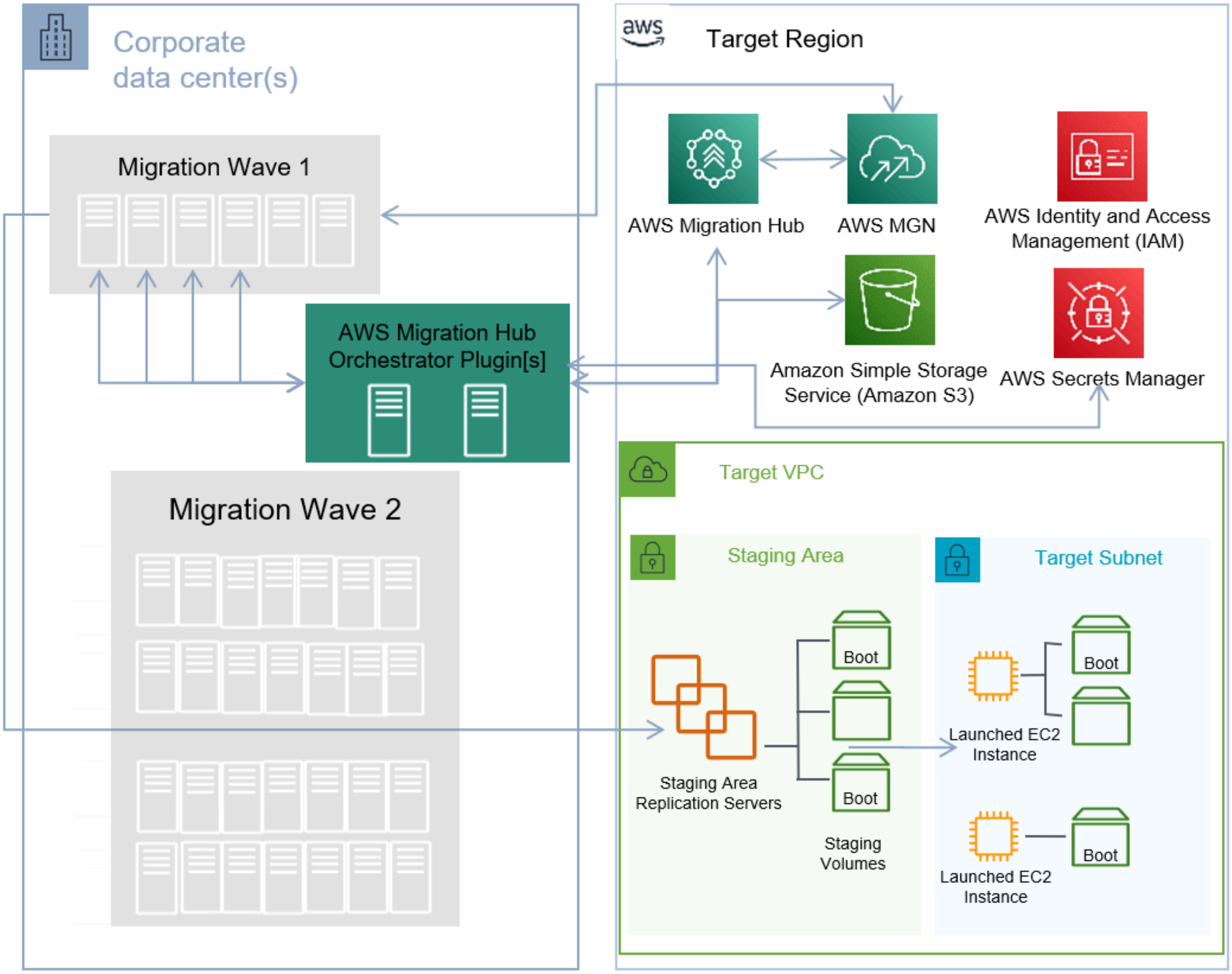
The following diagram illustrates an application migration with this template.



Migrate applications with any database

To migrate SAP NetWeaver based applications running on any database *other than SAP HANA*, use the Rehost applications on Amazon EC2 Migration Hub Orchestrator template.

The following diagram illustrates an application migration with this template.


Corporate data center(s)
Migration Wave 1
AWS Migration Hub Orchestrator Plugin[s]
Migration Wave 2
Target Region
AWS Migration Hub
AWS MGN
AWS Identity and Access Management (IAM)
Amazon Simple Storage Service (Amazon S3)
AWS Secrets Manager
Target VPC
Staging Area
Target Subnet
Staging Area Replication Servers
Boot
Boot
Staging Volumes
Launched EC2 Instance
Boot
Launched EC2 Instance
Boot

Oracle for SAP NetWeaver on Amazon Deployment and Operations Guide for Linux

SAP specialists, Amazon Web Services

December 2021

This guide is part of a content series that provides detailed information about hosting, configuring, and using SAP technologies in the Amazon Web Services Cloud. For more information, see SAP on Amazon Technical Documentation.

Overview

The purpose of this guide is to provide an overview of how to implement and operate Oracle database for SAP NetWeaver applications on Amazon Elastic Compute Cloud (Amazon EC2). It is for users who are responsible for planning, architecting, and deploying Oracle database for SAP NetWeaver applications on Amazon. You should have a good understanding of Amazon services, general network concepts, Oracle Enterprise Linux (OEL) OS, and Oracle database administration. This guide provides guidance to successfully launch and configure the resources required for Oracle database on Amazon.

It doesn’t provide guidance on how to setup network and security components like Amazon Virtual Private Cloud (Amazon VPC), subnets, route tables, ACLs, NAT gateway, Amazon Identity and Access Management (IAM) roles, and Amazon security groups. It focuses on how to configure and maintain the compute, storage, and OS components for the Oracle database on Linux on Amazon. It is not intended to replace the standard installation and administration guides from SAP or Oracle.

Prerequisites

We recommend familiarizing yourself with these guides:

- SAP on Amazon Overview and Planning
- Getting Started with Architecting SAP on the Amazon Cloud
- Best practices for Amazon EC2
- Migrating Oracle Database Workloads to Oracle Linux on Amazon
- Determining the IOPS Needs for Oracle Database on Amazon

- SAP Note 2606828 - Oracle Database Roadmap for SAP NetWeaver (SAP portal access required)

Technical requirements

Before you begin deploying Oracle database for SAP applications on Amazon, ensure that you meet the following requirements:

- If necessary, request a service limit increase by creating a support ticket. This is to ensure that the Amazon services required for Oracle database deployment are not constrained by the default limit. For more information, see Amazon service quotas. For example, you may have to increase the Amazon EC2 instance limit before your Oracle deployment.
- You will need the following information for your existing resources while running the Amazon CLI commands to create Amazon EC2 and Amazon Elastic Block Store (Amazon EBS) resources.

Information

Information	Description
Amazon Region	Region where you want to deploy your Amazon resources.
Availability Zone (AZ)	Availability Zone within your target Region where you want to deploy your resources.
Amazon VPC id	Amazon VPC where you want to deploy your Amazon EC2 instances for SAP installation.
VPS subnet id	Subnet where you want to deploy your Amazon EC2 instances.
Linux AMI id	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instances. You can find the latest Linux AMIs on Amazon Marketplace.
Key pair	Make sure that you have generated the key pair in your target Region and that you have access to the private key.

Security group id	Name of the security group that you want to assign to your Amazon EC2 instances.
Access key ID	Access key for your Amazon account that will be used with Amazon CLI tools.
Secret access key	Secret key for your Amazon account that will be used with Amazon CLI tools.

- Create security groups and open ports to enable communication. For existing security groups, ensure that the required ports are open. For a list of ports, refer to TCP/IP ports of all SAP products and Managing Oracle Database Port Numbers.
- Ensure that you have installed and configured Amazon CLI with required credentials, if you plan to use it to launch instances. For more information, see Installing the Amazon CLI.
- If you plan to use the Amazon Management Console, ensure that you have the essential credentials and permissions to launch and configure Amazon services. For more information, see Access management for Amazon resources.
- Ensure that you have the software files required for installation readily available. You can stage these in Amazon S3 or Amazon Elastic File System (Amazon EFS). Amazon EFS can be easily shared on all of your installation hosts. For more information, see Create your Amazon EFS file system.
- Oracle for SAP on Amazon is supported on an OEL OS. For more information, see SAP Note 1656099 and SAP Note 2358420 (login required). If you are currently using a different OS, you can procure licenses and perform a migration. For more information, see Migrating Oracle Database Workloads. To use AMIs published by Oracle, see Launch an Oracle Linux instance in Amazon.

Planning

Plan your SAP system landscape according to the SAP Master Guide for your version of SAP NetWeaver for Linux OS and Oracle database.

- Deployment options
- Sizing
- Amazon Machine Image (AMI)

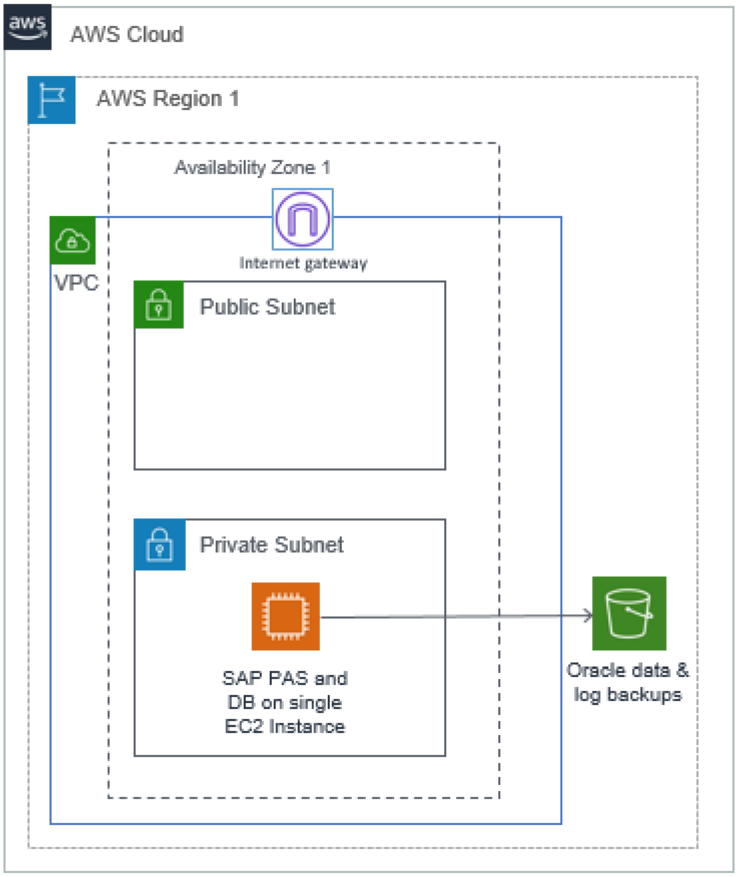
- Security and compliance
- Storage for Oracle

Deployment options

To install Oracle for SAP NetWeaver, you have four deployment options:

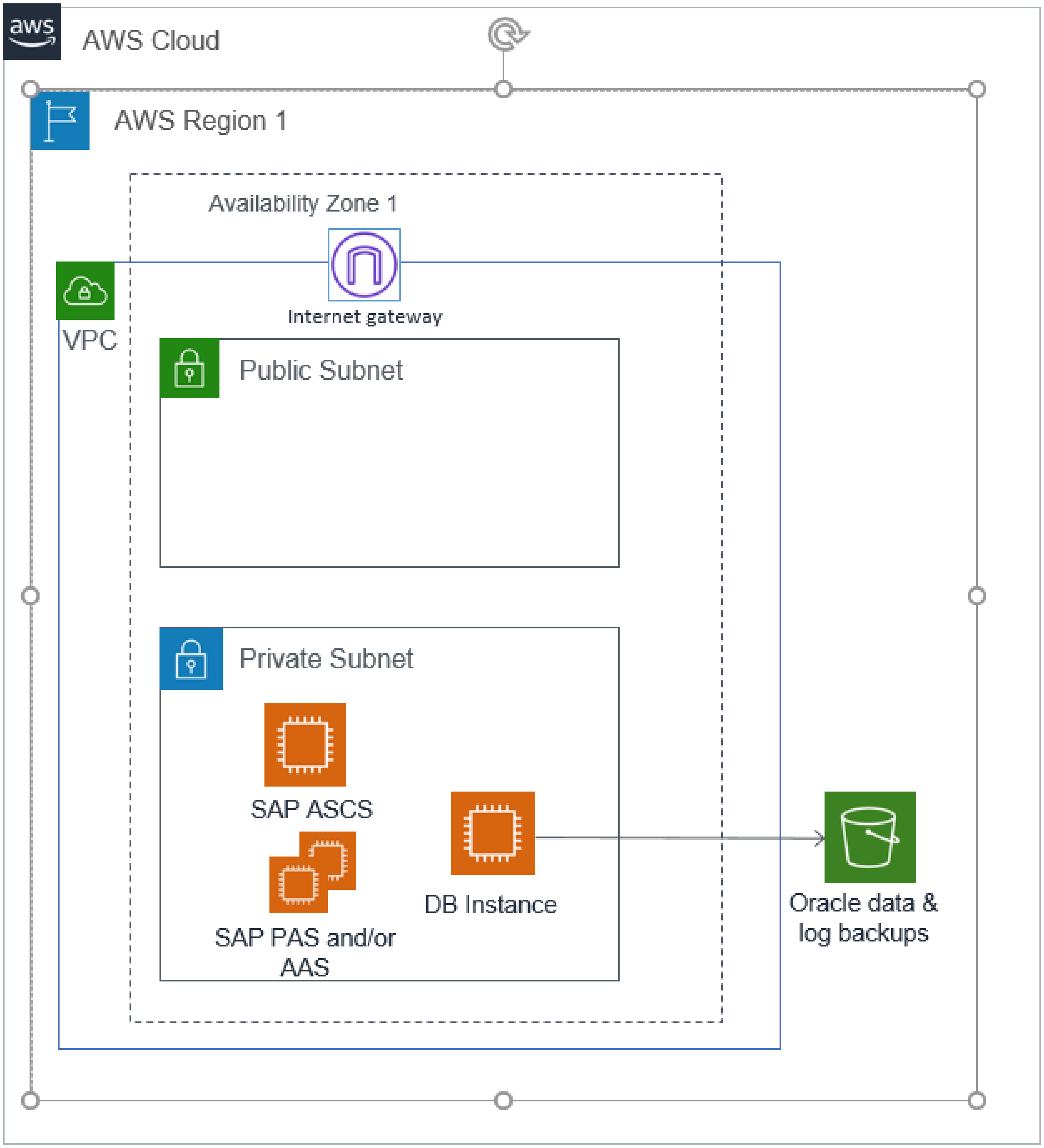
Standalone deployment

In standalone deployment (also known as single host installation), all components of the SAP NetWevaer, ABAP SAP Central Services (ASCS), and database Primary Application Server (PAS) run on one Amazon EC2 instance. One Amazon EC2 instance in a single Availability Zone in a single Region runs the Oracle database. This option can be optimal for non-production workloads. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power. However, this solution is not database state aware and does not protect your database against storage failure, OS issues, Availability Zone or Region failure.



Distributed deployment

In distributed deployment, every instance of SAP NetWeaver (ASCS/SCS, database, PAS, and optionally AAS) can run on a separate Amazon EC2 instance. This system also deploys Oracle database in a single Availability Zone. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power.



High availability deployment

In high availability deployment, you deploy two Amazon EC2 instances across two Availability Zones within a Region, and the Oracle database with Oracle Data Guard or a third-party high availability solution.

> **Note**
>
> When using native Oracle with SAP and Amazon features, the design must be a subset of supported features, as described in SAP Note 105047 and SAP Note 2358420.

Option 1: high availability with Oracle Data Guard

Oracle Data Guard is a feature of the Oracle database enterprise edition. It provides a set of tools to manage one or more Oracle standby databases for high availability and disaster recovery. To create an Oracle standby database, replicate the Oracle primary database to a secondary server by backup/restore or RMAN duplicate method. When the standby database is set up, any changes to the primary database are replicated on the standby database. This ensures that both the databases are in sync. The following table describes the replication methods associated with the Oracle Data Guard protection modes.

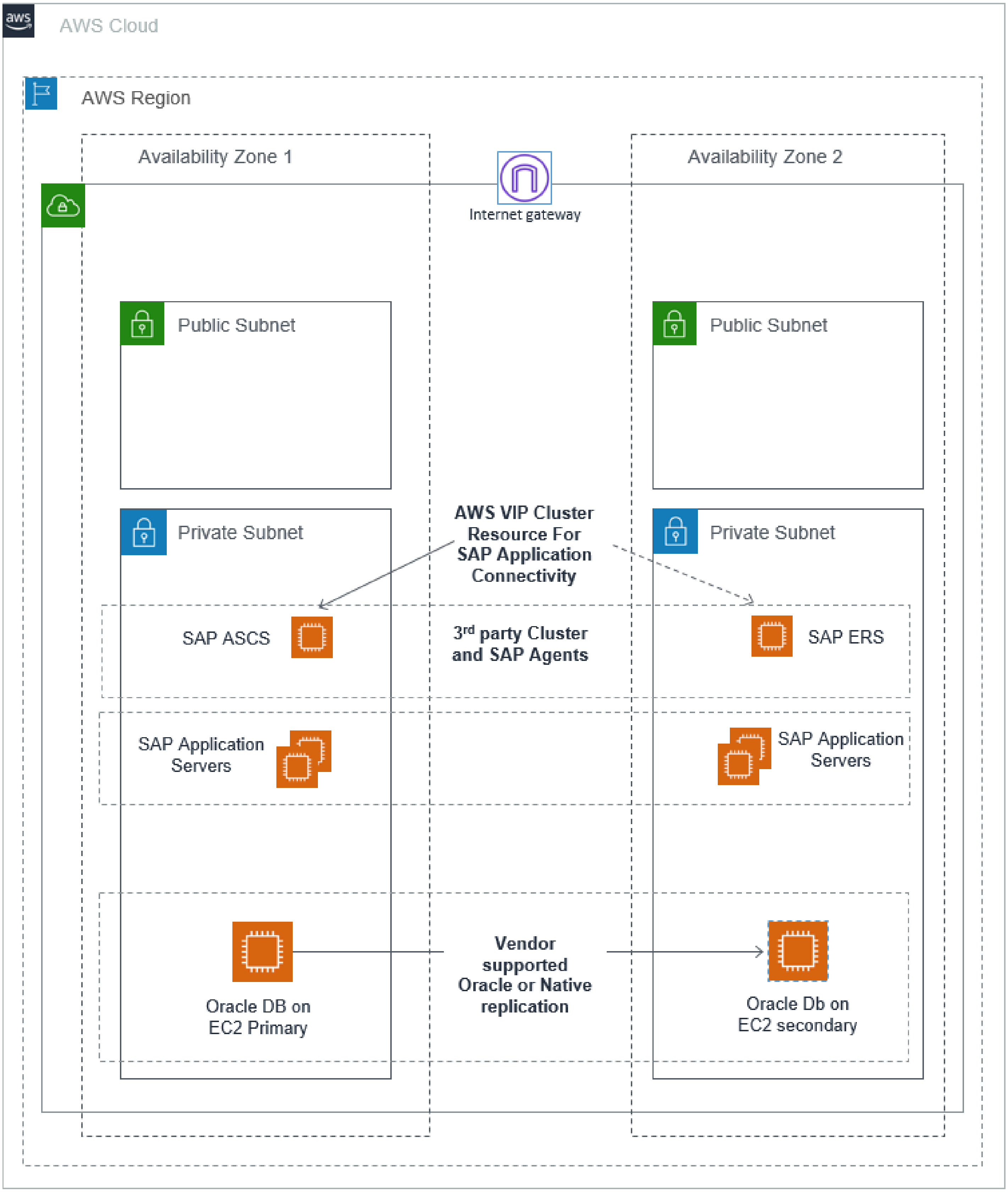
Oracle Data Guard protection modes

Protection mode	Replication	Behavior
Maximum performance	Asynchronous	Primary database is not affected by any delays in writing redo data to the standby database.
Maximum availability	Synchronous	Commit occurs when all the redo data needed to recover transactions has been written to the online redo log and to at least one synchronized standby database. If Data Guard is not able to write to the standby database, the

		behavior will be similar to the maximum performance protection mode.
Maximum protection	Synchronous	Changes must be written to both, the online redo log and to the standby database for every transaction. If Data Guard is unable to write the redo stream to at least one standby, it will shut down the primary database.

All Availability Zones within an Amazon Region are connected with high-bandwidth over fully redundant and dedicated metro fiber, providing high-throughput and low-latency networking between Availability Zones. For a high availability configuration, you can set up a primary and standby relationship between two Oracle databases with synchronous replication, running on Amazon EC2 instances in different Availability Zones within the same Region.

The maximum protection mode can cause a shutdown of the primary database in case of a standby database failure. Unless you need to meet a compliance requirement, we recommend using the maximum availability option for high availability.


AWS Cloud
AWS Region
Availability Zone 1
Availability Zone 2
Internet gateway
Public Subnet
Public Subnet
Private Subnet
Private Subnet
AWS VIP Cluster Resource For SAP Application Connectivity
SAP ASCS
3rd party Cluster and SAP Agents
SAP ERS
SAP Application Servers
SAP Application Servers
Oracle DB on EC2 Primary
Vendor supported Oracle or Native replication
Oracle Db on EC2 secondary

You can use manual failover or switchover to the standby database by following the steps in the Data Guard Broker Switchover and Failover Operations. Alternatively, you can automate this process. For more information, see Oracle Data Guard Fast-Start Failover. To reconnect the SAP applications after the failover is complete, refer to the Reconnect SAP instance to database section in the https://www.sap.com/documents/2016/12/a67bac51-9a7c-0010-82c7-eda71af511fa.html.

Option 2: high availability using third-party products

You can use third-party products to achieve Oracle database high availability in your SAP on Amazon environment. Here are two examples:

- Using SIOS to Protect your Critical Core on Amazon
- High Availability Configuration in Amazon Cloud using InfoScale Enterprise

For a complete list of certified partners, see the SAP wiki.

These products provide end-to-end high availability for SAP applications and databases. They also detect failures and perform automatic failovers, making them a good option for production environments with low recovery time objective. Using a virtual IP address makes the user or application redirection automatic in case of failover. For more information, see the vendor documentation.

Both of the preceding mentioned third-party examples are using their own storage (Data Keeper for SIOS and Veritas Volume Replicator for Veritas) and not database native replication. In option 1, the database is replicated using the Oracle Data Guard. The Guard supports SIOS but is not controlled by the SIOS application recovery kits, that is, a database failover is handled by the Guard. The Guard also supports Veritas, you can replicate using either the Guard or the Veritas Volume Replicator.

