Amazon CloudWatch 事件

用户指南

Amazon CloudWatch 事件: 用户指南

Table of Contents

什么是 Amazon CloudWatch Events？

Amazon CloudWatch Events 提供近乎实时的系统事件流，该流描述 Amazon Web Services (AWS) 资源的变化。通过使用可快速设置的简单规则，您可以匹配事件并将事件路由到一个或多个目标函数或流。CloudWatch Events 会在发生操作更改时感知到这些更改。CloudWatch Events 将响应这些操作更改并在必要时采取纠正措施，方式是发送消息以响应环境、激活函数、进行更改并捕获状态信息。

您还可以使用 CloudWatch Events 来计划使用 Cron 或 rate 表达式在某些时间自行触发的自动化操作。有关更多信息，请参阅 规则的计划表达式 (p. 27)。

您可以将以下 AWS 服务配置为 CloudWatch Events 的目标：

- Amazon EC2 实例
- AWS Lambda 函数
- Amazon Kinesis Data Streams 中的流
- Amazon Kinesis Data Firehose 中的传输流
- Amazon ECS 任务
- Systems Manager Run Command
- Systems Manager Automation
- AWS Batch 作业
- Step Functions 状态机
- AWS CodePipeline 中的管道
- AWS CodeBuild 项目
- Amazon Inspector 评估模板
- Amazon SNS 主题
- Amazon SQS 队列
- 内置目标 - `EC2 CreateSnapshot API call`、`EC2 RebootInstances API call`、`EC2 StopInstances API call` 和 `EC2 TerminateInstances API call`。
- 另一个 AWS 账户的默认事件总线

概念

在您开始使用之前 CloudWatch Events，应了解以下概念：

- 事件—事件表示 AWS 环境中的更改。AWS 资源可以在其状态更改时生成事件。例如，Amazon EC2 会在 EC2 实例的状态从待处理更改为正在运行时生成事件，Amazon EC2 Auto Scaling 会在启动或终止实例时生成事件。AWS CloudTrail 在您执行 API 调用时发布事件。您可以生成自定义应用程序级事件并将它们发布到 CloudWatch Events。您还可以设置定期生成的计划事件。有关生成事件的服务的列表，以及来自每项服务的示例事件，请参阅 每个支持服务的 CloudWatch Events 事件示例 (p. 34)。
- 目标—目标负责处理事件。目标可以包括 Amazon EC2 实例、AWS Lambda 函数、Kinesis 流、Amazon ECS 任务、Step Functions 状态机、Amazon SNS 主题、Amazon SQS 队列和内置目标。目标接收 JSON 格式的事件。

- 规则—规则匹配传入事件并将其路由到目标进行处理。单个规则可路由到多个目标，所有这些目标将并行处理。规则不按特定顺序处理。这可让组织的不同部门能够查找和处理他们感兴趣的事件。规则可以定制发送到目标的 JSON，方法是仅传递特定部分或使用常量来覆盖 JSON。

相关 AWS 服务

以下服务可与 CloudWatch Events 一起使用：

- AWS CloudTrail 使您能够监控您的账户对 CloudWatch Events API 的调用 (包括由 AWS 管理控制台、AWS CLI 和其他服务进行的调用)。当 CloudTrail 日志记录打开时，CloudWatch Events 会将日志文件写入 S3 存储桶。每个日志文件包含一个或多个记录，具体取决于为满足某个请求要执行的操作数量。有关更多信息，请参阅 在 AWS CloudTrail 中记录 Amazon CloudWatch Events API 调用 (p. 114)。
- AWS CloudFormation 可让您对 AWS 资源进行建模和设置。您可创建一个模板来描述所需的 AWS 资源，而 AWS CloudFormation 则可为您预配和配置这些资源。您可以在 AWS CloudFormation 模板中使用 CloudWatch Events 规则。有关更多信息，请参阅 AWS CloudFormation 用户指南中的 AWS::Events::Rule。
- AWS Config 使您能够记录您的 AWS 资源发生的配置更改。这些信息包括资源之间的关联方式以及资源以前的配置方式，让您了解资源的配置和关系如何随着时间的推移而更改。您还可以创建 AWS Config 规则，以检查您的资源是否符合组织的策略。有关更多信息，请参阅 AWS Config 开发人员指南。
- AWS Identity and Access Management (IAM) 可以帮助您安全地控制用户对 AWS 资源的访问权限。通过 IAM 可以控制哪些人可以使用您的 AWS 资源 (身份验证) 以及他们可以使用的资源和采用的方式 (授权)。有关更多信息，请参阅 Amazon CloudWatch Events 的身份验证和访问控制 (p. 89)。
- Amazon Kinesis Data Streams 可实现快速而近乎持续的数据接收和汇总。使用的数据类型包括 IT 基础设施日志数据、应用程序日志、社交媒体、市场数据源和 Web 点击流数据。由于数据引入和处理的响应时间是实时的，因此处理通常是轻量级的。有关更多信息，请参阅 Amazon Kinesis Data Streams 开发人员指南。
- AWS Lambda 使您能够构建快速响应新信息的应用程序。将您的应用程式代码上传为 Lambda 函数，Lambda 将在高可用性计算基础设施上运行您的代码。Lambda 执行计算资源的所有管理任务，包括服务器和操作系统维护、容量配置、自动扩展、代码和安全补丁部署以及代码监控和日志记录。有关更多信息，请参阅 AWS Lambda Developer Guide。

CloudWatch Events 限制

CloudWatch Events 有以下限制：

资源	默认限制
API 请求	对于除 PutEvents 之外的所有 CloudWatch Events API 操作，每秒最多 50 个请求。默认情况下，PutEvents 限制为每秒 400 个请求。
默认事件总线	从 AWS 服务或其他 AWS 账户接收事件的速率没有限制。如果您使用 `PutEvents` API 向事件总线发送自定义事件，则存在 `PutEvents` API 限制。发送到您的账户中规则的目标上的事件将计入您的调用限制。 默认事件总线的策略大小限制为 10240 个字符。每当您授予对另一个账户的访问权限时，此策略大小都会增加。 您可以通过使用 `DescribeEventBus` API 查看您的当前策略及其大小。您可以请求提高限制。有关说明，请参阅 AWS 服务限制。
事件模式	最多 2048 个字符。

资源	默认限制
调用	一个调用是一个事件与规则匹配并发送到规则的目标上。限制为每秒 750 个调用 (在达到 750 个调用后，调用会受到限制；即，它们仍将发生，但会延迟)。如果目标的调用由于目标服务、账户限制等问题而失败，则对于特定的调用，会在不超过 24 小时的时间内尝试新的调用。 如果您从其他账户收到了事件，这些事件中的每一个与您账户中的一条规则匹配并发送到规则的目标，则这些事件将计入到您的账户的每秒 750 个调用的限制中。 您可以请求提高限制。有关说明，请参阅 AWS 服务限制。
ListRuleNamesByTarget	对于 请求，每页最多 100 条结果。
ListRules	对于请求，每页最多 100 条结果。
ListTargetsByRule	对于 请求，每页最多 100 条结果。
PutEvents	每个请求 10 个条目以及每秒 360 请求。每个请求的大小最多为 256 KB 字节。 您可以请求提高限制。有关说明，请参阅 AWS 服务限制。
PutTargets	每个请求 10 个条目。
RemoveTargets	每个请求 10 个条目。
规则	每账户每区域 100 个. 您可以请求提高限制。有关说明，请参阅 AWS 服务限制。 在请求提高限制之前，检查您的规则。您可以具有多个规则，每个规则均匹配极具针对性的事件。可考虑通过减少在CloudWatch Events 中的事件模式 (p. 30)中使用的标识符来扩大此范围。此外，每次规则匹配事件时都可以调用多个目标。请考虑向规则添加更多目标。
Systems Manager Run Command 目标	1 个目标键和 1 个目标值 Systems Manager Run Command 目前不支持多个目标值。
目标	每个规则 5 个。

设置 Amazon CloudWatch Events

您需要有 AWS 账户才能使用 Amazon CloudWatch Events。利用您的 AWS 账户，可以使用服务 (例如 Amazon EC2) 生成可在 CloudWatch 控制台 (一种基于 Web 的界面) 中查看的事件。此外，您还可以安装和配置 AWS Command Line Interface (AWS CLI) 以使用命令行界面。

注册 Amazon Web Services (AWS)

创建 AWS 账户时，我们会自动为所有 AWS 服务注册您的账户。您只需为使用的服务付费。

如果您已有一个 AWS 账户，请跳到下一个步骤。如果您还没有 AWS 账户，请使用以下步骤创建。

如需注册 AWS 账户

1. 打开 http://amazonaws.cn/，然后选择 Create an AWS Account。

   Note

   如果您之前已登录 AWS 管理控制台，则可能无法在浏览器中执行此操作。在此情况下，请选择 Sign in to a different account，然后选择 Create a new AWS account。
2. 按照屏幕上的说明进行操作。

   作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

登录 Amazon CloudWatch 控制台

登录 Amazon CloudWatch 控制台的步骤

1. 登录 AWS 管理控制台并通过以下网址打开 CloudWatch 控制台 https://console.amazonaws.cn/cloudwatch/。
2. 如果需要，可以更改区域。从导航栏中，选择 AWS 资源所在的区域。
3. 在导航窗格中，选择 Events。

账户凭证

虽然您可以使用根用户凭证访问 CloudWatch Events，但建议您使用 AWS Identity and Access Management (IAM) 账户。如果您使用 IAM 账户访问 CloudWatch，则必须拥有以下权限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
```

```
  ]
}
```

有关更多信息，请参阅 身份验证 (p. 89)。

设置命令行界面

可以使用 AWS CLI 执行 CloudWatch Events 操作。

有关如何安装和配置 AWS CLI 的信息，请参阅 AWS Command Line Interface 用户指南 中的使用 AWS Command Line Interface 进行设置。

区域终端节点

必须启用区域终端节点 (默认) 才能使用 CloudWatch Events。有关详细信息，请参阅 IAM 用户指南 中的在 AWS 区域中激活和停用 AWS STS。

Amazon CloudWatch Events 入门

使用本部分中的过程来创建和删除 CloudWatch Events 规则。这些是可用于任何事件源或目标的一般过程。有关针对特定场景和特定目标编写的教程，请参阅教程。

内容

限制

- 某些目标类型可能并非在所有区域都可用。有关更多信息，请参阅 Amazon Web Services 一般参考 中的区域和终端节点。
- 只能在 AWS 管理控制台中创建带内置目标的规则。
- Amazon SQS FIFO (先进先出) 队列不受支持。
- 如果您创建的规则使用加密的 Amazon SQS 队列作为目标，则您必须在您的 KMS 密钥策略中包含以下部分，事件才能成功传送到加密的队列。

```
{
                "Sid": "Allow CWE to use the key",
                "Effect": "Allow",
                "Principal": {
                                "Service": "events.amazonaws.com"
                },
                "Action": [
                                "kms:Decrypt",
                                "kms:GenerateDataKey"
                ],
                "Resource": "*"
}
```

创建对事件触发的 CloudWatch Events 规则

可以使用以下步骤创建对 AWS 服务发出的事件进行触发的 CloudWatch Events 规则。

创建对事件触发的规则：

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
   a. 依次选择 Event Pattern 和 Build event pattern to match events by service。
   b. 对于 Service Name ，选择用于发出应触发此规则的事件的服务。
   c. 对于 Event Type ，选择用于触发此规则的特定事件。如果只有 AWS API Call via CloudTrail 一个选项，则选定服务不会发出事件且规则只能基于对此服务进行的 API 调用。有关创建此类规则的更多信息，请参阅 使用 AWS CloudTrail 创建对 AWS API 调用触发的 CloudWatch Events 规则 (p. 7)。

d. 根据发出事件的服务，您可能会看到 Any... 和 Specific... 选项。选择 Any... 可对任何类型的选定事件触发事件，而选择 Specific... 可选择一个或多个特定事件类型。

4. 对于 Targets，选择 Add Target，然后选择当检测到选定类型的事件时要执行的 AWS 服务。
5. 在此部分的其他字段中，根据需要输入此目标类型的特定信息。
6. 对于许多目标类型，CloudWatch Events 需要权限将事件发送到目标。在这些情况下，CloudWatch Events 可以创建运行事件所需的 IAM 角色：

   - 若要自动创建 IAM 角色，请选择 Create a new role for this specific resource。
   - 若要使用您之前创建的 IAM 角色，请选择 Use existing role。

7. 根据需要，可以重复步骤 4 至 6 为此规则添加另一目标。
8. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
9. 选择 Create rule。

使用 AWS CloudTrail 创建对 AWS API 调用触发的 CloudWatch Events 规则

要创建对由不发出事件的 AWS 服务所进行的操作触发的规则，您可使此规则基于该服务进行的 API 调用 (由 AWS CloudTrail 记录)。CloudTrail 一般会检测所有 AWS API 调用，以 Get、List 或 Describe 开头的调用除外。有关可用作规则触发器的完整 API 列表，请参阅 CloudTrail 事件历史记录所支持的服务。

通过 CloudTrail 创建对 API 调用触发的规则：

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：

   a. 依次选择 Event Pattern 和 Build event pattern to match events by service。
   b. 对于 Service Name，选择要将 API 操作用作触发器的服务。
   c. 对于 Event Type，选择 AWS API Call via CloudTrail。
   d. 要在调用此服务的任何 API 操作时触发您的规则，请选择 Any operation。要仅在调用特定 API 操作时触发您的规则，请选择 Specific operation(s)，在下一个框中键入操作名称，然后按 ENTER。要添加更多操作，请选择 +。

4. 对于 Targets，选择 Add Target，然后选择当检测到选定类型的事件时要执行的 AWS 服务。
5. 在此部分的其他字段中，根据需要输入此目标类型的特定信息。
6. 对于许多目标类型，CloudWatch Events 需要权限将事件发送到目标。在这些情况下，CloudWatch Events 可以创建运行事件所需的 IAM 角色：

   - 若要自动创建 IAM 角色，请选择 Create a new role for this specific resource。
   - 若要使用您之前创建的 IAM 角色，请选择 Use existing role。

7. 根据需要，可以重复步骤 4 至 6 为此规则添加另一目标。
8. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
9. 选择 Create rule。

创建按计划触发的 CloudWatch Events 规则

可以使用以下步骤创建定期触发的 CloudWatch Events 规则。

创建定期触发的规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，选择 Schedule。
4. 选择 Fixed rate of，然后指定运行任务的频率，或选择 Cron expression 并指定一个用于定义何时触发任务的 Cron 表达式。有关 Cron 表达式语法的更多信息，请参阅 规则的计划表达式 (p. 27)。
5. 对于 Targets，选择 Add Target，然后选择当检测到选定类型的事件时要执行的 AWS 服务。
6. 在此部分的其他字段中，根据需要输入此目标类型的特定信息。
7. 对于许多目标类型，CloudWatch Events 需要权限将事件发送到目标。在这些情况下，CloudWatch Events 可以创建运行事件所需的 IAM 角色：
   - 若要自动创建 IAM 角色，请选择 Create a new role for this specific resource。
   - 若要使用您之前创建的 IAM 角色，请选择 Use existing role。
8. 根据需要，可以重复步骤 5 至 7 为此规则添加另一目标。
9. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
10. 选择 Create rule。

删除或禁用 CloudWatch Events 规则

可以使用以下步骤删除或禁用 CloudWatch Events。

删除或禁用规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Rules。
3. 执行以下任一操作：
   a. 要删除规则，请选择规则旁边的按钮，然后依次选择 Actions、Delete 和 Delete。
   b. 要临时禁用规则，请选择规则旁边的按钮，然后依次选择 Actions、Disable 和 Disable。

CloudWatch 事件教程

以下教程为您介绍如何为特定任务和目标创建 CloudWatch Events 规则。

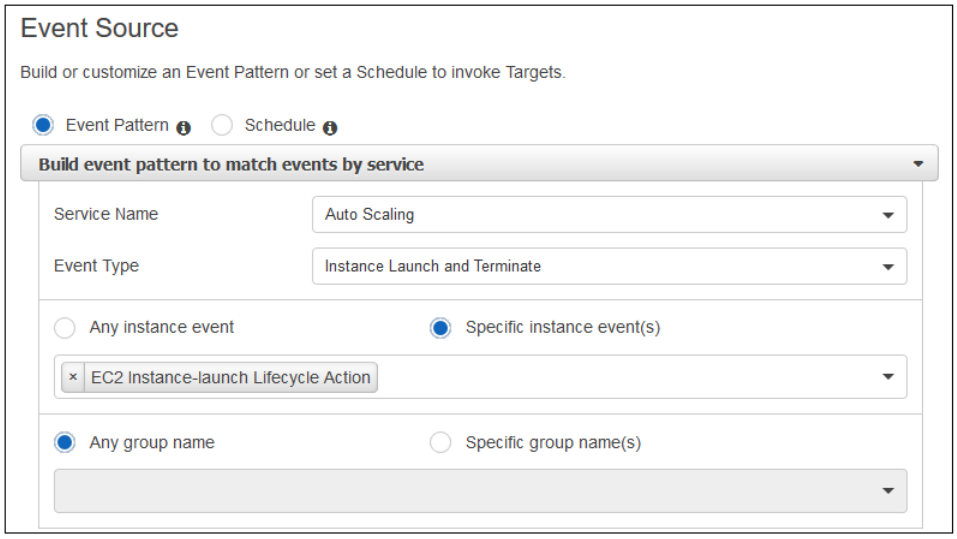
教程:

教程：使用 CloudWatch Events 将事件中继到 Amazon EC2 Run Command

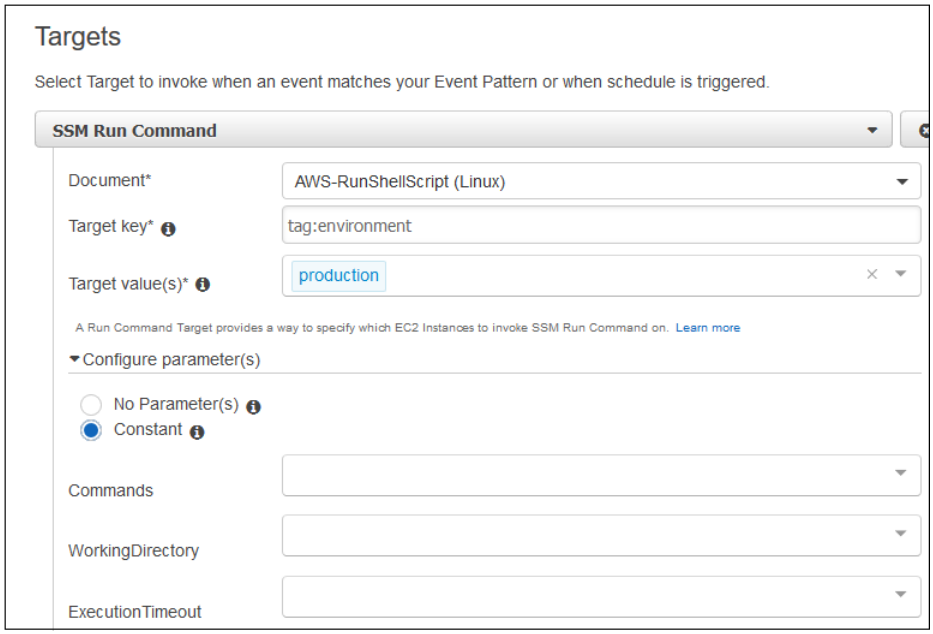
当某些事件发生时，您可以使用 Amazon CloudWatch Events 来调用 AWS Systems Manager Run Command 并对 Amazon EC2 实例操作执行。在本教程中，将设置 Run Command 以运行 shell 命令并配置在 Amazon EC2 Auto Scaling 组中启动的每个新实例。本教程假设您已向 Amazon EC2 Auto Scaling 组分配一个标签，其中使用 `environment` 作为键并使用 `production` 作为值。

创建 CloudWatch Events 规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
    a. 依次选择 Event Pattern 和 Build event pattern to match events by service。
    b. 对于 Service Name，选择 Auto Scaling。对于 Event Type，选择 Instance Launch and Terminate。
    c. 依次选择 Specific instance event(s) 和 EC2 Instance-launch Lifecycle Action。
    d. 默认情况下，该规则与区域中任何 Amazon EC2 Auto Scaling 组匹配。若要使该规则与特定组匹配，请选择 Specific group name(s)，然后选择一个或多个组。



4. 对于 Targets，依次选择 Add Target 和 SSM Run Command。
5. 对于 Document，选择 AWS-RunShellScript (Linux)。(请注意，有许多其他同时适用于 Linux 和 Windows 实例的 Document 选项。)对于 Target key，键入 **`tag:environment`**。对于 Target value(s)，键入 **`production`** 并选择 Add。
6. 在 Configure parameter(s) 下，选择 Constant。
7. 对于 Commands，键入 shell 命令并选择 Add。对所有要在实例启动时运行的命令重复此步骤。
8. 如果需要，请在 WorkingDirectory 和 ExecutionTimeout 中键入适当的信息。
9. CloudWatch Events 可以创建要运行的事件所需的 IAM 角色：
   - 若要自动创建 IAM 角色，请选择 Create a new role for this specific resource。
   - 若要使用您之前创建的 IAM 角色，请选择 Use existing role。



10. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
11. 选择 Create rule。

教程：使用 CloudWatch Events 记录 Amazon EC2 实例的状态

您可以创建 AWS Lambda 函数来记录 Amazon EC2 实例的状态更改。您可以选择创建一个规则，以便在状态发生任何转换时或者在状态转换为一个或多个相关状态时运行您前面创建的函数。在此教程中，您将记录任何新实例的启动。

步骤 1：创建 AWS Lambda 函数

创建 Lambda 函数以记录状态更改事件。在创建规则时，您可以指定此函数。

创建 Lambda 函数

1. 通过以下网址打开 AWS Lambda 控制台：https://console.amazonaws.cn/lambda/。
2. 如果您是首次使用 Lambda，则会看到一个欢迎页面；请选择 Get Started Now；否则，请选择 Create a Lambda function。
3. 在 Select blueprint 页面上，为筛选条件键入 `hello`，然后选择 hello-world 蓝图。
4. 在 Configure triggers 页面上，选择 Next。
5. 在 Configure function 页面上，执行以下操作：
   a. 键入 Lambda 函数的名称和说明。(例如，将函数命名为“LogEC2InstanceStateChange”)。
   b. 编辑 Lambda 函数的示例代码。例如：

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2InstanceStateChange');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. 对于 Role，选择 Choose an existing role，然后从 Existing role 中选择您的基本执行角色。否则，创建新的基本执行角色。
   d. 选择 Next。
6. 在 Review 页面上，选择 Create function。

步骤 2：创建规则

创建一个规则，以便每当您启动 Amazon EC2 实例时，就将运行您的 Lambda 函数。

创建 CloudWatch Events 规则

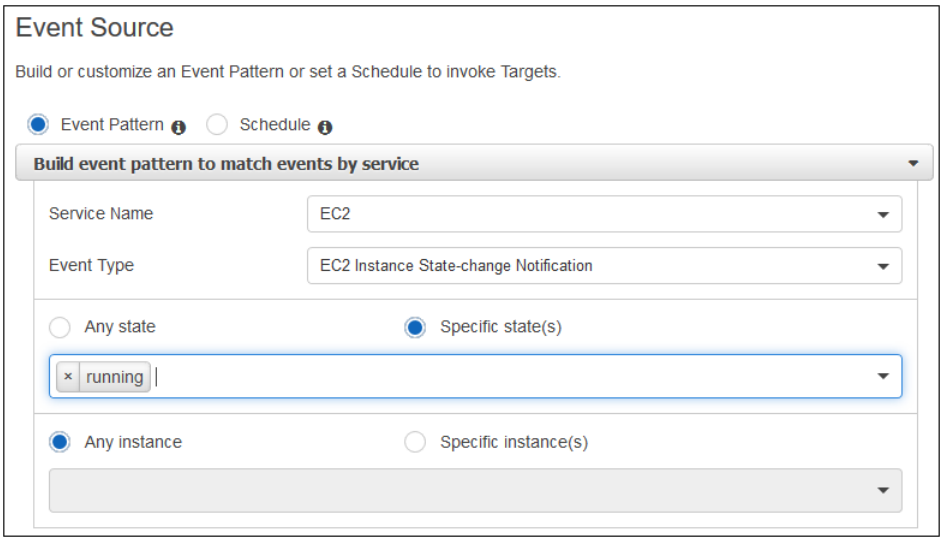
1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
   a. 选择 Event Pattern。

b. 选择 Build event pattern to match events by service。

c. 选择 EC2，然后选择 EC2 Instance State-change Notification。

d. 选择 Specific state(s)，然后选择 Running。

e. 默认情况下，该规则与区域中任何实例匹配。要使该规则匹配某个特定实例，请选择 Specific instance(s)，然后选择一个或多个实例。



4. 对于 Targets，选择 Add target，然后选择 Lambda function。
5. 对于 Function，选择您创建的 Lambda 函数。
6. 选择 Configure details。
7. 对于 Rule definition，键入规则的名称和描述。
8. 选择 Create rule。

步骤 3：测试规则

为了测试规则，启动一个 Amazon EC2 实例。等待几分钟，在该实例启动并初始化之后，您可以验证您的 Lambda 函数是否已调用。

通过启动实例测试规则

1. 打开 Amazon EC2 控制台 https://console.amazonaws.cn/ec2/。
2. 启动 实例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的启动实例。
3. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
4. 在导航窗格中，依次选择 Events 和 Rules，再选择所创建规则的名称，然后选择 Show metrics for the rule。
5. 若要查看 Lambda 函数的输出，请执行以下操作：

   a. 在导航窗格中，选择 Logs。

   b. 选择您的 Lambda 函数 (/aws/lambda/function-name) 的日志组的名称。

   c. 选择日志流的名称，以查看您启动的实例的函数提供的数据。

6. (可选) 完成后，您可以打开 Amazon EC2 控制台并停止或终止您启动的实例。想要了解更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的终止您的实例。

教程：使用 CloudWatch Events 记录 Auto Scaling 组的状态

可以运行 AWS Lambda 函数，只要 Auto Scaling 组启动或终止 Amazon EC2 实例，此函数就会记录一个事件，而不管该启动或终止事件是否成功。

有关使用 Amazon EC2 Auto Scaling 事件的其他 CloudWatch Events 方案，请参阅 Amazon EC2 Auto Scaling 用户指南中的在 Auto Scaling 组扩展时获取 CloudWatch Events。

步骤 1：创建 AWS Lambda 函数

创建一个 Lambda 函数，以记录您的 Auto Scaling 组的扩展和缩减事件。在创建规则时，您可以指定此函数。

创建 Lambda 函数

1. 通过以下网址打开 AWS Lambda 控制台：https://console.amazonaws.cn/lambda/。
2. 如果您是首次使用 Lambda，则会看到一个欢迎页面；请选择 Get Started Now；否则，请选择 Create a Lambda function。
3. 在 Select blueprint 页面上，为筛选条件键入 `hello`，然后选择 hello-world 蓝图。
4. 在 Configure triggers 页面上，选择 Next。
5. 在 Configure function 页面上，执行以下操作：
   a. 键入 Lambda 函数的名称和说明。(例如，将函数命名为“LogAutoScalingEvent”。)
   b. 编辑 Lambda 函数的示例代码。例如：

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. 对于 Role，选择 Choose an existing role，然后从 Existing role 中选择您的基本执行角色。否则，创建新的基本执行角色。
   d. 选择 Next。
6. 选择 Create function。

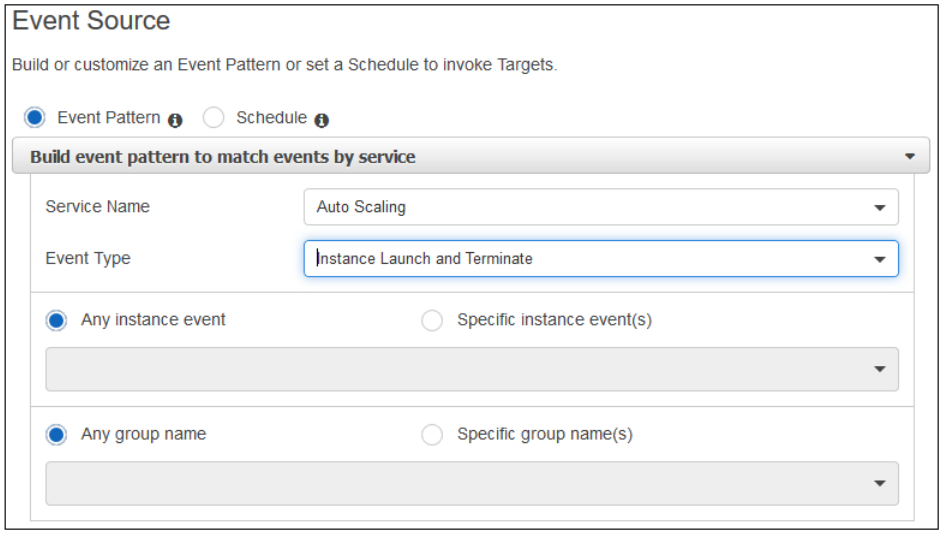
步骤 2：创建规则

创建一个规则，以便每当您的 Auto Scaling 组启动或终止一个实例时，就将运行您的 Lambda 函数。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
   a. 选择 Event Pattern。
   b. 选择 Build event pattern to match events by service。
   c. 选择 Auto Scaling，然后选择 Instance Launch and Terminate。

d. 选择 Any instance event 以捕获所有成功和失败的实例启动和终止事件。



4. 默认情况下，该规则与区域中任何 Auto Scaling 组匹配。要使该规则与特定 Auto Scaling 组匹配，请选择 Specific group name(s)，然后选择一个或多个 Auto Scaling 组。
5. 对于 Targets，选择 Add target，然后选择 Lambda function。
6. 对于 Function，选择您创建的 Lambda 函数。
7. 选择 Configure details。
8. 对于 Rule definition，键入规则的名称和描述。(例如，将规则描述为“Log whenever an Auto Scaling group scales out or in”。)
9. 选择 Create rule。

步骤 3：测试规则

您可以通过手动扩展 Auto Scaling 组来测试您的规则，以便其启动实例。等待几分钟，在扩展事件发生之后，您可以验证您的 Lambda 函数是否已调用。

使用 Auto Scaling 组测试您的规则

1. 要增加您的 Auto Scaling 组的大小，请执行以下操作：
   a. 打开 Amazon EC2 控制台 https://console.amazonaws.cn/ec2/。
   b. 在导航窗格上，依次选择 Auto Scaling 和 Auto Scaling Groups。
   c. 选择您的 Auto Scaling 组所对应的复选框。
   d. 在 Details 选项卡上，选择 Edit。对于 Desired，将所需容量增加一。例如，如果当前值是 2，请键入 3。理想容量必须小于或等于组的最大容量。因此，如果您的 Desired 新值大于 Max，则必须更新 Max。完成后，选择 Save。
2. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
3. 在导航窗格中，依次选择 Events 和 Rules，再选择所创建规则的名称，然后选择 Show metrics for the rule。
4. 若要查看 Lambda 函数的输出，请执行以下操作：
   a. 在导航窗格中，选择 Logs。
   b. 选择您的 Lambda 函数 (/aws/lambda/function-name) 的日志组的名称。

c. 选择日志流的名称，以查看您启动的实例的函数提供的数据。

5. (可选) 完成后，您可以将所需的容量减一，这样 Auto Scaling 组就会返回到它之前的大小。

教程：使用 CloudWatch Events 记录 Amazon S3 对象级别操作

您可以在 S3 存储桶上记录对象级别 API 操作。在 Amazon CloudWatch Events 可以匹配这些事件之前，您必须使用 AWS CloudTrail 设置配置为接收这些事件的跟踪。

步骤 1：配置您的 AWS CloudTrail 跟踪

为了将 S3 存储桶的数据事件记录到 AWS CloudTrail 和 CloudWatch Events，应创建一个跟踪。跟踪会捕获您账户中的 API 调用和相关事件，并将日志文件传输到您指定的 S3 存储桶。您可以更新现有跟踪或创建一个新跟踪。

创建跟踪

1. 在 https://console.amazonaws.cn/cloudtrail/ 打开 CloudTrail 控制台。
2. 在导航窗格中，依次选择 Trails (跟踪) 和 Create trail (创建跟踪)。
3. 对于 Trail name，键入跟踪的名称。
4. 对于 Data events，键入存储桶的名称和前缀 (可选)。对于每个跟踪，您可以添加最多 250 个 Amazon S3 对象。
   - 要记录存储桶中所有 Amazon S3 对象的数据事件，请指定一个 S3 存储桶和一个空前缀。当事件在该存储桶中的对象上发生时，跟踪将处理和记录事件。
   - 要记录特定 Amazon S3 对象的数据事件，请指定一个 S3 存储桶和该对象前缀。当事件在该 存储桶中的对象上发生且对象以指定前缀开头时，跟踪将处理和记录事件。
5. 对于每个资源，指定是否记录 Read-only、Write-only 或 All 事件。
6. 对于 Storage location，创建或选择要用于日志文件存储的现有 S3 存储桶。
7. 选择 Create。

有关更多信息，请参阅 AWS CloudTrail User Guide 中的数据事件。

步骤 2：创建 AWS Lambda 函数

创建一个 Lambda 函数，以记录 S3 存储桶的数据事件。在创建规则时，您可以指定此函数。

创建 Lambda 函数

1. 通过以下网址打开 AWS Lambda 控制台：https://console.amazonaws.cn/lambda/。
2. 如果您是首次使用 Lambda，则会看到一个欢迎页面；请选择 Get Started Now；否则，请选择 Create a Lambda function。
3. 在 Select blueprint 页面上，为筛选条件键入 `hello`，然后选择 hello-world 蓝图。
4. 在 Configure triggers 页面上，选择 Next。
5. 在 Configure function 页面上，执行以下操作：
   a. 键入 Lambda 函数的名称和说明。(例如，将函数命名为“LogS3DataEvents”。)
   b. 编辑 Lambda 函数的代码。例如：

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

c. 对于 Role，选择 Choose an existing role，然后从 Existing role 中选择您的基本执行角色。否则，创建新的基本执行角色。

d. 选择 Next。

6. 在 Review 页面上，选择 Create function。

步骤 3：创建规则

创建一个规则以便运行您的 Lambda 函数来响应 Amazon S3 数据事件。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：

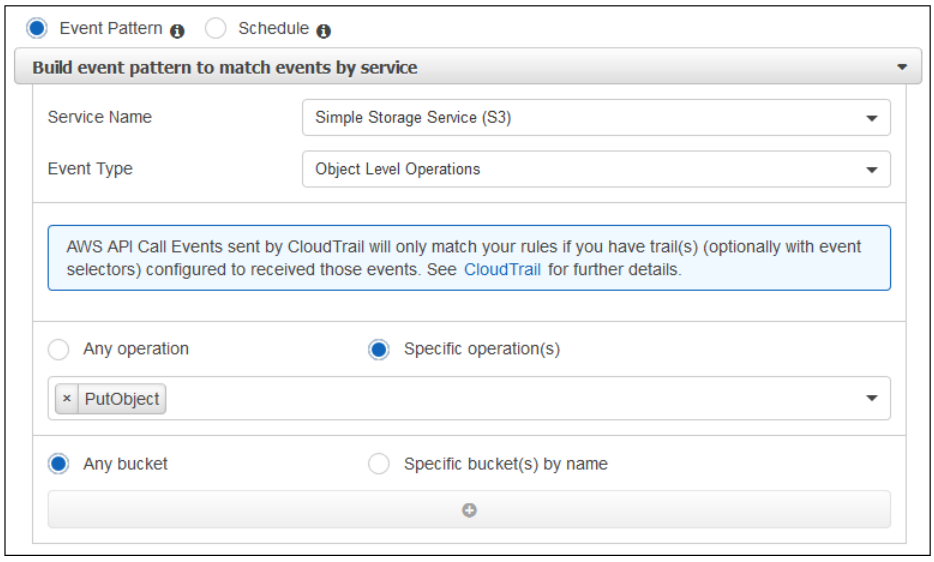
   a. 选择 Event Pattern。

   b. 选择 Build event pattern to match events by service。

   c. 选择 Simple Storage Service (S3)，然后选择 Object Level Operations。

   d. 选择 Specific operation(s)，然后选择 PutObject。

   e. 默认情况下，该规则与区域中所有存储桶的数据事件匹配。若要匹配特定存储桶的数据事件，请选择 Specify bucket(s) by name，然后指定一个或多个存储桶。



4. 对于 Targets，选择 Add target，然后选择 Lambda function。
5. 对于 Function，选择您创建的 Lambda 函数。
6. 选择 Configure details。

7. 对于 Rule definition，键入规则的名称和描述。
8. 选择 Create rule。

步骤 4：测试规则

为了测试规则，将一个对象置于 S3 存储桶中。您可以验证您的 Lambda 函数是否已调用。

查看 Lambda 函数的日志

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Logs。
3. 选择您的 Lambda 函数 (/aws/lambda/function-name) 的日志组的名称。
4. 选择日志流的名称，以查看您启动的实例的函数提供的数据。

您还可以在 S3 存储桶中检查您为跟踪指定的 CloudTrail 日志的内容。有关更多信息，请参阅 AWS CloudTrail User Guide 中的获取并查看您的 CloudTrail 日志文件。

教程：使用输入转换器自定义要传递给事件目标的内容

您可以使用 CloudWatch Events 的输入转换器功能自定义从事件获取的文本，然后再将文本输入规则目标。

您可以从事件中定义多个 JSON 路径，并将其输出分配给不同的变量。然后，您可以在输入模板中以 <*variable-name*> 形式使用这些变量。

如果指定一个变量来匹配事件中不存在的 JSON 路径，则该变量将替换为 null。不能对字符 < 和 > 进行转义。

在本教程中，我们从实例状态更改事件中提取 Amazon EC2 实例的实例 ID 和状态。我们使用输入转换器将这些数据放入发送给 Amazon SNS 主题的易于阅读的消息中。当任何实例更改为任何状态时，均将触发该规则。例如，使用此规则，以下 Amazon EC2 实例状态更改通知事件将产生 Amazon SNS 消息：EC2 实例 i-1234567890abcdef0 将状态更改为已停止。

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2015-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/ i-1234567890abcdef0"
   ],
   "detail":{
      "instance-id":" i-1234567890abcdef0",
      "state":"stopped"
   }
}
```

我们通过将 *instance* 变量映射到事件中的 `$.detail.instance-id` JSON 路径，将 *state* 变量映射到 `$.detail.state` JSON 路径，来实现这一点。然后，我们将输入模板设置为“EC2 实例 <instance> 将状态更改为 <state>。”

创建规则

使用输入转换器来自定义发送到目标的 Amazon EC2 实例状态更改信息

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
   a. 选择 Event Pattern。
   b. 选择 Build event pattern to match events by service。
   c. 选择 EC2，然后选择 EC2 Instance State-change Notification。
   d. 选择任何状态和任何实例。
4. 对于目标，选择添加目标，然后选择 SNS 主题。
5. 对于主题，选择您希望在 Amazon EC2 实例更改状态时通知的 Amazon SNS 主题。
6. 选择配置输入、输入转换器。
7. 在下一个框中，键入 {"state" : "$.detail.state", "instance" : "$.detail.instance-id"}
8. 在接下来的框中，键入 "The EC2 instance <instance> has changed state to <state>."
9. 选择 Configure details。
10. 键入规则的名称和描述，然后选择创建规则。