Disaster recovery deployment

With disaster recovery deployment of your SAP systems on Amazon Cloud, you can achieve business continuity. Based on recovery time objective, recovery point objective, and cost, you can set up disaster recovery deployment with one of the following three scenarios:

- backup and restore
- pilot light
- hot standby

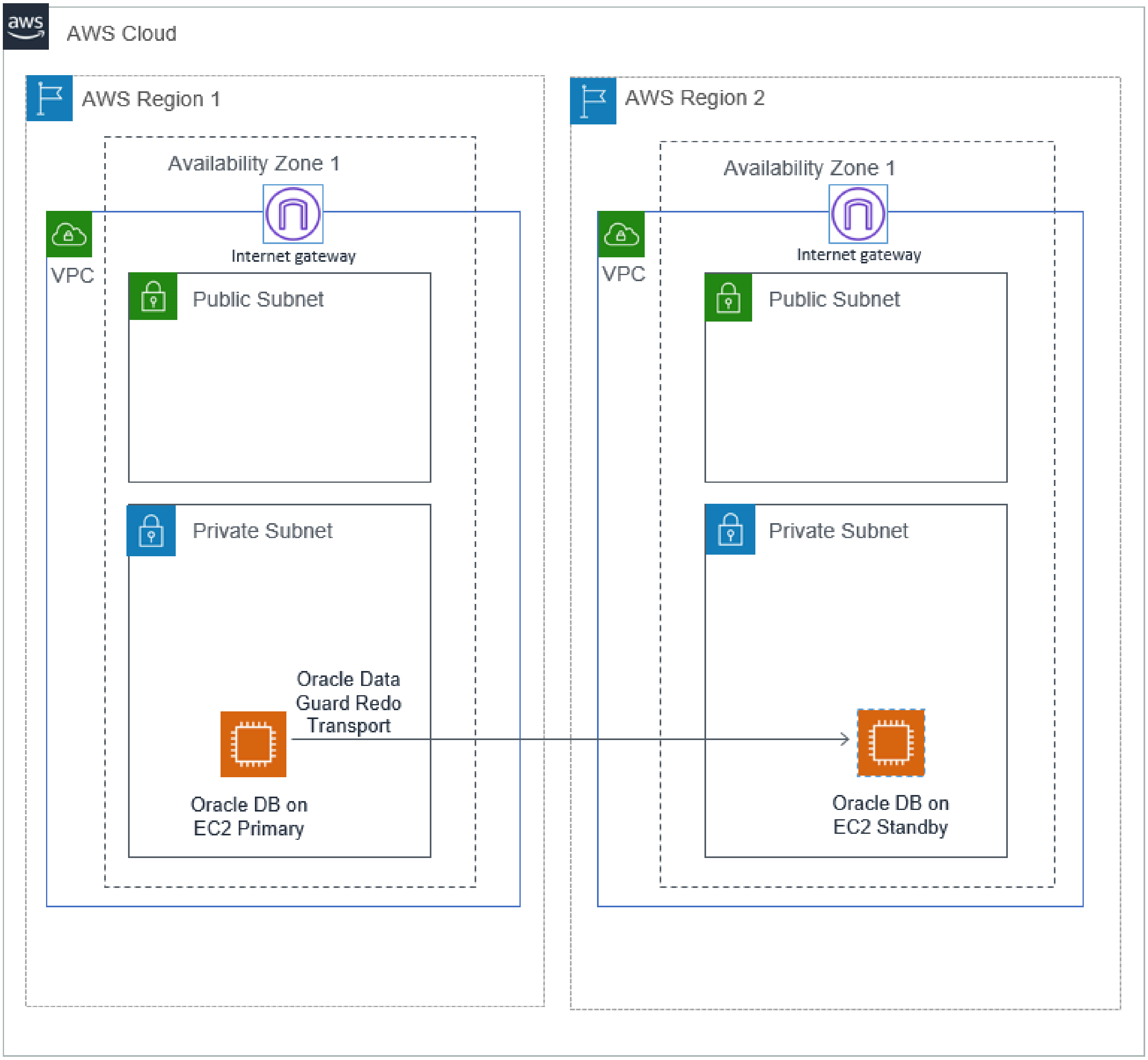
To setup disaster recovery across Amazon Regions, setup additional Amazon EC2 instances in a secondary Region for standby Oracle database. Also, configure Oracle Data Guard or a third-part solution for data replication across Amazon Regions.

Option 1:disaster recovery using Oracle Data Guard

Using Oracle Data Guard, you can set up pilot light or hot standby DR deployment in your Amazon Region. The maximum performance (asynchronous copy) option must be used in Data Guard.

Hot standby Amazon EC2 instance is of the same size as your production Amazon EC2 instance. This makes it more efficient to switch over during a DR test or event. Alternatively, you can use a pilot light approach wherein a smaller size Amazon EC2 instance is running in the Amazon DR Region as the target of data replication. This Amazon EC2 instance should have enough resources to take over the load of data replication. This option costs less than hot standby. However, during a DR test or event, you have to perform an additional step of resizing the Amazon EC2 instance. Before switchover, you must resize the DR Amazon EC2 instance to the same size as production, so that it can take over the production workloads. This step increases the time required to switchover to DR. You must also factor in the latency of running a non-production environment in another Region.

Pilot light option can run a non-production environment that is the same size as production in the DR Region. This ensures availability of Amazon EC2 instance in case of a DR event.

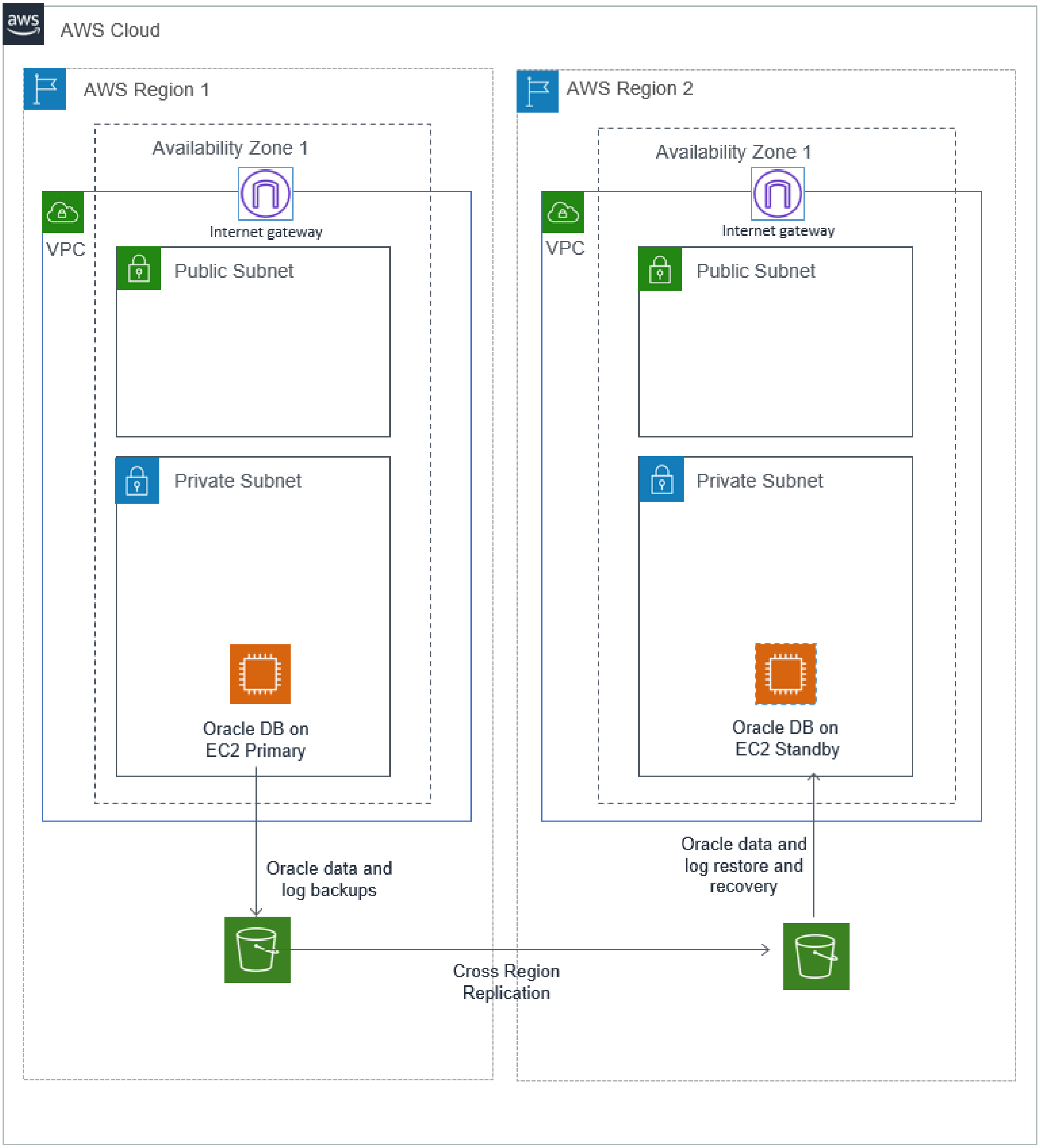


Option 2:passive disaster recovery using backup and recovery

This option uses Oracle database backup and recovery feature to set up your DR. You can store your database backups in Amazon Simple Storage Service (Amazon S3) and use the Amazon S3 Cross-region replication (CRR) to replicate your backup to your target Region. It enables automatic, asynchronous copying of objects across buckets in different Amazon Regions. You can install and configure Oracle database on an Amazon EC2 instance in your target DR Region and shut down the instance to save cost. You can then restart it and perform database restore and recovery as needed.

Alternatively, you can use automation such as Amazon CloudFormation or Cloud Development Kit or third-party automation tools to launch an Amazon EC2 instance and install and configure SAP applications Oracle database when needed. Any automation must be created and tested in advance and we recommend performing frequent DR drills. This helps you save cost for Amazon EC2 instances and Amazon EBS volumes.

Note that the time to recover your database is dependent on the size of database. Any log files that were not copied to the DR Regions are lost and cannot be used for recovery. This option typically has higher RTO and RPO as compared to other options that use data replication technologies. However, it offers lower TCO in comparison to other options.



You can choose to deploy high availability and disaster recovery for the same production database instance.

If you want to use Oracle Data Guard for HA and DR, see Multiple Standby Databases Best Practices .

Sizing

Sizing applies to three key areas - compute, network, and storage.

Compute

Amazon has certified multiple instance families with different sizes to run SAP workloads. For more details, see Amazon EC2 Instance Types for SAP.

To provision instances based on your requirements, you can use the Right sizing process. This process can help you optimize costs. Although it is ideal to use the right sizing approach when you move your SAP workloads to Amazon Cloud, it is an ongoing process. We recommend you to use the latest generation of your selected instance family.

For a greenfield (new) deployment of SAP workloads, you can use the Quick Sizer tool to calculate the compute requirement in SAPS. This helps you to select the closest matching Amazon EC2 instance for a price that is most economical for you. Before completing your selection, ensure that the selected Amazon EC2 instance provides enough Amazon EBS and overall network throughput to meet your application requirements.

For migrations, you can use any of the following data sources to decide the right size of your instance:

- Source system utilization and workload patterns, such as EarlyWatch alert reports.
- Source system specification: CPU, memory, storage size + throughput + IOPS, network.
- Source system SAPS rating.

For more details, see Compute & storage.

Network

Network performance is often not explicitly stated as a requirement in SAP sizing. Amazon enables you to check the network performance of all Amazon EC2 Instance Types.

Ensure that you have your network components setup to deploy resources related to your SAP workload. If you haven't already setup network components like Amazon VPC, subnets, route tables etc., you can use the, Amazon Quick Start Modular and Scalable VPC Architecture to most effectively deploy scalable Amazon VPC architecture in minutes. After setting up your Amazon VPC, you must set up Amazon EC2 instances within the Amazon VPC for your SAP workloads.

You also must set up a secured network connection between the corporate data center and the Amazon VPC along with the appropriate route table configuration, if it isn't already configured.

Storage

Deploying SAP workloads on Amazon required a minimum storage size and layout, based on your choice of OS/DB platform. For further details, refer to the relevant SAP documentation. You need to create Amazon EBS volumes that match these requirements.

You must check that the storage required is enough to provide sufficient I/O performance. The new `gp3` volume is ideal for Oracle workloads that require smaller volume size. With `gp3`, the storage throughput and IOPS are decoupled from the size and can scale independently.

The io2 volume is well-suited for I/O-intensive database workloads that require sustained IOPS performance or more than 16,000 IOPS. The `io2 Block Express` is another provisioned IOPS SSD volume for workloads that require sub-millisecond latency, sustained IOPS performance, and more than 64,000 IOPS or 1,000 MiB/s of throughput.

For more details, see Storage for Oracle.

Amazon Machine Image (AMI)

You can deploy your SAP Oracle workload on Oracle Enterprise Linux 6.4 or later. A base AMI is required to launch an Amazon EC2 instance. You can create your own AMIs or obtain an Oracle Linux AMI from Oracle. For using AMIs from Oracle, see Launch an Oracle Linux instance in Amazon. You can create your own Oracle Enterprise Linux image or use other images available at Amazon Marketplace.

Security and compliance

The following are additional Amazon security resources to help you achieve the optimum level of security for your SAP NetWeaver environment on Amazon:

- Amazon Cloud Security
- CIS Amazon Foundations
- Amazon Well-Architected Framework

OS Hardening

Check the following resources to strengthen the security of your workloads. You must have access to the SAP portal to view the SAP Notes.

- Refer to Security in Amazon EC2.
- Use Amazon Inspector.
- SAP Note 1635808
- SAP Note 2069760
- SAP Note 2936683
- SAP Note 1565179

To follow the CIS Benchmarks, see Securing Oracle Linux.

Encryption

The important aspect of securing your workloads is encrypting your data, both at rest and in transit. For more details, refer to the following:

- Amazon EBS encryption
- Data encryption in Amazon EFS
- Data encryption in Amazon S3

In addition to Amazon encryption features, you can also use Oracle Transparent Data Encryption, as described in SAP Note 974876.

Security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

Customers often separate the SAP system into multiple subnets, with the database in a separate subnet to the application servers, and other components, such as a web dispatcher in another subnet, possibly with external access.

If workloads are scaled horizontally, or high availability is necessary, you may choose to include multiple, functionally similar, Amazon EC2 instances in the same security group. In this case, you must add a rule to your security groups.

If Linux is used, some configuration changes may be necessary in the security groups, route tables, and network ACLs. For more information, see Security group rules for different use cases.

Network ACL

A network access control list (ACL) is an optional layer of security for your Amazon VPC that acts as a firewall for controlling traffic in and out of one or more subnets (they're stateless firewalls at the subnet level). You may set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your Amazon VPC.

See Amazon VPC Subnet Zoning Patterns for SAP on Amazon to understand the network considerations for SAP workloads.

API call logging

Amazon CloudTrail is a web service that records Amazon API calls for your account and delivers log files to you. The recorded information includes the identity of the caller, time of the call, source IP address, request parameters, and response elements returned by the Amazon service. With CloudTrail, you can get a history of Amazon API calls for your account, including API calls made via Amazon Management Console, Amazon SDKs, command line tools, and higher-level Amazon services (such as, Amazon CloudFormation). The Amazon API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

Notification on access

You can use Amazon SNS or any third-party application to set up notifications on SSH login to your email address or mobile phone.

Storage for Oracle

This section describes the key considerations for designing storage layout of Oracle for SAP NetWeaver on Amazon. Before defining the layout, we recommend familiarizing yourself with IOPS and throughput offered by Amazon EBS volume types and learning to calculate the baseline and burstable IOPS and throughput for these volumes. Amazon EC2 instances also have IOPS and throughput limits. For more details, see Amazon EBS-optimized instances.

Amazon FSx for NetApp ONTAP

FSx for ONTAP is certified for Oracle databases on SAP NetWeaver. For more information, see SAP Note 1656250 - SAP on Amazon: Support prerequisites (portal access required).

File system

The file system structure for SAP Oracle deployment may differ with the database version. Refer to the following SAP Notes for individual Oracle database versions:

- SAP Note 2660017
- SAP Note 1915301
- SAP Note 1524205

The directory structure for database installation requires several file systems. This section only focuses on the storage layout of the file systems mentioned in the following table. The other file systems (used for storing Oracle software binaries, trace, and log files) are critical for operations but do not have heavy performance requirements as compared to the following files.

Description	File system
Database data files	/ oracle/<DBSID>/ sapdata(1,2….n)
Database online redo logs	/ oracle/<DBSID>/ origlog(A,B…)
Database mirror redo logs	/ oracle/<DBSID>/ mirrlog(A,B…)
Database offline redo logs	/ oracle/<DBSID>/ oraarch

You can calculate the capacity requirements from your existing database for migrations or using the SAP Quick Sizer tool for new implementations.

Calculate the IOPS

The most efficient way to estimate the actual IOPS that is necessary for your database is to query the system tables over a period of time and find the peak IOPS usage of your existing database. To perform this task, measure IOPS over a period of time and select the highest value.

You can access this information from the GV$SYSSTAT dynamic performance view, a special view in the Oracle database that provides performance information. The view is continuously updated while the database is open and in use. Oracle Enterprise Manager and Automatic Workload Repository reports access this view to gather data.

Alternatively, you can use the native storage tools to calculate the IOPS requirements. If storage tools are not available, you can use a script. For more information, see Determining the IOPS Needs for Oracle Database on Amazon.

Amazon EBS volume types

Amazon has multiple options for database storage, based on your throughput and IOPS requirements.

Two options for general purpose SSD:

- gp3 volumes deliver a consistent baseline rate of 3,000 IOPS and 125 MiB/s, included with the price of storage. You can provision additional IOPS (up to 16,000) and throughput (up to 1,000 MiB/s) for an additional cost.
- gp2 volumes deliver performance linked to the size of the volume. We recommend new customers to use gp3 volumes. Existing gp2 users can migrate to gp3 easily with Amazon EBS Elastic Volumes. It enables modification of volume types, IOPS or throughput of your existing Amazon EBS volumes without interrupting the Amazon EC2 instances.

Three options for provisioned IOPS SSD:

- `io1` is designed to deliver a consistent baseline performance of up to 50 IOPS/GB to a maximum of 64,000 IOPS, and provide up to 1,000 MiB/s of throughput per volume.
- `io2` is designed to deliver a consistent baseline performance of up to 500 IOPS/GB to a maximum of 64,000 IOPS, and provide up to 1,000 MiB/s of throughput per volume.
- `io2 Block Express` is designed for workloads that require sub-millisecond latency, sustained IOPS performance, more than 64,000 IOPS, and 1,000 MiB/s of throughput per volume.

Comparisons

Choosing between general purpose and provisioned IOPS SSD

We recommend using gp3 on Oracle for SAP on Amazon workloads. It provides a better option for price over performance. You can dynamically switch from one volume type to another, if needed.

Choosing between `io1` *and* `io2`

We recommend you to use `io2` for provisioned IOPS use case. It provides lower annual failure rate and higher configurable IOPS per GB.

Choosing between `gp3` *and* `io2`

`io2` provides lower annual failure rate and higher maximum IOPS per volume but costs more than `gp3`. You can decide to use either of the two based your requirements regarding failure rate, IOPS, and cost.

Choosing between `io2` *and* `io2 Block Express`

`io2 Block Express` should be chosen over `io2` for workloads that require sub-millisecond latency and more than 64,000 IOPS and 1,000 MiB/s of throughput from a single volume. If `io2 Block Express` doesn't support your Amazon EC2 instance and Amazon EBS volume, use `io2`.

> **Note**
>
> Check the Amazon EBS volume types to ensure that your chosen volume supports the Amazon EC2 instance in use.

Best practices

Follow these best practices for maximizing your database performance and storage resilience:

- Use Amazon EBS-optimized instances for database storage. They have a dedicated path between Amazon EC2 and Amazon EBS volumes.
- To achieve higher IOPS and throughput, you can use Linux Volume Manager (LVM) to create Linux file systems with striping across multiple Amazon EBS volumes or Oracle Automatic Storage Management. For Automatic Storage Management, you can use multiple Amazon EBS volumes for creating disk groups, as recommended in the SAP installation guide for Oracle database.
- In case of backup to local file system, overall data throughput of the Amazon EBS volumes used to create the file system is crucial for backup performance.In Amazon, you can use throughput optimized (`st1`) type Amazon EBS volumes for database backups to local file system. For larger file systems, `st1` type volumes have higher maximum throughput per volume at a lower cost when compared with `gp3` or `io1`/`io2`. Other volume types can be considered if `st1` doesn't meet your requirement. Ensure that the backup storage window can meet the required backup window available for database backups.

- When running SAP NetWeaver on Oracle database, you are required to have the `/sapmnt/[SID]` directory mounted on the database server. We recommend that you use Amazon EFS to host the `/sapmnt` directory and mount this on all SAP servers using the NFS protocol.
- Data and log files should be on separate volumes.
- Origlog and mirrlog should be on separate volumes.
- Copies of control files should be stored on file systems that are created on separate volumes.
- Redo log files are written sequentially by the Oracle database instance Log Writer (LGWR) process. Log file systems must be designed to support such I/O activity.

Example configuration

You need to set up the Oracle database with the following requirements:

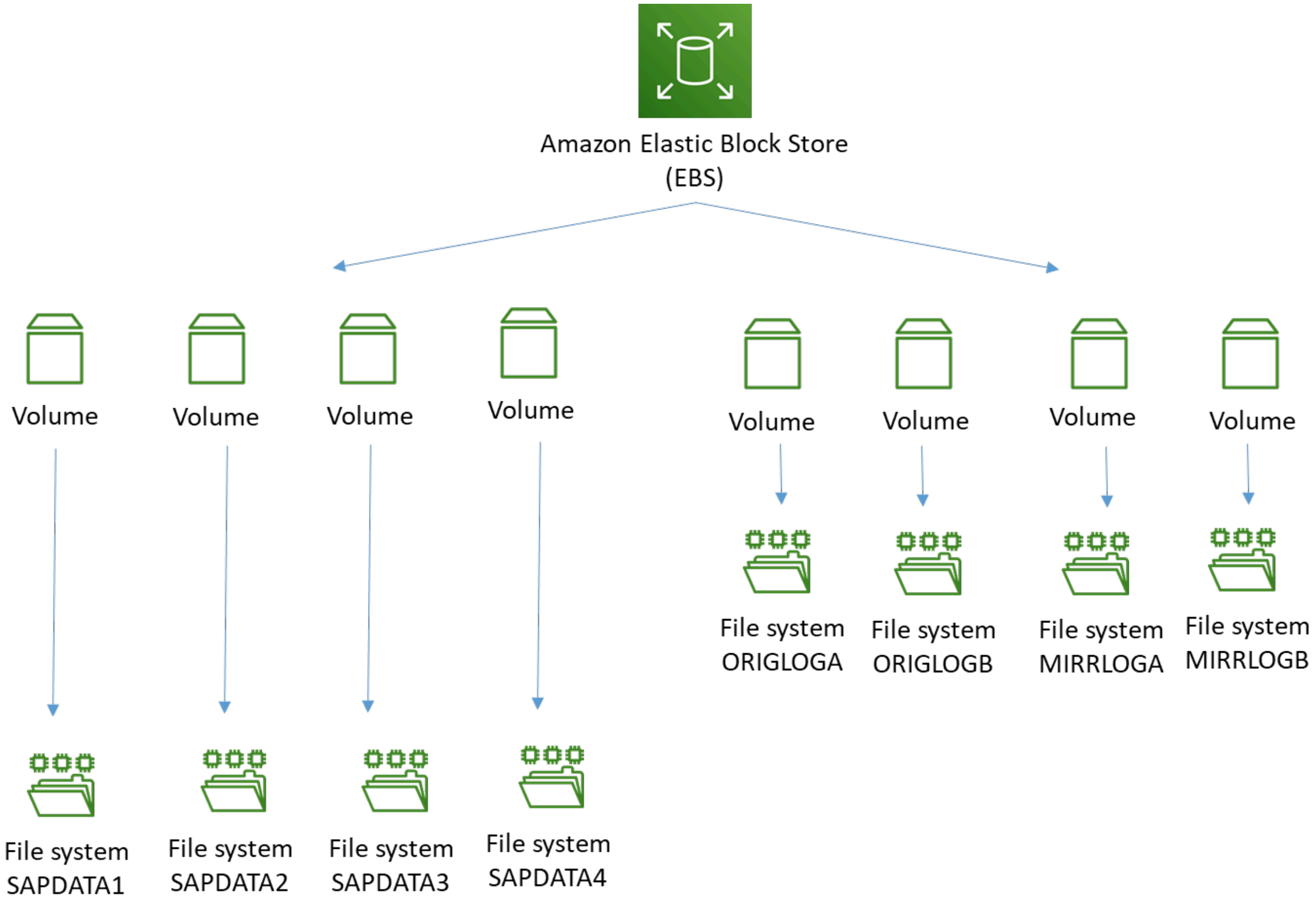
- Data file system with 48,000 peak IOPS, 3,000 MiB/s throughput and 12 TB capacity
- Each of the archive log, online and mirror redo log files with 3,800 peak IOPS and 25 GB capacity
- Oracle based file system for all other directories under `/oracle` with 300 GB capacity

The following is an example storage design to achieve the previously mentioned performance and capacity requirements.

Data file system with provisioned IOPS using gp3 or `io2` and without LVM:

Create one Amazon EBS volume for each file system size and use provisioned IOPS and throughput, as per the individual file system requirements. For data volumes, higher size and IOPS can be assigned to application tablespace volumes as needed. Since size, type, and IOPS can be changed dynamically, you can adjust these parameters with changing requirements.

This is a simpler approach and doesn’t require any volume striping using Linux LVM or similar technology. However, with this approach, you are still limited by the maximum size and IOPS supported by individual Amazon EBS volumes.



Data file system with provisioned IOPS using gp3 or `io2` and with LVM:

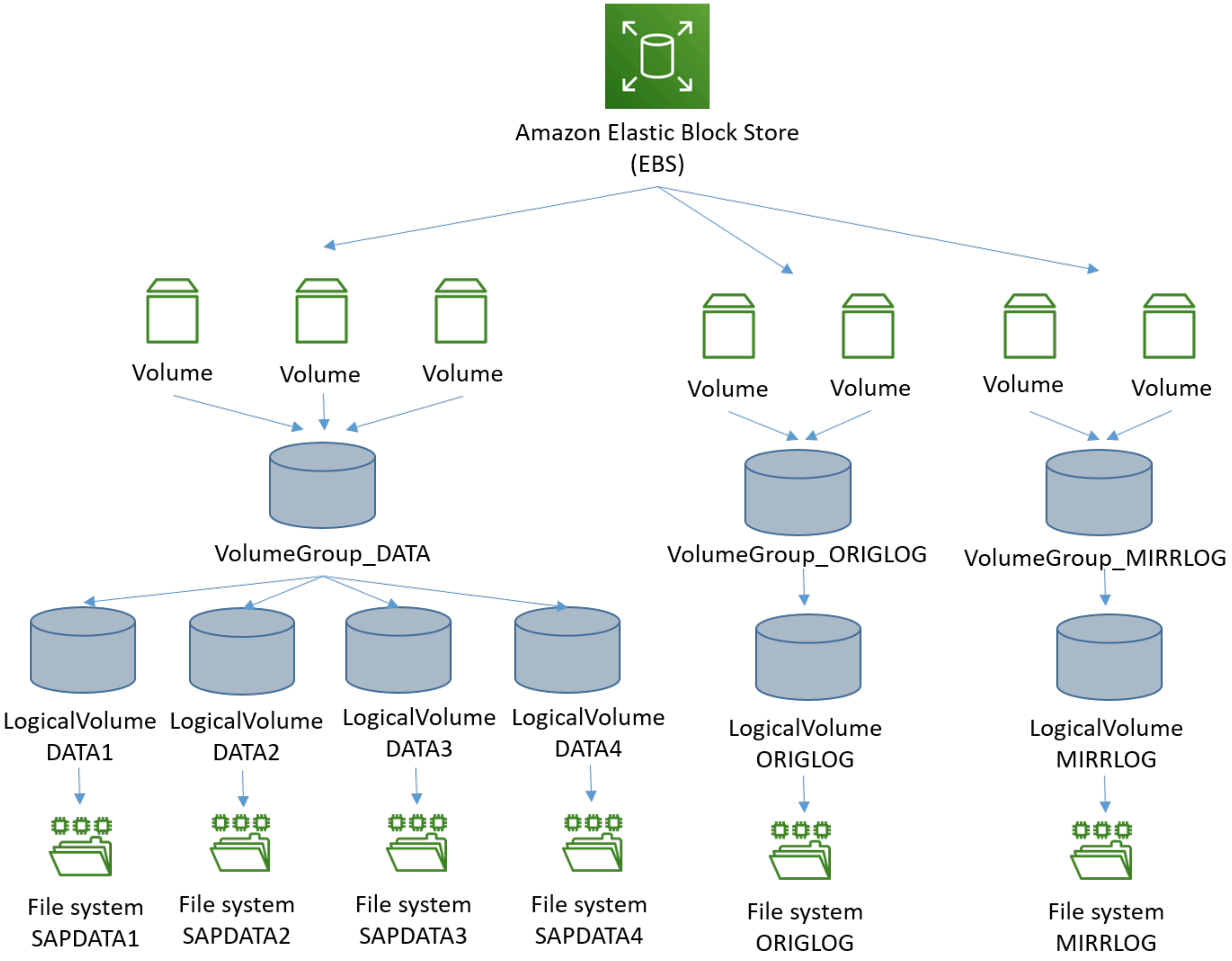
Create a single data volume group using stripping and four Amazon EBS volumes of 3 TB/each. This provides 12 TB of file system capacity. Each data volume can be configured with 12,000 IOPS and 750 MiB/s throughput to provide combined IOPS of 48,000 and throughput of 3,000 MiB/s. Multiple `sapdata` logical volumes and file systems can be created under this volume group. You can also create multiple volume groups for data file systems alone.

Creating volume groups increases the recovery time, in case one of the underlying Amazon EBS volumes becomes unusable. In this case, many data files residing on this volume group may be impacted as all `sapdata` file systems are stripped across all Amazon EBS volumes.

The benefit of using this approach is that you don’t need to have the IOPS and throughput requirements of individual `sapdata` file systems. The requirements of the Oracle database data file system are sufficient. Also, `sapdata` shares the IOPS and throughput, leading to better utilization of baseline performance when using gp3 volumes.

Online redo logs

Use 25 GB gp3 Amazon EBS volumes with 3,800 provisioned IOPS.



Deployment

- Standalone deployment
- HA/DR deployment

Standalone deployment

In this example, we set up a sample environment for installation. It includes a public subnet for RDP and SSH access via the internet. We use Amazon Launch Wizard for SAP in a single Availability

Zone deployment to create the Amazon VPC, subnets, security groups, and IAM roles. You can refer to this example set up but should also follow your own network layout and comply with security standards, such as the following:

- Using a Landing Zone solution like Amazon Control Tower.
- Working with a cloud team like Cloud Center of Excellence to use existing standards.

Step 1: Prepare your Amazon account

Check the Region where you want to deploy your Amazon resources:

- You pick your region for deployment during the planning phase.
- Display the Amazon Command Line Interface configuration data:

```
$ aws configure list
```

Ensure that the default region listed in the command output is the same as the target region where you want to deploy your Amazon resources and install SAP workloads. In this deployment, we provision an Amazon EC2 instance.

Note

In this section, the syntax used for the Amazon CLI and Linux commands is specific to the scope of this document. Each command supports many additional options. To learn more, use the `aws help` command.

Step 2: Create a JSON file for Amazon EBS storage

Create a JSON file containing the storage requirements for SAP Oracle database server volumes. The following is an example `JSON` file with two Amazon EBS volumes for swap and installation directories. You can add more volumes as per your storage design.

```
[
  {
    "DeviceName": "/dev/sdh",
    "Ebs": {
      "VolumeSize": 32,
      "VolumeType": "gp3",
      "DeleteOnTermination": true
    }
  },
  {
    "DeviceName": "/dev/sdg",
    "Ebs": {
      "VolumeSize": 50,
      "VolumeType": "gp3",
      "DeleteOnTermination": true
    }
  }
]
```

In the preceding example, the device name `/dev/shd` is for non-nitro based hypervisors. On Nitro-based instances, Amazon EBS volumes are presented as NVME block devices. You need to perform additional mapping when configuring these volumes. For more information, see Operating system and storage configuration.

Step 3: Launch the Amazon EC2 instance

Launch the Amazon EC2 instance for the SAP Oracle database installation in your target region, using the information gathered in Step 1. You must create the required storage volumes and attach them to the Amazon EC2 instance for the SAP installation, based on the `JSON` file you created in the Amazon EBS storage (Step 2).

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--count <number-of-EC2-instances> \
--instance-type <instance-type> \
--key-name=<name-of-key-pair>
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings file://C:\Users\<file>.json \
--region <region-ID>
```

Use this command in a single line format, as shown in the following example.

```
aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxx --count 1 --instance-type m5.large
 --key-name=my_key --security-group-ids sg-xxxxxxxx --subnet-id subnet-xxxxxx  --block-
device-mappings file://C:\Users\<file>.json
```

You can also launch Amazon EC2 instances using the Amazon Management Console. For more information, see Launch an instance.

Step 4: Prepare the Oracle Linux OS

Before starting the installation, you need to perform Oracle Enterprise Linux specific prerequisite tasks. For more information, refer to SAP Notes 1635808, 2069760, and 2936683 (login required).

Step 5: Prepare each Amazon EC2 instance for SAP Oracle installation

Download the SAP installation media as per the SAP installation guide, for the version of SAP NetWeaver you want to install on your Amazon EBS volumes. Locate your installation guide on the Guide Finder for SAP NetWeaver.

If you choose to install the SAP Oracle database with high availability deployment across two Availability Zones, repeat the preceding steps for Oracle database standby HA instance in the second Availability Zone.

If you choose to install SAP Oracle database with high availability and disaster recovery deployment across two Amazon Regions, repeat the preceding steps in the second Amazon Region in which you want to run the Oracle database standby DR instance.

Step 6: Installing SAP Oracle on Amazon EC2 instances

You are now ready to install the SAP Oracle software on your Amazon EC2 instances. For more information, see the Oracle Database Software Installation section of your SAP NetWeaver installation guide. Locate your installation guide on the Guide Finder for SAP NetWeaver. Instructions are available for all supported Oracle database versions.

HA/DR deployment

Installing SAP Oracle on Amazon EC2 instances and configuring HA/DR

Create an additional Amazon EC2 instance and perform the installation in a secondary Availability Zone. The steps for creating a HA or DR instance in a secondary Availability Zone are the same as described in Standalone deployment. You can simplify this step by using the following methods.

- If you have built any automation using Amazon CloudFormation or other tools to create the primary Amazon EC2 instance and install database software, you can use the same automation to build the HA instance.

- You can create an Amazon Machine Image of the primary Amazon EC2 instance and launch another instance in the secondary Availability Zone.

The configuration of high availability or disaster recovery depends on the tools you use. See the next sections for more details.

> **Note**
>
> The preceding steps are not applicable to passive DR.

Third-party references

To configure the SAP Oracle system with HA/DR using the Oracle Data Guard, refer to the following documents.

- Setting up Oracle 12c Data Guard for SAP
- Setting up Oracle 12c Data Guard for SAP
- Oracle Standby Databases
- Configuring Oracle Data Guard

For information about configuring HA/DR using a third-party product, see the vendor-specific documentation, such as the following.

- SIOS Oracle High Availability
- Veritas InfoScale™ 7.4.1 Solutions

> **Note**
>
> You need to configure cross-regional Amazon VPC peering or Transit Gateway to enable SAP Oracle asynchronous replication between the two Regions.

To perform a manual failover or switchover, see HA/DR operations.

Operations

Tagging Amazon resources

A tag is a label that you assign to an Amazon resource. Each tag consists of a key and an optional value, both defined by you. Adding tags to various Amazon resources will make managing SAP environments more efficient, and help you search for resources quickly. Many Amazon EC2 API calls can be used in conjunction with a special tag filter. For more information, see Tagging Amazon resources. The following are some examples of how you can use tags for your operational needs.

- You can tag your Amazon EBS volumes to identify their environment and use the same tags to create backup policies. For instance, Environment=DEV/QAS/PRD.
- You can use similar tags (DEV/QAS/PRD) for Amazon EC2 instances and use them for patching your OS or running scripts to stop/start applications or Amazon EC2 instances.

Monitoring

Amazon provides multiple native services to monitor and manage your SAP environment. CloudWatch and CloudTrail can be used to monitor your underlying infrastructure and APIs. CloudWatch provides ready-to-use KPIs for CPU, disk utilization, and enables you to create custom metrics for KPIs that you want to monitor. CloudTrail allows you to log the API calls made to your Amazon infrastructure components.

Operating system maintenance

In general, operating system maintenance across large estates of Amazon EC2 instances can be managed by using:

- Tools specific to the operating system, like Oracle Enterprise Manager
- Third-party products, such as those available on Amazon Marketplace.
- Amazon Systems Manager

The following are some key operating system maintenance tasks

Patching

You can follow SAP recommended patching process to update your landscape on Amazon. With Amazon Systems Manager Patch Manager, you can roll out OS patches according to your corporate policies. It has multiple benefits:

- Scheduling based on tags
- Defining patch baselines
- Auto-approving patches with lists of approved and rejected patches

Amazon Systems Patch Manager integrates with IAM, CloudTrail, and CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager operations work. Third-party products are available on Amazon Marketplace.

Maintenance Windows

Amazon Systems Manager Maintenance Windows lets you define a schedule to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, installing software or patches.

Administrator access

For administrative purposes, you can access the backend of your SAP systems via SSH or Amazon Systems Manager Session Manager.

Automation

Amazon Systems Manager Automation simplifies common maintenance and deployment tasks of Amazon EC2 instances and other Amazon resources. For more information, see Amazon Systems Manager Automation.

Automation using Infrastructure-as-Code with Amazon CloudFormation

We recommend following the principle of Infrastructure-as-Code (IaC) for automating and maintaining your workloads on Amazon. Amazon CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment in a repeatable and automated manner.

Cost optimization

We recommend cost optimization as an ongoing process. There are many Amazon services that help with budgeting, cost control and optimization. For more details, see Cost Optimization Pillar - Amazon Well-Architected Framework

Compute & storage

Compute

Amazon EBS volumes are exposed as NVMe block devices on Nitro-based instances. When changing Amazon EC2 instance types from a previous generation to a Nitro generation, NVMe device IDs associated with the volume can change. To avoid mount errors during change of instance type or instance reboots, you need to create a label for your file systems and mount it by the label, *and not* the NVMe IDs. For more details, see support article.

Aside from operating system maintenance, you should consider maintenance for your Amazon EC2 instances. It can be driven by using Automation runbook. The following are some examples.

- Use `Amazon-StopEC2InstanceWithApproval` to request one or more IAM users approve the instance stop action. After the approval is received, runbook stops the instance.
- Use `Amazon-StopEC2Instance` to automatically stop instances on a schedule, using CloudWatch Events or a Maintenance Window task. For example, you can configure an Automation workflow to stop instances every Friday evening and restart on Monday mornings. Note that this automation will only stop and start the Amazon EC2 instance. You must create additional document to gracefully stop and start SAP applications and database and then use the Amazon Systems Manager to run such automations.
- Use `Amazon-UpdateCloudFormationStackWithApproval` to update resources that were deployed using Amazon CloudFormation template. The update applies a new template. You can configure the Automation to request approval by one or more IAM users before the update begins.

You can also use Amazon Instance Scheduler to configure custom start and stop schedules for Amazon EC2 and Amazon RDS instances.

Storage

The following are the storage services used across this guide.

- Amazon EBS provides persistent storage for SAP applications and database. Amazon EBS volumes can be resized and even have the volume type changed without disrupting the applications. For more details, see Amazon EBS Elastic Volumes. After modifying the Amazon EBS volume, you need to extend the file system to match the extended volume size. For more details, see Extend a Windows file system after resizing a volume.
- Amazon EFS does not require you to explicitly provision storage, you pay only for your usage. It is built to scale on demand, without disrupting applications, growing and shrinking automatically as you add and remove files. This ensures that your applications have the required storage.
- Amazon S3 also does not require you to explicitly provision storage, you pay only for your usage. You can use Object lifecycle management to set rules that define when objects are transitioned or archived to colder storage (Amazon S3 IA or S3 Glacier) and when they expire. For more information, see Managing your storage lifecycle.

Backup & restore

Snapshots and AMIs

A common approach for backing up your SAP NetWeaver application servers is using snapshots and AMIs.

The SAP application data is stored on Amazon EBS volumes attached to the SAP NetWeaver application servers. You can backup the data on these volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups of Amazon EBS volumes, which means that only the blocks on the device that have changed after your most recent snapshot are saved. For more information, see Create Amazon EBS snapshots.

An Amazon Machine Image (AMI) provides the information required to launch an instance along with a block device mapping of all Amazon EBS volumes attached to it.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you’re confident that your instance is in a consistent state appropriate for AMI creation, you can check the *No Reboot* option.

You can use Amazon Backup to centrally configure backup policies and monitor backup activity for these snapshots. Once you have completed the SAP installation and post installation steps, create an image of the instance.