教程：使用 CloudWatch Events 记录 AWS API 调用

可以使用 AWS Lambda 函数来记录每个 AWS API 调用。例如，可以创建一个规则来记录 Amazon EC2 中的任何操作，也可以将此规则限制为仅记录特定的 API 调用。在此教程中，每当 Amazon EC2 实例停止时就记录。

先决条件

在您可以匹配这些事件之前，您必须使用 AWS CloudTrail 设置跟踪。如果您没有跟踪，请完成以下步骤。

创建跟踪

1. 在 https://console.amazonaws.cn/cloudtrail/ 打开 CloudTrail 控制台。
2. 依次选择 Trails 和 Add new trail。
3. 对于 Trail name，键入跟踪的名称。
4. 对于 S3 bucket，键入新存储桶的名称，CloudTrail 将在其中传输日志。
5. 选择 Create。

步骤 1：创建 AWS Lambda 函数

创建 Lambda 函数以记录 API 调用事件。在创建规则时，您可以指定此函数。

创建 Lambda 函数

1. 通过以下网址打开 AWS Lambda 控制台：https://console.amazonaws.cn/lambda/。
2. 如果您是首次使用 Lambda，则会看到一个欢迎页面；请选择 Get Started Now；否则，请选择 Create a Lambda function。
3. 在 Select blueprint 页面上，为筛选条件键入 `hello`，然后选择 hello-world 蓝图。
4. 在 Configure triggers 页面上，选择 Next。

5. 在 Configure function 页面上，执行以下操作：
   a. 键入 Lambda 函数的名称和说明。(例如，将函数命名为“LogEC2StopInstance”。)
   b. 编辑 Lambda 函数的示例代码。例如：

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2StopInstance');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. 对于 Role，选择 Choose an existing role，然后从 Existing role 中选择您的基本执行角色。否则，创建新的基本执行角色。
   d. 选择 Next。
6. 在 Review 页面上，选择 Create function。

步骤 2：创建规则

创建一个规则，以便每当您停止 Amazon EC2 实例时，就将运行您的 Lambda 函数。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 选择 。
4. 对于 Event source，执行以下操作：
   a. 选择 Event Pattern。
   b. 选择 Build event pattern to match events by service。
   c. 选择 EC2，然后选择 AWS API Call via CloudTrail。
   d. 选择 Specific operation(s)，然后在下面的框中键入 `StopInstances`。
5. 对于 Targets，选择 Add target，然后选择 Lambda function。
6. 对于 Function，选择您创建的 Lambda 函数。
7. 选择 Configure details。
8. 对于 Rule definition，键入规则的名称和描述。
9. 选择 Create rule。

步骤 3：测试规则

可使用 Amazon EC2 控制台停止 Amazon EC2 实例来测试您的规则。在等待几分钟以便实例停止后，检查 CloudWatch 控制台中的 AWS Lambda 指标，以验证您的函数是否已被调用。

通过停止一个实例来测试您的规则

1. 打开 Amazon EC2 控制台 https://console.amazonaws.cn/ec2/。
2. 启动 实例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的启动实例。
3. 停止实例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的停止并启动实例。
4. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。

5. 在导航窗格中，选择 Events，再选择所创建规则的名称，然后选择 Show metrics for the rule。
6. 若要查看 Lambda 函数的输出，请执行以下操作：

   a. 在导航窗格中，选择 Logs。
   b. 选择您的 Lambda 函数 (/aws/lambda/function-name) 的日志组的名称。
   c. 选择日志流的名称，以查看您停止的实例的函数提供的数据。
7. (可选) 完成后，您可以终止已停止的实例。想要了解更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的终止您的实例。

教程：使用 CloudWatch Events 计划自动化 Amazon EBS 快照

可以按照计划运行 CloudWatch Events 规则。在此教程中，您按照计划为现有 Amazon Elastic Block Store (Amazon EBS) 卷创建自动化快照。您可以选择一个固定速度，每隔几分钟创建一个快照；或者使用 cron 表达式来指定在每天的特定时间创建快照。

Important

只能在 AWS 管理控制台中创建带内置目标的规则。

步骤 1：创建规则

创建按照计划拍摄快照的规则。可以使用 rate 表达式或 Cron 表达式来指定计划。有关更多信息，请参阅 规则的计划表达式 (p. 27)。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event Source，执行以下操作：

   a. 选择 Schedule。
   b. 选择 Fixed rate of 并指定计划间隔 (例如，5 分钟)。或者，选择 Cron expression 并指定一个 Cron 表达式 (例如，从现在开始，周一至周五每 15 分钟一次)。
4. 对于 Targets (目标)，选择 Add target (添加目标)，然后选择 EC2 CreateSnapshot API call (EC2 CreateSnapshot API 调用)。您可能必须在可能目标的列表中向上滚动以查找 EC2 CreateSnapshot API 调用。
5. 对于卷 ID，输入目标 Amazon EBS 卷的卷 ID。
6. 选择 Create a new role for this specific resource。新的角色将向目标授予代表您访问资源的权限。
7. 选择 Configure details。
8. 对于 Rule definition，键入规则的名称和描述。
9. 选择 Create rule。

步骤 2：测试规则

在拍摄第一张快照后，您可以通过查看这张快照来验证您的规则。

测试您的规则

1. 打开 Amazon EC2 控制台 https://console.amazonaws.cn/ec2/。

2. 在导航窗格中，依次选择 Elastic Block Store 和 Snapshots。
3. 验证第一张快照是否在列表中显示。
4. (可选) 完成后，您可以禁用该规则，以防止拍摄其他快照。

   a. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
   b. 在导航窗格中，依次选择 Events 和 Rules。
   c. 选择规则，然后依次选择 Actions 和 Disable。
   d. 当系统提示确认时，选择 Disable。

教程：使用 CloudWatch Events 计划 AWS Lambda 函数

您可以设置规则以按计划运行 AWS Lambda 函数。本教程演示如何使用 AWS 管理控制台或 AWS CLI 创建规则。如果您想使用 AWS CLI 但尚未安装，请参阅 AWS Command Line Interface 用户指南。

CloudWatch Events 不在计划表达式中提供第二级精度。使用 cron 表达式的最高解析精度是一分钟。由于 CloudWatch Events 和目标服务的分布式特性，计划规则触发时间与目标服务实际执行目标资源的时间之间的延迟可能有几秒钟。您的计划规则会在这一分钟内触发，但不会精确到在第 0 秒时触发。

步骤 1：创建 AWS Lambda 函数

创建 Lambda 函数来记录计划的事件。在创建规则时，您可以指定此函数。

创建 Lambda 函数

1. 通过以下网址打开 AWS Lambda 控制台：https://console.amazonaws.cn/lambda/。
2. 如果您是首次使用 Lambda，则会看到一个欢迎页面；请选择 Get Started Now；否则，请选择 Create a Lambda function。
3. 在 Select blueprint 页面上，为筛选条件键入 `hello`，然后选择 hello-world 蓝图。
4. 在 Configure triggers 页面上，选择 Next。
5. 在 Configure function 页面上，执行以下操作：

   a. 键入 Lambda 函数的名称和说明。(例如，将函数命名为“LogScheduledEvent”。)
   b. 编辑 Lambda 函数的示例代码。例如：

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. 对于 Role，选择 Choose an existing role，然后从 Existing role 中选择您的基本执行角色。否则，创建新的基本执行角色。
   d. 选择 Next。
6. 在 Review 页面上，选择 Create function。

步骤 2：创建规则

创建按计划运行 Lambda 函数的规则。

使用控制台创建规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event Source，执行以下操作：
   a. 选择 Schedule。
   b. 选择 Fixed rate of 并指定计划间隔 (例如，5 分钟)。
4. 对于 Targets，选择 Add target，然后选择 Lambda function。
5. 对于 Function，选择您创建的 Lambda 函数。
6. 选择 Configure details。
7. 对于 Rule definition，键入规则的名称和描述。
8. 选择 Create rule。

如果您愿意，可以使用 AWS CLI 创建规则。首先，您必须向该规则授予调用您的 Lambda 函数的权限。然后，您可以创建规则并将该 Lambda 函数添加为目标。

使用 AWS CLI 创建规则

1. 使用以下 put-rule 命令以创建按计划触发其自身的规则：

```
aws events put-rule \
--name my-scheduled-rule \
--schedule-expression 'rate(5 minutes)'
```

   当此规则触发时，它会生成一个事件，该事件可作为此规则的目标的输入。以下是示例事件：

```
{
    "version": "0",
    "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
    "detail-type": "Scheduled Event",
    "source": "aws.events",
    "account": "123456789012",
    "time": "2015-10-08T16:53:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule"
    ],
    "detail": {}
}
```

2. 使用以下 add-permission 命令信任 CloudWatch Events 服务委托方 (events.amazonaws.com) 并使用指定的 Amazon 资源名称 (ARN) 设置规则的权限范围：

```
aws lambda add-permission \
--function-name LogScheduledEvent \
--statement-id my-scheduled-event \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule
```

3. 使用以下 put-targets 命令将您创建的 Lambda 函数添加到该规则，使其每 5 分钟运行一次：

```
aws events put-targets --rule my-scheduled-rule --targets file://targets.json
```

   创建文件 `targets.json` 并输入以下内容：

```
[
  {
    "Id": "1",
    "Arn": "arn:aws:lambda:us-east-1:123456789012:function:LogScheduledEvent"
  }
]
```

步骤 3：测试规则

您可以验证您的 Lambda 函数是否已调用。

测试您的规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，依次选择 Events 和 Rules，再选择所创建规则的名称，然后选择 Show metrics for the rule。
3. 若要查看 Lambda 函数的输出，请执行以下操作：
   a. 在导航窗格中，选择 Logs。
   b. 选择您的 Lambda 函数 (/aws/lambda/function-name) 的日志组的名称。
   c. 选择日志流的名称，以查看您启动的实例的函数提供的数据。
4. (可选) 完成后，可禁用该规则。
   a. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
   b. 在导航窗格中，依次选择 Events 和 Rules。
   c. 选择规则，然后依次选择 Actions 和 Disable。
   d. 当系统提示确认时，选择 Disable。

教程：将 AWS Systems Manager Automation 设置为 CloudWatch Events 目标

可以使用 CloudWatch Events 定期安排时间调用 AWS Systems Manager Automation，也可以在检测到指定事件时调用。本教程假定您根据特定事件调用 Systems Manager Automation。

创建 CloudWatch Events 规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
   a. 选择 Event Pattern，然后选择 Build event pattern to match events by service。
   b. 对于 Service Name 和 Event Type，选择要用作触发器的服务和事件类型。

   根据所选的服务和事件类型，您可能需要在 Event Source 下指定其他选项。
4. 对于 Targets，依次选择 Add Target 和 SSM Automation。
5. 对于 Document，选择在触发目标后要运行的 Systems Manager 文档。
6. (可选) 要指定文档的特定版本，请选择 Configure document version。
7. 在 Configure parameter(s) 下，选择 No Parameter(s) 或 Constant。

如果您选择 Constant，则指定要传递到文档执行的常量。

8. CloudWatch Events 可以创建要运行的事件所需的 IAM 角色：
   - 若要自动创建 IAM 角色，请选择 Create a new role for this specific resource。
   - 若要使用您之前创建的 IAM 角色，请选择 Use existing role。
9. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
10. 选择 Create rule。

教程：使用 CloudWatch Events 将事件中继到 Amazon Kinesis 流

可以将 CloudWatch Events 中的 AWS API 调用事件中继到 Amazon Kinesis 中的流。

先决条件

安装 AWS CLI。 有关更多信息，请参阅 AWS Command Line Interface 用户指南。

步骤 1：创建 Amazon Kinesis 流

使用下面的 create-stream 命令创建流。

```
aws kinesis create-stream --stream-name test --shard-count 1
```

当流状态为 ACTIVE 时，表示流已就绪。使用以下 describe-stream 命令检查流状态：

```
aws kinesis describe-stream --stream-name test
```

步骤 2：创建规则

例如，创建一条规则，以在您停止 Amazon EC2 实例时将事件发送到流。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source，执行以下操作：
   a. 选择 Event Pattern。
   b. 选择 Build event pattern to match events by service。
   c. 选择 EC2，然后选择 Instance State-change Notification。
   d. 选择 Specific state(s)，然后选择 Running。
4. 对于 Targets，选择 Add target，然后选择 Kinesis stream。
5. 对于 Stream，选择您创建的流。
6. 选择 Create a new role for this specific resource。
7. 选择 Configure details。
8. 对于 Rule definition，键入规则的名称和描述。
9. 选择 Create rule。

步骤 3：测试规则

为了测试规则，停止一个 Amazon EC2 实例。等待几分钟，在该实例停止之后，检查 CloudWatch 指标，以验证您的函数是否已调用。

通过停止一个实例来测试您的规则

1. 打开 Amazon EC2 控制台 https://console.amazonaws.cn/ec2/。
2. 启动 实例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的启动实例。
3. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
4. 在导航窗格中，依次选择 Events 和 Rules，再选择所创建规则的名称，然后选择 Show metrics for the rule。
5. (可选) 完成后，您可以终止实例。想要了解更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的终止您的实例。

步骤 4：验证事件是否已中继

您可以从流中获取记录，以验证事件是否已中继。

获取记录

1. 使用以下 get-shard-iterator 命令开始从 Kinesis 流中读取数据：

```
aws kinesis get-shard-iterator --shard-id shardId-000000000000 --shard-iterator-type
 TRIM_HORIZON --stream-name test
```

下面是示例输出：

```
{
    "ShardIterator": "AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg="
}
```

2. 使用以下 get-records 命令获取记录。分区迭代器是您在上一步获取的：

```
aws kinesis get-records --shard-
iterator AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp+KEd9I6AJ9ZG4lNR1EMi
+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd+efGN2aHFdkH1rJl4BL9Wyrk
+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg=
```

如果命令成功，它将从指定分区的流中请求记录。您可能会收到零个或多个记录。返回的任何记录都不能表示流中的所有记录。如果您未收到预期的数据，请继续调用 `get-records`。

Kinesis 中的记录是经过 Base64 编码的。但是，AWS CLI 中的流支持不提供 Base64 解码。如果您使用 Base64 解码程序手动解码数据，您会发现它是以 JSON 格式中继到流的事件。

教程：使用 AWS CodeBuild 安排自动构建

在本教程的示例中，您安排 AWS CodeBuild 在每个工作日晚上 8 点 (GMT) 运行构建任务。您还可以将一个常量传递到 AWS CodeBuild 以用于该计划构建。

创建规则，安排每晚 8 点构建 AWS CodeBuild 项目

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event Source，执行以下操作：

   a. 选择 Schedule。

   b. 选择 Cron 表达式，并将以下内容指定为表达式：0 20 ? * MON-FRI *. 有关 Cron 表达式的更多信息，请参阅规则的计划表达式 (p. 27)。
4. 对于目标，请选择添加目标，然后选择 CodeBuild 项目。
5. 对于项目 ARN ，请键入构建项目的 ARN。
6. 在本教程中，我们添加一个可选的步骤，将一个参数传递到 AWS CodeBuild 以覆盖默认值。在将 AWS CodeBuild 设置为目标时，不需要执行该步骤。要传递参数，请选择配置输入，然后选择常量 (JSON 文本)。

   在常量 (JSON 文本) 下面的框中，键入以下内容以将这些计划构建的超时覆盖设置为 30 分钟：{ "timeoutInMinutesOverride": 30 }

   有关可传递的参数的更多信息，请参阅 StartBuild。您无法在该字段中传递 `projectName` 参数。您可以在项目 ARN 中使用 ARN 指定项目。
7. CloudWatch Events 可以创建运行您的构建项目所需的 IAM 角色：

   - 若要自动创建 IAM 角色，请选择 Create a new role for this specific resource。
   - 若要使用您之前创建的 IAM 角色，请选择 Use existing role。这必须是已具有足够权限调用构建的角色。CloudWatch Events 不会为您选择的角色授予额外的权限。
8. 选择配置详细信息。
9. 对于 Rule definition，键入规则的名称和描述。
10. 选择 Create rule。

规则的计划表达式

可以使用 Cron 或 rate 表达式在 CloudWatch Events 中创建按自动化计划自行触发的规则。所有计划的事件都使用 UTC 时区，计划的最小精度为 1 分钟。

CloudWatch Events 不在计划表达式中提供第二级精度。使用 cron 表达式的最高解析精度是一分钟。由于 CloudWatch Events 和目标服务的分布式特性，计划规则触发时间与目标服务实际执行目标资源的时间之间的延迟可能有几秒钟。您的计划规则会在这一分钟内触发，但不会精确到在第 0 秒时触发。

CloudWatch Events 支持计划表达式的以下格式。

格式

Cron 表达式

Cron 表达式有六个必填字段，之间以空格分隔。

语法

```
cron(fields)
```

字段	值	通配符
分钟	0-59	, - * /
小时	0-23	, - * /
日期	1-31	, - * ? / L W
月	1-12 或 JAN-DEC	, - * /
星期几	1-7 或 SUN-SAT	, - * ? L #
年代	1970-2199	, - * /

通配符

- , (逗号) 通配符包含其他值。在“月份”字段中，JAN、FEB 和 MAR 将包含 January、February 和 March。
- - (破折号) 通配符用于指定范围。在“日”字段中，1-15 将包含指定月份的 1 - 15 日。
- * (星号) 通配符包含该字段中的所有值。在“小时”字段中，* 将包含每个小时。
- / (正斜杠) 通配符用于指定增量。在“分钟”字段中，您可以输入 1/10 以指定从一个小时的第一分钟开始的每个第十分钟 (例如，第 11 分钟、第 21 分钟和第 31 分钟，依此类推)。
- ? (问号) 通配符用于指定一个或另一个。在“日期”字段中，您可以输入 7，如果您不介意 7 日是星期几，则可以在“星期几”字段中输入 ?。
- “日期”或“星期几”字段中的 L 通配符用于指定月或周的最后一天。
- “日期”字段中的 W 通配符用于指定工作日。在“日期”字段中，3W 用于指定最靠近当月的第三周的日。
- “星期几”字段中的 # 通配符用于指定一个月内所指定星期几的特定实例。例如，3#2 指该月的第二个星期二：3 指的是星期二，因为它是每周的第三天，2 是指该月内该类型的第二天。

限制

- 您无法在同一 Cron 表达式中为日期和星期几字段同时指定值。如果您在其中一个字段中指定了值 (或 *)，则必须在另一个字段中使用 ? (问号)。
- 不支持产生的速率快于 1 分钟的 Cron 表达式。

示例

在创建带计划的规则时，可以使用以下示例 cron 字符串。

分钟	小时	日期	月	星期几	年代	意义
0	10	*	*	?	*	每天上午的 10:00 (UTC) 运行
15	12	*	*	?	*	每天在下午 12:15 (UTC) 运行
0	18	?	*	MON-FRI	*	每星期一到星期五的下午 6:00 (UTC) 运行
0	8	1	*	?	*	每月第 1 天的上午 8:00 (UTC) 运行
0/15	*	*	*	?	*	每 15 分钟运行一次
0/10	*	?	*	MON-FRI	*	从星期一到星期五，每 10 分钟运行一次
0/5	8-17	?	*	MON-FRI	*	每星期一到星期五的上午 8:00 和下午 5:55 (UTC) 之间，每 5 分钟运行一次

以下示例说明如何将 Cron 表达式与 AWS CLI put-rule 命令结合使用。第一个示例创建在每天中午 12:00 (UTC) 触发的规则。

```
aws events put-rule --schedule-expression "cron(0 12 * * ? *)" --name MyRule1
```

下一个示例创建在每天下午 2:00 过后的 5:35 (UTC) 触发的规则。

```
aws events put-rule --schedule-expression "cron(5,35 14 * * ? *)" --name MyRule2
```

下一个示例创建从 2002 到 2005 年在每个月最后一个周五的上午 10:15 (UTC) 触发的规则。

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2002-2005)" --name MyRule3
```

Rate 表达式

Rate 表达式在创建计划事件规则时启动，然后按照其定义的计划运行。

Rate 表达式有两个必需字段。这些字段用空格分隔。

语法

```
rate(value unit)
```

value

正数。

unit

时间单位。

有效值：minute | minutes | hour | hours | day | days

限制

如果值等于 1，则单位必须为单数。同样，对于大于 1 的值，单位必须为复数。例如，rate(1 hours) 和 rate(5 hour) 无效，而 rate(1 hour) 和 rate(5 hours) 有效。

示例

以下示例说明如何将 Rate 表达式与 AWS CLI put-rule 命令结合使用。

```
aws events put-rule --schedule-expression "rate(5 minutes)" --name MyRule3
```

```
aws events put-rule --schedule-expression "rate(1 hour)" --name MyRule4
```

```
aws events put-rule --schedule-expression "rate(1 day)" --name MyRule5
```

CloudWatch Events 中的事件模式

Amazon CloudWatch Events 中的事件表示为 JSON 对象。有关 JSON 对象的详细信息，请参阅 RFC 7159。以下是示例事件：

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/ i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": " i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

请务必记住以下有关事件的详细信息：

- 所有事件都具有相同的顶级字段 (上述示例中显示的字段)，这些字段永远不能缺少。
- detail 顶级字段的内容因生成事件的服务以及所生成的事件而异。source 字段和 detail-type 字段的组合用于标识在 detail 字段中找到的字段和值。有关由 AWS 服务生成的事件的示例，请参阅 CloudWatch Events 的事件类型。

下面描述了每个事件字段。

version

默认情况下，在所有事件中设置为 0 (零)。

id

为每个事件生成一个唯一值。在事件通过规则移到目标时以及处理事件时，这对于跟踪事件非常有用。

detail-type

与 source 字段组合起来标识显示在 detail 字段中的字段和值。

source

标识发起事件的服务。从 AWS 中发起的所有事件均以“aws.”开头。客户生成的事件可具有任意值，前提是它不以“aws.”开头。建议使用 Java 包名样式反向域名字符串。

要查找 AWS 服务的正确 `source` 值，请参阅 AWS 服务命名空间中的表。例如，Amazon CloudFront 的 `source` 值是 `aws.cloudfront`。

account

标识 AWS 账户的 12 位数字。

time

事件时间戳，可由发起事件的服务指定。如果事件跨时间间隔，则服务可能选择报告开始时间，因此该值会明显早于实际接收事件的时间。

region

标识事件源自的 AWS 区域。

resources

此 JSON 数组包含用于标识事件中涉及的资源的 ARN。是否包含这些 ARN 由服务决定。例如，Amazon EC2 实例状态更改包含 Amazon EC2 实例 ARN，Auto Scaling 事件包含实例和 Auto Scaling 组的 ARN，而对 AWS CloudTrail 的 API 调用不包含资源 ARN。

detail

一个 JSON 对象，其内容由发起事件的服务决定。上述示例中的 detail 内容非常简单，仅为两个字段。AWS API 调用事件的 detail 对象具有约 50 个字段，可嵌套多个级别。

事件模式

规则使用事件模式来选择事件并将事件路由到目标。模式匹配或不匹配事件。事件模式表示为 JSON 对象，其结构类似于事件的结构，例如：

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "detail": {
    "state": [ "running" ]
  }
}
```

请务必记住以下有关事件模式匹配的事项：

- 要使模式匹配事件，事件必须包含模式中列出的所有字段名。字段名必须显示在具有相同嵌套结构的事件中。
- 模式中未提及的事件的其他字段将被忽略；实际上，有一个 "*"：未提及字段的 "*" 通配符。
- 匹配是精确的 (逐个字符)，不进行小写化或任何其他字符串标准化。
- 要匹配的值遵循 JSON 规则：用引号引起来的字符串、数字以及不带引号的关键字 `true`、`false` 和 `null`。
- 数字匹配在字符串表示级别进行。例如，300、300.0 和 3.0e2 不相等。

在编写模式来匹配事件时，可以使用 `TestEventPattern` API 或 `test-event-pattern` CLI 命令以确保模式将匹配所需的事件。有关详细信息，请参阅 TestEventPattern 或 test-event-pattern。

以下事件模式将匹配此页面顶部的事件。第一个模式匹配的原因是该模式中指定的实例值之一匹配事件 (且该模式未指定事件中未包含的任何附加字段)。第二个模式匹配的原因是时间中包含“已终止”状态。

```
{
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcdefgh"
  ]
}
```

```
{
  "detail": {
    "state": [ "terminated" ]
  }
```

```
}
```

这些事件模式不匹配此页面顶部的事件。第一个模式不匹配的原因是该模式为状态指定了“待处理”值，且此值未在事件中显示。第二个模式不匹配的原因是该模式中指定的资源值未在事件中显示。

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "detail": {
    "state": [ "pending" ]
  }
}
```

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1::image/ami-12345678" ]
}
```

在事件模式中匹配 Null 值和空字符串。

您可以创建一种与具有 null 值或空字符串的事件字段匹配的模式。要了解其工作原理，请考虑下面的示例事件：

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [
   ],
  "detail": {
    "eventVersion": "",
    "responseElements": null
   }
}
```

要匹配其 `eventVersion` 值为空字符串的事件，请使用下面的模式，它可匹配该事件示例。

```
{
  "detail": {
     "eventVersion": [""]
  }
}
```

要匹配其 `responseElements` 值为 null 的事件，请使用下面的模式，它可匹配该事件示例。

```
{
  "detail": {
     "responseElements": [null]
  }
}
```

在模式匹配中，Null 值和空字符串是不可互换的。编写为检测空字符串的模式不会捕获 `null` 值。

CloudWatch Events 模式下的数组

模式中每个字段的值均为一个包含一个或多个值的数组，如果数组中的任一值匹配事件中的值，则模式匹配。如果事件中的值为数组，则在模式数组与事件数组的交集不为空时，模式匹配。

例如，某个示例事件模式包含以下文本：

```
"resources": [
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:111122223333:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:444455556666:instance/i-b188560f",
]
```

示例模式将与包括以下文本的事件相匹配，因为模式数组中的第一项与事件数组中的第二项匹配。

```
"resources": [
   "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/ASGTerminate",
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
]
```

每个支持服务的 CloudWatch Events 事件示例

以下 AWS 服务会发出 CloudWatch Events 可检测到的事件：

事件类型

未列出的服务的事件

您也可以对并不发出事件且不在前面列表中的服务使用 CloudWatch Events。AWS CloudTrail 是一个服务，可用于自动记录事件，例如 AWS 服务 API 调用。您可以创建对 CloudTrail 所捕获的信息触发的

CloudWatch Events 规则。有关 CloudTrail 的更多信息，请参阅什么是 AWS CloudTrail？。有关如何创建使用 CloudTrail 的 CloudWatch Events规则的更多信息，请参阅 使用 AWS CloudTrail 创建对 AWS API 调用触发的 CloudWatch Events 规则 (p. 7)。

Amazon EC2 Auto Scaling 事件

以下是 Amazon EC2 Auto Scaling 事件的示例。有关更多信息，请参阅 Amazon EC2 Auto Scaling 用户指南 中的在 Auto Scaling 组扩展时获取 CloudWatch 事件。

EC2 实例启动生命周期操作

由于生命周期挂钩，Amazon EC2 Auto Scaling 已将实例移入 `Pending:Wait` 状态。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:59fcbb81-
bd02-485d-80ce-563ef5b237bf:autoScalingGroupName/sampleASG"
  ],
  "detail": {
    "LifecycleActionToken": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
    "NotificationMetadata": "additional-info"
  }
}
```

EC2 实例启动成功

Amazon EC2 Auto Scaling 已成功启动实例。

```
{
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [
      "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/ASGLaunchSuccess",
      "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
      ],
  "detail": {
      "StatusCode": "InProgress",
      "AutoScalingGroupName": "ASGLaunchSuccess",
      "ActivityId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
      "Details": {
            "Availability Zone": "us-east-1b",
            "Subnet ID": "subnet-95bfcebe"
      },
      "RequestId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
```

```
      "EndTime": "2015-11-11T21:31:47.208Z",
      "EC2InstanceId": "i-b188560f",
      "StartTime": "2015-11-11T21:31:13.671Z",
      "Cause": "At 2015-11-11T21:31:10Z a user request created an Auto Scaling group
 changing the desired capacity from 0 to 1. At 2015-11-11T21:31:11Z an instance was started
 in response to a difference between desired and actual capacity, increasing the capacity
 from 0 to 1."
      }
}
```

EC2 实例启动失败

Auto Scaling 未能启动实例。

```
{
  "id": "1681ab87-4a09-459f-95a2-7fa09403c4b7",
  "detail-type": "EC2 Instance Launch Unsuccessful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:42:36Z",
  "region": "us-east-1",
  "resources": [
      "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:528ffce5-
ef9f-4c1d-8d18-5d005b4a438c:autoScalingGroupName/brokenASG",
      "arn:aws:ec2:us-east-1:123456789012:instance/"
      ],
  "detail": {
      "StatusCode": "Failed",
      "AutoScalingGroupName": "brokenASG",
      "ActivityId": "06076c51-4874-487d-b15b-7895a713ab55",
      "Details": {
            "Availability Zone": "us-east-1e",
            "Subnet ID": "subnet-16c5df2c"
       },
      "RequestId": "06076c51-4874-487d-b15b-7895a713ab55",
      "EndTime": "2015-11-11T21:42:36.000Z",
      "EC2InstanceId": "",
      "StartTime": "2015-11-11T21:42:36.698Z",
      "Cause": "At 2015-11-11T21:42:09Z a user request update of Auto Scaling group
 constraints to min: 0, max: 10, desired: 2 changing the desired capacity from 0 to 2. At
 2015-11-11T21:42:35Z an instance was started in response to a difference between desired
 and actual capacity, increasing the capacity from 0 to 2."
      }
}
```

EC2 实例终止生命周期操作

由于生命周期挂钩，Auto Scaling 已将实例移至 `Terminating:Wait` 状态。

```
{
  "version": "0",
  "id": "468fe059-f4b7-445f-bb22-2a271b94974d",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-12-22T18:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:59fcbb81-
bd02-485d-80ce-563ef5b237bf:autoScalingGroupName/sampleASG"
  ],
  "detail": {
    "LifecycleActionToken": "630aa23f-48eb-45e7-aba6-799ea6093a0f",
```

```
    "AutoScalingGroupName": "sampleASG",
    "LifecycleHookName": "SampleLifecycleHook-6789",
    "EC2InstanceId": "i-12345678",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING"
  }
}
```

EC2 实例终止成功

Auto Scaling 已成功终止实例。

```
{
  "id": "156d01c9-a6c3-4d7e-b883-5758266b95af",
  "detail-type": "EC2 Instance Terminate Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:36:57Z",
  "region": "us-east-1",
  "resources": [
      "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/ASGTerminate",
      "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
      ],
  "detail": {
      "StatusCode": "InProgress",
      "AutoScalingGroupName": "ASGTerminate",
      "ActivityId": "56472e79-538a-4ba7-b3cc-768d889194b0",
      "Details": {
            "Availability Zone": "us-east-1b",
            "Subnet ID": "subnet-95bfcebe"
            },
      "RequestId": "56472e79-538a-4ba7-b3cc-768d889194b0",
      "EndTime": "2015-11-11T21:36:57.498Z",
      "EC2InstanceId": "i-b188560f",
      "StartTime": "2015-11-11T21:36:12.649Z",
      "Cause": "At 2015-11-11T21:36:03Z a user request update of Auto Scaling group
 constraints to min: 0, max: 1, desired: 0 changing the desired capacity from 1 to
 0. At 2015-11-11T21:36:12Z an instance was taken out of service in response to a
 difference between desired and actual capacity, shrinking the capacity from 1 to 0. At
 2015-11-11T21:36:12Z instance i-b188560f was selected for termination."
      }
}
```

EC2 实例终止失败

Auto Scaling 未能终止实例。

```
{
  "id": "5e3df53a-0239-4e31-7d15-087ebef903ce",
  "detail-type": "EC2 Instance Terminate Unsuccessful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-12-01T23:34:57Z",
  "region": "us-east-1",
  "resources": [
      "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:cf5ebd9c-8e2a-4197-
abe2-2fb94e8d1f87:autoScalingGroupName/ASGTermFail",
      "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
      ],
  "detail": {
      "StatusCode": "InProgress",
      "Description": "Terminating EC2 instance: i-b188560f",
      "AutoScalingGroupName": "ASGTermFail",
```

```
      "ActivityId": "c1a8f6ce-82e8-4517-96ba-67d1999ceee4",
      "Details": {
            "Availability Zone": "us-east-1e",
            "Subnet ID": "subnet-915643ba"
            },
      "RequestId": "c1a8f6ce-82e8-4517-96ba-67d1999ceee4",
      "StatusMessage": "",
      "EndTime": "2015-12-01T23:34:57.721Z",
      "EC2InstanceId": "i-b188560f",
      "StartTime": "2015-12-01T23:33:48.489Z",
      "Cause": "At 2015-12-01T23:33:41Z a user request explicitly set group desired
 capacity changing the desired capacity from 2 to 0. At 2015-12-01T23:33:47Z an instance
 was taken out of service in response to a difference between desired and actual capacity,
 shrinking the capacity from 2 to 0. At 2015-12-01T23:33:47Z instance i-0867b4292c0cff474
 was selected for termination. At 2015-12-01T23:33:48Z instance i-b188560f was selected for
 termination."
      }
}
```

AWS API 调用事件

以下是 Amazon S3 的 AWS API 调用事件示例，用以创建存储桶：

```
{
    "version": "0",
    "id": "36eb8523-97d0-4518-b33d-ee3579ff19f0",
    "detail-type": "AWS API Call via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2016-02-20T01:09:13Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "Root",
            "principalId": "123456789012",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2016-02-20T01:05:59Z"
                }
            }
        },
        "eventTime": "2016-02-20T01:09:13Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "CreateBucket",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "100.100.100.100",
        "userAgent": "[S3Console/0.4]",
        "requestParameters": {
            "bucketName": "bucket-test-iad"
        },
        "responseElements": null,
        "requestID": "9D767BCC3B4E7487",
        "eventID": "24ba271e-d595-4e66-a7fd-9c16cbf8abae",
        "eventType": "AwsApiCall"
    }
}
```

仅支持以下服务的读/写事件。只读操作 (例如以 List、Get 或 Describe 开头) 不受支持。此外，大于 256KB 的 AWS API 调用事件不受支持。

- Amazon EC2 Auto Scaling
- AWS Certificate Manager
- AWS CloudFormation
- Amazon CloudFront
- AWS CloudHSM
- Amazon CloudSearch
- AWS CloudTrail
- Amazon CloudWatch
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- AWS CodeDeploy
- AWS CodePipeline
- Amazon Cognito 身份
- Amazon Cognito Sync
- AWS Config
- AWS Data Pipeline
- AWS Device Farm
- AWS Direct Connect
- AWS Directory Service
- AWS Database Migration Service
- Amazon DynamoDB
- Amazon Elastic Container Registry
- Amazon Elastic Container Service
- Amazon EC2 Systems Manager
- Amazon ElastiCache
- AWS Elastic Beanstalk
- Amazon Elastic Compute Cloud
- Amazon Elastic File System
- Elastic Load Balancing
- Amazon EMR
- Amazon Elastic Transcoder
- Amazon Elasticsearch Service
- Amazon GameLift
- Amazon Glacier
- AWS Identity and Access Management [美国东部（弗吉尼亚北部） only]
- Amazon Inspector
- AWS IoT
- AWS Key Management Service
- Amazon Kinesis
- Amazon Kinesis Data Firehose
- AWS Lambda
- Amazon Machine Learning
- AWS OpsWorks

- Amazon Polly
- Amazon Redshift
- Amazon Relational Database Service
- Amazon Route 53
- AWS Security Token Service
- Amazon Simple Email Service
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Simple Storage Service
- Amazon Simple Workflow Service
- AWS Step Functions
- AWS Storage Gateway
- AWS Support
- AWS WAF
- Amazon WorkDocs
- Amazon WorkSpaces

AWS Batch 事件

有关 AWS Batch 生成的事件示例，请参阅 AWS Batch 事件。

AWS CodeBuild 事件

有关 AWS CodeBuild 示例事件的信息，请参阅 AWS CodeBuild 用户指南 中的构建通知输入格式参考。

AWS CodeCommit 事件

以下是 AWS CodeCommit 事件的示例。

referenceCreated 事件

```
{
      "version": "0",
      "id": "01234567-0123-0123-0123-012345678901",
      "detail-type": "CodeCommit Repository State Change",
      "source": "aws.codecommit",
      "account": "123456789012",
      "time": "2017-06-12T10:23:43Z",
      "region": "us-east-1",
      "resources": [
        "arn:aws:codecommit:us-east-1:123456789012:myRepo"
      ],
      "detail": {
        "event": "referenceCreated",
        "repositoryName": "myRepo",
        "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
        "referenceType": "tag",
        "referenceName": "myTag",
        "referenceFullName": "refs/tags/myTag",
        "commitId": "3e5983EXAMPLE"
```

```
      }
 }
```

referenceUpdated 事件

```
{
      "version": "0",
      "id": "01234567-0123-0123-0123-012345678901",
      "detail-type": "CodeCommit Repository State Change",
      "source": "aws.codecommit",
      "account": "123456789012",
      "time": "2017-06-12T10:23:43Z",
      "region": "us-east-1",
      "resources": [
        "arn:aws:codecommit:us-east-1:123456789012:myRepo"
      ],
      "detail": {
        "event": "referenceUpdated",
        "repositoryName": "myRepo",
        "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
        "referenceType": "branch",
        "referenceName": "myBranch",
        "referenceFullName": "refs/heads/myBranch",
        "commitId": "26a8f2EXAMPLE",
        "oldCommitId": "3e5983EXAMPLE"
      }
}
```

referenceDeleted 事件

```
{
     "version": "0",
      "id": "01234567-0123-0123-0123-012345678901",
      "detail-type": "CodeCommit Repository State Change",
      "source": "aws.codecommit",
      "account": "123456789012",
      "time": "2017-06-12T10:23:43Z",
      "region": "us-east-1",
      "resources": [
        "arn:aws:codecommit:us-east-1:123456789012:myRepo"
      ],
      "detail": {
        "event": "referenceDeleted",
        "repositoryName": "myRepo",
        "repositoryId": "12345678-1234-5678-abcd-12345678abcd",
        "referenceType": "branch",
        "referenceName": "myBranch",
        "referenceFullName": "refs/heads/myBranch",
        "oldCommitId": "26a8f2EXAMPLE"
      }
}
```

AWS CodeDeploy 事件

以下是 AWS CodeDeploy 事件的示例。有关更多信息，请参阅 AWS CodeDeploy User Guide 中的使用 CloudWatch Events 监视部署。

CodeDeploy 部署状态更改通知

部署状态发生更改。

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Deployment State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T22:06:31Z",
  "id": "c071bfbf-83c4-49ca-a6ff-3df053957145",
  "resources": [
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup"
  ],
  "detail": {
    "instanceGroupId": "9fd2fbef-2157-40d8-91e7-6845af69e2d2",
    "region": "us-east-1",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "state": "SUCCESS",
    "deploymentGroup": "myDeploymentGroup"
  }
}
```

CodeDeploy 实例状态更改通知

属于部署组的实例状态发生更改。

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Instance State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T23:18:50Z",
  "id": "fb1d3015-c091-4bf9-95e2-d98521ab2ecb",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup",
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication"
  ],
  "detail": {
    "instanceId": "i-0000000aaaaaaaaaa",
    "region": "us-east-1",
    "state": "SUCCESS",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "instanceGroupId": "8cd3bfa8-9e72-4cbe-a1e5-da4efc7efd49",
    "deploymentGroup": "myDeploymentGroup"
  }
}
```