+

```
aws ec2 create-image --instance-id i-1234567890abcdef0 --name "My server" --description
 "An AMI for my server"
```

Amazon provides a very simple and quick way to copy an SAP system. You can use the Amazon Console Home or the Amazon CLI to create a new AMI of an existing SAP system. You can then launch exact copies of the original system from the new AMI. For more details, see Amazon Machine Images (AMI).

Backup to Amazon S3

You can perform traditional file-based backup to Amazon S3 from your Amazon EBS volumes. One way to take backup is to use Amazon CLI and initiate it by using Amazon Systems Manager Run command, so that you can centrally manage the backups.

Backup with third-party products

There are many third-party products for Amazon services, including a number that have been certified by SAP. Go to SAP Partner Services and Solutions Directory, select **ISV Solutions** in *Service/Solution Type*, then **Backup and Recovery** in *Software Solution*.

Amazon EFS backup

Using Amazon Backup, you can centrally configure backup policies and monitor backup activity for Amazon resources, including Amazon EFS file systems.

Alternatively, you can perform a file-level backup of your Amazon EFS file system to Amazon S3. You can do this by running a file-level copy to Amazon S3 from any Amazon EC2 instance running in the same region. This can be automated and scheduled using Amazon Systems Manager Run Command in combination with CloudWatch Events.

Backup and restore for Oracle database

You must to regularly backup your operating system and database to recover them in case of any failure. Amazon Cloud provides various services and tools that you can use to backup your Oracle database.

Storage snapshots

You can backup your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only blocks on the device that have changed after your most recent snapshot are saved. Snapshots of Amazon EBS volumes can be created for backup of Oracle database file systems like Oracle home and stage directories.

For complete Oracle database file backups using snapshots, you can use the Storage Snapshots Optimization feature by Oracle, supported from version 12c. For more details, see Making Backups with Third-Party Snapshot Technologies. Amazon customers can use this feature in combination with Amazon EBS snapshots to perform Oracle database backups. Using snapshot backups may reduce the backup window as compared to the full database backup approach. To set up snapshot-based based database backup, you can use sample scripts and steps published in the blog Improving Oracle backup and recovery performance with Amazon EBS multi-volume crash-consistent snapshots.

Oracle database backups

Any one of the following methods can be used for Oracle database backup.

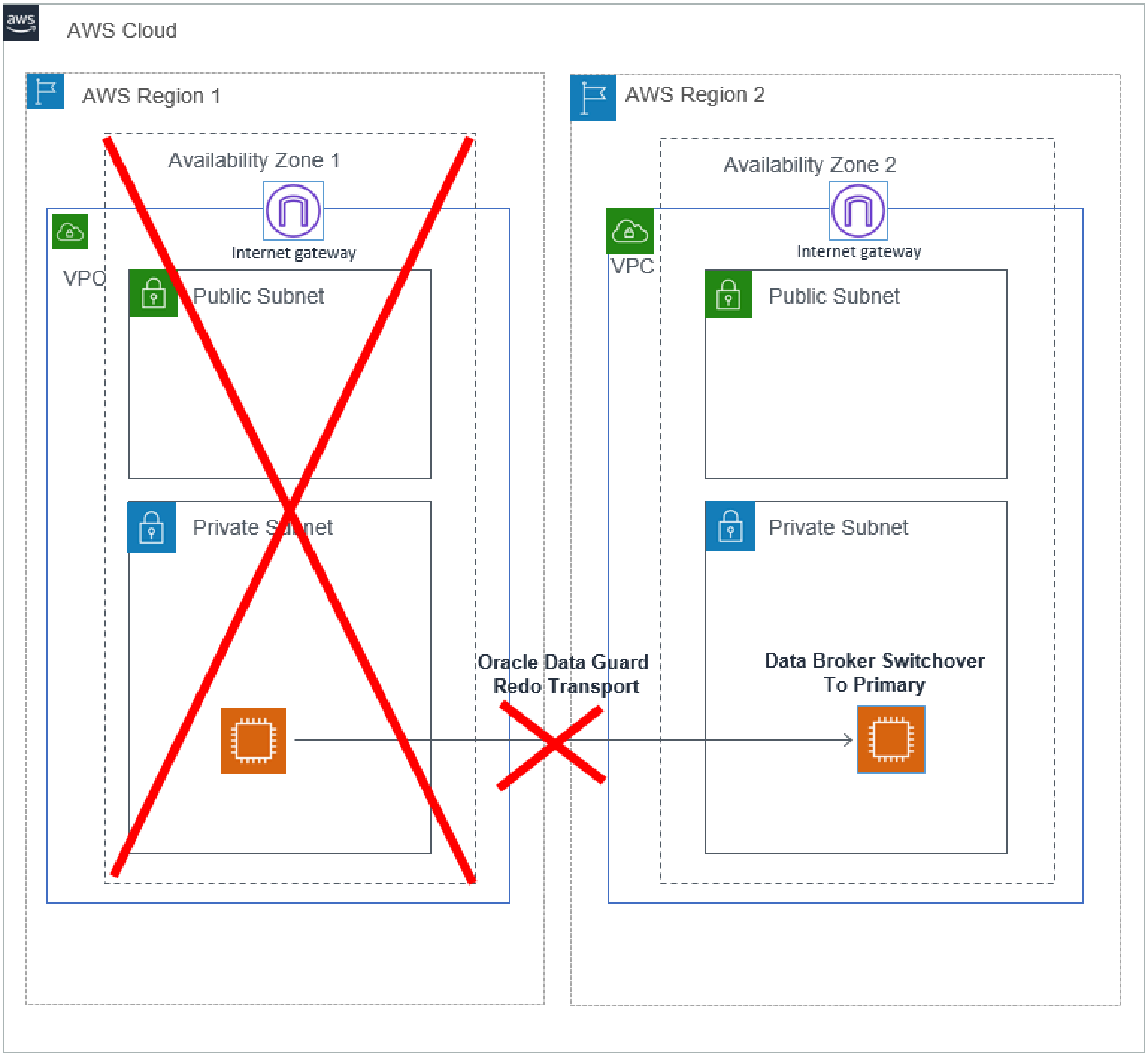
- Oracle database native tools (BRTOOLS) can be used to take backups on local storage. Once the backup is complete on local storage, it can be moved to Amazon S3 bucket via scripts.
- Oracle Secure Backup Cloud Module to backup your database directly to Amazon S3 using RMAN. For setup, see Oracle Database Backup To Cloud: Amazon Simple Storage Service (S3). For licence requirements, see Oracle Secure Backup Licensing Information.
- You can backup your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. For more information, see the preceding Storage snapshots section.
- There are many third-party tools from partner like Commvault, NetBackup, etc. that use the SAP backint interface and have Oracle database agents, with the capability to backup the database directly to Amazon S3.

To configure and tune backups for your Oracle database, see SAP on Oracle – Backup and Recovery and Database Backup and Recovery User’s Guide - Tuning RMAN Performance.

HA/DR operations

Oracle Data Guard

You can use manual failover or switchover to the standby database using steps described in the Switchover and Failover Operations. You can also automate this process by following the steps in Oracle Data Guard Fast-Start Failover. When using this feature with the observer node, you must place the observer node in the third Availability Zone. Ensure that you have the license to use the fast-start failover, it may not be included with the Data Guard. You can also use supported third-party products that provide automatic failover operation with the Data Guard. To reconnect the SAP applications post-failover, see the *Reconnect SAP instance to database* section-premises in https://www.sap.com/documents/2016/12/a67bac51-9a7c-0010-82c7-eda71af511fa.html.



Perform a DNS change

In case of manual failover, you may install SAP application servers using a virtual hostname and perform a DNS change to direct the SAP application servers to the new primary database server. For a DNS resolution in Amazon, you can use any of the following options.

- Amazon Route 53 enables you to create a private hosted zone for your environment and an A record for the virtual hostname used for Oracle database. Initially, this A record is mapped to the IP address of the primary Oracle database instance.
- You can maintain your own DNS server on-premise or on your Amazon EC2 instances. You can create an A record there for your virtual hostname used for Oracle database. Initially, this A record is mapped to the IP address of the primary Oracle database instance.
- With the Amazon Directory Service, you can create an A record for the virtual hostname used for Oracle database.

With any of the previously mentioned options, you can change the A record to a private IP address of the primary database instance in case of a failover. This DNS change can also be automated using Amazon services and scripts.

Resources

SAP on Amazon customers have the flexibility to deploy SAP Oracle database on the scalable, on-demand Amazon EC2 platform in a highly available manner. They don’t have to invest in costly capital expenditures for the underlying infrastructure. By combining the Amazon platform flexibility and SAP installation techniques, our customers greatly improve the availability of their deployments. For more details, see SAP on Amazon Case Studies.

Support

Amazon offers two levels of support. Amazon Business Support provides resources and technical support for customers running SAP workloads on Amazon. Amazon Enterprise Support offers support to customers running mission critical SAP production workloads on Amazon.

Document revisions

Date	Change
December 2021	Initial publication

SAP ASE for SAP NetWeaver on Amazon Deployment and Operations Guide for Linux

This guide provides information about configuring SAP ASE database for SAP NetWeaver on Amazon.

Prerequisites

The following information is required to deploy SAP Adaptive Server Enterprise (ASE) for SAP NetWeaver applications on Amazon. This pertains to your existing resources, using Amazon CLI to create Amazon EC2 and Amazon EBS resources.

Information

Information	Description
Amazon Region	Region where you want to deploy your Amazon resources.
Availability Zone (AZ)	Availability Zone within your target Region where you want to deploy your resources.
Amazon VPC id	Amazon VPC where you want to deploy your Amazon EC2 instances for SAP installation.
VPS subnet id	Subnet where you want to deploy your Amazon EC2 instances.
Linux AMI id	Amazon Machine Image (AMI) that will be used to launch your Amazon EC2 instances. You can find the latest Linux AMIs on Amazon Marketplace.
Key pair	Make sure that you have generated the key pair in your target Region and that you have access to the private key.

Security group id	Name of the security group that you want to assign to your Amazon EC2 instances.
Access key ID	Access key for your Amazon account that will be used with Amazon CLI tools.
Secret access key	Secret key for your Amazon account that will be used with Amazon CLI tools.

- Create security groups and open ports to enable communication. For existing security groups, ensure that the required ports are open. For a list of ports, refer to TCP/IP ports of all SAP products.
- Ensure that you have installed and configured Amazon CLI with required credentials, if you plan to use it to launch instances. For more information, see Installing the Amazon CLI.
- If you plan to use the Amazon Management Console, ensure that you have the essential credentials and permissions to launch and configure Amazon services. For more information, see Access management for Amazon resources.
- Ensure that you have the software files required for installation readily available. You can stage these in Amazon S3 or Amazon Elastic File System (Amazon EFS). Amazon EFS can be easily shared on all of your installation hosts. For more information, see Create your Amazon EFS file system.
- You can request a service limit increase by creating a support ticket. For more information, see Amazon service quotas.

References

You can refer to the following resources before deploying SAP ASE on Amazon. If you are new to Amazon, see Get started with Amazon.

- What is Amazon EC2?
- Amazon Elastic Block Store (Amazon EBS)
- What is Amazon S3?
- What is Amazon VPC?
- What is IAM?

- SAP on Amazon Overview and Planning
- SAP Lens - Amazon Well-Architected Framework
- Storage options for Oracle Database

  The storage options for Oracle are also valid for ASE.
- Performance and Tuning Series: Basics
- Installation of SAP Systems Based on the Application Server ABAP of SAP NetWeaver 7.3 EHP1 to 7.52 on UNIX: SAP Adaptive Server Enterprise
- Installation of SAP Systems Based on the Application Server Java of SAP NetWeaver 7.5 and SAP Solution Manager 7.2 SR2 Java of SAP NetWeaver 7.5 on UNIX: SAP Adaptive Server Enterprise
- SAP Note 2922454 - SAP Adaptive Server Enterprise (SAP ASE) on Cloud Platforms (requires SAP portal access)
- SAP Note 1941500 - Certification information for Linux and other Operating Systems - SAP ASE (requires SAP portal access)

Planning

Plan your SAP system landscape according to the SAP Master Guide for your version of SAP system running ASE on Linux. We recommend referring to the following SAP Notes (require SAP portal access).

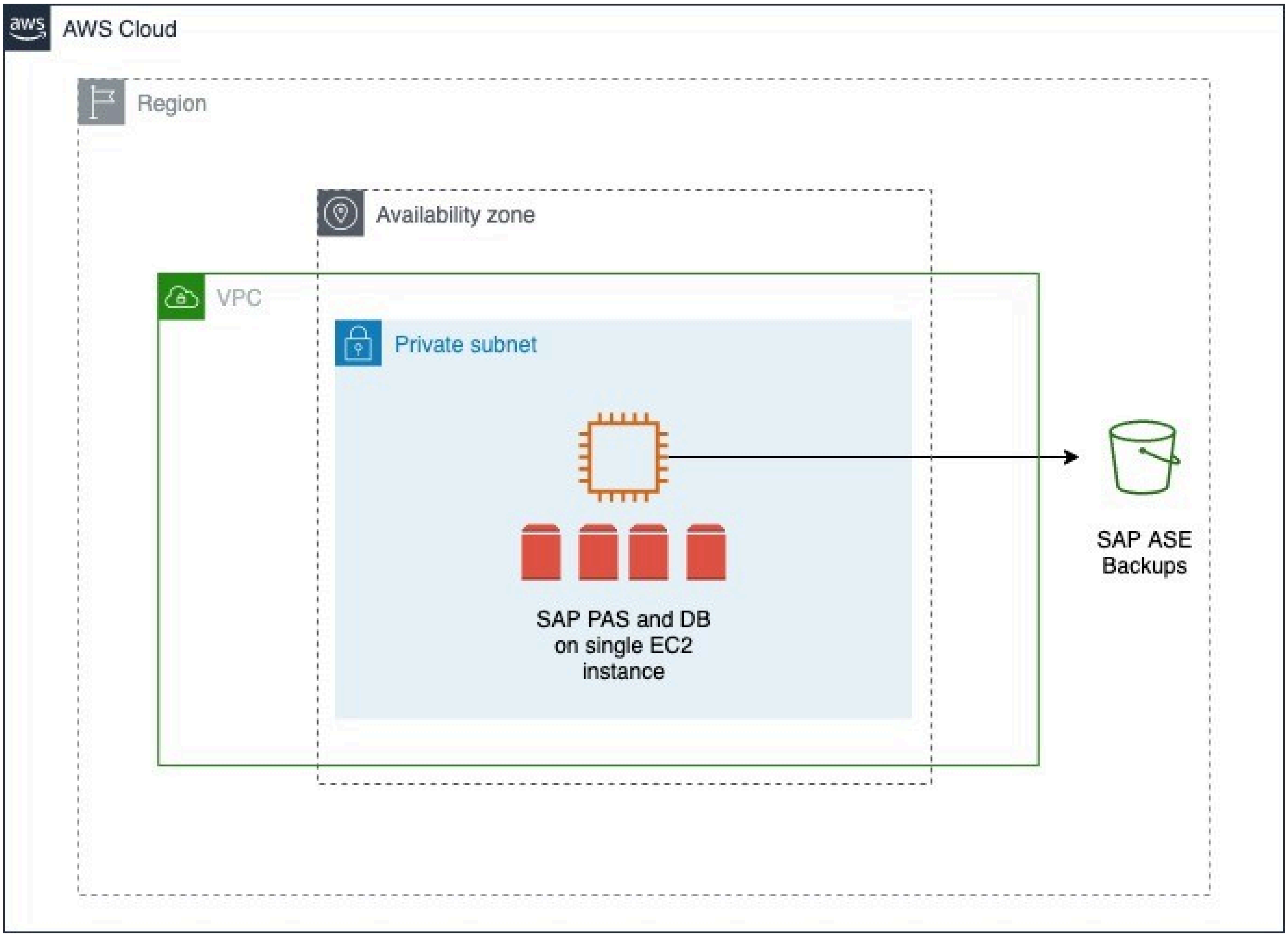
- SAP Note 1748888 - SYB: Inst.Systems Based on NW 7.3 and Higher: SAP ASE
- SAP Note 1554717 - SYB: Planning information for SAP on SAP ASE
- SAP Note 1656250 - SAP on Amazon: Support prerequisites

Deployment options

To install SAP ASE for SAP NetWeaver, you have four deployment options:

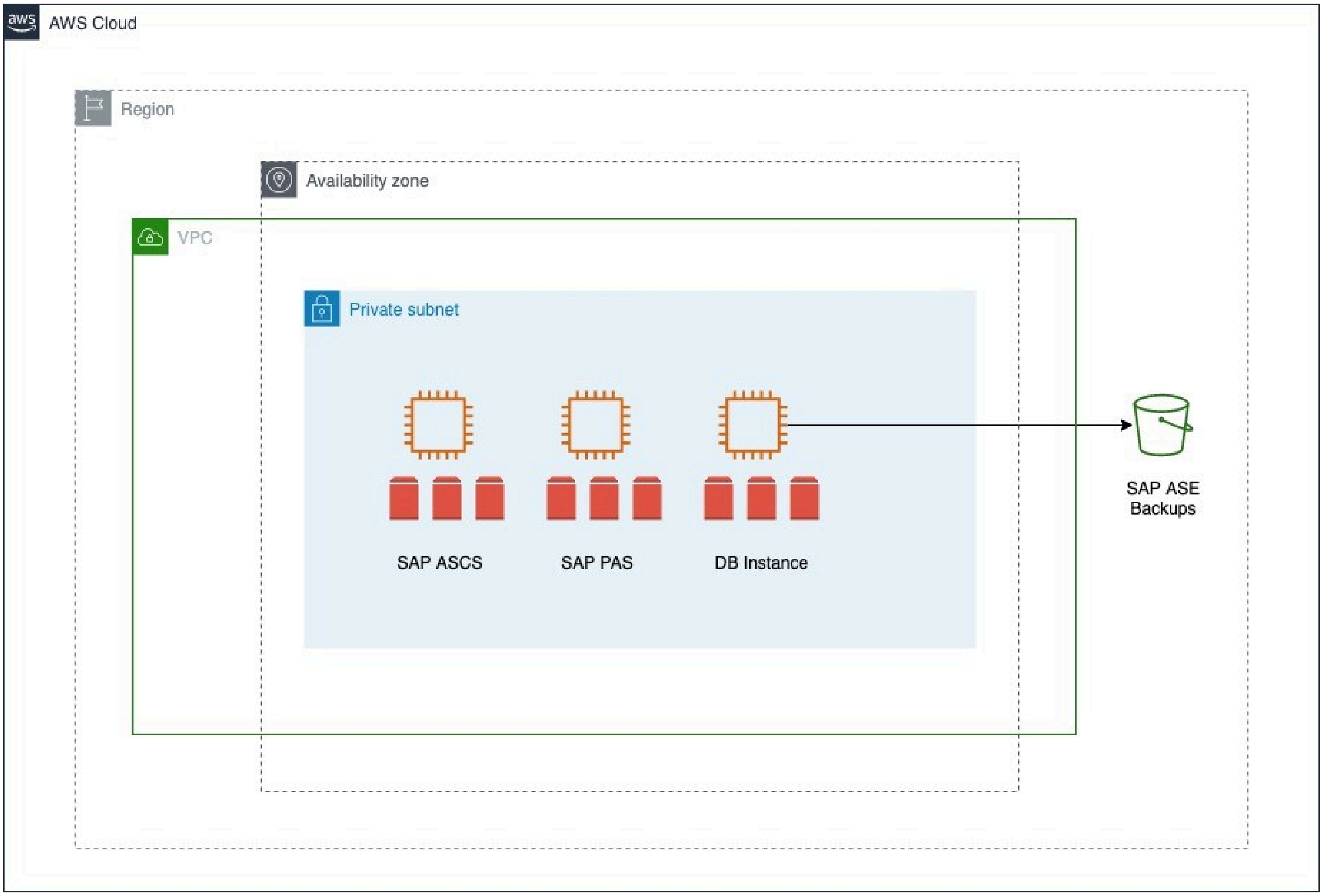
Standalone deployment

In standalone deployment (also known as single host installation), all components of the SAP NetWevaer, ABAP SAP Central Services (ASCS), and database Primary Application Server (PAS) run on one Amazon EC2 instance using a single Availability Zone in an Amazon Region. This option is recommended for non-production workloads. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power. This solution is not database state aware, and does not protect your database against storage failure, OS issues, Availability Zone or Region failure.



Distributed deployment

In distributed deployment, every instance of SAP NetWeaver (ASCS/SCS, database, PAS, and optionally AAS) can run on a separate Amazon EC2 instance. This system also deploys SAP ASE database in a single Availability Zone. You can use Amazon EC2 auto recovery feature to protect your instance against infrastructure issues like loss of network connectivity or system power. This solution is not database state aware, and does not protect your database against storage failure, OS issues, Availability Zone or Region failure.

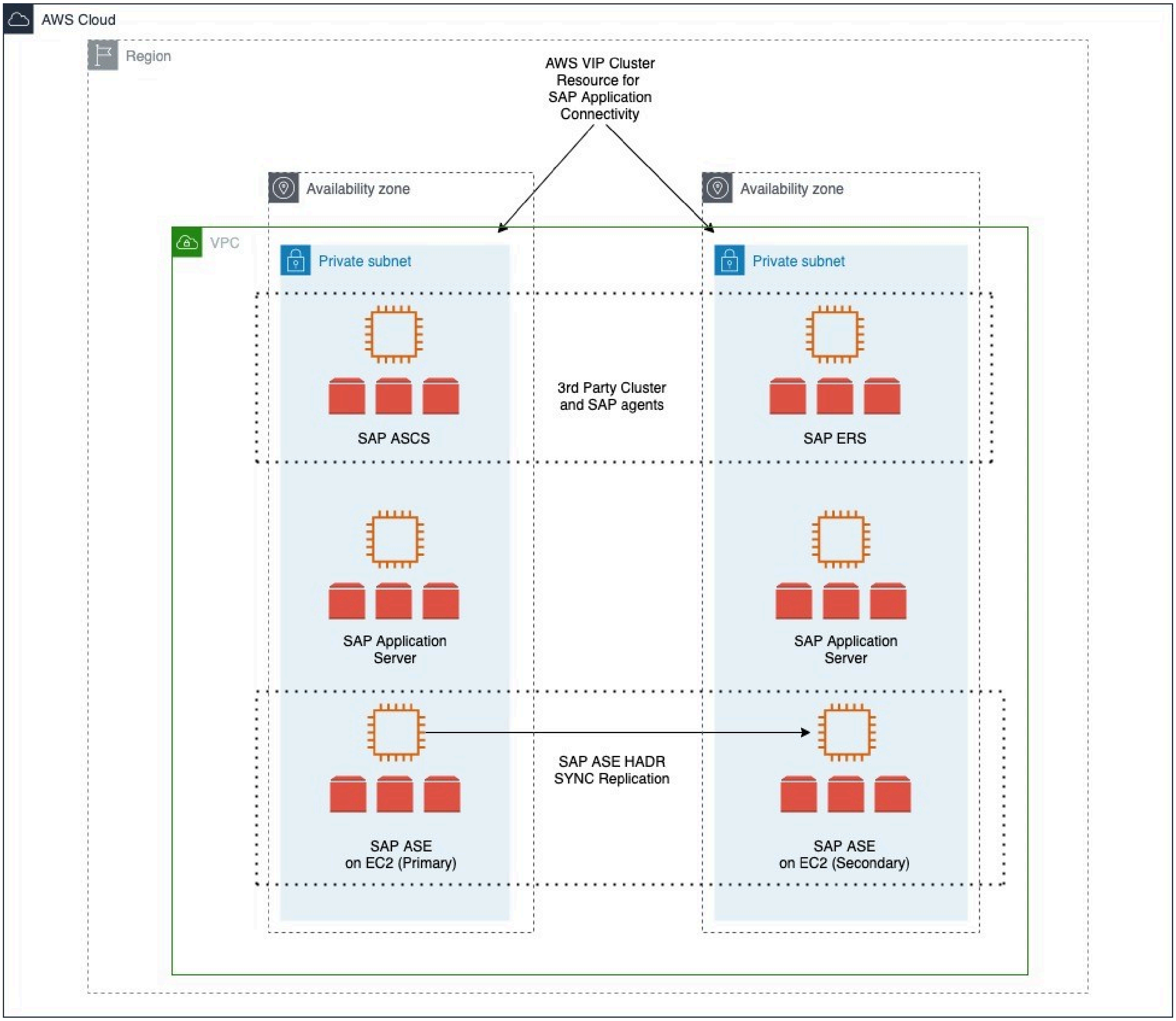


High availability deployment

In a high availability deployment, you deploy two Amazon EC2 instances across two Availability Zones within a Region, and the SAP ASE database with a combination of the Data Movement Component and Database Fault Manager.

All Availability Zones within an Amazon Region are connected with high-bandwidth over fully redundant and dedicated metro fiber, providing high-throughput and low-latency networking between Availability Zones. For a high availability configuration, you can set up a primary and standby relationship between two SAP ASE databases with synchronous replication, running on Amazon EC2 instances in different Availability Zones within the same Region.

The SAP ASE always-on option is a high availability/disaster recovery system that contains two or more SAP ASE servers – the primary server where all of the transaction processing takes place, and the warm standby (companion) server. The primary and standby nodes are deployed in different Availability Zones, providing protection against zonal failures. You can also integrate Fault Manager to automatically failover the system in case of failures. You can learn more about this on the SAP Help Portal's SAP Adaptive Servers Enterprise HADR Users Guide.



We recommend referring to the following SAP Notes (SAP portal access required) for a high availability deployment.

- *SAP Note 1650511 – SYB: High Availability Offerings with SAP Adaptive Server Enterprise*
- *SAP Note 2808173 – Special Instructions when Installing and Upgrading HADR with SAP Business Suite on SAP ASE*

Disaster recovery deployment

You can increase business continuity with disaster recovery deployment of your SAP systems on Amazon Cloud. Based on recovery time objective, recovery point objective, and cost, you can set up disaster recovery deployment with one of the following three options.

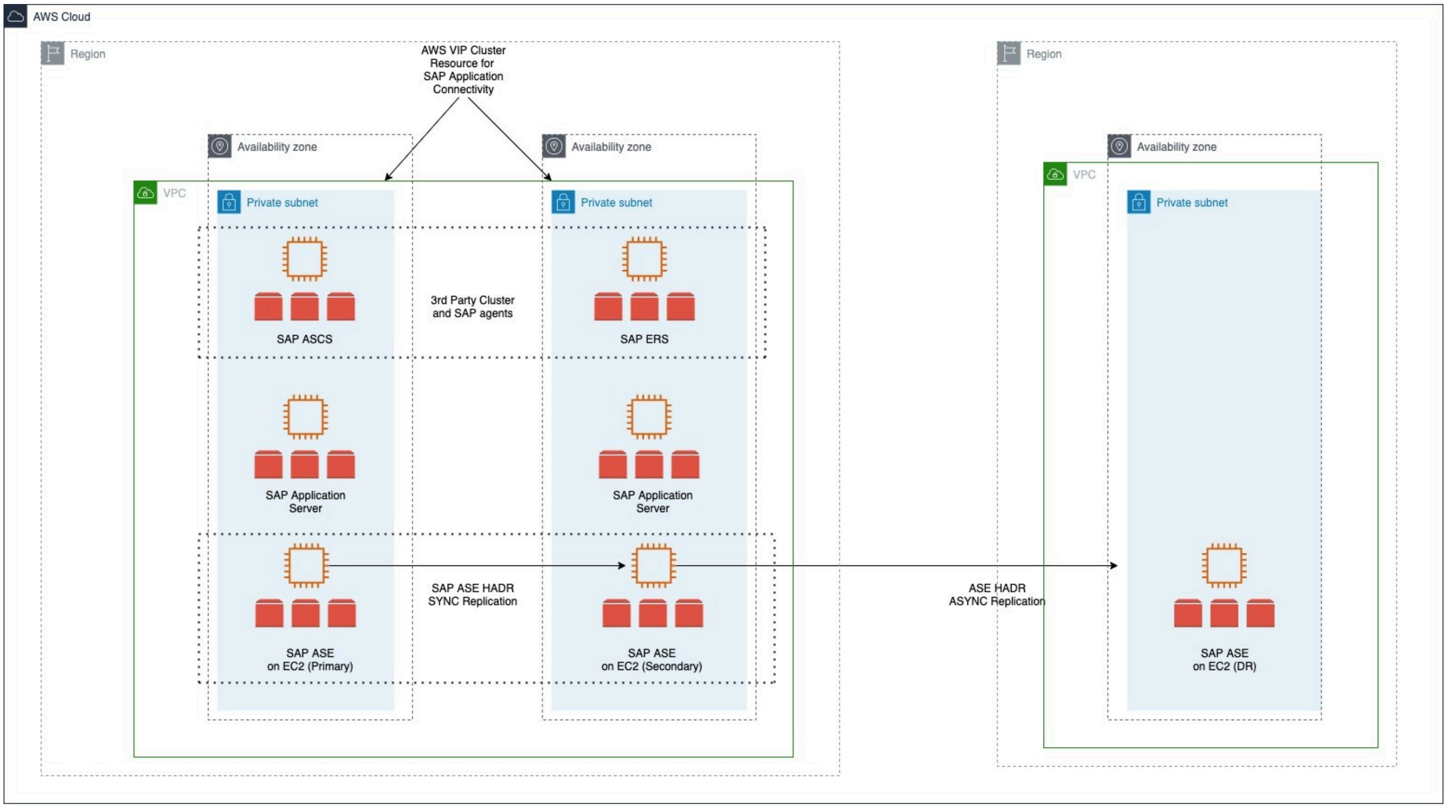
Option 1 – disaster recovery using the SAP ASE HADR feature

You can use the SAP ASE HADR feature to replicate your database in a secondary Amazon Region or Availability Zone, based on your business and audit requirements. You can also integrate this DR node in an existing HADR landscape. This setup enables you to increase the overall system resiliency.

With this option, you can either choose pilot light, where the recovery instance is smaller than the current instance, or hot standby, where the recovery instance is of the same size as the current instance. You must consider your recovery time objectives and manual effort required when choosing between pilot light or hot standby. The recovery instance for the pilot light option must be resized before assuming disaster recovery.

For more details, check the following SAP resources.

- *HADR System with DR Node Users Guide*
- *2934459 - HADR support of two ASE servers on Primary and Companion machines - SAP ASE (requires SAP portal access)*

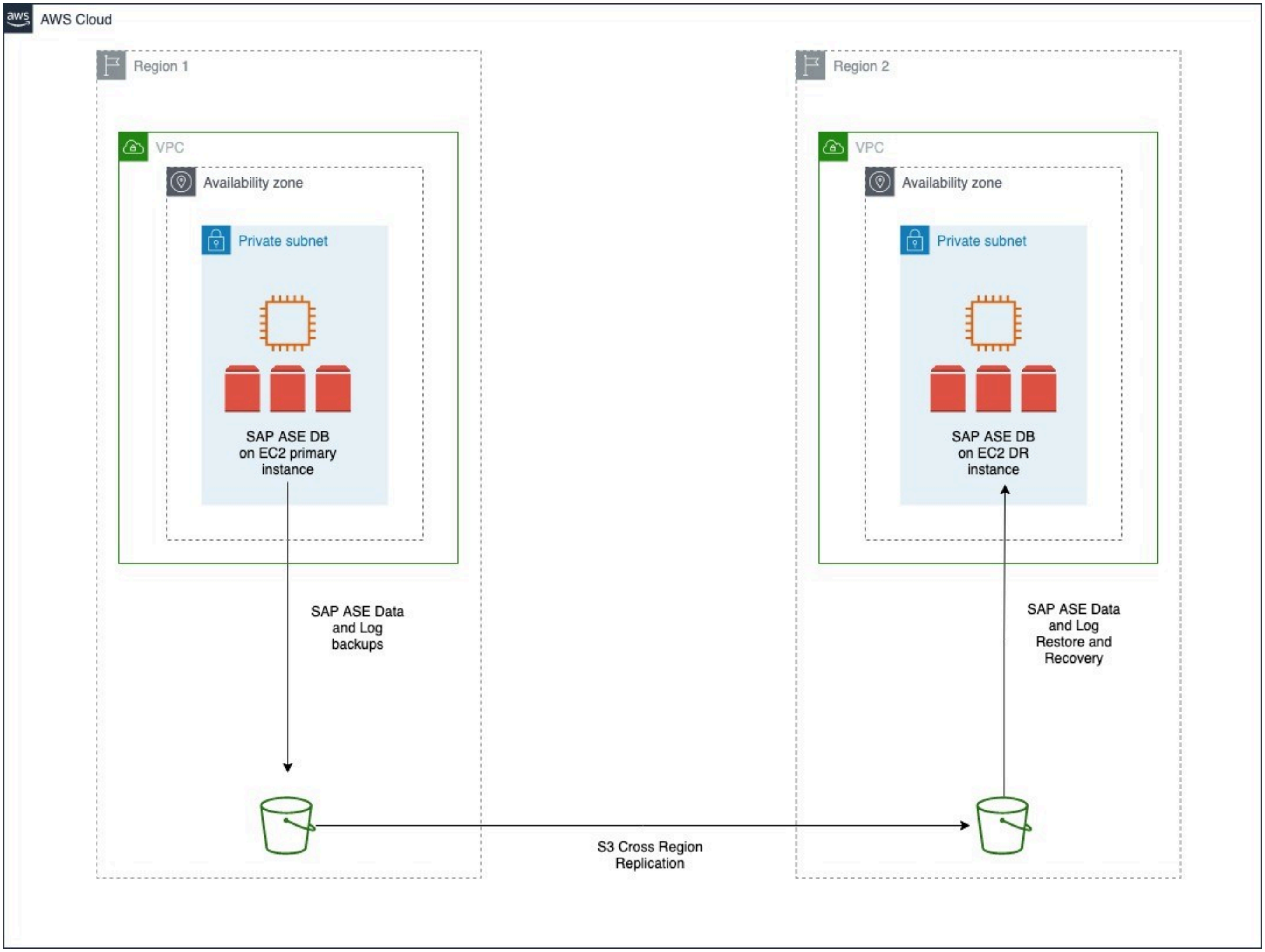


Option 2 – passive disaster recovery using backup and recovery

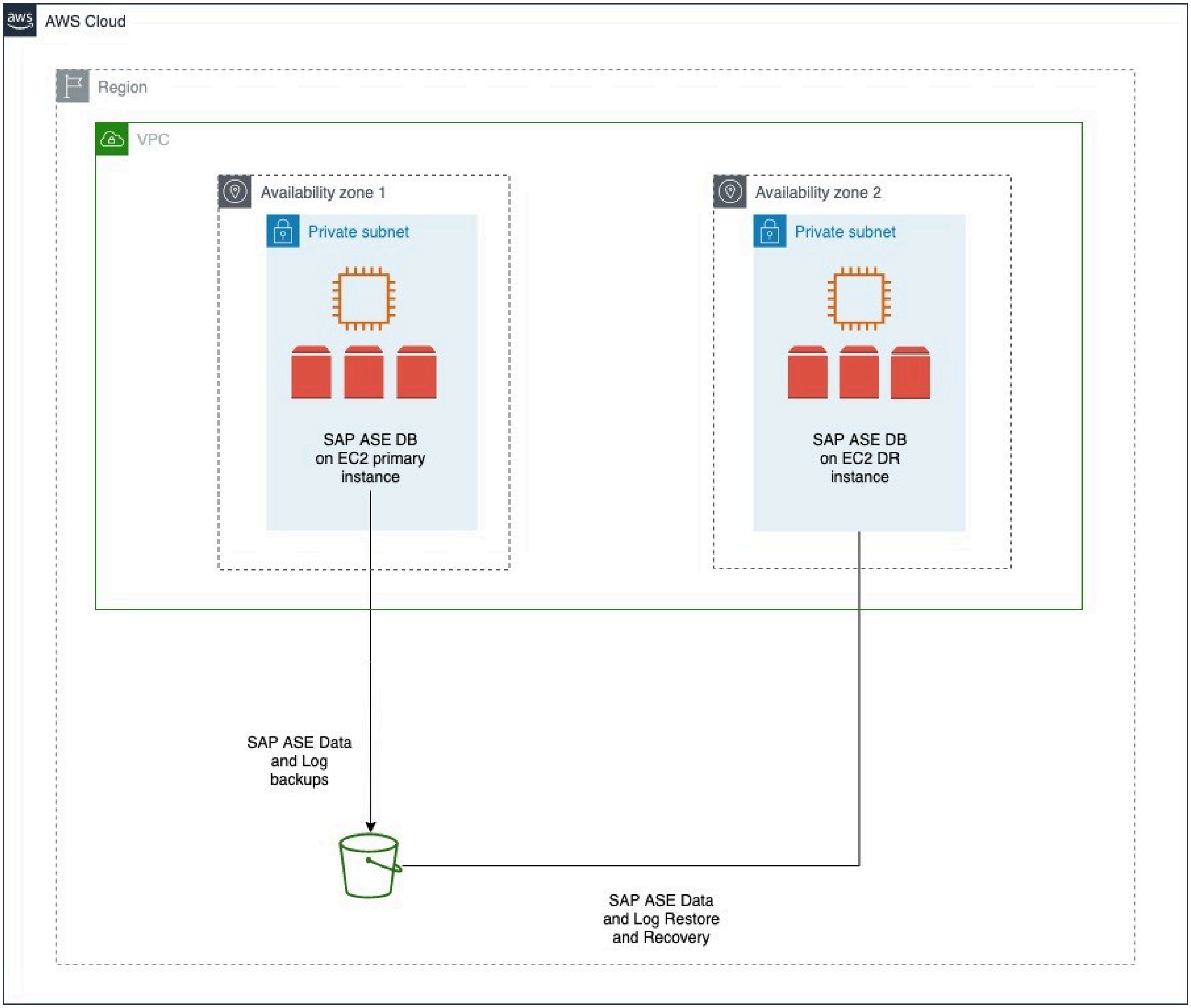
You can store database backups in Amazon S3 and use Amazon S3 Cross-Region Replication (CRR) to replicate your backup in target Region. This method enables automatic, asynchronous copying of objects across Amazon S3 buckets in different Amazon Regions. To save costs, you can install and configure SAP ASE database on an Amazon EC2 instance in your disaster recovery Region, and shut the instance. Restart the instance to restore and recover database from the replicated Amazon S3 bucket as needed.

Alternatively, you can use Amazon CloudFormation, Amazon Cloud Development Kit (Amazon CDK) or third-party automation tools to launch an Amazon EC2 instance and to install and configure the SAP ASE database when needed. This helps save costs on Amazon EC2 and Amazon EBS. You must create and test automations before implementation. We recommend performing frequent disaster recovery drills on automations.

The time to recover the database is dependent on the size of the database. Any log files that are not copied over to the disaster recovery Regions are lost and cannot be used for recovery. This option has higher recovery time and point objectives but offers lower costs in comparison to other options. You can use Amazon S3 Replication Time Control to reduce your recovery point objective. For more information, see Using Amazon S3 Replication Time Control.


AWS Cloud
Region 1
VPC
Availability zone
Private subnet
SAP ASE DB on EC2 primary instance
SAP ASE Data and Log backups
Region 2
VPC
Availability zone
Private subnet
SAP ASE DB on EC2 DR instance
SAP ASE Data and Log Restore and Recovery
S3 Cross Region Replication

You can also recover the SAP ASE database backups in the same Amazon Region, in case of an Availability Zone failure. Amazon EBS snapshots and Amazon S3 bucket data is automatically replicated within the Region. In the event of an Availability Zone failure, an Amazon EC2 instance can be created in a different Availability Zone of the same Region. It is created from the Amazon EBS snapshots of the source Amazon EC2 instance. The SAP ASE database is restored from the backups in the Amazon S3 bucket. Amazon S3 One Zone-IA is the only exception to automatic replication. For more information, see Amazon S3 Storage Classes.

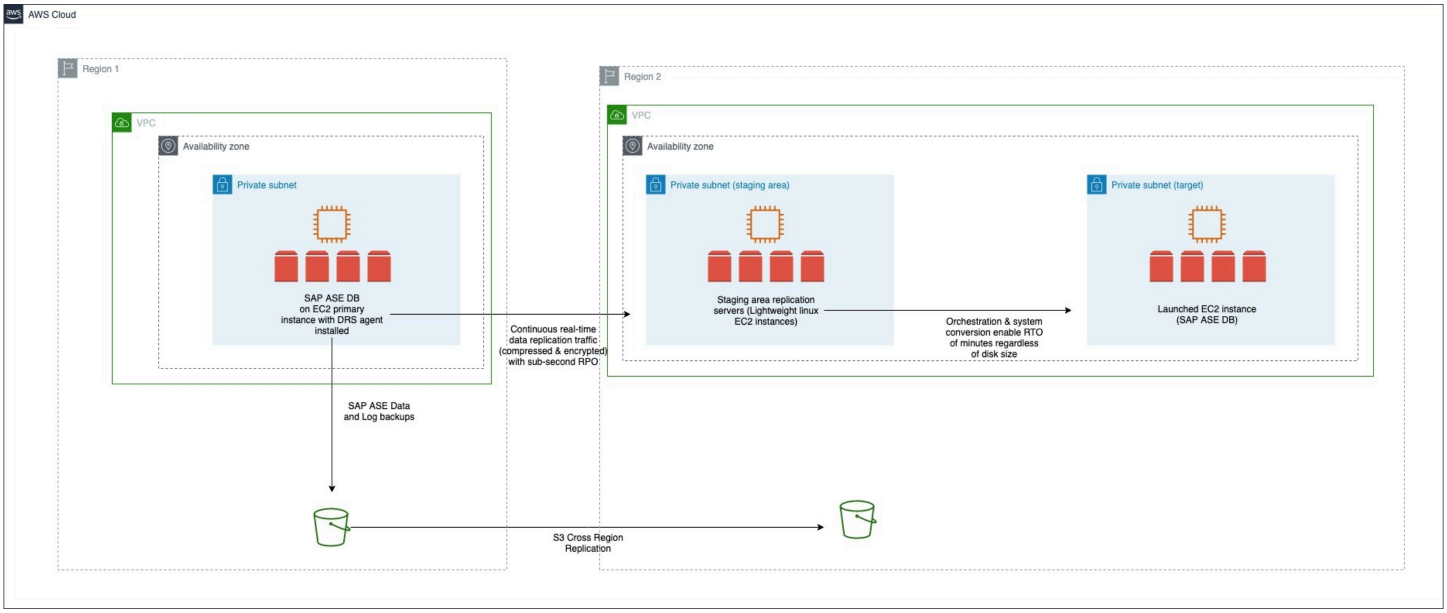

AWS Cloud
Region
VPC
Availability zone 1
Private subnet
SAP ASE DB on EC2 primary instance
Availability zone 2
Private subnet
SAP ASE DB on EC2 DR instance
SAP ASE Data and Log backups
SAP ASE Data and Log Restore and Recovery

Option 3 – disaster recovery using Amazon Elastic Disaster Recovery

You can use Amazon Elastic Disaster Recovery to replicate source servers from the primary Region to a secondary Region. Elastic Disaster Recovery uses block-level replication, and is not application-aware.

Elastic Disaster Recovery is only used for disaster recovery. You can use Amazon S3 Cross Region Replication for backup.

For more information, see Disaster recovery for SAP workloads on Amazon using Amazon Elastic Disaster Recovery.


AWS Cloud
Region 1
VPC
Availability zone
Private subnet
SAP ASE DB on EC2 primary instance with DRS agent installed
Continuous real-time data replication traffic (compressed & encrypted) with sub-second RPO
SAP ASE Data and Log backups
S3 Cross Region Replication
Region 2
VPC
Availability zone
Private subnet (staging area)
Staging area replication servers (Lightweight linux EC2 instances)
Orchestration & system conversion enable RTO of minutes regardless of disk size
Private subnet (target)
Launched EC2 instance (SAP ASE DB)

Sizing

Sizing applies to three key areas - compute, network, and storage.

Compute

Amazon has certified multiple instance families with different sizes to run SAP workloads. For more details, see Amazon EC2 Instance Types for SAP.

To provision instances based on your requirements, you can use the Right sizing process. This process can help you optimize costs. Although it is ideal to use the right sizing approach when you move your SAP workloads to Amazon Cloud, it is an ongoing process. We recommend you to use the latest generation of your selected instance family.

For a greenfield (new) deployment of SAP workloads, you can use the Quick Sizer tool to calculate the compute requirement in SAPS. This helps you to select the closest matching Amazon EC2 instance for a price that is most economical for you. Before completing your selection, ensure that the selected Amazon EC2 instance provides enough Amazon EBS and overall network throughput to meet your application requirements.

For migrations, you can use any of the following data sources to decide the right size of your instance:

- Source system utilization and workload patterns, such as EarlyWatch alert reports.
- Source system specification: CPU, memory, storage size + throughput + IOPS, network.

- Source system SAPS rating.

Network

Network performance is often not explicitly stated as a requirement in SAP sizing. Amazon enables you to check the network performance of all Amazon EC2 Instance Types.

Ensure that you have your network components setup to deploy resources related to your SAP workload. If you haven't already setup network components like Amazon VPC, subnets, route tables etc., you can use the, Amazon Quick Start Modular and Scalable VPC Architecture to most effectively deploy scalable Amazon VPC architecture in minutes. After setting up your Amazon VPC, you must set up Amazon EC2 instances within the Amazon VPC for your SAP workloads.

Storage

Amazon Elastic Block Store (Amazon EBS) volumes are designed to be highly available and reliable. Amazon EBS volume data is replicated across multiple servers in an Availability Zone to prevent the loss of data from the failure of any single component. Owing to this built-in protection, you can skip configuring `RAID 1` for these volumes.

You must check that the storage required is enough to provide sufficient I/O performance. The new `gp3` volume is ideal for SAP ASE workloads that require smaller volume size. With `gp3`, the storage throughput and IOPS are decoupled from the size and can scale independently.

The io2 volume is well-suited for I/O-intensive database workloads that require sustained IOPS performance or more than 16,000 IOPS. The `io2 Block Express` is another provisioned IOPS SSD volume for workloads that require sub-millisecond latency, sustained IOPS performance, and more than 64,000 IOPS or 1,000 MiB/s of throughput.

Note

`io2 Block Express` is only supported on select Amazon EC2 instance types. For more information, see Provisioned IOPS SSD volumes.

The following table lists the main directories for SAP ASE database.

Usage	Directory

Database instance root files	/sybase/<SID>
Database data files	/sybase/<SID>/sapdata_1 /sybase/<SID>/ sapdata_X
Database log files	/sybase<SID>/saplog_1
Database temporary tablespace	/sybase/<SID>/saptmp
Diagnostic tablespace for SAPTOOLS	/sybase/<SID>/sapdiag
Directory for ASE backup	/sybasebackup

Operating system

You can deploy your SAP ASE workload on SLES, SLES for SAP, RHEL for SAP with High Availability and Update Services (RHEL for SAP with HA and US) or RHEL for SAP Solutions.

SLES for SAP and RHEL for SAP with HA and US products are available on Amazon Marketplace under an hourly or an annual subscription model.

SLES for SAP

SLES for SAP provides additional benefits, including Extended Service Pack Overlap Support (ESPOS), configuration and tuning packages for SAP applications, and High Availability Extensions (HAE). For details, see the SUSE Linux Enterprise Server for SAP Applications product page to learn more about the benefits of using SLES for SAP. We strongly recommend using SLES for SAP instead of SLES for all your SAP workloads.

If you plan to use Bring Your Own Subscription (BYOS) images provided by SUSE, ensure that you have the registration code required to register your instance with SUSE to access repositories for software updates.

RHEL for SAP

RHEL for SAP with High Availability and Update Services provides access to Red Hat Pacemaker cluster software for High Availability, extended update support, and the libraries that are required to run SAP HANA. For details, see Red Hat Enterprise Linux for SAP offerings on Amazon Web Services FAQ in the Red Hat Knowledgebase.

If you plan to use the BYOS model with RHEL, either through the Red Hat Cloud Access program or other means, ensure that you have access to a RHEL for SAP Solutions subscription. For details, see Overview of Red Hat Enterprise Linux for SAP Solutions subscription in the Red Hat Knowledgebase.

Security and compliance

The following are additional Amazon security resources to help you achieve the optimum level of security for your SAP NetWeaver environment on Amazon:

- Amazon Cloud Security
- CIS Amazon Foundations
- Amazon Well-Architected Framework

Infrastructure hardening

In some cases, you can further lock down the operating system configuration. For instance, to avoid sharing the credentials of your Amazon account with an SAP administrator who needs to log on to an Amazon EC2 instance. Refer to Security in Amazon EC2 and Best Practice 6.2 – Build and protect the operating system to learn more.

You can also use an automated solution provided by Amazon – Amazon Inspector.

Encryption

The important aspect of securing your workloads is encrypting your data, both at rest and in transit. For more details, refer to the following resources.

- Amazon EBS encryption
- Data encryption in Amazon EFS
- Data encryption in Amazon S3
- Protect your SAP data at rest and in transit

You can also refer to the following SAP resources.

- *SAP Note 2481596 – SYB: Encrypted data transfer between SAP system and SAP ASE database (requires SAP portal access)*

- SAP Adaptive Server Enterprise – Database Encryption

Security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. Security groups act at the instance level, not the subnet level.

SAP system is often separated into multiple subnets, with the database in a separate subnet to the application servers, and other components, such as a web dispatcher in another subnet, possibly with external access.

If workloads are scaled horizontally, or high availability is necessary, you may choose to include multiple, functionally similar, Amazon EC2 instances in the same security group. In this case, you must add a rule to your security groups.

If Linux is used, some configuration changes may be necessary in the security groups, route tables, and network ACLs. For more information, see Security group rules for different use cases.

Network ACL

A network access control list (ACL) is an optional layer of security for your Amazon VPC that acts as a firewall for controlling traffic in and out of one or more subnets (they're stateless firewalls at the subnet level). You may set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your Amazon VPC.

See Amazon VPC Subnet Zoning Patterns for SAP on Amazon to understand the network considerations for SAP workloads.

API call logging

Amazon CloudTrail is a web service that records Amazon API calls for your account and delivers log files to you. The recorded information includes the identity of the caller, time of the call, source IP address, request parameters, and response elements returned by the Amazon service. With CloudTrail, you can get a history of Amazon API calls for your account, including API calls made via Amazon Management Console, Amazon SDKs, command line tools, and higher-level Amazon services (such as, Amazon CloudFormation). The Amazon API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

For more information, see What Is Amazon CloudTrail?

Notification on access

You can use Amazon SNS or any third-party application to set up notifications on SSH login to your email address or mobile phone.

Deployment

This section provides information about example deployments.

Standalone deployment

In this example, we set up a sample environment for installation. It includes a public subnet for RDP and SSH access via the internet. We use the Amazon Quick Start for Modular and Scalable Amazon VPC Architecture in a single Availability Zone deployment to create the Amazon VPC, subnets, security groups, and IAM roles. You can refer to this example set up but should also follow your own network layout and comply with security standards, such as the following:

- Using a Landing Zone solution like Amazon Control Tower.
- Working with a cloud team like Cloud Center of Excellence to use existing standards.

Step 1: Prepare your Amazon account

Check the Region where you want to deploy your Amazon resources:

- You pick your region for deployment during the planning phase.
- Display the Amazon Command Line Interface configuration data:

```
                $ aws configure list
```

Ensure that the default region listed in the command output is the same as the target region where you want to deploy your Amazon resources and install SAP workloads. In this deployment, we provision an Amazon EC2 instance.

Note

In this section, the syntax used for the Amazon CLI and Linux commands is specific to the scope of this document. Each command supports many additional options. To learn more, use the `aws help` command.

Step 2: Create a JSON file for Amazon EBS storage

Create a `JSON` file containing the storage requirements for SAP ASE database server volumes. The following is an example `JSON` file with two Amazon EBS volumes for swap and installation directories. You can add more volumes as per your storage design.

```
[
    {
        "DeviceName": "/dev/nvme2n1",
        "Ebs": {
        "VolumeSize": 32,
        "VolumeType": "gp3",
        "DeleteOnTermination": true
        }
    },
    {
        "DeviceName": "/dev/nvme3n1",
        "Ebs": {
        "VolumeSize": 50,
        "VolumeType": "gp3",
        "DeleteOnTermination": true
    }
  }
]
```

Note

In the preceding example, the device name `/dev/nvme2n1` is for Nitro based hypervisors. It differs for non-Nitro based hypervisors. For more information, see Storage configuration.

Step 3: Launch the Amazon EC2 instance

Launch the Amazon EC2 instance for the SAP ASE database installation in your target Amazon Region, using the information gathered in Step 1. You must create the required storage volumes and attach them to the Amazon EC2 instance for the SAP installation, based on the `JSON` file you created in the Amazon EBS storage (Step 2).

```
$ aws ec2 run-instances \
--image-id <AMI-ID> \
--count <number-of-EC2-instances> \
--instance-type <instance-typ> \
--key-name=<name-of-key-pair> \
--security-group-ids <security-group-ID> \
--subnet-id <subnet-ID> \
--block-device-mappings file://<PATH>\<file>.json \
--region <region-ID>
```

Use this command in a single line format, as shown in the following example.