AWS CodePipeline 事件

以下是 AWS CodePipeline 事件的示例。

管道执行状态更改

```
{
  "version": "0",
```

```
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Pipeline Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": "1",
    "state": "STARTED",
    "execution-id": "01234567-0123-0123-0123-012345678901"
  }
}
```

阶段执行状态更改

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Stage Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": "1",
    "execution-id": "01234567-0123-0123-0123-012345678901",
    "stage": "Prod",
    "state": "STARTED"
  }
}
```

操作执行状态更改

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Action Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": 1,
    "execution-id": "01234567-0123-0123-0123-012345678901",
    "stage": "Prod",
    "action": "myAction",
    "state": "STARTED",
    "type": {
      "owner": "AWS",
      "category": "Deploy",
      "provider": "CodeDeploy",
```

```
      "version": 1
    }
  }
}
```

AWS 管理控制台登录事件

以下是控制台登录事件的示例：

```
{
  "id": "6f87d04b-9f74-4f04-a780-7acf4b0a9b38",
  "detail-type": "AWS Console Sign In via CloudTrail",
  "source": "aws.signin",
  "account": "123456789012",
  "time": "2016-01-05T18:21:27Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "Root",
            "principalId": "123456789012",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012"
            },
      "eventTime": "2016-01-05T18:21:27Z",
      "eventSource": "signin.amazonaws.com",
      "eventName": "ConsoleLogin",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "0.0.0.0",
      "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36",
      "requestParameters": null,
      "responseElements": {
            "ConsoleLogin": "Success"
            },
      "additionalEventData": {
            "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
            "MobileVersion": "No",
            "MFAUsed": "No" },
      "eventID": "324731c0-64b3-4421-b552-dfc3c27df4f6",
      "eventType": "AwsConsoleSignIn"
      }
}
```

Amazon EBS 事件

以下是 Amazon Elastic Block Store (Amazon EBS) 事件的示例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的 Amazon CloudWatch Events for Amazon EBS。

EBS 快照通知

Amazon EBS 创建了快照 (`createSnapshot`)，复制了快照 (`copySnapshot`)，或共享了快照 (`shareSnapshot`)。`detail` 字段中的 `source` 字段不会将账户 ID 作为卷 ARN 的一部分包含在内。

```
{
  "version": "0",
```

```
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2016-11-14T01:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "createSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::us-west-2:volume/vol-01234567",
    "StartTime": "2016-11-14T00:00:00Z",
    "EndTime": "2016-11-ddT01:30:00Z"
  }
}
```

EBS 卷通知

在 Amazon EBS 创建或删除卷、未能创建卷、未能附加卷或未能重新附加卷时，将会生成事件。

Amazon EBS 卷创建

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"EBS Volume Notification",
   "source":"aws.ec2",
   "account":"0123456789ab",
   "time":"2017-12-29T17:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567"
   ],
   "detail":{
      "result":"available",
      "cause":"",
      "event":"createVolume",
      "request-id":"01234567-0123-0123-0123-0123456789ab"
   }
}
```

Amazon EBS 卷删除

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"EBS Volume Notification",
   "source":"aws.ec2",
   "account":"0123456789ab",
   "time":"2017-12-29T17:28:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1: 0123456789ab:volume/vol-01234567"
   ],
   "detail":{
      "result":"deleted",
      "cause":"",
      "event":"deleteVolume",
```

```
      "request-id":"01234567-0123-0123-0123-0123456789ab"
   }
}
```

Amazon EBS 卷创建失败

以下实例展示了一个创建卷的失败尝试。连接失败或重新连接的事件类似，只不过 "event" 字段的值分别是 attachVolume 或 reattachVolume。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "2016-11-14T00:30:07Z",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
 is disabled.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",

}
```

Amazon EC2 事件

以下是 Amazon EC2 事件的示例。

EC2 实例状态更改通知

此 EC2 实例状态更改通知事件示例说明处于 pending 状态的实例。state 其他可能的值包括 running、shutting-down、stopped、stopping 和 terminated。

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2015-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
   ],
   "detail":{
      "instance-id":"i-abcd1111",
      "state":"pending"
   }
}
```

EC2 Spot 实例中断

以下是在 Amazon EC2 中断 Spot 实例时发出的事件的示例。

```
{
```

```
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "EC2 Spot Instance Interruption Warning",
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ"
    "region": "us-east-2",
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],
    "detail": {
        "instance-id": "i-1234567890abcdef0",
        "instance-action": "action"
    }
}
```

AWS OpsWorks Stacks 事件

以下是 AWS OpsWorks Stacks 事件的示例。

AWS OpsWorks Stacks 实例状态更改

指示 AWS OpsWorks Stacks 实例的状态更改。以下是实例状态。

- booting
- connection_lost
- online
- pending
- rebooting
- requested
- running_setup
- setup_failed
- shutting_down
- start_failed
- stopping
- stop_failed
- stopped
- terminating
- terminated

```
{
  "version": "0",
  "id": "dc5fa8df-48f1-2108-b1b9-1fe5ebcf2296",
  "detail-type": "OpsWorks Instance State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-25T11:12:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50z3e4z500z"
  ],
  "detail": {
    "initiated_by": "user",
    "hostname": "testing1",
    "stack-id": "acd3df16-e859-4598-8414-377b12a902da",
```

```
    "layer-ids": [
      "d1a0cb7f-c7e9-4a63-811c-976f0267b2c8"
    ],
    "instance-id": "a648d98f-fdd8-4323-952a-a50z3e4z500z",
    "ec2-instance-id": "i-08b1c2b67aa292276",
    "status": "requested"
  }
}
```

只有当实例处于 requested、terminating 或 stopping 状态时，才会填充 initiated_by 字段。initiated_by 字段可以包含以下值之一。

- user - 用户使用 API 或 AWS 管理控制台请求的实例状态更改。
- auto-scaling - AWS OpsWorks Stacks 自动扩展功能启动的实例状态更改。
- auto-healing - AWS OpsWorks Stacks 自动修复功能启动的实例状态更改。

AWS OpsWorks Stacks 命令状态更改

AWS OpsWorks Stacks 命令的状态中出现的更改。命令状态如下。

- expired - 命令超时。
- failed - 出现一般命令故障。
- skipped - 由于实例在 AWS OpsWorks Stacks 中与在 Amazon EC2 中具有不同的状态，跳过了命令。
- successful - 命令成功。
- superseded - 由于命令将应用已经应用过的配置更改，跳过了命令。

```
{
  "version": "0",
  "id": "96c778b6-a40e-c8c1-aafc-c9852a3a7b52",
  "detail-type": "OpsWorks Command State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-26T08:54:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50a3e4e500f"
  ],
  "detail": {
    "command-id": "acc9f4f3-a3ec-4fab-b70f-c7d04e71e3ec",
    "instance-id": "a648d98f-fdd8-4323-952a-a50a3e4e500f",
    "type": "setup",
    "status": "successful"
  }
}
```

AWS OpsWorks Stacks 部署状态更改

AWS OpsWorks Stacks 部署的状态中出现的更改。部署状态如下。

- running
- successful
- failed

```
{
  "version": "0",
```

```
  "id": "b8230afa-60c7-f43f-b632-841c1cfb22ff",
  "detail-type": "OpsWorks Deployment State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-25T11:15:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50a3e4e500f"
  ],
  "detail": {
    "duration": 16,
    "stack-id": "acd3df16-e859-4598-8414-377b12a902da",
    "instance-ids": [
      "a648d98f-fdd8-4323-952a-a50a3e4e500f"
    ],
    "deployment-id": "606419dc-418e-489c-8531-bff9770fc346",
    "command": "configure",
    "status": "successful"
  }
}
```

只有在部署完成后才填充 `duration` 字段，以秒为单位显示时间。

AWS OpsWorks Stacks 警报

引发了 AWS OpsWorks Stacks 服务错误。

```
{
  "version": "0",
  "id": "f99faa6f-0e27-e398-95bb-8f190806d275",
  "detail-type": "OpsWorks Alert",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-20T16:51:29Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "stack-id": "2f48f2be-ac7d-4dd5-80bb-88375f94db7b",
    "instance-id": "986efb74-69e8-4c6d-878e-5b77c054cbb0",
    "type": "InstanceStop",
    "message": "The shutdown of the instance timed out. Please try stopping it again."
  }
}
```

AWS Systems Manager 事件

以下是 AWS Systems Manager 事件的示例。有关更多信息，请参阅 Amazon EC2 用户指南（适用于 Linux 实例）中的记录 Run Command 的命令执行状态更改。

运行命令状态更改通知

```
{
    "version": "0",
    "id": "51c0891d-0e34-45b1-83d6-95db273d1602",
    "detail-type": "EC2 Command Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
```

```
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "expire-after": "2016-07-14T22:01:30.049Z",
        "parameters": {
            "executionTimeout": ["3600"],
            "commands": ["date"]
        },
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

运行命令调用状态更改通知

```
{
    "version": "0",
    "id": "4780e1b8-f56b-4de5-95f2-95db273d1602",
    "detail-type": "EC2 Command Invocation Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "instance-id": "i-9bb89e2b",
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

自动化步骤状态更改通知

```
{
  "version": "0",
  "id": "eeca120b-a321-433e-9635-dab369006a6b",
  "detail-type": "EC2 Automation Step Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-1:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-1:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "EndTime": "Nov 29, 2016 7:43:25 PM",
    "StartTime": "Nov 29, 2016 7:43:23 PM",
    "Time": 2630.0,
    "StepName": "runFixedCmds",
    "Action": "aws:runCommand"
  }
}
```

自动化执行状态更改通知

```
{
```

```
  "version": "0",
  "id": "d290ece9-1088-4383-9df6-cd5b4ac42b99",
  "detail-type": "EC2 Automation Execution Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-1:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-1:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "StartTime": "Nov 29, 2016 7:43:20 PM",
    "EndTime": "Nov 29, 2016 7:43:26 PM",
    "Time": 5753.0,
    "ExecutedBy": "arn:aws:iam::123456789012:user/userName"
  }
}
```

状态管理器关联状态更改

```
{
   "version":"0",
   "id":"db839caf-6f6c-40af-9a48-25b2ae2b7774",
   "detail-type":"EC2 State Manager Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-05-16T23:01:10Z",
   "region":"us-west-1",
   "resources":[
      "arn:aws:ssm:us-west-1::document/AWS-RunPowerShellScript"
   ],
   "detail":{
      "association-id":"6e37940a-23ba-4ab0-9b96-5d0a1a05464f",
      "document-name":"AWS-RunPowerShellScript",
      "association-version":"1",
      "document-version":"Optional.empty",
      "targets":"[{\"key\":\"InstanceIds\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-13T17:22:54.458Z",
      "last-successful-execution-date":"2017-05-16T23:00:01Z",
      "last-execution-date":"2017-05-16T23:00:01Z",
      "last-updated-date":"2017-02-13T17:22:54.458Z",
      "status":"Success",
      "association-status-aggregated-count":"{\"Success\":1}",
      "schedule-expression":"cron(0 */30 * * * ? *)",
      "association-cwe-version":"1.0"
   }
}
```

State Manager 实例关联状态更改

```
{
   "version":"0",
   "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
   "detail-type":"EC2 State Manager Instance Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-02-23T15:23:48Z",
   "region":"us-east-1",
   "resources":[
```

```
      "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
      "arn:aws:ssm:us-east-1:123456789012:document/my-custom-document"
   ],
   "detail":{
      "association-id":"34fcb7e0-9a14-4984-9989-0e04e3f60bd8",
      "instance-id":"i-12345678",
      "document-name":"my-custom-document",
      "document-version":"1",
      "targets":"[{\"key\":\"instanceids\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-23T15:23:48Z",
      "last-successful-execution-date":"2017-02-23T16:23:48Z",
      "last-execution-date":"2017-02-23T16:23:48Z",
      "status":"Success",
      "detailed-status":"",
      "error-code":"testErrorCode",
      "execution-summary":"testExecutionSummary",
      "output-url":"sampleurl",
      "instance-association-cwe-version":"1"
   }
}
```

AWS Systems Manager Parameter Store 事件

以下是 Amazon EC2 Systems Manager (SSM) Parameter Store 事件的示例。

创建参数

```
{
  "version": "0",
  "id": "6a7e4feb-b491-4cf7-a9f1-bf3703497718",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:43:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Create",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

更新参数

```
{
  "version": "0",
  "id": "9547ef2d-3b7e-4057-b6cb-5fdf09ee7c8f",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:44:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
```

```
  "detail": {
    "operation": "Update",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

删除参数

```
{
  "version": "0",
  "id": "80e9b391-6a9b-413c-839a-453b528053af",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:45:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Delete",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

AWS Systems Manager 配置合规性事件

以下是 Amazon EC2 Systems Manager (SSM) 配置合规性事件的示例。

关联合规

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

关联不合规

```
{
```

```
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "non_compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

补丁合规

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

补丁不合规

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "non_compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
```

```
}
```

Amazon EC2 维护时段事件

以下是 Amazon EC2 维护时段事件的示例。

注册目标

状态也可能是 DEREGISTERED。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Target Registration Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:001312665065:maintenancewindow/mw-0ed7251d3fcf6e0c2",
      "arn:aws:ssm:us-west-2:001312665065:windowtarget/
e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6"
   ],
   "detail":{
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "status":"REGISTERED"
   }
}
```

时段执行类型

其他可能的状态为 PENDING、IN_PROGRESS、SUCCESS、FAILED、TIMED_OUT 和 SKIPPED_OVERLAPPING。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Execution State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

任务执行类型

其他可能的状态为 IN_PROGRESS、SUCCESS、FAILED 和 TIMED_OUT。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Execution State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:56Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.759Z",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "end-time":"2016-11-16T01:00:56.847Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

已处理的任务目标

其他可能的状态为 IN_PROGRESS、SUCCESS、FAILED 和 TIMED_OUT。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Target Invocation State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-123456789012",
      "status":"TIMED_OUT",
      "owner-information":"Owner"
   }
}
```

时段状态更改

可能的状态为 ENABLED 和 DISABLED。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
```

```
   ],
   "detail":{
      "window-id":"mw-123456789012",
      "status":"DISABLED"
   }
}
```

Amazon ECS 事件

对于 Amazon ECS 示例事件，请参阅 Amazon Elastic Container Service Developer Guide 中的 Amazon ECS 事件。

Amazon EMR 事件

以下是 Amazon EMR 事件的示例。

Amazon EMR Auto Scaling 策略状态更改

```
{
   "version":"0",
   "id":"2f8147ab-8c48-47c6-b0b6-3ee23ec8d300",
   "detail-type":"EMR Auto Scaling Policy State Change",
   "source":"aws.emr",
   "account":"123456789012",
   "time":"2016-12-16T20:42:44Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "resourceId":"ig-X2LBMHTGPCBU",
      "clusterId":"j-1YONHTCP3YZKC",
      "state":"PENDING",
      "message":"AutoScaling policy modified by user request",
      "scalingResourceType":"INSTANCE_GROUP"
   }
}
```

Amazon EMR 集群状态更改 - 正在启动

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "STARTING",
    "message": "Amazon EMR cluster j-123456789ABCD (Development Cluster) was requested at
 2016-12-16 20:42 UTC and  is being created."
  }
```

```
}
```

Amazon EMR 集群状态更改 - 已终止

```
{
  "version": "0",
  "id": "1234abb0-f87e-1234-b7b6-000000123456",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T21:00:23Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user
 request\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "TERMINATED",
    "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has terminated at
 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
  }
}
```

Amazon EMR 实例组状态更改

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Instance Group State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:57:47Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "market": "ON_DEMAND",
    "severity": "INFO",
    "requestedInstanceCount": "2",
    "instanceType": "m3.xlarge",
    "instanceGroupType": "CORE",
    "instanceGroupId": "ig-ABCDEFGHIJKL",
    "clusterId": "j-123456789ABCD",
    "runningInstanceCount": "2",
    "state": "RUNNING",
    "message": "The resizing operation for instance group ig-ABCDEFGHIJKL in Amazon EMR
 cluster j-123456789ABCD (Development Cluster) is complete. It now has an instance count of
 2. The resize started at 2016-12-16 20:57 UTC and took 0 minutes to complete."
  }
}
```

Amazon EMR 步骤状态更改

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Step Status Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:53:09Z",
  "region": "us-east-1",
```

```
  "resources": [],
  "detail": {
    "severity": "ERROR",
    "actionOnFailure": "CONTINUE",
    "stepId": "s-ZYXWVUTSRQPON",
    "name": "CustomJAR",
    "clusterId": "j-123456789ABCD",
    "state": "FAILED",
    "message": "Step s-ZYXWVUTSRQPON (CustomJAR) in Amazon EMR cluster j-123456789ABCD
 (Development Cluster) failed at 2016-12-16 20:53 UTC."
  }
}
```

Amazon GameLift 事件

以下是 Amazon GameLift 事件的示例。有关更多信息，请参阅 Amazon GameLift 开发人员指南 中的 FlexMatch 事件引用。

对战搜索

```
{
  "version": "0",
  "id": "cc3d3ebe-1d90-48f8-b268-c96655b8f013",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T21:15:36.421Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T21:15:35.676Z",
        "players": [
          {
            "playerId": "player-1"
          }
        ]
      }
    ],
    "estimatedWaitMillis": "NOT_AVAILABLE",
    "type": "MatchmakingSearching",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  }
}
```

潜在的对战游戏已创建

```
{
  "version": "0",
  "id": "fce8633f-aea3-45bc-aeba-99d639cad2d4",
  "detail-type": "GameLift Matchmaking Event",
```

```
"source": "aws.gamelift",
"account": "123456789012",
"time": "2017-08-08T21:17:41.178Z",
"region": "us-west-2",
"resources": [
  "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
],
"detail": {
  "tickets": [
    {
      "ticketId": "ticket-1",
      "startTime": "2017-08-08T21:15:35.676Z",
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    },
    {
      "ticketId": "ticket-2",
      "startTime": "2017-08-08T21:17:40.657Z",
      "players": [
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    }
  ],
  "acceptanceTimeout": 600,
  "ruleEvaluationMetrics": [
    {
      "ruleName": "EvenSkill",
      "passedCount": 3,
      "failedCount": 0
    },
    {
      "ruleName": "EvenTeams",
      "passedCount": 3,
      "failedCount": 0
    },
    {
      "ruleName": "FastConnection",
      "passedCount": 3,
      "failedCount": 0
    },
    {
      "ruleName": "NoobSegregation",
      "passedCount": 3,
      "failedCount": 0
    }
  ],
  "acceptanceRequired": true,
  "type": "PotentialMatchCreated",
  "gameSessionInfo": {
    "players": [
      {
        "playerId": "player-1",
        "team": "red"
      },
      {
        "playerId": "player-2",
        "team": "blue"
      }
    ]
```

```
    },
    "matchId": "3faf26ac-f06e-43e5-8d86-08feff26f692"
  }
}
```

接受对战游戏

```
{
  "version": "0",
  "id": "b3f76d66-c8e5-416a-aa4c-aa1278153edc",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:04:42.660Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T20:04:16.637Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue",
            "accepted": false
          }
        ]
      }
    ],
    "type": "AcceptMatch",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue",
          "accepted": false
        }
      ]
    },
    "matchId": "848b5f1f-0460-488e-8631-2960934d13e5"
  }
}
```

接受对战游戏已完成

```
{
```

```
  "version": "0",
  "id": "b1990d3d-f737-4d6c-b150-af5ace8c35d3",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T20:43:14.621Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T20:30:40.972Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-08T20:33:14.111Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue"
          }
        ]
      }
    ],
    "acceptance": "TimedOut",
    "type": "AcceptMatchCompleted",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    },
    "matchId": "a0d9bd24-4695-4f12-876f-ea6386dd6dce"
  }
}
```


对战已成功


```
{
  "version": "0",
  "id": "5ccb6523-0566-412d-b63c-1569e00d023d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T19:59:09.159Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
```

```
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:58:59.277Z",
        "players": [
          {
            "playerId": "player-1",
            "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T19:59:08.663Z",
        "players": [
          {
            "playerId": "player-2",
            "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
            "team": "blue"
          }
        ]
      }
    ],
    "type": "MatchmakingSucceeded",
    "gameSessionInfo": {
      "gameSessionArn": "arn:aws:gamelift:us-west-2:123456789012:gamesession/836cf48d-
bcb0-4a2c-bec1-9c456541352a",
      "ipAddress": "192.168.1.1",
      "port": 10777,
      "players": [
        {
          "playerId": "player-1",
          "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
          "team": "blue"
        }
      ]
    },
    "matchId": "c0ec1a54-7fec-4b55-8583-76d67adb7754"
  }
}
```


对战超时


```
{
  "version": "0",
  "id": "fe528a7d-46ad-4bdc-96cb-b094b5f6bf56",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:11:35.598Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "reason": "TimedOut",
    "tickets": [
      {
        "ticketId": "ticket-1",
```

```
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      }
    ],
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 3,
        "failedCount": 0
      }
    ],
    "type": "MatchmakingTimedOut",
    "message": "Removed from matchmaking due to timing out.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    }
  }
}
```

对战已取消

```
{
  "version": "0",
  "id": "8d6f84da-5e15-4741-8d5c-5ac99091c27f",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:00:07.843Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "reason": "Cancelled",
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:59:26.118Z",
        "players": [
          {
```

```
            "playerId": "player-1"
          }
        ]
      }
    ],
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 0,
        "failedCount": 0
      }
    ],
    "type": "MatchmakingCancelled",
    "message": "Cancelled by request.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  }
}
```

对战已失败

```
{
  "version": "0",
  "id": "025b55a4-41ac-4cf4-89d1-f2b3c6fd8f9d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-16T18:41:09.970Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-16T18:41:02.631Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      }
    ],
```

```
    "customEventData": "foo",
    "type": "MatchmakingFailed",
    "reason": "UNEXPECTED_ERROR",
    "message": "An unexpected error was encountered during match placing.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    },
    "matchId": "3ea83c13-218b-43a3-936e-135cc570cba7"
  }
}
```

AWS Glue 事件

以下是 AWS Glue 事件的格式。

成功的作业运行

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T18:57:21Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"INFO",
        "state":"SUCCEEDED",
        "jobRunId":"jr_abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789",
        "message":"Job run succeeded"
    }
}
```

失败的作业运行

```
{
    "version":"0",
    "id":"abcdef01-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T06:02:03Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"ERROR",
        "state":"FAILED",
        "jobRunId":"jr_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef",
        "message":"JobName:MyJob and
 JobRunId:jr_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef failed to
 execute with exception Role arn:aws:iam::123456789012:role/Glue_Role should be given
 assume role permissions for Glue Service."
```

```
    }
}
```

停止的作业运行

```
{
"version":"0",
"id":"abcdef00-1234-5678-9abc-def012345678",
"detail-type":"Glue Job State Change",
"source":"aws.glue",
"account":"123456789012",
"time":"2017-11-20T20:22:06Z",
"region":"us-east-1",
"resources":[],
"detail":{
"jobName":"MyJob",
"severity":"INFO",
"state":"STOPPED",
"jobRunId":"jr_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
"message":"Job run stopped"
}
}
```

爬网程序已启动

```
{
   "version":"0",
   "id":"05efe8a2-c309-6884-a41b-3508bcdc9695",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"561226563745",
   "time":"2017-11-11T01:09:46Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "accountId":"561226563745",
      "crawlerName":"S3toS3AcceptanceTestCrawlera470bd94-9e00-4518-8942-e80c8431c322",
      "startTime":"2017-11-11T01:09:46Z",
      "state":"Started",
      "message":"Crawler Started"
   }
}
```

爬网程序成功

```
{
   "version":"0",
   "id":"3d675db5-59b9-6388-b8e8-e0a9b6d567a9",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"561226563745",
   "time":"2017-11-11T01:25:00Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "tablesCreated":"0",
      "warningMessage":"N/A",
      "partitionsUpdated":"0",
```

```
      "tablesUpdated":"0",
      "message":"Crawler Succeeded",
      "partitionsDeleted":"0",
      "accountId":"561226563745",
      "runningTime (sec)":"7",
      "tablesDeleted":"0",
      "crawlerName":"SchedulerTestCrawler51fb3a8b-1015-49f0-a969-ca126680b94b",
      "completionDate":"2017-11-11T01:25:00Z",
      "state":"Succeeded",
      "partitionsCreated":"0",
      "cloudWatchLogLink":"https://console.aws.amazon.com/
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws-glue/
crawlers;stream=SchedulerTestCrawler51fb3a8b-1015-49f0-a969-ca126680b94b"
   }
}
```

爬网程序失败

```
{
   "version":"0",
   "id":"f7965b59-470f-2e06-bb89-a8cebaabefac",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"782104008917",
   "time":"2017-10-20T05:10:08Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "crawlerName":"test-crawler-notification",
      "errorMessage":"Internal Service Exception",
      "accountId":"1234",
      "cloudWatchLogLink":"https://console.aws.amazon.com/cloudwatch/home?region=us-
east-1#logEventViewer:group=/aws-glue/crawlers;stream=test-crawler-notification",
      "state":"Failed",
      "message":"Crawler Failed"
   }
}
```

作业运行处于正在启动状态

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"STARTING",
          "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in STARTING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

作业运行处于正在运行状态

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"RUNNING",
    "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in RUNNING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

作业运行处于正在停止状态

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"STOPPING",
    "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in STOPPING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

Amazon GuardDuty 事件

有关示例 Amazon GuardDuty 事件的信息，请参阅 Amazon GuardDuty 用户指南中的使用 Amazon CloudWatch Events 监控 Amazon GuardDuty。

AWS Health 事件

以下是 AWS Personal Health Dashboard (AWS Health) 事件的格式。有关更多信息，请参阅 AWS Health 用户指南 中的使用 Amazon CloudWatch Events 管理 AWS Health 事件。

AWS Health 事件格式

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "region",
  "resources": [],
  "detail": {
    "eventArn": "arn:aws:health:region::event/id",
    "service": "service",
    "eventTypeCode": "AWS_service_code",
    "eventTypeCategory": "category",
    "startTime": "Sun, 05 Jun 2016 05:01:10 GMT",
    "endTime": "Sun, 05 Jun 2016 05:30:57 GMT",
    "eventDescription": [{
      "language": "lang-code",
      "latestDescription": "description"
    }]
    ...
  }
}
```

eventTypeCategory

事件的类别代码。可能的值为issue、accountNotification和scheduledChange。

eventTypeCode

事件类型的唯一标识符。示例包括 AWS_EC2_INSTANCE_NETWORK_MAINTENANCE_SCHEDULED 和 AWS_EC2_INSTANCE_REBOOT_MAINTENANCE_SCHEDULED。通常在 startTime 之前两周左右推送出包含 MAINTENANCE_SCHEDULED 的事件。

id

事件的唯一标识符。

service

受事件影响的 AWS 服务。例如，EC2、S3、REDSHIFT 或 RDS。

Elastic Load Balancing API 问题

```
{
  "version": "0",
  "id": "121345678-1234-1234-1234-123456789012",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "ap-southeast-2",
  "resources": [],
  "detail": {
    "eventArn": "arn:aws:health:ap-southeast-2::event/
AWS_ELASTICLOADBALANCING_API_ISSUE_90353408594353980",
    "service": "ELASTICLOADBALANCING",
    "eventTypeCode": "AWS_ELASTICLOADBALANCING_API_ISSUE",
    "eventTypeCategory": "issue",
    "startTime": "Sat, 11 Jun 2016 05:01:10 GMT",
    "endTime": "Sat, 11 Jun 2016 05:30:57 GMT",
    "eventDescription": [{
```

```
      "language": "en_US",
      "latestDescription": "A description of the event will be provided here"
  }
}
```

Amazon EC2 实例存储驱动器性能下降

```
{
  "version": "0",
  "id": "121345678-1234-1234-1234-123456789012",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "us-west-2",
  "resources": [
    "i-abcd1111"
  ],
  "detail": {
    "eventArn": "arn:aws:health:us-west-2::event/
AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED_90353408594353980",
    "service": "EC2",
    "eventTypeCode": "AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED",
    "eventTypeCategory": "issue",
    "startTime": "Sat, 05 Jun 2016 15:10:09 GMT",
    "eventDescription": [{
      "language": "en_US",
      "latestDescription": "A description of the event will be provided here"
    }],
    "affectedEntities": [{
      "entityValue": "i-abcd1111",
      "tags": {
        "stage": "prod",
        "app": "my-app"
  }
}
```

AWS KMS 事件

以下是 AWS Key Management Service (AWS KMS) 事件的示例。有关更多信息，请参阅 AWS Key Management Service Developer Guide 中的 AWS KMS 事件。

KMS CMK 轮换

AWS KMS 自动轮换了 CMK 的密钥材料。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-25T21:05:33Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
```

```
}
```

KMS 导入的密钥材料过期

AWS KMS 删除了 CMK 的过期密钥材料。

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-22T20:12:19Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

KMS CMK 删除

AWS KMS 完成了计划的 CMK 删除。

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-19T03:23:45Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Amazon Macie 事件

以下是 Amazon Macie 事件的示例。

警报已创建

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Macie Alert",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2017-04-24T22:28:49Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id/alert/alert_id",
    "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id"
```

```
  ],
  "detail": {
    "notification-type": "ALERT_CREATED",
    "name": "Scanning bucket policies",
    "tags": [
      "Custom_Alert",
      "Insider"
    ],
    "url": "https://lb00.us-east-1.macie.aws.amazon.com/111122223333/posts/alert_id",
    "alert-arn": "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id/alert/alert_id",
    "risk-score": 80,
    "trigger": {
      "rule-arn": "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id",
      "alert-type": "basic",
      "created-at": "2017-01-02 19:54:00.644000",
      "description": "Alerting on failed enumeration of large number of bucket policies",
      "risk": 8
    },
    "created-at": "2017-04-18T00:21:12.059000",
    "actor": "555566667777:assumed-role:superawesome:aroaidpldc7nsesfnheji",
    "summary": {
      "Description": "Alerting on failed enumeration of large number of bucket policies",
      "IP": {
        "34.199.185.34": 121,
        "34.205.153.2": 2,
        "72.21.196.70": 2
      },
      "Time Range": [
        {
          "count": 125,
          "start": "2017-04-24T20:23:49Z",
          "end": "2017-04-24T20:25:54Z"
        }
      ],
      "Source ARN": "arn:aws:sts::123456789012:assumed-role/RoleName",
      "Record Count": 1,
      "Location": {
        "us-east-1": 125
      },
      "Event Count": 125,
      "Events": {
        "GetBucketLocation": {
          "count": 48,
          "ISP": {
            "Amazon": 48
          }
        },
        "ListRoles": {
          "count": 2,
          "ISP": {
            "Amazon": 2
          }
        },
        "GetBucketPolicy": {
          "count": 37,
          "ISP": {
            "Amazon": 37
          },
          "Error Code": {
            "NoSuchBucketPolicy": 22
          }
        },
        "GetBucketAcl": {
          "count": 37,
          "ISP": {
            "Amazon": 37
```

```
          }
        },
        "ListBuckets": {
          "count": 1,
          "ISP": {
            "Amazon": 1
          }
        }
      },
      "recipientAccountId": {
        "123456789012": 125
      }
    }
  }
}
```

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Macie Alert",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2017-04-18T18:15:41Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id/alert/alert_id",
    "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id"
  ],
  "detail": {
    "notification-type": "ALERT_CREATED",
    "name": "Bucket is writable by all authenticated users",
    "tags": [
      "Custom_Alert",
      "Audit"
    ],
    "url": "https://lb00.us-east-1.macie.aws.amazon.com/111122223333/posts/alert_id",
    "alert-arn": "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id/alert/alert_id",
    "risk-score": 70,
    "trigger": {
      "rule-arn": "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id",
      "alert-type": "basic",
      "created-at": "2017-04-08 00:21:30.749000",
      "description": "Bucket is writable by all authenticated users",
      "risk": 7
    },
    "created-at": "2017-04-18T18:16:17.046454",
    "actor": "444455556666",
    "summary": {
      "Description": "Bucket is writable by all authenticated users",
      "Bucket": {
        "secret-bucket-name": 1
      },
      "Record Count": 1,
      "ACL": {
        "secret-bucket-name": [
          {
            "Owner": {
              "DisplayName": "bucket_owner",
              "ID": "089d2842f4b392f5c5c61f073bd2e4a37b3bb2e62659318c6960e8981648a17e"
            },
            "Grants": [
              {
                "Grantee": {
                  "Type": "Group",
```

```
                  "URI": "http://acs.amazonaws.com/groups/global/AuthenticatedUsers"
                },
                "Permission": "WRITE"
              }
            ]
          }
        ]
      },
      "Event Count": 1,
      "Timestamps": {
        "2017-01-10T22:48:06.784937": 1
      }
    }
  }
}
```

警报已更新

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Macie Alert",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2017-04-18T17:47:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id/alert/alert_id",
    "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id"
  ],
  "detail": {
    "notification-type": "ALERT_UPDATED",
    "name": "Public bucket contains high risk object",
    "tags": [
      "Custom_Alert",
      "Audit"
    ],
    "url": "https://lb00.us-east-1.macie.aws.amazon.com/111122223333/posts/alert_id",
    "alert-arn": "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id/alert/alert_id",
    "risk-score": 100,
    "trigger": {
     "rule-arn": "arn:aws:macie:us-east-1:123456789012:trigger/trigger_id",
      "alert-type": "basic",
      "created-at": "2017-04-08 00:23:39.138000",
      "description": "Public bucket contains high risk object",
      "risk": 10
    },
    "created-at": "2017-04-08T00:36:26.270000",
    "actor": "public_bucket",
    "summary": {
      "Description": "Public bucket contains high risk object",
      "Object": {
        "public_bucket/secret_key.txt": 1,
        "public_bucket/financial_summary.txt": 1
      },
      "Record Count": 2,
      "Themes": {
        "Secret Markings": 1,
        "Corporate Proposals": 1,
        "Confidential Markings": 1
      },
      "Event Count": 2,
      "DLP risk": {
        "7": 2
      },
```

```
      "Owner": {
        "bucket_owner": 2
      },
      "Timestamps": {
        "2017-04-03T16:12:53+00:00": 2
      }
    }
  }
}
```

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Macie Alert",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:macie:us-east-1:123456789012:trigger/macie/alert/alert_id",
    "arn:aws:macie:us-east-1:123456789012:trigger/macie"
  ],
  "detail": {
    "notification-type": "ALERT_UPDATED",
    "name": "Lists the instance profiles that have the specified associated IAM role, Lists
 the names of the inline policies that are embedded in the specified IAM role",
    "tags": [
      "Predictive",
      "Behavioral_Anomaly"
    ],
    "url": "https://lb00.us-east-1.macie.aws.amazon.com/111122223333/posts/alert_id",
    "alert-arn": "arn:aws:macie:us-east-1:123456789012:trigger/macie/alert/alert_id",
    "risk-score": 20,
    "created-at": "2017-04-22T03:08:35.256000",
    "actor": "123456789012:assumed-role:rolename",
    "trigger": {
      "alert-type": "predictive",
      "features": {
        "distinctEventName": {
          "name": "distinctEventName",
          "description": "Event Names executed during a user session",
          "narrative": "A sudden increase in event names utilized by a user can be an
 indicator of a change in user behavior or account risk",
          "risk": 3
        },
        "ListInstanceProfilesForRole": {
          "name": "ListInstanceProfilesForRole",
          "description": "Lists the instance profiles that have the specified associated
 IAM role",
          "narrative": "Information collection activity suggesting the start of a
 reconnaissance or exfiltration campaign",
          "anomalous": true,
          "multiplier": 8.420560747663552,
          "excession_times": [
            "2017-04-21T18:00:00Z"
          ],
          "risk": 1
        },
        "ListRolePolicies": {
          "name": "ListRolePolicies",
          "description": "Lists the names of the inline policies that are embedded in the
 specified IAM role",
          "narrative": "Information collection activity suggesting the start of a
 reconnaisance or exfiltration campaign",
```

```
          "anomalous": true,
          "multiplier": 12.017441860465116,
          "excession_times": [
            "2017-04-21T18:00:00Z"
          ],
          "risk": 2
        }
      }
    }
  }
}
```

计划的事件

下面是一个计划事件示例：

```
{
  "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2015-10-08T16:53:06Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:events:us-east-1:123456789012:rule/MyScheduledRule" ],
  "detail": {}
}
```

AWS Server Migration Service 事件

以下是 AWS Server Migration Service 事件的示例。

已删除复制作业通知

```
{
    "version": "0",
    "id": "5630992d-92cd-439f-f2a8-92c8212aee24",
    "detail-type": "Server Migration Job State Change",
    "source": "aws.sms",
    "account": "123456789012",
    "time": "2018-02-07T22:30:11Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:sms:us-west-1:123456789012:sms-job-21a64348"
    ],
    "detail": {
        "state": "Deleted",
        "replication-run-id": "N/A",
        "replication-job-id": "sms-job-21a64348",
        "version": "1.0"
    }
}
```

已完成复制作业通知

```
{
    "version": "0",
```

```
    "id": "3f9c59cc-f941-522a-be6d-f08e44ff1715",
    "detail-type": "Server Migration Job State Change",
    "source": "aws.sms",
    "account": "123456789012",
    "time": "2018-02-07T22:54:00Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:sms:us-west-1:123456789012:sms-job-2ea64347",
        "arn:aws:sms:us-west-1:123456789012:sms-job-2ea64347/sms-run-e1a64388"
    ],
    "detail": {
        "state": "Completed",
        "replication-run-id": "sms-run-e1a64388",
        "replication-job-id": "sms-job-2ea64347",
        "ami-id": "ami-746d6314",
        "version": "1.0"
    }
}
```

AWS Trusted Advisor 事件

以下是 AWS Trusted Advisor 事件的示例。有关更多信息，请参阅 AWS Support User Guide 中的使用 Amazon CloudWatch Events 监控 Trusted Advisor 检查结果。

低使用率 Amazon EC2 实例

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T20:07:49Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "check-name": "Low Utilization Amazon EC2 Instances",
    "check-item-detail": {
      "Day 1": "0.1%  0.00MB",
      "Day 2": "0.1%  0.00MB",
      "Day 3": "0.1%  0.00MB",
      "Region/AZ": "ca-central-1a",
      "Estimated Monthly Savings": "$9.22",
      "14-Day Average CPU Utilization": "0.1%",
      "Day 14": "0.1%  0.00MB",
      "Day 13": "0.1%  0.00MB",
      "Day 12": "0.1%  0.00MB",
      "Day 11": "0.1%  0.00MB",
      "Day 10": "0.1%  0.00MB",
      "14-Day Average Network I/O": "0.00MB",
      "Number of Days Low Utilization": "14 days",
      "Instance Type": "t2.micro",
      "Instance ID": "i-01234567890abcdef",
      "Day 8": "0.1%  0.00MB",
      "Instance Name": null,
      "Day 9": "0.1%  0.00MB",
      "Day 4": "0.1%  0.00MB",
      "Day 5": "0.1%  0.00MB",
      "Day 6": "0.1%  0.00MB",
      "Day 7": "0.1%  0.00MB"
    },
    "status": "WARN",
```

```
    "resource_id": "arn:aws:ec2:ca-central-1:123456789012:instance/i-01234567890abcdef",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