```
aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxx --count 1 --instance-type m5.large
 --key-name=my_key --security-group-ids sg-xxxxxxxx --subnet-id subnet-xxxxxx --block-
device-mappings file://<PATH>\<file>.json
```

In this example, *m5.large* is the value for the `instance-type` parameter. You must select an Amazon EC2 instance type based on your business requirements.

You can also launch Amazon EC2 instances using the Amazon Management Console. For more information, see Launch an instance.

Step 4: Prepare the Linux Operating System

Before starting the installation, you need to perform Linux specific prerequisite tasks. For more information, refer to the following SAP Notes (requires SAP portal access).

- SAP Note 1554717 – SYB: Planning information for SAP on ASE
- SAP Note 1748888 – SYB: Inst.Systems Based on NW 7.3 and Higher: SAP ASE

Step 5: Prepare each Amazon EC2 instance for SAP ASE installation

Download the SAP installation media as per the SAP installation guide, for the version of SAP NetWeaver you want to install on your Amazon EBS volumes. Locate your installation guide on the Guide Finder for SAP NetWeaver and ABAP Platform. You can store the SAP installation media using Amazon EFS or an Amazon S3 bucket for later reuse.

If you choose to install the SAP ASE database with high availability deployment across two Availability Zones, repeat the preceding steps for SAP ASE database standby high availability instance in the second Availability Zone.

If you choose to install SAP ASE database with high availability and disaster recovery deployment across two Amazon Regions, repeat the preceding steps in the second Amazon Region in which you want to run the ASE database standby disaster recovery instance.

Step 6: Installing SAP ASE on Amazon EC2 instances

You are now ready to install the SAP ASE software on your Amazon EC2 instances. For more information, see the SAP ASE Database Software Installation section of your SAP NetWeaver installation guide. Locate your installation guide on the Guide Finder for SAP NetWeaver and ABAP Platform.

The following is a non-exhaustive list of post-installation tasks for your SAP ASE database.

- Updating to the most recent patch available
- Installation of additional components
- Configure the SAP ASE backup

For more information, see the Operations section.

High availability disaster recovery deployment

Create an additional Amazon EC2 instance and perform the installation in a secondary Availability Zone. The steps for creating a high availability or disaster recovery instance in a secondary Availability Zone are the same as described in Standalone deployment. You can simplify this step by using the following methods.

- If you have built any automation using Amazon CloudFormation or other tools to create the primary Amazon EC2 instance and install database software, you can use the same automation to build the HA instance.
- You can create an Amazon Machine Image of the primary Amazon EC2 instance and launch another instance in the secondary Availability Zone.

The configuration of high availability or disaster recovery depends on the tools you use. See the next sections for more details.

Note

You must configure cross-regional Amazon VPC peering or Transit Gateway to enable SAP ASE asynchronous replication between two Regions.

Operations

Tagging Amazon resources

A tag is a label that you assign to an Amazon resource. Each tag consists of a key and an optional value, both defined by you. Adding tags to various Amazon resources will make managing SAP environments more efficient, and help you search for resources quickly. Many Amazon EC2 API calls can be used in conjunction with a special tag filter. For more information, see Tagging Amazon resources. The following are some examples of how you can use tags for your operational needs.

Tag name	Tag value
Name	SAP server’s virtual (host) name
Environment	SAP server’s landscape role; for example: SBX, DEV, QAT, STG, PRD.
Application	SAP solution or product; for example: ECC, CRM, BW, PI, SCM, SRM, EP
Owner	SAP point of contact
Service level	Known uptime and downtime schedule

After tagging your resources, you can apply specific security restrictions, such as access control (as seen in the following example policy), based on tag values.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
        "Sid": "LaunchEC2Instances",
         "Effect": "Allow",
        "Action": [
            "ec2:Describe*",
             "ec2:RunInstances"
        ],
        "Resource": [
            "*"
        ]
        },
        {
        "Sid": "AllowActionsIfYouAreTheOwner",
        "Effect": "Allow",
        "Action": [
            "ec2:StopInstances",
            "ec2:StartInstances",
            "ec2:RebootInstances",
             "ec2:TerminateInstances"
        ],
        "Condition": {
            "StringEquals": {
            "ec2:ResourceTag/PrincipalId": "${aws:userid}"
            }
        },
        "Resource": [
            "*"
        ]
        }
    ]
}
```

IAM only allows specific permissions based on the tag value. In this scenario, the current ID must match the tag value to enable permissions for the user. For more information, see Tag your Amazon EC2 resources.

Monitoring

Amazon provides multiple native services to monitor and manage your SAP environment. CloudWatch and CloudTrail can be used to monitor your underlying infrastructure and APIs. CloudWatch provides ready-to-use KPIs for CPU, disk utilization, and enables you to create custom metrics for KPIs that you want to monitor. CloudTrail allows you to log the API calls made to your Amazon infrastructure components.

Operating system maintenance

In general, operating system maintenance across large estates of Amazon EC2 instances can be managed by using:

- Third-party products, such as those available on Amazon Marketplace.
- Amazon Systems Manager

The following are some key operating system maintenance tasks.

Patching

You can follow SAP recommended patching process to update your landscape on Amazon. With Amazon Systems Manager Patch Manager, you can roll out OS patches according to your corporate policies. It has multiple benefits:

- Scheduling based on tags
- Defining patch baselines
- Auto-approving patches with lists of approved and rejected patches

Amazon Systems Patch Manager integrates with IAM, CloudTrail, and CloudWatch Events to provide a secure patching experience that includes event notifications and the ability to audit usage. For details about the process, see How Patch Manager operations work. Third-party products are available on Amazon Marketplace.

Maintenance Windows

Amazon Systems Manager Maintenance Windows lets you define a schedule to perform potentially disruptive actions on your instances, such as patching an operating system, updating drivers, installing software or patches.

Administrator access

For administrative purposes, you can access the backend of your SAP systems via SSH or Amazon Systems Manager Session Manager.

Automation

Amazon Systems Manager Automation simplifies common maintenance and deployment tasks of Amazon EC2 instances and other Amazon resources. For more information, see Amazon Systems Manager Automation.

Automation using Infrastructure-as-Code with Amazon CloudFormation

We recommend following the principle of Infrastructure-as-Code (IaC) for automating and maintaining your workloads on Amazon. Amazon CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment in a repeatable and automated manner.

Cost optimization

We recommend cost optimization as an ongoing process. There are many Amazon services that help with budgeting, cost control and optimization. For more details, see Cost Optimization Pillar - Amazon Well-Architected Framework and Cost Optimization Pillar -SAP Lens.

Compute & storage

Compute

Amazon EBS volumes are exposed as NVMe block devices on Instances built on the Nitro System. When changing Amazon EC2 instance types from a previous generation to a Nitro generation, NVMe device IDs associated with the volume can change. To avoid mount errors during change of instance type or instance reboots, you need to create a label for your file systems and mount it by the label, *and not* the NVMe IDs. For more details, see support article.

Aside from operating system maintenance, you should consider maintenance for your Amazon EC2 instances. It can be driven by using Creating your own runbooks. The following are some examples.

- Use `Amazon-StopEC2InstanceWithApproval` to request one or more IAM users approve the instance stop action. After the approval is received, runbook stops the instance.

- Use `Amazon-StopEC2Instance` to automatically stop instances on a schedule, using CloudWatch Events or a Maintenance Window task. For example, you can configure an Automation workflow to stop instances every Friday evening and restart on Monday mornings. Note that this automation will only stop and start the Amazon EC2 instance. You must create additional document to gracefully stop and start SAP applications and database and then use the Amazon Systems Manager to run such automations.
- Use `Amazon-UpdateCloudFormationStackWithApproval` to update resources that were deployed using Amazon CloudFormation template. The update applies a new template. You can configure the Automation to request approval by one or more IAM users before the update begins.

You can also use Amazon Instance Scheduler to configure custom start and stop schedules for Amazon EC2 and Amazon RDS instances.

Storage

The following are the storage services used across this guide.

- Amazon EBS provides persistent storage for SAP applications and database. Amazon EBS volumes can be resized and even have the volume type changed without disrupting the applications. For more details, see Amazon EBS Elastic Volumes. After modifying the Amazon EBS volume, you need to extend the file system to match the extended volume size. For more details, see Extend a Linux file system after resizing a volume.
- Amazon EFS does not require you to explicitly provision storage, you pay only for your usage. It is built to scale on demand, without disrupting applications, growing and shrinking automatically as you add and remove files. This ensures that your applications have the required storage.
- Amazon S3 also does not require you to explicitly provision storage, you pay only for your usage. You can use Object lifecycle management to set rules that define when objects are transitioned or archived to colder storage (Amazon S3 IA or S3 Glacier) and when they expire. For more information, see Managing your storage lifecycle.

Backup & restore

Snapshots and AMIs

A common approach for backing up your SAP NetWeaver application servers is using snapshots and AMIs.

The SAP application data is stored on Amazon EBS volumes attached to the SAP NetWeaver application servers. You can back up the data on these volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups of Amazon EBS volumes, which means that only the blocks on the device that have changed after your most recent snapshot are saved. For more information, see Create Amazon EBS snapshots.

An Amazon Machine Image (AMI) provides the information required to launch an instance along with a block device mapping of all Amazon EBS volumes attached to it.

Amazon EC2 powers down the instance before creating the AMI to ensure that everything on the instance is stopped and in a consistent state during the creation process. If you're confident that your instance is in a consistent state appropriate for AMI creation, you can check the *No Reboot* option.

You can use Amazon Backup to centrally configure backup policies and monitor backup activity for these snapshots. Once you have completed the SAP installation and post-installation steps, create an image of the instance.

```
aws ec2 create-image --instance-id i-1234567890abcdef0 --name "My server" --description
 "An AMI for my server"
```

Amazon provides a very simple and quick way to copy an SAP system. You can use the Amazon Console Home or the Amazon CLI to create a new AMI of an existing SAP system. You can then launch exact copies of the original system from the new AMI. For more details, see Amazon Machine Images (AMI).

Backup to Amazon S3

You can perform traditional file-based backup to Amazon S3 from your Amazon EBS volumes. One way to take backup is to use Amazon CLI and initiate it by using Amazon Systems Manager Run command, so that you can centrally manage the backups.

Backup with third-party products

Many third-part products for Amazon services are certified by SAP. For more information, see Amazon SAP Competency Partners.

Amazon EFS backup

Using Amazon Backup, you can centrally configure backup policies and monitor backup activity for Amazon resources, including Amazon EFS file systems.

Alternatively, you can perform a file-level backup of your Amazon EFS file system to Amazon S3. You can do this by running a file-level copy to Amazon S3 from any Amazon EC2 instance running in the same region. This can be automated and scheduled using Amazon Systems Manager Run Command in combination with CloudWatch Events.

Backup and restore for ASE database

You must to regularly backup your operating system and database to recover them in case of any failure. Amazon Cloud provides various services and tools that you can use to backup your SAP ASE database.

Storage snapshots

You can backup your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only blocks on the device that have changed after your most recent snapshot are saved. Snapshots of Amazon EBS volumes can be created for backup of SAP ASE database file systems.

See How to use snapshots to create an automated recovery procedure for SAP ASE databases to learn more.

SAP ASE database backups

You can configure your SAP ASE database to store backups on Amazon EFS or local Amazon EBS volumes. You must configure regular backups for Amazon EFS. For more information, see Backing up your Amazon EFS file systems. You can reduce costs by enabling Amazon EFS storage classes to retain cold backups in infrequent access. For more information, see Amazon EFS Infrequent Access.

You can also configure backups to be store on Amazon EFS volumes and to be regularly uploaded to Amazon S3. Use `DBACOCKPIT` to schedule backup frequency. You can also use Amazon Systems Manager Maintenance Windows to schedule backup frequency.

Amazon SNS enables you to setup push notifications for success or failure. Once backups are stored in Amazon S3, you can use lifecycle policies to define data retention timeline. For more information, see Managing your storage lifecycle.

You can improve Amazon S3 data upload performance with Gateway endpoints and Amazon CLI. For more information, see Gateway endpoints for Amazon S3 and Amazon CLI S3 Configuration.

Review the following SAP Notes (portal access required) for more details.

- SAP Note 1585981 - SYB: Ensuring Recoverability for SAP ASE
- SAP Note 1887068 - SYB: Using external backup and restore with SAP ASE
- SAP Note 1588316 - SYB: Configure automatic database and log backups
- SAP Note 1618817 - SYB: How to restore an SAP ASE database server (UNIX)

To use third-party tools to backup your SAP ASE database, see Amazon Storage Competency Partners.

Disaster recovery

See Disaster recovery deployment to learn about disaster recovery for your SAP ASE database.

Perform a DNS change

In case of manual failover, you may install SAP application servers using a virtual hostname and perform a DNS change to direct the SAP application servers to the new primary database server. For a DNS resolution in Amazon, you can use any of the following options.

- Amazon Route 53 enables you to create a private hosted zone for your environment and an A record for the virtual hostname used for SAP ASE database. Initially, this A record is mapped to the IP address of the primary SAP ASE database instance.
- You can maintain your own DNS server on-premise or on your Amazon EC2 instances. You can create an A record there for your virtual hostname used for SAP ASE database. Initially, this A record is mapped to the IP address of the primary SAP ASE database instance.
- With the Amazon Directory Service, you can create an A record for the virtual hostname used for SAP ASE database.

With any of the previously mentioned options, you can change the A record to a private IP address of the primary database instance in case of a failover. This DNS change can also be automated using Amazon services and scripts.

Resources

SAP on Amazon customers have the flexibility to deploy SAP ASE database on the scalable, on-demand Amazon EC2 platform in a highly available manner. They don't have to invest in costly capital expenditures for the underlying infrastructure. By combining the Amazon platform flexibility and SAP installation techniques, our customers greatly improve the availability of their deployments. For more details, see SAP on Amazon Case Studies.

Support

Amazon offers three levels of support. Amazon Business Support provides resources and technical support for customers running SAP workloads on Amazon. Amazon Enterprise Support and Amazon Enterprise On-Ramp Support offers support to customers running mission critical SAP production workloads on Amazon.

To learn more about this, see SAP Note 1656250 – SAP on Amazon: Support prerequisites (requires SAP portal access).

SAP NetWeaver on Amazon Automation

Amazon Systems Manager is a collection of capabilities that help you manage your applications and infrastructure running in Amazon Cloud. Systems Manager simplifies application and resource management, shortens the time to detect and resolve operational problems, and helps you manage your Amazon resources securely at scale.

This chapter contains information about how to use Systems Manager to automate management of your SAP applications.

Automation prerequisites

Because SAP automation in Amazon Cloud relies on Systems Manager, you must satisfy the Systems Manager prerequisites. In addition, there are prerequisites specified in this chapter for specific tasks, such as SAP installation and operating system patching. Those prerequisites are listed in their respective sections.

Before you begin, verify the following prerequisites, which apply to all of the automation tasks described in this chapter:

- You must have the latest SSM agent installed on your Amazon EC2 instances. For more information, see Manually installing SSM Agent on EC2 instances for Linux in the *Amazon Systems Manager User Guide*.
- You must satisfy the prerequisites for Systems Manager. For more information, see Systems Manager prerequisites in the *Amazon Systems Manager User Guide*.

Automated SAP installation

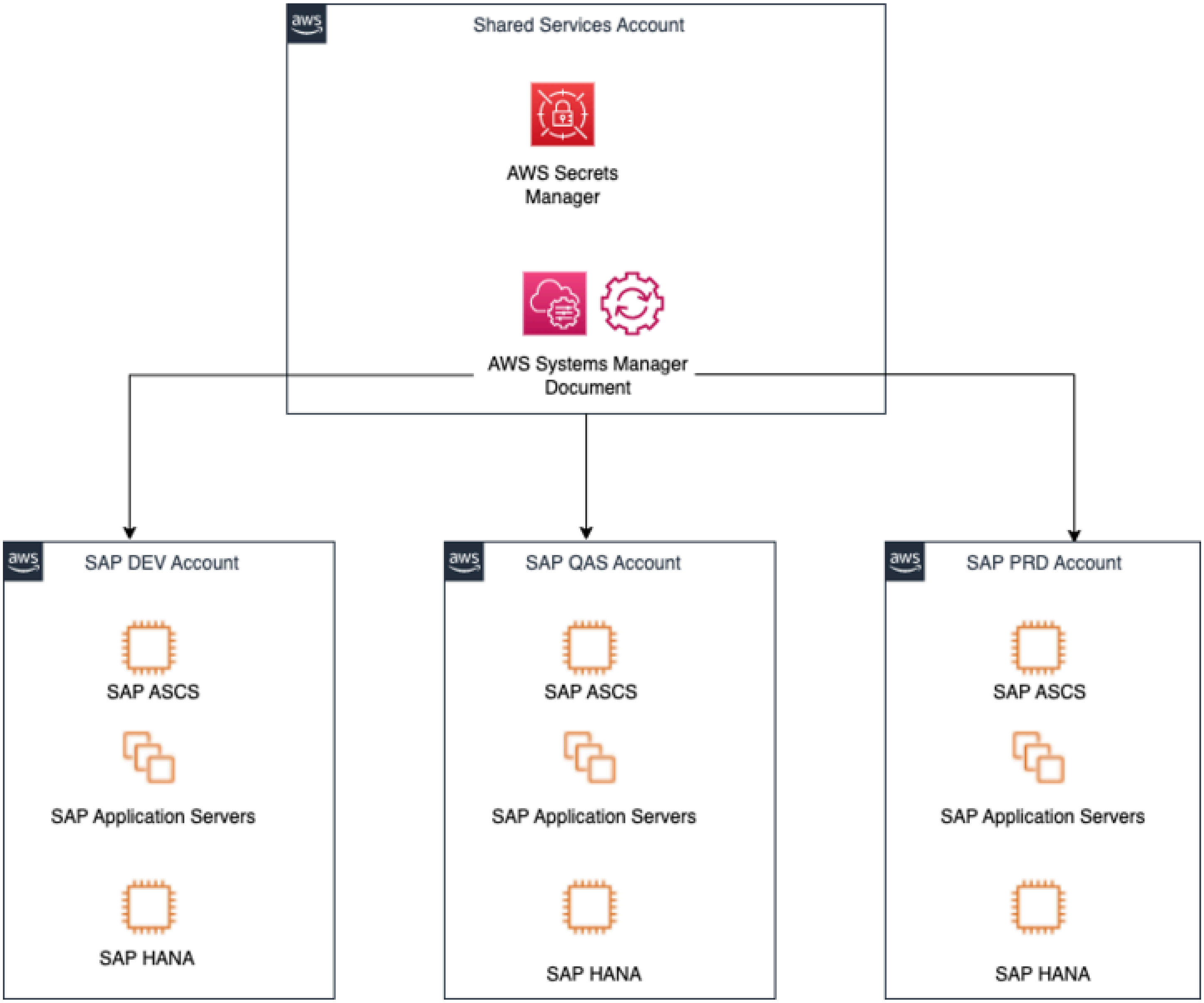
Deploying an SAP system requires significant effort in building an infrastructure that conforms to SAP specifications. Installation, operating system configuration, and configuration of parameters based on the type of SAP workload must be repeated for the development, quality, and production landscape. You can automate this installation and configuration using Amazon Systems Manager. Automating the installation and configuration of your SAP landscape helps your team stay compliant with auditable policies related to configuration as code. In addition, it turns the SAP installation into an easily repeatable process, which makes the quality of the outcome easier to improve because you can simulate it and run it multiple times using the same source of information.

The solution described here uses Systems Manager documents to install a distributed SAP landscape that contains the following:

- ABAP SAP Central Services (ASCS)
- Database instance
- Primary Application Server (PAS)
- Additional Application Servers (AAS)

Automated SAP installation architecture

The example architecture shown in the diagram below uses a centralized Amazon account that stores the Amazon Systems Manager document (SSM document). The document is shared with Amazon accounts that host Amazon EC2 instances running SAP HANA workloads.



You can use multiple Amazon accounts and Amazon organizations to arrange the accounts into a hierarchy and group them into organizational units. These organizational units can be used for things such as consolidated billing, workload isolation, and administrative isolation. You can create separate Amazon accounts for development, testing, staging, and production on a per-application basis as part of an organization. For more information, see the https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html *Amazon Organizations">User Guide*.

Systems Manager automation provides multi-account and multi-Amazon Region support that allows you to execute your own automation documents across multiple accounts from a central Amazon account. You can centralize the SSM documents into a Shared Services account or use an automation account. The automation account can be the Amazon account that runs SAP workloads or a dedicated account that only runs SSM documents. Using a centralized Amazon for automation reduces administration overhead by maintaining the SSM document and its dependencies in a single account. For more information about Shared Services, see Infrastructure OU - Shared Services account in the *Amazon Security Reference Architecture*.

In order for Systems Manager to trigger automation documents from a centralized Amazon account to the connected accounts, IAM permissions are required in the automation and child accounts. For more information, see Running automations in multiple Amazon Regions and accounts in the *Amazon Systems Manager User Guide*.

You can share SSM documents privately or publicly with accounts in the same Region. To privately share a document, modify the document permissions and allow specific individuals to access it based on their Amazon account ID. For more information, see Sharing SSM documents in the *Amazon Systems Manager User Guide*.

Components

The installation automation workflow includes automation runbooks and SSM command documents.

Automation runbook

An automation runbook defines the actions that Systems Manager performs on your managed instances and other Amazon resources. A runbook contains one or more steps that run in sequential order. For more information, see the following documentation:

- What is an automation? in the *Amazon Systems Manager User Guide*
- Systems Manager Automation runbook reference

SSM command document

If a task must be repeated multiple times on multiple hosts, you can create it as an SSM command document. These documents are usable across multiple runbooks. For more information, see Systems Manager Command document plugin reference in the *Amazon Systems Manager User Guide*.

You can make the SSM command document as granular as you need, based on factors such as:

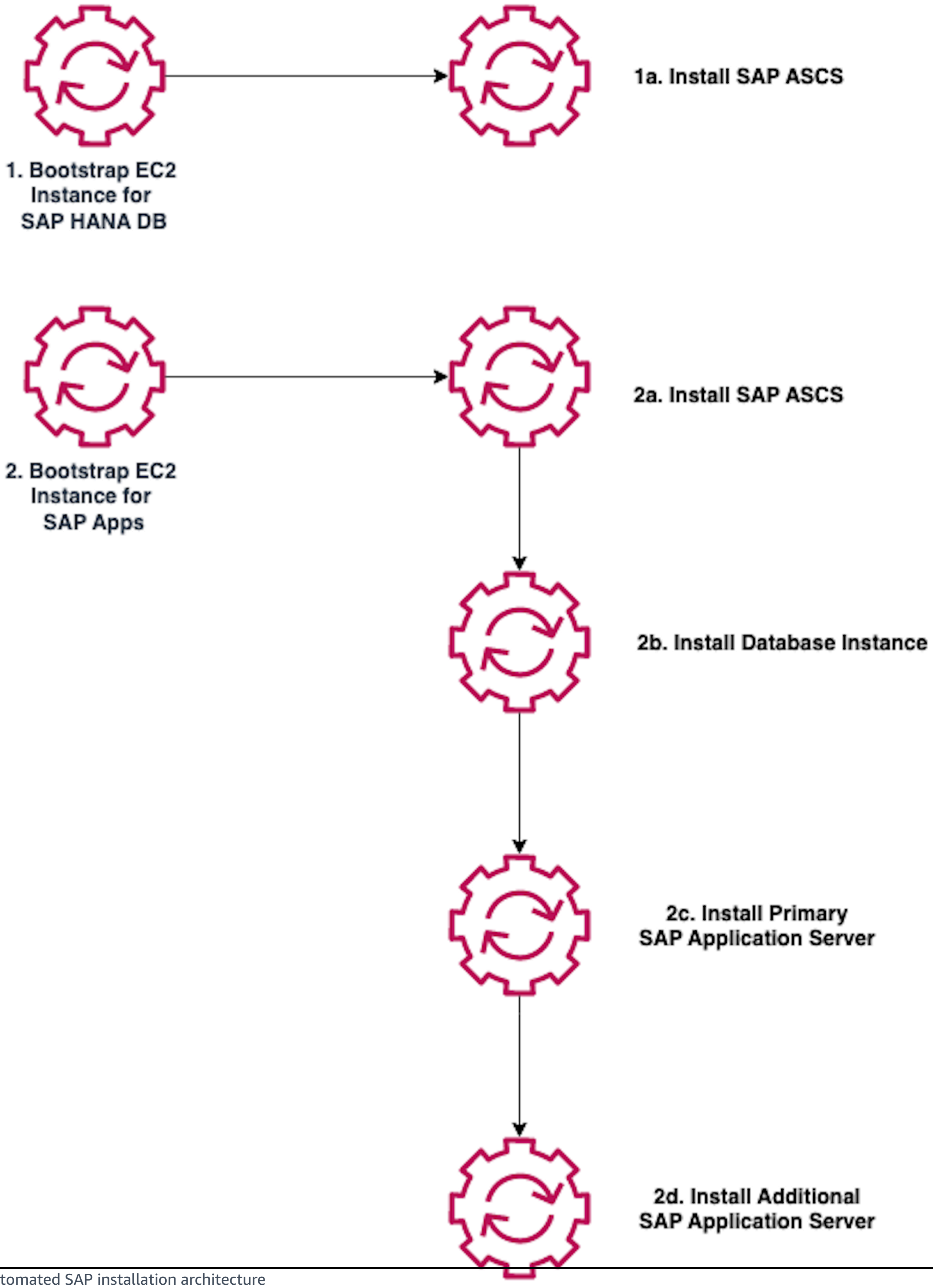
- Segregation of duties

- Types of SAP systems that are being deployed
- Complexity of SAP systems that are being deployed
- Security

Workflow

As an example, each runbook can be made up of several SSM documents that perform a specific configuration. The following runbooks can be used, which are illustrated in the diagram below.