负载均衡器优化

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T20:07:03Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "check-name": "Load Balancer Optimization ",
    "check-item-detail": {
      "Instances in Zone a": "1",
      "Status": "Yellow",
      "Instances in Zone b": "0",
      "# of Zones": "2",
      "Region": "eu-central-1",
      "Load Balancer Name": "my-load-balance",
      "Instances in Zone e": null,
      "Instances in Zone c": null,
      "Reason": "Single AZ",
      "Instances in Zone d": null
    },
    "status": "WARN",
    "resource_id": "arn:aws:elasticloadbalancing:eu-central-1:123456789012:loadbalancer/my-
load-balancer",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

已泄露的访问密钥

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T19:38:24Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "check-name": "Exposed Access Keys",
    "check-item-detail": {
      "Case ID": "12345678-1234-1234-abcd-1234567890ab",
      "Usage (USD per Day)": "0",
      "User Name (IAM or Root)": "my-username",
      "Deadline": "1440453299248",
      "Access Key ID": "AKIAIOSFODNN7EXAMPLE",
      "Time Updated": "1440021299248",
      "Fraud Type": "Exposed",
      "Location": "www.example.com"
    },
    "status": "ERROR",
    "resource_id": "",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
```

```
    }
}
```

在 AWS 账户之间发送和接收事件

您可以设置您的 AWS 账户将事件发送到另一个 AWS 账户，或接收来自另一个账户的事件。如果两个账户属于同一个组织，或属于具有合作伙伴关系或类似关系的组织，这可能会很有用。

如果您将账户设置为发送或接收事件，则可以指定它向哪些 AWS 账户发送事件或从哪些账户接收事件。

整体过程如下所述：

- 编辑接收方 的默认事件总线 上的接收方账户权限，以允许一个或多个指定的账户 (或所有 AWS 账户) 将事件发送到接收方账户。
- 在发送方 账户中，设置一个或多个将接收方账户的默认事件总线作为目标的规则。
- 在接收方 账户中，设置一个或多个匹配来自发送方账户的事件的规则。

接收方账户在其中将权限添加到默认事件总线的 AWS 区域必须与发送方账户在其中创建向接收方账户发送事件的规则的区域相同。

从一个账户发送到另一个账户的事件将作为自定义事件向发送账户收取费用。不向接收账户收费。有关 CloudWatch Events 定价的更多信息，请参阅 Amazon CloudWatch 定价。

接收方账户可以设置一条规则，将从发送方账户收到的事件发送到第三个账户，但是这些事件不会实际发送到此第三个账户。

允许您的 AWS 账户从其他 AWS 账户接收事件

要接收其他账户的事件，您必须先编辑您的账户的默认事件总线 上的权限。默认事件总线接受来自 AWS 服务、其他授权 AWS 账户和 `PutEvents` 调用的事件。

当您编辑默认事件总线的权限以向其他 AWS 账户授予权限时，可以按账户 ID 指定账户。或者您可以选择从所有 AWS 账户接收事件。

> Warning
>
> 如果您选择从所有 AWS 账户接收事件，请注意创建仅匹配要从其他账户接收的事件的规则。要创建更安全的规则，请确保每个规则的事件模式包含一个 `account` 字段，其中包含您要从其接收事件的一个或多个账户的账户 ID。其事件模式包含账户字段的规则与从其他账户发送的事件不匹配。有关更多信息，请参阅 CloudWatch Events 中的事件模式 (p. 30)。

使用控制台允许您的账户从其他 AWS 账户接收事件

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，依次选择 Event Buses 和 Add Permission。
3. 对于 Principal，键入从其接收事件的账户的 12 位 AWS 账户 ID。要接收来自所有其他 AWS 账户的事件，请选择 Everybody(*)。
4. 选择 Add。

使用 AWS CLI 允许您的账户从其他 AWS 账户接收事件

1. 要允许一个特定 AWS 账户发送事件，请运行以下命令：

```
aws events put-permission --action events:PutEvents --statement-id MySid --
principal SenderAccountID
```

要允许所有其他 AWS 账户发送事件，请运行以下命令：

```
aws events put-permission --action events:PutEvents --statement-id MySid --principal \*
```

2. 为您的默认事件总线设置权限后，您可以选择使用 `describe-event-bus` 命令检查权限。

```
aws events describe-event-bus
```

将事件发送到另一个 AWS 账户

要将事件发送到另一个账户，可配置一个 CloudWatch Events 规则，该规则将另一个 AWS 账户的默认事件总线作为目标。该接收账户的默认事件总线也必须配置为从您的账户接收事件。

使用控制台从您的账户向另一个 AWS 账户发送事件

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create Rule。
3. 对于 Event Source，选择 Event Pattern，然后选择要发送至另一个账户的服务名称和事件类型。
4. 选择 Add Target。
5. 在下拉列表中，选择 Event bus in another AWS account。然后在 Account ID 中，键入要向其发送事件的 AWS 账户的 12 位账户 ID。
6. 在页面底部，选择配置详细信息。
7. 键入规则的名称和描述，然后选择创建规则。

使用 AWS CLI 将事件发送到另一个 AWS 账户

1. 使用 `put-rule` 命令创建一条规则，此规则应与要发送至其他账户的事件类型相匹配。
2. 将其他账户的默认事件总线作为规则的目标添加：

```
aws events put-targets --rule NameOfRuleMatchingEventsToSend --targets
 "Id"="MyId","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/default"
```

编写与来自另一个 AWS 账户的事件进行匹配的规则

如果您的账户设置为从其他 AWS 账户接收事件，则可以编写与那些事件进行匹配的规则。将规则的事件模式设置为与您从其他账户接收的事件相匹配。

除非您在规则的事件模式中指定 `Account`，否则您的账户中与您从其他账户收到的事件进行匹配的任何规则 (包括新规则和现有规则) 都将基于这些事件触发。如果您要从另一个账户接收事件，并且希望仅对从您自己的账户生成的事件模式触发规则，则必须添加 `Account` 并将您自己的账户 ID 指定为规则的事件模式。

如果您将您的 AWS 账户设置为接受来自所有 AWS 账户的事件，我们强烈建议您将 `Account` 添加到您的账户的每一个 CloudWatch Events 规则中。这可以防止账户中的规则对来自未知 AWS 账户的事件触发。在规则中指定 `Account` 字段时，可以在该字段中指定多个 AWS 账户的账户 ID。

使用控制台编写与来自另一个账户的事件进行匹配的规则

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Events 和 Create Rule。

3. 对于 Event Source，选择 Event Pattern，然后选择规则应匹配的服务名称和事件类型。
4. 选择 Event Pattern Preview 旁的 Edit。
5. 在编辑窗口中，添加一个 `Account` 行，指定发送此事件的 AWS 账户应与规则匹配。例如，编辑窗口最初显示以下内容：

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
}
```

   您可以添加以下内容以使规则与 AWS 账户 123456789012 和 111122223333 发送的 EBS 卷通知进行匹配：

```
{
  "account": [
    "123456789012","111122223333"
  ],
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
}
```

6. 编辑事件模式后，选择 Save。
7. 像往常一样完成规则的创建，在您的账户中设置一个或多个目标。

使用 AWS CLI 编写与来自另一个 AWS 账户的事件进行匹配的规则

- 使用 `put-rule` 命令，并在规则事件模式的 `Account` 字段中指定规则要匹配的其他 AWS 账户。以下示例规则与 AWS 账户 123456789012 和 111122223333 中的 Amazon EC2 实例状态更改进行匹配：

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"account\":
["123456789012", "111122223333"],\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2
 Instance State-change Notification\"]}"  --role-arn "arn:aws:iam::123456789012:role/
MyRoleForThisRule"
```

使用 PutEvents 添加事件

PutEvents 操作在一次请求中将多个事件发送到 CloudWatch Events。有关更多信息，请参阅 Amazon CloudWatch Events API 参考 中的 PutEvents 和 AWS CLI Command Reference 中的 put-events。

每个 PutEvents 请求可支持有限数目的条目。有关更多信息，请参阅 CloudWatch Events 限制 (p. 2)。PutEvents 操作将尝试按请求的自然顺序处理所有条目。在调用 PutEvents 后，每个事件均将获得由 CloudWatch Events 分配的唯一 ID。

以下示例 Java 代码将两个相同的事件发送到 CloudWatch Events：

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

PutEventsRequest request = new PutEventsRequest()
        .withEntries(requestEntry, requestEntry);

PutEventsResult result = awsEventsClient.putEvents(request);

for (PutEventsResultEntry resultEntry : result.getEntries()) {
    if (resultEntry.getEventId() != null) {
        System.out.println("Event Id: " + resultEntry.getEventId());
    } else {
        System.out.println("Injection failed with Error Code: " +
 resultEntry.getErrorCode());
    }
}
```

PutEvents 结果包含响应条目的数组。响应数组中的每个条目按自然顺序 (从请求和响应的顶部到底部) 直接与请求数组中的一个条目关联。响应 Entries 数组包含的条目数量始终与请求数组相同。

处理使用 PutEvents 时出现的失败情况

默认情况下，请求内的单个条目的失败不会中止对请求中后续条目的处理。这意味着，响应条目数组包含处理成功和不成功的条目。您必须删除处理不成功的条目并在后续调用中包括它们。

成功的结果条目包含 ID 值，不成功的结果条目包含 ErrorCode 和 ErrorMessage 值。ErrorCode 参数反映错误的类型。ErrorMessage 提供有关错误的更多详细信息。以下示例具有针对 PutEvents 请求的三个结果条目。第二个条目失败，并且反映在响应中。

示例：PutEvents 响应语法

```
{
    "FailedEntryCount": 1,
    "Entries": [
        {
            "EventId": "11710aed-b79e-4468-a20b-bb3c0c3b4860"
        },
        {   "ErrorCode": "InternalFailure",
            "ErrorMessage": "Internal Service Failure"
        },
```

```
            "EventId": "d804d26a-88db-4b66-9eaf-9a11c708ae82"
        }
    ]
}
```

处理不成功的条目可包含在后续 `PutEvents` 请求中。首先，查看 `PutEventsResult` 中的 `FailedRecordCount` 参数以确认请求中是否存在失败的记录。如果存在，则应将每个具有 `ErrorCode` (不为空) 的 `Entry` 添加到后续请求中。有关此类处理程序的示例，请参阅以下代码。

示例：PutEvents 失败处理程序

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

List<PutEventsRequestEntry> putEventsRequestEntryList = new ArrayList<>();
for (int i = 0; i < 3; i++) {
    putEventsRequestEntryList.add(requestEntry);
}

PutEventsRequest putEventsRequest = new PutEventsRequest();
putEventsRequest.withEntries(putEventsRequestEntryList);
PutEventsResult putEventsResult = awsEventsClient.putEvents(putEventsRequest);

while (putEventsResult.getFailedEntryCount() > 0) {
    final List<PutEventsRequestEntry> failedEntriesList = new ArrayList<>();
    final List<PutEventsResultEntry> PutEventsResultEntryList =
 putEventsResult.getEntries();
    for (int i = 0; i < PutEventsResultEntryList.size(); i++) {
        final PutEventsRequestEntry putEventsRequestEntry =
 putEventsRequestEntryList.get(i);
        final PutEventsResultEntry putEventsResultEntry = PutEventsResultEntryList.get(i);
        if (putEventsResultEntry.getErrorCode() != null) {
            failedEntriesList.add(putEventsRequestEntry);
        }
    }
    putEventsRequestEntryList = failedEntriesList;
    putEventsRequest.setEntries(putEventsRequestEntryList);
    putEventsResult = awsEventsClient.putEvents(putEventsRequest);
    }
```

使用 AWS CLI 发送事件

可以使用 AWS CLI 发送自定义事件。以下示例将一个自定义事件放入 CloudWatch Events 中：

```
aws events put-events \
--entries '[{"Time": "2016-01-14T01:02:03Z", "Source": "com.mycompany.myapp", "Resources":
 ["resource1", "resource2"], "DetailType": "myDetailType", "Detail": "{ \"key1\":
 \"value1\", \"key2\": \"value2\" }"}]'
```

您还可以创建文件，例如 entries.json，如下所示：

```
[
  {
    "Time": "2016-01-14T01:02:03Z",
    "Source": "com.mycompany.myapp",
```

```
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }"
  }
]
```

可以使用 AWS CLI 读取该文件中的条目并发送事件。在命令提示符处，输入：

```
aws events put-events --entries file://entries.json
```

计算 PutEvents 事件条目大小

可以使用 `PutEvents` 操作将自定义事件注入 CloudWatch Events中。您可使用 `PutEvents` 操作注入多个事件，前提是总条目大小不到 256KB。可以执行以下步骤来预先计算事件条目大小。随后，可将多个事件条目批量注入到一个请求中以提高效率。

> **Note**
>
> 已对条目施加大小限制。即使条目大小低于大小限制，也并不意味着 CloudWatch Events 中的事件也将小于此大小。相反，事件大小始终大于条目大小，因为事件的 JSON 表示形式有一些必要的字符和键。有关更多信息，请参阅 CloudWatch Events 中的事件模式 (p. 30)。

`PutEventsRequestEntry` 大小的计算方式如下：

- 如果指定 `Time` 参数，则按 14 字节来度量。
- `Source` 和 `DetailType` 参数按其 UTF-8 编码形式的字节数来度量。
- 如果指定 `Detail` 参数，则按其 UTF-8 编码形式的字节数来度量。
- 如果指定 `Resources` 参数，则每个实体按其 UTF-8 编码形式的字节数来度量。

以下示例 Java 代码计算给定 `PutEventsRequestEntry` 对象的大小：

```
int getSize(PutEventsRequestEntry entry) {
    int size = 0;
    if (entry.getTime() != null) {
        size += 14;
    }
    size += entry.getSource().getBytes(StandardCharsets.UTF_8).length;
    size += entry.getDetailType().getBytes(StandardCharsets.UTF_8).length;
    if (entry.getDetail() != null) {
        size += entry.getDetail().getBytes(StandardCharsets.UTF_8).length;
    }
    if (entry.getResources() != null) {
        for (String resource : entry.getResources()) {
            if (resource != null) {
                size += resource.getBytes(StandardCharsets.UTF_8).length;
            }
        }
    }
    return size;
}
```

将 CloudWatch Events 和接口 VPC 终端节点一起使用

如果您使用 Amazon Virtual Private Cloud (Amazon VPC) 托管 AWS 资源，则可以在您的 VPC 和 CloudWatch Events 之间建立私有连接。您可以使用此连接实现 CloudWatch Events 与您的 VPC 上的资源的通信而不用访问公共 Internet。

Amazon VPC 是一项 AWS 服务，可用来启动在虚拟网络中定义的 AWS 资源。借助 VPC，您可以控制您的网络设置，如 IP 地址范围、子网、路由表和网络网关。要将您的 VPC 连接到 CloudWatch Events，请为 CloudWatch Events 定义一个接口 VPC 终端节点。这种类型的终端节点使您能够将 VPC 连接到 AWS 服务。该终端节点提供了到 CloudWatch Events 的可靠、可扩展的连接，无需 Internet 网关、网络地址转换 (NAT) 实例或 VPN 连接。有关更多信息，请参阅 Amazon VPC 用户指南 中的什么是 Amazon VPC。

接口 VPC 终端节点由 AWS PrivateLink 提供支持，后者是一种 AWS 技术，可将弹性网络接口与私有 IP 地址结合使用来支持 AWS 服务之间的私有通信。有关更多信息，请参阅新增 – 适用于 AWS 服务的 AWS PrivateLink。

以下步骤适用于 Amazon VPC 的用户。有关更多信息，请参阅 Amazon VPC 用户指南 中的入门。

可用性

目前，CloudWatch Events 以下区域支持 VPC 终端节点：

- 美国东部（俄亥俄州）
- 美国东部（弗吉尼亚北部）
- 美国西部（加利福尼亚北部）
- 美国西部（俄勒冈）
- 亚太地区（孟买）
- 亚太区域（首尔）
- 亚太区域（新加坡）
- 亚太区域（悉尼）
- 亚太区域（东京）
- 加拿大 (中部)
- 欧洲（法兰克福）
- 欧洲（爱尔兰）
- 欧洲 (伦敦)
- 欧洲 (巴黎)
- 南美洲（圣保罗）

为 CloudWatch Events 创建 VPC 终端节点

要开始将您的 VPC 与 CloudWatch Events 一起使用，请为 CloudWatch Events 创建接口 VPC 终端节点。有关更多信息，请参阅 Amazon VPC 用户指南中的创建接口终端节点。

您不需要更改 CloudWatch Events 的设置。CloudWatch Events 使用公有终端节点或私有接口 VPC 终端节点 (二者中在使用中的那个) 调用其他 AWS 服务。例如，如果您为 CloudWatch Events 创建了一个接口 VPC 终端节点，并且您已经有一个在触发后向 Amazon SNS 发送通知的 CloudWatch Events 规则，通知将开始流过接口 VPC 终端节点。

Amazon CloudWatch Events 的身份验证和访问控制

访问 Amazon CloudWatch Events 时需要有 AWS 可以用来验证您的请求的凭证。这些凭证必须有权访问 AWS 资源，例如从其他 AWS 资源检索事件数据。下面几节提供详细的信息来说明如何使用 AWS Identity and Access Management (IAM) 和 CloudWatch Events 控制谁能访问您的资源，从而对这些资源进行保护：

身份验证

您可以下面任一类型的身份访问 AWS：

- AWS 账户根用户 – 注册 AWS 时，您需要提供与您的 AWS 账户关联的电子邮件地址和密码。这些是您的根凭证，它们提供对您所有 AWS 资源的完全访问权限。

  Important

  出于安全考虑，我们建议您仅使用根凭证创建管理员，它是对您的 AWS 账户具有完全访问权限的 IAM 用户。随后，您可以使用此管理员来创建具有有限权限的其他 IAM 用户和角色。有关更多信息，请参阅 IAM 用户指南 中的 IAM 最佳实践和创建管理员用户和组。
- IAM 用户 - IAM 用户就是您的 AWS 账户中的一种身份，它具有特定的自定义权限 (例如，用于在 CloudWatch Events 中向目标发送事件数据的权限)。您可以使用 IAM 用户名和密码来登录以保护 AWS 网页，如 AWS 管理控制台、AWS 开发论坛或 AWS Support Center。

  除了用户名和密码之外，您还可以为每个用户生成访问密钥。在通过多个软件开发工具包之一或使用 AWS Command Line Interface (AWS CLI) 以编程方式访问 AWS 服务时，可以使用这些密钥。SDK 和 AWS CLI 工具使用访问密钥对您的请求进行加密签名。如果您不使用 AWS 工具，则必须自行对请求签名。CloudWatch Events supports 签名版本 4，后者是一种用于对入站 API 请求进行身份验证的协议。有关验证请求的更多信息，请参阅 AWS General Reference 中的签名版本 4 签名流程。

- IAM 角色 – IAM 角色是可在账户中创建的另一种具有特定权限的 IAM 身份。它类似于 IAM 用户，但未与特定人员关联。利用 IAM 角色，您可以获得可用于访问 AWS 服务和资源的临时访问密钥。具有临时凭证的 IAM 角色在以下情况下很有用：

  - 联合用户访问 - 您可以不创建 IAM 用户，而是使用来自 AWS Directory Service、您的企业用户目录或 Web 身份提供商 (IdP) 的既有身份。他们被称为联合身份用户。在通过身份提供商请求访问权限时，AWS 将为联合身份用户分配角色。有关联合身份用户的更多信息，请参阅 IAM 用户指南 中的联合身份用户和角色。

  - 跨账户访问 – 可以使用您账户中的 IAM 角色向另一个 AWS 账户授予对您账户的资源的访问权限。有关示例，请参阅 &guide-iam-user; 中的教程：使用 IAM 角色委派跨 AWS 账户的访问权限。

- AWS 服务访问 - 您可以在您的账户中使用 IAM 角色为 AWS 服务授予访问您的账户的资源所需的权限。例如，您可以创建一个角色，此角色允许 Amazon Redshift 代表您访问 Amazon S3 存储桶，然后将存储在该存储桶中的数据加载到 Amazon Redshift 群集中。有关更多信息，请参阅 IAM 用户指南 中的创建向 AWS 服务委派权限的角色。

- 在 Amazon EC2 上运行的应用程序 – 您不用将访问密钥存储在 EC2 实例中以供实例上运行的应用程序使用并发出 AWS API 请求，而是可以使用 IAM 角色管理这些应用程序的临时凭证。要将 AWS 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅 IAM 用户指南 中的对 Amazon EC2 上的应用程序使用角色。

访问控制

您可以使用有效的凭证来对自己的请求进行身份验证，但您还必须拥有权限才能创建或访问 CloudWatch Events 资源。例如，您必须拥有调用与您的 CloudWatch Events 规则关联的 AWS Lambda、Amazon Simple Notification Service (Amazon SNS) 和 Amazon Simple Queue Service (Amazon SQS) 目标的权限。

下面几节介绍如何管理 CloudWatch Events 的权限。我们建议您先阅读概述。

管理您的 CloudWatch Events 资源的访问权限概述

每个 AWS 资源都归某个 AWS 账户所有，创建和访问资源的权限由权限策略进行管理。账户管理员可以向 IAM 身份（即：用户、组和角色）挂载权限策略，某些服务（如 AWS Lambda）也支持向资源挂载权限策略。

> Note
>
> 账户管理员 (或管理员 IAM 用户) 是具有管理员权限的用户。有关更多信息，请参阅 IAM 用户指南 中的 IAM 最佳实践。

在授予权限时，您要决定谁获得权限，获得对哪些资源的权限，以及您允许对这些资源执行的具体操作。

主题

CloudWatch Events 资源和操作

在 CloudWatch Events 中，规则是主要资源。CloudWatch Events 支持可与主资源一起使用的其他资源，例如事件。这些资源称为子资源。这些资源和子资源具有与其关联的唯一 Amazon 资源名称 (ARN)。有关

ARN 的详细信息，请参阅 Amazon Web Services 一般参考 中的 Amazon 资源名称 (ARN) 和 AWS 服务命名空间。

资源类型	ARN 格式
规则	`arn:aws:events:region:account:rule/rule-name`
所有 CloudWatch Events 资源	`arn:aws:events:*`
指定账户在指定地区拥有的所有 CloudWatch Events 资源	`arn:aws:events:region:account:*`

Note

AWS 中的大多数服务将 ARN 中的冒号 (:) 或正斜杠 (/) 视为相同的字符。不过，CloudWatch Events 在事件模式和规则中使用精确匹配。请在创建事件模式时务必使用正确的 ARN 字符，以使其匹配需要匹配的事件中的 ARN 语法。

例如，您可以使用某个特定规则 (*myRule*) 的 ARN 在语句中指定该规则，如下所示：

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/myRule"
```