- Bootstrap Amazon EC2 instances for SAP HANA database
- Bootstrap Amazon EC2 instances for SAP application servers
- Install SAP HANA database
- Install ABAP SAP Central Services (ASCS)
- Install a database instance
- Install a primary application server
- Install an additional application server


1a. Install SAP ASCS
1. Bootstrap EC2 Instance for SAP HANA DB
2a. Install SAP ASCS
2. Bootstrap EC2 Instance for SAP Apps
2b. Install Database Instance
2c. Install Primary SAP Application Server
2d. Install Additional SAP Application Server

Automated SAP NetWeaver on Amazon installation prerequisites

In addition to the prerequisites described in the Automation prerequisites section of this guide, verify the following prerequisites that are specific to automated SAP installation:

- You must have an existing infrastructure deployed.

  The example described in this guide uses a SAP HANA database, an SAP Central Services (ASCS) instance, and a database instance. The *Amazon for SAP* blog has a Terraform your SAP Infrastructure on Amazon example.
- SAP media files must be available.

  You must provide the SAP installation media files, which are obtained from SAP, in an Amazon S3 bucket. For more information, see Make SAP application software available for Amazon Launch Wizard for SAP to deploy SAP in the *Amazon Launch Wizard User Guide*. If you use the sample code provided in this guide, the media files are copied to local Amazon Elastic Block Store volumes.

SAP Notes

Read the following SAP Note:

- SAP Note: 2230669 - System Provisioning Using a Parameter Input File

Additional references

Before you begin, you can also familiarize yourself with how SAP works on Amazon by reading the following documentation:

- SAP on Amazon Planning in the *General SAP Guides*
- Amazon EC2 instance types for SAP on Amazon in the *General SAP Guides*
- SAP NetWeaver Environment Setup for Linux on Amazon in the *SAP NetWeaver Guides*

Configuring automated SAP installation

The sections below contain detailed instructions on how to configure automated SAP NetWeaver on Amazon installation.

Customize the Systems Manager document

This section shows you how to customize the Amazon Systems Manager document (SSM document) for the automated SAP installation. For more information about SSM documents, see Amazon Systems Manager Documents in the *Amazon Systems Manager User Guide*.

This section details the content that goes into the SSM document. For information about how to create the document, see Create an SSM document (console) in the *Amazon Systems Manager User Guide*.

As you create your SSM document, we recommend you do the following:

- Use schema version 2.2. For more information, see SSM document schema features and examples in the *Amazon Systems Manager User Guide*.
- Use Parameter Store to easily reference parameters that you use often. For more information, see Amazon Systems Manager Parameter Store in the *Amazon Systems Manager User Guide*.

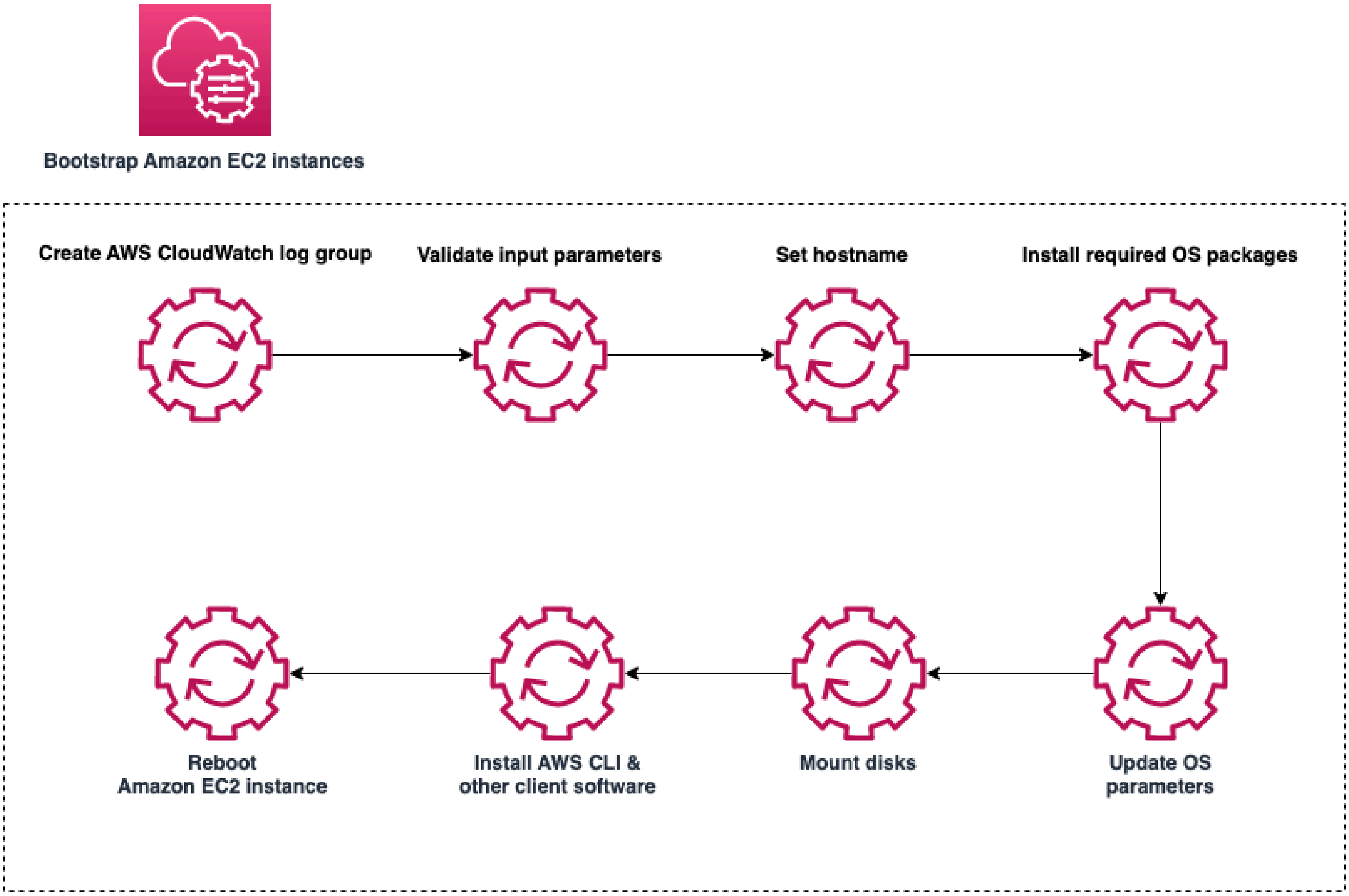
> **Tip**
>
> You can find sample SSM documents and parameter files in the aws-samples/terraform-aws-sap-netweaver-on-hana GitHub repository.

Bootstrap Amazon EC2 instances

Bootstrapping in Amazon EC2 consists of adding commands or scripts to the user data section of the instance. These commands and scripts can be executed when the instance starts. This simplifies configuration tasks. For more information, see Run commands on your Linux instance at launch in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

For SAP installation, bootstrapping includes several tasks, such as setting the hostname, installing operating system packages, setting operating system parameters, installing Amazon Data Provider for SAP, installing agents for monitoring, logging, and alerting, and mounting disks for the SAP HANA database instance and SAP application servers.

The image below shows the steps required for the bootstrap instance SSM document.


Bootstrap Amazon EC2 instances
Create AWS CloudWatch log group
Validate input parameters
Set hostname
Install required OS packages
Update OS parameters
Mount disks
Install AWS CLI & other client software
Reboot Amazon EC2 instance

The SSM document accepts required and optional parameters. The code below is an example parameter section for bootstrapping an SAP HANA database instance or any SAP NetWeaver application server instance:

```
parameters:
    AutomationAssumeRole:
        type: String
        description: "(Optional) The ARN of the role that allows Automation to perform
 the actions on your behalf. "
        default: ''
    InstanceId:
        type: String
        description: "(Required) The instance ids to bootstrap before SAP HANA
 installation"
        default: ''
    HostnameTagKey:
        type: String
        description: "(Required) The tag key where the hostname is stored"
        default: 'Hostname'
    DnsPrivateZoneName:
        type: String
        description: "(Optional) DNS Zone name to specify FQDN in hosts"
        default: 'sapteam.net'
    EfsFileSystemId:
        type: String
        description: (Required) The EFS file system id for /sapmnt folder
        default: ‘fs-7df7edae’
    MasterPassword:
        type: String
        description: '(Required) SAP NetWeaver Master Password'
        default: ''
    IniFile:
        type: String
        description: '(Required) Path to INI file'
        default: '/sapmnt/software/sapinstall.params'
    CloudWatchLogGroupName:
        type: String
        description: "(Required) Cloud Watch log group for the log output"
        default: '/customer/SAP/dev-setup-logs'
```

The next section of the SSM document is the `mainSteps` section.

A composite SSM document is a custom document that performs a series of actions by running one or more secondary SSM documents. Composite documents promote infrastructure as code by allowing you to create a standard set of SSM documents for common tasks, such as bootstrapping software or domain-joining instances. For example, you can create a composite document with secondary SSM documents for each bootstrap item, as listed below:

- Setting the hostname
- Installing operating system packages for SAP HANA
- Setting the operating system parameters for SAP HANA
- Mounting disks for SAP HANA
- Installing the Amazon Data Provider agent for SAP

Composite and secondary documents can be stored in Systems Manager, private and public GitHub repositories, or Amazon S3. They can be created in JSON or YAML. For more information, see Creating composite documents in the *Amazon Systems Manager User Guide*.

The code below shows the `mainSteps` section of the SSM document with the composite and secondary documents:

```
mainSteps:
- name: Prepare_logs
  action: aws:runCommand
  inputs:
    DocumentName: d4h-prepare-sap-installation-logs
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Set_hostname
  action: aws:runCommand
  inputs:
    DocumentName: d4h-set-hostname
    InstanceIds:
    - '{{ InstanceId }}'
    Parameters:
      PrivateZone: '{{ DnsPrivateZoneName }}'
      Hostname: '{{ Get_hostname.Hostname }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Install_Packages
  action: aws:runCommand
  inputs:
    DocumentName: d4h-install-sap-packages
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Set_OS_Parameters
  action: aws:runCommand
  inputs:
    DocumentName: d4h-set-sap-hana-parameters
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
```

```
- name: Mount_Disks
  action: aws:runCommand
  inputs:
    DocumentName: d4h-mount-hana-disks
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
- name: Install_Aws_Sap_Data_Provider
  action: aws:runCommand
  isCritical: false
  inputs:
    DocumentName: d4h-install-sap-aws-data-provider
    InstanceIds:
    - '{{ InstanceId }}'
    CloudWatchOutputConfig:
      CloudWatchLogGroupName: '{{ CloudWatchLogGroupName }}'
      CloudWatchOutputEnabled: True
```

Install the SAP HANA database

After you bootstrap the Amazon EC2 instances, you must install the SAP HANA database. For this installation, you can store the SAP HANA master password in the SSM document Parameter Store or use it as an input to the SSM document and reference it in the `passfile.xml` file.

The code below is an example SSM document for an SAP HANA installation:

```
mainSteps:
- action: "aws:runShellScript"
  name: "Run_installer"
  inputs:
    runCommand:
    - #!/bin/bash
    - HANA_MEDIA=`find /software/hana -name "DATA_UNITS"`
    - if [ -z "$HANA_MEDIA" ]
    - then
    -   echo "Could not find the DATA_UNITS folder in /software/hana. Check if
 everything was downloaded successfully. Exiting..." | tee -a $SSM_LOG_FILE
    -   exit 1
    - fi
    - PASSFILE=$HANA_MEDIA/../passfile.xml
    - chmod +x $HANA_MEDIA/HDB_LCM_LINUX_X86_64/hdblcm
    - HOSTNAME=`(hostname)`
    - INSTANCE=`(instancenumber)`
    - SID=`echo "{{sid}}" | tr a-z A-Z`
    - echo "Executing installation from $HANA_MEDIA/HDB_LCM_LINUX_X86_64/hdblcm for SID
 $SID, instance $INSTANCE, hostname $HOSTNAME..."
    - cat $PASSFILE | $HANA_MEDIA/HDB_LCM_LINUX_X86_64/hdblcm --action=install --
components=client,server --batch --autostart=1 -sid=$SID  --hostname=$HOSTNAME --
number=$INSTANCE  --read_password_from_stdin=xml | tee -a $SSM_LOG_FILE
    - echo "`date` Installation finished. Please check logs..." | tee -a $SSM_LOG_FILE
    - rm $INIFILE
```

Install SAP

Installing SAP includes ABAP SAP Central Services (ASCS), the database instance, and the primary and additional application server installation.

First, you create a parameter file with the required parameters. Refer to the SAP installation guide for the parameters that are specific to your installation. The code below is an example parameter file:

```
mainSteps:
- action: "aws:runShellScript"
  name: "Prepare_sapinstall_ini"
  inputs:
    runCommand:
    - #!/bin/bash
    - SAPINSTALL_INI_FILE={{ IniFile }}
    - SID=`echo "{{Sid}}" | tr a-z A-Z`
    - SAPSYSUID=`sapsysuid`
    - SIDADMUID=`sidadmuid`
    - SWTARGET=/sapmnt/software/
    - DOMAINNAME={{ DnsPrivateZoneName }}
    - HOSTNAME=`hostname`
    - FQDN=${LHOSTNAME}.${DOMAINNAME}
    - sed -i "s|default_scsVirtualHostname|${HOSTNAME}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_scsInstanceNumber|00|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_ssmpass|{{ MasterPassword }}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_sid|${SID}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_fqdn|${DOMAINNAME}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_sapsysGID|${SAPSYSUID}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_AdmUID|${SIDADMUID}|g" ${SAPINSTALL_INI_FILE}
    - sed -i "s|default_downloadBasket|${SWTARGET}|g" ${SAPINSTALL_INI_FILE}
    - echo '`date` Prepared the Ini File:...' | tee -a $SSM_LOG_FILE
```

The next step is to start the installation using the SAP silent, or unattended, installation mode, referring to the parameter file as in the example code below:

```
mainSteps:
- action: "aws:runShellScript"
  name: "Execute_installation"
  inputs:
    runCommand:
    - #!/bin/bash
    - echo '`date` Starting the Installation process...' | tee -a $
    - SYSTEMNUMBER=`systemnumber`
    - SAPAliasName=`hostname`
    - SWPMFILE=`find /sapmnt/software/SWPM-SUM/ -name SWPM*SAR`
    - chmod 775 /sapmnt/software/utils/sapcar
    - /sapmnt/software/utils/sapcar -xvf $SWPMFILE -R /sapmnt/software/SWPM
    - chmod 755 /sapmnt/software/SWPM/sapinst
    - cd /sapmnt/software/SWPM
    - ./sapinst SAPINST_INPUT_PARAMETERS_URL=/sapmnt/software/sapinstall.params
 SAPINST_EXECUTE_PRODUCT_ID={{ProductId}} SAPINST_USE_HOSTNAME=${SAPAliasName}
 SAPINST_SKIP_DIALOGS="true" SAPINST_START_GUISERVER=false | tee -a $SSM_LOG_FILE
```

You can add additional sections in the SSM document to validate the SAP installation by checking the SAP process running on the host and sending the results to the SSM document log file. The following code is an example of how to do this:

```
- action: "aws:runShellScript"
  name: "Validate_Installation"
  inputs:
    runCommand:
    - #!/bin/bash
    - sid=`echo {{ Sid }} | tr '[:upper:]' '[:lower:]'}`
    - SID=`echo {{ Sid }} | tr '[:lower:]' '[:upper:]'}`
    - HOSTNAME=`hostname`
    - SIDADM=${sid}adm
    - su - $SIDADM -c "stopsap $HOSTNAME" | tee -a $SSM_LOG_FILE
    - su - $SIDADM -c "startsap $HOSTNAME" | tee -a $SSM_LOG_FILE
    - sleep 15
    - _SAP_UP=$(netstat -an | grep 3200 | grep tcp | grep LISTEN | wc -l )
    - echo "This is the value of SAP_UP - $_SAP_UP" | tee -a $SSM_LOG_FILE
    - if [ "$_SAP_UP" -eq 1 ]
    - then
    -   echo "$(date) __ done installing ASCS." | tee -a $SSM_LOG_FILE
    -   exit 0
    - else
    -   echo "$(date) __ ASCS could not be installed successfully. Fix the issue and
 rerun the automation" | tee -a $SSM_LOG_FILE
    -   exit 1
    - fi
- action: "aws:runShellScript"
  name: "Save_services_file"
  inputs:
    runCommand:
    - #!/bin/bash
    - grep -i sap /etc/services > /sapmnt/services
    - if [ -s /sapmnt/services ]
    - then
    -   echo "Services file copied to sapmnt" | tee -a $SSM_LOG_FILE
    -   exit 0
    - else
    -   echo "Services file could not be copied" | tee -a $SSM_LOG_FILE
    -   exit 1
    - fi
```

Tag the Systems Manager document

A tag is a label that you assign to an Amazon resource. Each tag consists of a key and a value, both of which you define. For an overview of tagging Systems Manager resources, see Tagging Systems Manager resources in the *Amazon Systems Manager User Guide*.

For detailed instructions on how to tag SSM documents, see Tagging Systems Manager documents in the *Amazon Systems Manager User Guide*.

Example - tags and access management

You can use tagging for a variety of purposes. For example, if you’re using Amazon Identity and Access Management (IAM), you can control which users in your account can create, edit, or delete tags, and you can implement attribute-based access control (ABAC). For more information, see Grant permission to tag resources during creation and Control access to Amazon EC2 resources using resource tags in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

Example - tags and billing

You can use tags to organize your Amazon bill in a way that reflects your cost structure. To do this, sign up to get your Amazon account bill with tag key values included. For more information about setting up a cost allocation report with tags, see Monthly cost allocation report in the *Amazon Billing User Guide*. To see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see Using cost allocation tags in the *Amazon Billing User Guide*.

Automated operating system patching

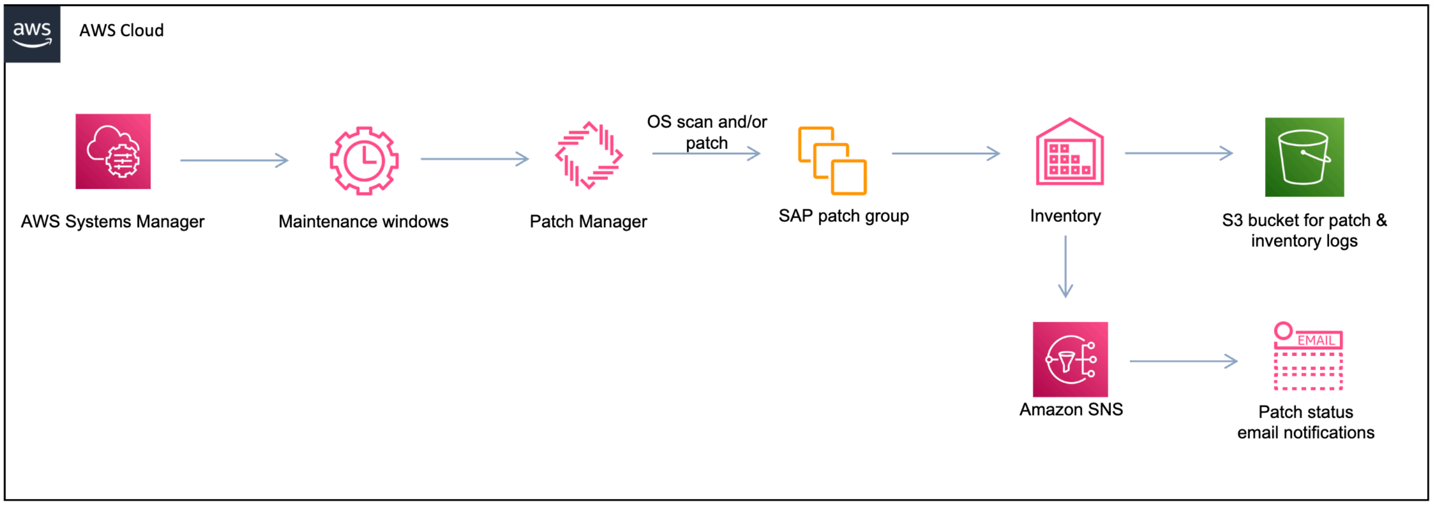
Patch management is an ongoing activity that is a key part of the SAP software lifecycle. It is a critical step in improving security, minimizing risk, remaining compliant, and reducing unplanned downtime. You can use Amazon Systems Manager to automate system patching activities. Systems Manager can reduce the manual effort that is required to manage the SAP landscape, which saves time and IT resources.

An operating system patching strategy should adhere to SAP software lifecycle best practices. Patches should be applied downstream across the landscape, from development, to test, to production. This allows the patches to be tested in less-critical systems before deploying them into production. Because patching is a repeatable process, it can be automated with Systems Manager and can be documented as a standard operating procedures (SOP). This will ensure consistent patch management across the SAP landscape. The SOP should be updated continuously for future maintenance activities.

The sections below describe how to use Systems Manager to apply regular patches that are released by operating system vendors.

Automated operating system patching architecture

The diagram below highlights the Amazon services that you can use to set up automated operating system patching and optional notifications on the patch status using Amazon Simple Notification Service (Amazon SNS).



The topics below contain descriptions of key components of the automated operating system patching setup. Familiarize yourself with them before continuing to the prerequisites.

Topics

- Patch Manager
- Lifecycle hooks

Patch Manager

Patch Manager is a capability of Amazon Systems Manager that automates the process of patching managed nodes with security-related and general operating system updates. You can use Patch Manager to apply patches for operating systems and applications, such as installing service packs on Microsoft Windows nodes and performing minor version upgrades on Linux nodes.

Patch Manager helps to patch fleets of Amazon EC2 instances according to operating system type. This includes versions of Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Oracle Linux, and Microsoft Windows Server that are supported by SAP on Amazon. You can patch your instances on a schedule or on-demand by creating a patching configuration. You can also scan instances to see a report of missing patches or to automatically install missing patches.

Patch Manager integrates with Amazon Identity and Access Management (IAM), Amazon CloudWatch Events, and Amazon Security Hub to provide a secure patching experience that includes event notifications and the ability to audit usage.

Lifecycle hooks

Patch Manager allows you to add lifecycle hooks that enable a multi-step, custom patching process. These hooks let you perform a custom action on instances when the corresponding lifecycle event occurs.

When you patch the operating system of an SAP application, lifecycle hooks can help you perform SAP-specific operations and automate the operating system patching lifecycle. You can automate the following tasks using lifecycle hooks:

- Stop the SAP application and necessary database services
- Initiate database or storage snapshot backup
- Patch the operating system and reboot if necessary
- Start the SAP application and the database after successful operating system patch update

For more information about lifecycle hooks, see the following documentation:

- About the Amazon-RunPatchBaselineWithHooks SSM document in the *Amazon Systems Manager User Guide*
- Orchestrating multi-step in the *Amazon Cloud Operations & Migrations Blog*

Automated operating system patching prerequisites

In addition to the prerequisites described in the Automation prerequisites section of this guide, verify the following prerequisites that are specific to automated operating system patching:

- Verify the Patch Manager prerequisites.

  Because the solution described here uses Amazon Systems Manager Patch Manager, you must verify that you have satisfied all of the Patch Manager prerequisites. For more information, see Patch Manager prerequisites in the *Amazon Systems Manager User Guide*.
- Ensure you have a backup of your SAP system.

  Before you make changes to the SAP system, verify that a backup is available to support rollback in case you encounter problems. You should have the following backups:

- Operating system backup – You should have an Amazon Machine Image (AMI) backup of the Amazon EC2 instance that consists of the base operating file system (`root` for Linux and `C:\` for Microsoft Windows) and the SAP application and database file systems.
- Database backup – If patching will occur on the database server, ensure you have the most recent database backup.

For data recovery recommendations, see Plan for data recovery in the *SAP Lens Amazon Well-Architected Framework*.

Supported operating systems

The following operating systems are supported by SAP and Patch Manager. Check the Patch Manager prerequisites for currently supported versions of the operating systems. For more information, see Patch Manager prerequisites in the *Amazon Systems Manager User Guide*.

- Oracle Linux

> **Note**
>
> Oracle Linux is required if you are running an Oracle database.

- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES)
- Microsoft Windows Server

> **Note**
>
> - SUSE Linux and Red Hat Linux have SAP versions of the Linux operating system. SAP recommends that you use RHEL for SAP Solutions/Applications or SLES for SAP Applications to run the SAP application.
> - Oracle Linux operating system is required for Oracle Database Server and SAP NetWeaver Application Servers with Oracle client installed. For more information, see SAP Note 2358420 - Oracle Database Support for Amazon Web Services EC2 (SAP portal access required).

For each of these operating systems, you can bring your own subscription to Amazon or use the Amazon Machine Images (AMIs) from the Amazon Marketplace.

SAP Notes

Review the following SAP Notes. You require SAP portal access to check these references from SAP.

- SAP Note: 1656099 - SAP Applications on Amazon: Supported DB/OS and Amazon EC2 products
- SAP Note: 2871484 - SAP supported variants of Red Hat Enterprise Linux
- SAP Note: 2358420 - Oracle Database Support for Amazon Web Services EC2
- SAP Note: 62988 - Service Packs for MS SQL Server
- SAP Note: 2235581 - SAP HANA: Supported Operating systems

Configuring automated operating system patching

The sections below contain detailed instructions on how to configure automated operating system patching.

Configure patch baselines

Patch Manager uses patch baselines, which include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches. For information about patch baselines, see About patch baselines in the *Amazon Systems Manager User Guide*. You can use predefined patch baselines or create custom patch baselines. The sections below contain instructions on how to use both.

For information about patch baselines that is specific to Linux, see How patch baseline rules work on Linux-based systems in the *Amazon Systems Manager User Guide*.

For information about the differences between Linux and Windows patching, see Key differences between Linux and Windows patching in the *Amazon Systems Manager User Guide*. If your system landscape has a combination of Windows Server and Linux operating systems, such as Windows Server for SAP application servers and Linux for database servers, you can define a baseline for each operating system type.

Predefined patch baselines

Patch manager provides predefined patch baselines for each of the supported operating systems. If your patching requirement patches the predefined baseline configuration, you might be able to use a predefined patch baseline for operating system patching. Alternatively, you can create your own custom patch baselines. This gives you greater control over which patches are approved or rejected for your environment.

For information about predefined patch baselines, see Viewing Amazon predefined patch baselines (console) in the *Amazon Systems Manager User Guide*.

Note

SUSE Linux Enterprise Server for SAP Applications and Red Hat Enterprise Linux for SAP Applications require custom patch baselines.

The following table is a subset of the predefined patch baselines in the Patch Manager documentation. To view the full list of predefined patch baselines, see About predefined baselines in the *Amazon Systems Manager User Guide*. The predefined patch baselines listed here are applicable to SAP.

Name	Supported operating system	Details
`Amazon-OracleLinuxDefaultPatchBaseline`	Oracle Linux	Approves all operating system patches that are classified as "Security" and that have a severity level of "Important" or "Moderate". Also approves all patches that are classified as "Bugfix" 7 days after release. Patches are auto-approved 7 days after they are released or updated.[1]
`Amazon-RedHatDefaultPatchBaseline`	Red Hat Enterprise Linux (RHEL)	Approves all operating system patches that are classified as "Security" and that have a severity level of "Critical" or "Important". Also approves all patches that are classified as "Bugfix". Patches are auto-

Name	Supported operating system	Details
		approved 7 days after they are released or updated.[1]
`Amazon-SuseDefaultPatchBaseline`	SUSE Linux Enterprise Server (SLES)	Approves all operating system patches that are classified as "Security" and with a severity of "Critical" or "Important". Patches are auto-approved 7 days after they are released or updated.[1]
`Amazon-DefaultPatchBaseline`	Windows Server	Approves all Windows Server operating system patches that are classified as "CriticalUpdates" or "Security Updates" and that have an MSRC severity of "Critical" or "Important". Patches are auto-approved 7 days after they are released or updated.[1]

[1] For Amazon Linux and Amazon Linux 2, the 7-day wait before patches are auto-approved is calculated from an `Updated Date` value in `updateinfo.xml`, not a `Release Date` value. Various factors can affect the `Updated Date` value. Other operating systems handle release and update dates differently. For information to help you avoid unexpected results with auto-approval delays, see How package release dates and update dates are calculated in the *Amazon Systems Manager User Guide*.

Custom patch baselines

Unlike predefined patch baselines, custom patch baselines do not have default patch approvals and compliance levels. This gives you greater control over which patches are approved or rejected for your environment and allows you to define your custom repositories. For example, you can assign specific approval rules and compliance values. It is also possible to create a custom patch baseline by copying a predefined patch baseline and specifying the compliance values that you want to assign to patches.

You can use Patch Manager to create a custom patch baseline for Linux-based managed nodes, such as Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), Oracle Linux. You can also specify patch source repositories for each of these operating systems. See the sections below for additional information about patch sources for each.

For instructions on how to create a custom patch baseline for Linux and Windows, see the following documentation:

- Creating a custom patch baseline (Linux) in the *Amazon Systems Manager User Guide*
- Creating a custom patch baseline (Windows) in the *Amazon Systems Manager User Guide*

Patch sources

When you use the default repositories that are configured on a managed node for patching operations, Patch Manager scans for security-related patches or installs them. This is the default behavior for Patch Manager. On Linux systems, you can also use Patch Manager to install patches that aren't related to security or that are in a different source repository than the default repository that is configured on the managed node.

In the procedure to create a custom patch baseline, there is an option to specify alternative patch source repositories if you are not using the default repository configuration. In each custom patch baseline, you can specify patch source configurations for up to 20 versions of a supported Linux operating system. For more information about alternative patch sources, see How to specify an alternative patch source repository (Linux) in the *Amazon Systems Manager User Guide*.

> **Note**
>
> If you specify alternative repositories, you must also specify the default repositories as part of the alternative patch source configuration if you want those updates to be applied.

The sections below contain information about how to obtain patch source details for SLES for SAP Applications, RHEL for SAP Applications, and Oracle Linux. You can use this information to specify a patch source when you create a custom patch baseline.

Patch sources for SLES for SAP Applications

You can use one of the following patch repositories for SUSE Linux Enterprise Server (SLES) for SAP Applications:

- SUSE public cloud update infrastructure
- Private repository

  For information about how to use a private patch repository, see Private and local repositories in this guide.

The public cloud update infrastructure is a global network of update servers maintained by SUSE on Amazon Cloud that provides low-latency access to patches from on-demand instances. Customers that use SUSE on-demand instances in Amazon automatically connect to the public cloud update infrastructure on boot. You can view the SUSE patch source server details in the `/etc/hosts` directory.

You can connect to the public cloud update infrastructure through an internet gateway in a public subnet, NAT gateway in a private subnet, or through a local data center. To see the repository list, run the command `zypper ls`.

By default, all repositories are considered for patching. If you want to only patch certain repositories or if you are using multiple patch sources for repositories, you must explicitly add patch sources based on repository configuration.

Complete the following steps to identify the patch source for the repository that you would like to use for patching:

1. Navigate to the following directory to view the repository files:

   ```
   /etc/zypp/repos.d
   ```

2. Save the name and configuration for each repository file. For example, you might save the following:

   - Name – `SUSE_Linux_Enterprise_Server_for_SAP_Applications_x86_64:SLE-Product-SLES_SAPXX-SPX-Updates`
   - Configuration –

     ```
     name=SLE-Product-SLES_SAPXX-SPX-Updates
     enabled=1
     autorefresh=1
     baseurl=plugin:/susecloud?
     credentials=SUSE_Linux_Enterprise_Server_for_SAP_Applications_x86_64&path=/repo/
     SUSE/Updates/SLE-Product-SLES_SAP/XX-SPX/x86_64/update/
     ```

```
service=SUSE_Linux_Enterprise_Server_for_SAP_Applications_x86_64
```

3. Enter this information when you create the custom patch baseline in the **Patch sources** section of **Patch Manager**. For the full list of steps, see Creating a custom patch baseline (Linux) in the *Amazon Systems Manager User Guide*.
4. If you add a patch source for any repository, you must add patch sources for all the repositories that you would like to patch, including the default repositories.

> **Important**
>
> Before you deploy the patch, you must accept the license agreement in the `zypper.conf` configuration file. You can find the file in the following directory:
>
> ```
> /etc/zypp/zypper.conf
> ```
>
> To accept the license agreement, uncomment the license agreement property and save it as:
>
> ```
> autoAgreeWithLicenses = yes
> ```

Patch sources for RHEL for SAP Applications

You can use one of the following patch repositories for Red Hat Enterprise Linux (RHEL) for SAP Applications:

- Red Hat update infrastructure
- Local repository

  For information about how to use a private patch repository, see Private and local repositories in this guide.

Red Hat update infrastructure is a global network of update servers maintained by Red Hat on Amazon Cloud that provides low-latency access to patches from on-demand instances. Customers that use Red Hat on-demand instances in Amazon automatically connect to the Red Hat update infrastructure on boot.

The RHEL repositories are stored in the following location:

```
/etc/yum.repos.d/
```

Complete the following steps to identify the patch source for the repository that you would like to use for patching:

1. Run the following command to view the default, enabled repositories:

   ```
   cat /etc/yum.repos.d/* | grep -B 4 -A 6 "enabled=1"
   ```

   This command returns four lines before and six lines after each repository that is enabled. For example, the command might return something like this:

   ```
   [rhui-client-config-server-8-sap-bundle]
   name=Red Hat Update Infrastructure 3 Client Configuration for SAP Bundle
   mirrorlist=https://rhui3.REGION.ce.redhat.com/pulp/mirror/protected/rhui-client-
   config/rhel/server/8/$basearch/sap-bundle
   enabled=1
   gpgcheck=1
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
   sslverify=1
   sslcacert=/etc/pki/rhui/cdn.redhat.com-chain.crt
   sslclientcertexample=/etc/pki/rhui/product/rhui-client-config-server-8-sap-bundle.crt
   sslclientkeyexample=/etc/pki/rhui/rhui-client-config-server-8-sap-bundle.key
   ```

2. Save the name and configuration for each repository file. In this example, you would save the following:

   - Name – `rhui-client-config-server-8-sap-bundle`
   - Configuration

     ```
     name=Red Hat Update Infrastructure 3 Client Configuration for SAP Bundle
     mirrorlist=https://rhui3.REGION.ce.redhat.com/pulp/mirror/protected/rhui-client-
     config/rhel/server/8/$basearch/sap-bundle
     enabled=1
     gpgcheck=1
     gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
     sslverify=1
     sslcacertexample=/etc/pki/rhui/cdn.redhat.com-chain.crt
     ```

```
sslclientcertexample=/etc/pki/rhui/product/rhui-client-config-server-8-sap-bundle.crt
```

3. For each entry that was returned by the command in the previous step, create a new patch source when you create a custom patch baseline in the **Patch sources** section of **Patch Manager**. For the full list of steps, see Creating a custom patch baseline (Linux) in the *Amazon Systems Manager User Guide*.
4. If you add a patch source for any repository, you must add patch sources for all the repositories that you would like to patch, including the default repositories.

Patch sources for Oracle Linux

On Oracle Linux, the patch baseline uses preconfigured repositories on the managed node. All Oracle Linux Amazon Machine Images (AMIs) can access the public YUM repository. Only licensed Oracle Linux systems can access the Oracle ULN repository.

The Oracle Linux repositories are stored in the following location:

```
/etc/yum.repos.d/
```

Complete the following steps to identify the patch source for the repository that you would like to use for patching:

1. Run the following command to view the default, enabled repositories:

   ```
   cat /etc/yum.repos.d/* | grep -B 4 -A 6 "enabled=1"
   ```

   This command returns four lines before and six lines after each repository that is enabled. For example, the command might return something like this:

   ```
   [ol8-appsteream]
   name=Oracle Linux 8 Application Stream ($basearch)
   baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL8/appstream/$basearch/
   gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
   gpgcheck=1
   ```

2. Save the name and configuration for each repository file. In this example, you would save the following:
   - Name – `ol8-appsteream`

- Configuration

```
name=Oracle Linux 8 Application Stream ($basearch)
baseurl=https://yum$ociregion.$ocidomain/repo/OracleLinux/OL8/appstream/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
```

3. For each entry that was returned by the command in the previous step, create a new patch source when you create a custom patch baseline in the **Patch sources** section of **Patch Manager**. For the full list of steps, see Creating a custom patch baseline (Linux) in the *Amazon Systems Manager User Guide*.
4. If you add a patch source for any repository, you must add patch sources for all the repositories that you would like to patch, including the default repositories.

Oracle Linux 7 managed nodes use YUM as the package manager, while Oracle Linux 8 managed nodes use DNF as the package manager. Both package managers have an update notice, which is a file named `updateinfo.xml`. The update notice is a collection of packages that fix specific issues. Individual packages aren't assigned classifications or severity levels, so Patch Manager assigns the attributes of an update notice to the related packages and installs the packages based on the classification filters specified in the patch baseline.

Only patches specified in `updateinfo.xml` are applied if you are using the default patch baseline provided by Amazon or if you do not select the option to include non-security update patches when you create a custom baseline. If you create a custom baseline and you do select the option to include non-security update patches, the patches in `updateinfo.xml` and the patches that are not in `updateinfo.xml` are applied. For more information, see How patch baseline rules work on Oracle Linux in the *Amazon Systems Manager User Guide*.

Oracle Linux instances require internet access to the public YUM repository or Oracle ULN in order to download packages. If the Amazon EC2 instance is on a private subnet of an Amazon VPC, you can use a proxy server or a local YUM repository to download packages. For more information, see Configuring a System to Use a Proxy With a Yum Server in the Oracle documentation. Alternatively, Oracle Linux systems can work with Oracle Linux Manager for YUM package management. An Oracle Linux Manager system can be in a public subnet while Oracle Linux systems can be in a private subnet. For more information, see Oracle Linux Manager in the Oracle documentation.

Windows Server considerations

For additional information about security patches for Windows, see How security patches are selected and How patches are installed in the *Amazon Systems Manager User Guide*.

Create patch groups

You can use patch groups to organize instances for patching. This can help you ensure that you only deploy patches to the correct set of instances and that the patches have been adequately tested before they are deployed. After you create the patch group, you can tag your Amazon EC2 instances to add them to the patch group and then add the patch group to a patch baseline.

You might want to organize patch groups by:

- Operating system – such as Linux and Windows
- Environment – such as development, test, and production
- Server function – such as SAP database servers and SAP application servers

> **Note**
>
> An Amazon EC2 instance can only be in one patch group at a time.

For more information about patch groups, see About patch groups in the *Amazon Systems Manager User Guide*.

Tag Amazon EC2 instances to add to the patch group

After you create the patch group, use tags to add Amazon EC2 instances to the patch group. For detailed steps on how to do this, see Working with patch groups in the *Amazon Systems Manager User Guide*.

Add the patch group to a patch baseline

To ensure that the correct patches are installed during the patching execution, you must register the patch group with a patch baseline. When the system applies a patch baseline to an instance, the service checks to see if a patch group is defined for the instance. For detailed steps on how to add a patch group to a patch baseline, see Add a patch group to a patch baseline in the *Amazon Systems Manager User Guide*.

Note

Patch groups are not used in patching operations that are based on patch policies. For more information, see the following:

- Using Quick Setup patch policies
- Configure the home Amazon Region
- Creating a patch policy

Applying patches

After you have created the patch baseline and tagged your Amazon EC2 instances to the patch group, you can apply patches. You can schedule patches or run them on-demand.

Scheduled patching

SAP maintenance activities are usually scheduled in advance. The non-critical SAP systems can be patched in an ad-hoc manner, such as a sandbox system. The patching process should be documented in runbooks. After the system is successfully patched, the patching activities for the downstream SAP systems can be scheduled, either using maintenance windows or directly from Patch Manager.

For more information about patching schedules, see the following documentation:

- About patching schedules using maintenance windows in the *Amazon Systems Manager User Guide*
- Walkthrough: Creating a maintenance window for patching (console) in the *Amazon Systems Manager User Guide*

On-demand patching

The **Patch now** option in Patch Manager allows you to run on-demand patching operations directly from the Systems Manager console. With this option, you do not need to create a schedule to update the compliance status of your managed nodes or to install patches on non-compliant nodes.

Scanning the Amazon EC2 instances allows you to identify systems that are potentially non-compliant, vulnerable, or un-patched. We recommend that you schedule system scans frequently, such as weekly.

For detailed instructions on how to run on-demand patching, see Patching managed nodes on demand in the *Amazon Systems Manager User Guide*.

Patch summary

After the patch baseline has run, you can view the patch status in Patch Manager. For details about the patch summary and how to access it in Patch Manager, see Viewing patch Dashboard summaries (console) in the *Amazon Systems Manager User Guide*.

Patch compliance reports

Patch compliance reports allow you to view the status of managed nodes. For more information about compliance reports, including detailed instructions on how to view them, see the following documentation:

- Working with patch compliance reports in the *Amazon Systems Manager User Guide*
- Viewing patch compliance results (console) in the *Amazon Systems Manager User Guide*

Monitoring

You can view Patch Manager output after each patch is run. By default, Patch Manager stores the first 48,000 characters of the command output. In some cases, you might want to view the complete log, such as for troubleshooting. In this case, the log output can be stored in Amazon S3. For details about how to store log output in Amazon S3, see Configuring Amazon CloudWatch Logs Logs for Run Command in the *Amazon Systems Manager User Guide*.

Another option is to output the logs to Amazon CloudWatch Logs for unified logging. For more information, see Sending SSM Agent logs to CloudWatch Logs in the *Amazon Systems Manager User Guide*.

For information about how to set up detailed monitoring and notifications, see Monitoring Amazon Systems Manager in the *Amazon Systems Manager User Guide*.

Private and local repositories

If you would like to manage your operating system repository locally, either within your VPC on Amazon or an on-premises data center, without using an outbound internet connection for your instance, you can use a private or local repository.

Some reasons to use a private repository are:

- They provide access to repositories for Amazon EC2 instances that do not have access to the internet for security reasons.
- You have additional add-on products from vendors that are not provided through the public cloud update infrastructure.
- You want to deploy an organized and consistent set of patches across mission-critical workloads. Using an online repository might introduce new updates which could lead to inconsistency across the landscape.
- You want to improve software download times and reduce bandwidth overhead while patching a large fleet of infrastructure.

If you are on SUSE Linux Enterprise Server (SLES) and you want to use private repositories, make sure that the operating system repositories are pointing to the local repository instead of the respective vendor repositories before you use Patch Manager. If you are on Red Hat Enterprise Linux (RHEL) or Oracle Linux, you must use a custom baseline to point to local repositories.

Alternative tools for patching

In addition to Amazon Systems Manager, there are other automated patching tools that you might use, which are listed below. This list is not exhaustive, but is meant to give you a starting point for doing your own research if you decide to consider alternate tools.

SUSE Manager

SUSE Manager is an infrastructure management tool for Linux systems. With SUSE manager, you can automate software management of SLES< RHEL and OEL operating systems. For more information, and a list of Amazon EC2 instances, see SUSE Manager 4.0 Documentation.

Repository Mirroring Tool (For SUSE Linux)

Repository Monitoring Tool (RMT) is a service from SUSE Linux that helps manage private repositories by downloading updates and distributing them across the landscape. This reduces network bandwidth usage and allows you to set more restrictive firewall policies. For more information, see the SUSE Linux Repository Mirroring Tool Guide.

Red Hat Satellite (For Red Hat Linux)

Red Hat Satellite is a system management solution that enables you to deploy, configure, and maintain your systems across physical, virtual, and cloud environments. Satellite Server synchronizes the content from the Red Hat Customer Portal and other sources, and provides functionality such as fine-grained lifecycle management, user and group role-based access control, integrated subscription management, as well as advanced GUI, CLI, or API access. For more information, see the Red Hat Customer Portal.

KernelCare (For Red Hat Linux)

KernelCare is a live patching system that patches Linux kernel vulnerabilities automatically, with no reboots. It works with all major Linux distributions, such as RHEL, CentOS, Amazon Linux, and Ubuntu. It also interoperates with common vulnerability scanners such as Nessus, Tenable, Rapid7, and Qualys. For more information, see KernelCare on Amazon Marketplace.

Zypper Package Manager (For SUSE Linux)

Zypper is a command-line package manager for installing updating, and removing packages. It can also be used to manage repositories. Zypper offers advantages over graphical package managers such as scripting actions. For more information, see the Zypper package manager documentation.

Considerations for multiple accounts

When you run SAP workloads in Amazon, you must consider an Amazon account strategy that meets the security controls of your organization. For example, you might separate SAP from non-SAP workloads and separate production from non-production environments. Amazon Systems Manager does not support multi-account patching.

In every Amazon account with SAP workloads, patch baselines should be created and patch execution should be performed to ensure that patching is applied to all SAP systems. In a multi-account environment, this should also follow the SAP best practice of patching in the development account, then test, and finally in the production Amazon account.

Automation troubleshooting

If you encounter errors related to SAP automation, refer to the Troubleshooting Systems Manager Automation documentation in the *Amazon Systems Manager User Guide*. There, you will find an action-specific failures reference as well as information about common errors such as access denied errors and errors related to timed out or failed statuses after the execution started.

Logging installation steps

You can log individual automated installation steps with the code below. In this example, logs are added to `$SSM_LOG_FILE` for each `run` command.

```
action: "aws:runShellScript"
name: "Validate_Installation"
inputs:
runCommand:
- #!/bin/bash
- sid=echo {{ Sid }} | tr '[:upper:]' '[:lower:]'}``
- SID=echo {{ Sid }} | tr '[:lower:]' '[:upper:]'}``
- HOSTNAME=hostname``
- SIDADM=${sid}adm
- su - $SIDADM -c "stopsap $HOSTNAME" | tee -a $SSM_LOG_FILE
- su - $SIDADM -c "startsap $HOSTNAME" | tee -a $SSM_LOG_FILE
- sleep 15
- _SAP_UP=$(netstat -an | grep 3200 | grep tcp | grep LISTEN | wc -l )
- echo "This is the value of SAP_UP - $_SAP_UP" | tee -a $SSM_LOG_FILE
- if [ "$_SAP_UP" -eq 1 ]
- then
-   echo "$(date) __ done installing ASCS." | tee -a $SSM_LOG_FILE
-   exit 0
- else
-   echo "$(date) __ ASCS could not be installed successfully. Fix the issue and rerun
 the automation" | tee -a $SSM_LOG_FILE
-   exit 1
- fi
```