还可以使用星号 (*) 通配符指定属于特定账户的所有规则，如下所示：

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/*"
```

要指定所有资源，或者如果特定 API 操作不支持 ARN，请在 Resource 元素中使用星号 (*) 通配符，如下所示：

```
"Resource": "*"
```

有些 CloudWatch Events API 操作接受多个资源 (例如 `PutTargets`)。要在单个语句中指定多种资源，请使用逗号将它们隔开，如下所示：

```
"Resource": ["arn1", "arn2"]
```

CloudWatch Events 提供一组操作用来处理 CloudWatch Events 资源。有关可用操作的列表，请参阅 CloudWatch Events 权限参考 (p. 103)。

了解资源所有权

AWS 账户对在该账户下创建的资源具有所有权，而无论创建资源的人员是谁。具体而言，资源所有者是对资源创建请求进行身份验证的委托人实体 (即 AWS 账户根用户、IAM 用户或 IAM 角色) 的 AWS 账户。以下示例说明了它的工作原理：

- 如果您使用 AWS 账户的根用户凭证创建规则，则您的 AWS 账户即为该 CloudWatch Events 资源的所有者。
- 如果您在您的 AWS 账户中创建 IAM 用户并对该用户授予创建 CloudWatch Events 资源的权限，则该用户可以创建 CloudWatch Events 资源。但是，该用户所属的 AWS 账户拥有这些 CloudWatch Events 资源。

- 如果您在您的 AWS 账户中创建具有创建 CloudWatch Events 资源的权限的 IAM 角色，则能够担任该角色的任何人都可以创建 CloudWatch Events 资源。该角色所属的 AWS 账户拥有这些 CloudWatch Events 资源。

管理对资源的访问

权限策略 规定谁可以访问哪些内容。下一节介绍创建权限策略时的可用选项。

> Note
>
> 本节讨论如何在 CloudWatch Events 范围内使用 IAM。它不提供有关 IAM 服务的详细信息。有关完整的 IAM 文档，请参阅什么是 IAM？（在 IAM 用户指南 中）。有关 IAM 策略语法和说明的信息，请参阅 IAM 用户指南 中的 IAM 策略参考。

挂载到 IAM 身份的策略称作基于身份 的策略 (IAM 策略)，而挂载到资源的策略称作基于资源的策略。CloudWatch Events 同时支持基于身份的策略 (IAM 策略) 和基于资源的策略。

主题

基于身份的策略 (IAM 策略)

您可以向 IAM 身份挂载策略。例如，您可以执行以下操作：

- 将权限策略附加到您的账户中的用户或组 - 要授予查看 CloudWatch 控制台中的规则的用户权限，您可以将权限策略附加到用户或用户所属的组。
- 向角色挂载权限策略 (授予跨账户权限) - 您可以向 IAM 角色挂载基于身份的权限策略，以授予跨账户的权限。例如，账户 A 中的管理员可以创建一个角色，向另一 AWS 账户（如账户 B）或某项 AWS 服务授予跨账户权限，如下所述：
  1. 账户 A 管理员可以创建一个 IAM 角色，然后向该角色挂载权限策略，授予其访问账户 A 中资源的权限。
  2. 账户 A 管理员可以向角色挂载信任策略，将账户 B 标识为能够担任该角色的委托人。
  3. 之后，账户 B 管理员可以委派权限，指派账户 B 中的任何用户担任该角色。这样，账户 B 中的用户就可以创建或访问账户 A 中的资源了。如果您需要授予 AWS 服务担任该角色所需的权限，则信任策略中的委托人也可以是 AWS 服务委托人。

  有关使用 IAM 委派权限的更多信息，请参阅 IAM 用户指南 中的访问权限管理。

可以创建特定的 IAM 策略来限制您账户中的用户有权访问的调用和资源，然后将这些策略与 IAM 用户关联。有关创建 IAM 角色和探索 CloudWatch Events 的示例 IAM 策略语句的更多信息，请参阅 管理您的 CloudWatch Events 资源的访问权限概述 (p. 90)。

基于资源的策略 (IAM 策略)

在 CloudWatch Events 中触发一个规则时，将调用与该规则关联的所有目标。调用是指调用 AWS Lambda 函数，发布到 Amazon SNS 主题并将事件中继到 Kinesis 流。为了能对您拥有的资源执行 API 调用，CloudWatch Events 需要相应权限。对于 Lambda、Amazon SNS 和 Amazon SQS 资源，CloudWatch Events 依赖基于资源的策略。对于 Kinesis 流，CloudWatch Events 依赖 IAM 角色。

有关如何创建 IAM 角色和探索 CloudWatch Events 的基于资源的策略语句示例的更多信息，请参阅 使用 CloudWatch Events 的基于资源的策略 (p. 100)。

指定策略元素：操作、效果和委托人

对于每种 CloudWatch Events 资源，该服务都定义了一组 API 操作。为授予这些 API 操作的权限，CloudWatch Events 定义了一组您可以在策略中指定的操作。某些 API 操作可能需要多个操作的权限才能执行 API 操作。有关资源和 API 操作的更多信息，请参阅 CloudWatch Events 资源和操作 (p. 90) 和 CloudWatch Events 权限参考 (p. 103)。

以下是基本的策略元素：

- Resource - 您使用 Amazon 资源名称 (ARN) 来标识策略应用到的资源。有关更多信息，请参阅 CloudWatch Events 资源和操作 (p. 90)。
- Action - 您可以使用操作关键字标识要允许或拒绝的资源操作。例如，`events:Describe` 权限允许执行 `Describe` 操作的用户权限。
- Effect - 用于指定当用户请求特定操作时的效果 (可以是允许或拒绝)。如果没有显式授予 (允许) 对资源的访问权限，则隐式拒绝访问。您也可显式拒绝对资源的访问，这样可确保用户无法访问该资源，即使有其他策略授予了访问权限的情况下也是如此。
- Principal - 在基于身份的策略 (IAM 策略) 中，附加了策略的用户是隐式委托人。对于基于资源的策略，您可以指定要接收权限的用户、账户、服务或其他实体 (仅适用于基于资源的策略)。

要了解有关 IAM 策略语法和说明的更多信息，请参阅 IAM 用户指南 中的 AWS IAM 策略参考。

有关显示所有 CloudWatch Events API 操作及其适用资源的表，请参阅 CloudWatch Events 权限参考 (p. 103)。

在策略中指定条件

当您授予权限时，可使用访问策略语言来指定规定策略何时生效的条件。例如，您可能希望策略仅在特定日期后应用。有关使用策略语言指定条件的更多信息，请参阅 IAM 用户指南 中的条件。

要表示条件，您可以使用预定义的条件键。存在 AWS 范围内的条件键和特定于 CloudWatch Events 的键，您可以根据需要使用。有关 AWS 范围内的键的完整列表，请参阅 IAM 用户指南 中的条件的可用键。有关特定于 CloudWatch Events 的键的完整列表，请参阅使用 IAM 策略条件实现精细访问控制 (p. 105)。

为 CloudWatch Events 使用基于身份的策略 (IAM 策略)

本主题提供了基于身份的策略的示例，在这些策略中，账户管理员可以向 IAM 身份 (即：用户、组和角色) 挂载权限策略。

下面显示了允许用户将事件数据放入 Kinesis 的权限策略的示例。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchEventsInvocationAccess",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

本主题的各个部分涵盖以下内容：

主题

使用 CloudWatch 控制台所需的权限

用户要能够使用 CloudWatch 控制台中的 CloudWatch Events，则必须拥有一组为其 AWS 账户描述其他 AWS 资源的最低权限。要使用 CloudWatch 控制台中的 CloudWatch Events，您必须拥有以下服务的权限：

- 自动化
- Amazon EC2 Auto Scaling
- CloudTrail
- CloudWatch
- CloudWatch Events
- IAM
- Kinesis
- Lambda
- Amazon SNS
- Amazon SWF

如果创建比必需的最低权限更为严格的 IAM 策略，对于附加了该 IAM 策略的用户，控制台将无法按预期正常运行。为确保这些用户仍可使用 CloudWatch 控制台，同时向用户附加 `CloudWatchEventsReadOnlyAccess` 托管策略，请参阅 适用于 CloudWatch Events 的 AWS 托管 (预定义) 策略 (p. 95)。

对于只需要调用 AWS CLI 或 CloudWatch API 的用户，您无需为其提供最低控制台权限。

下面列出了使用 CloudWatch 控制台所需的一整套权限：

- `automation:CreateAction`
- `automation:DescribeAction`
- `automation:UpdateAction`
- `autoscaling:DescribeAutoScalingGroups`
- `cloudtrail:DescribeTrails`
- `ec2:DescribeInstances`
- `ec2:DescribeVolumes`
- `events:DeleteRule`
- `events:DescribeRule`
- `events:DisableRule`
- `events:EnableRule`

- `events:ListRuleNamesByTarget`
- `events:ListRules`
- `events:ListTargetsByRule`
- `events:PutEvents`
- `events:PutRule`
- `events:PutTargets`
- `events:RemoveTargets`
- `events:TestEventPattern`
- `iam:ListRoles`
- `kinesis:ListStreams`
- `lambda:AddPermission`
- `lambda:ListFunctions`
- `lambda:RemovePermission`
- `sns:GetTopicAttributes`
- `sns:ListTopics`
- `sns:SetTopicAttributes`
- `swf:DescribeAction`
- `swf:ReferenceAction`
- `swf:RegisterAction`
- `swf:RegisterDomain`
- `swf:UpdateAction`

适用于 CloudWatch Events 的 AWS 托管 (预定义) 策略

AWS 通过提供由 AWS 创建和管理的独立 IAM 策略来解决很多常用案例。托管策略可授予常用案例的必要权限，因此，您可以免去调查都需要哪些权限的工作。有关更多信息，请参阅 IAM 用户指南 中的 AWS 托管策略。

以下 AWS 托管策略 (您可以将它们挂载到自己账户中的用户) 是特定于 CloudWatch Events 的：

- CloudWatchEventsFullAccess - 授予对 CloudWatch Events 的完全访问权限。
- CloudWatchEventsInvocationAccess - 允许 CloudWatch Events 将事件中继到您账户的 Amazon Kinesis Data Streams 中的流。
- CloudWatchEventsReadOnlyAccess - 授予对 CloudWatch Events 的只读访问权限。
- CloudWatchEventsBuiltInTargetExecutionAccess - 允许 CloudWatch Events 中的内置目标代表您执行 Amazon EC2 操作。

用于发送事件的 IAM 角色

为了让 CloudWatch Events 将事件中继到您的 Kinesis 流目标，您必须创建 IAM 角色。

创建用于发送 CloudWatch Events 的 IAM 角色;

1. 通过以下网址打开 IAM 控制台：https://console.amazonaws.cn/iam/。

2. 按照 IAM 用户指南 中的创建角色以向 AWS 服务委派权限中的步骤创建 IAM 角色。按步骤创建角色时，请执行以下操作：

   - 在 Role Name 中，使用在 AWS 账户内唯一的名称 (例如，CloudWatchEventsSending)。
   - 在 Select Role Type 中，选择 AWS Service Roles，然后选择 Amazon CloudWatch Events。这会授予代入该角色的 CloudWatch Events 权限。
   - 在 Attach Policy 中，选择 CloudWatchEventsInvocationAccess。

此外，您还可以创建自己的自定义 IAM 策略，以授予 CloudWatch Events 操作和资源的相关权限。您可以将这些自定义策略挂载到需要这些权限的 IAM 用户或组。有关 IAM 策略的详细信息，请参阅 IAM 用户指南 中的 IAM 策略概述。有关管理和创建自定义 IAM 策略的详细信息，请参阅 IAM 用户指南 中的管理 IAM 策略。

CloudWatch Events 访问特定目标所需的权限

为了让 CloudWatch Events 访问特定目标，您必须指定访问该目标的 IAM 角色，且该角色必须附加了特定策略。

如果目标是 Kinesis 流，则用于将事件数据发送到该目标的角色必须包含以下策略：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

如果目标是 Run Command 且您正在为命令指定一个或多个 InstanceIds 值，则您指定的角色必须包含以下策略：

```
{
     "Version": "2012-10-17",
     "Statement": [
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ec2:{{region}}:{{accountId}}:instance/[[instanceIds]]",
                     "arn:aws:ssm:{{region}}:*:document/{{documentName}}"
                ]
           }
     ]
}
```

如果目标是 Run Command 且您正在为命令指定一个或多个标签，则您指定的角色必须包含以下策略：

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

```
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ec2:{{region}}:{{accountId}}:instance/*"
                ],
                "Condition": {
                    "StringEquals": {
                        "ec2:ResourceTag/*": [
                            "[[tagValues]]"
                        ]
                    }
                }
            },
            {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ssm:{{region}}:*:document/{{documentName}}"
                ]
            }
     ]
}
```

如果目标是 Step Functions 状态机，则您指定的角色必须包含以下策略：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "states:StartExecution" ],
            "Resource": [ "arn:aws:states:*:*:stateMachine:*" ]
        }
     ]
}
```

如果目标是 ECS 任务，则您指定的角色必须包含以下策略：

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ecs:RunTask"
        ],
        "Resource": [
            "arn:aws:ecs:*:{{account-id}}:task-definition/{{task-definition-name}}"
        ],
        "Condition": {
            "ArnLike": {
                "ecs:cluster": "arn:aws:ecs:*:{{account-id}}:cluster/{{cluster-name}}"
            }
        }
    }]
}
```

客户托管策略示例

本节的用户策略示例介绍如何授予各 CloudWatch Events 操作的权限。当您使用 CloudWatch Events API、AWS 软件开发工具包或 AWS CLI 时，可以使用这些策略。

Note

所有示例都使用 美国西部（俄勒冈）区域 (us-west-2) 和虚构的账户 ID。

您可以使用列出的以下示例 IAM 策略来限制 IAM 用户和角色对 CloudWatch Events 的访问。

示例

示例 1：CloudWatchEventsBuiltInTargetExecutionAccess

以下策略允许 CloudWatch Events 中的内置目标代表您执行 Amazon EC2 操作。

Important

只能在 AWS 管理控制台中创建带内置目标的规则。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchEventsBuiltInTargetExecutionAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:RebootInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "ec2:CreateSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

示例 2：CloudWatchEventsInvocationAccess

以下策略允许 CloudWatch Events 将事件中继到您账户的 Kinesis 流中的流。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchEventsInvocationAccess",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

示例 3：CloudWatchEventsConsoleAccess

以下策略确保 IAM 用户可使用 CloudWatch Events 控制台。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchEventsConsoleAccess",
            "Effect": "Allow",
            "Action": [
                "automation:CreateAction",
                "automation:DescribeAction",
                "automation:UpdateAction",
                "autoscaling:DescribeAutoScalingGroups",
                "cloudtrail:DescribeTrails",
                "ec2:DescribeInstances",
                "ec2:DescribeVolumes",
                "events:*",
                "iam:ListRoles",
                "kinesis:ListStreams",
                "lambda:AddPermission",
                "lambda:ListFunctions",
                "lambda:RemovePermission",
                "sns:GetTopicAttributes",
                "sns:ListTopics",
                "sns:SetTopicAttributes",
                "swf:DescribeAction",
                "swf:ReferenceAction",
                "swf:RegisterAction",
                "swf:RegisterDomain",
                "swf:UpdateAction"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMPassRoleForCloudWatchEvents",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
                "arn:aws:iam::*:role/AWS_Events_Actions_Execution"
            ]
        }
    ]
}
```

示例 4：CloudWatchEventsFullAccess

以下策略允许通过 AWS CLI 和 SDK 对 CloudWatch Events 执行操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchEventsFullAccess",
            "Effect": "Allow",
            "Action": "events:*",
            "Resource": "*"
        },
        {
            "Sid": "IAMPassRoleForCloudWatchEvents",
```

```
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/AWS_Events_Invoke_Targets"
        }
    ]
}
```

示例 5：CloudWatchEventsReadOnlyAccess

以下策略提供了对 CloudWatch Events 的只读访问权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchEventsReadOnlyAccess",
            "Effect": "Allow",
            "Action": [
                "events:Describe*",
                "events:List*",
                "events:TestEventPattern"
            ],
            "Resource": "*"
        }
    ]
}
```

使用 CloudWatch Events 的基于资源的策略

在 CloudWatch Events 中触发一个规则时，将调用与该规则关联的所有目标。调用是指调用 AWS Lambda 函数，发布到 Amazon SNS 主题并将事件中继到 Kinesis 流。为了能对您拥有的资源执行 API 调用，CloudWatch Events 需要相应权限。对于 Lambda、Amazon SNS 和 Amazon SQS 资源，CloudWatch Events 依赖基于资源的策略。对于 Kinesis 流，CloudWatch Events 依赖 IAM 角色。

您可以使用下列权限调用与您的 CloudWatch Events 规则相关联的目标。下面的过程使用 AWS CLI 将权限添加到您的目标。有关如何安装和配置 AWS CLI 的信息，请参阅 AWS Command Line Interface 用户指南中的使用 AWS 命令行界面进行设置。

主题

AWS Lambda 权限

要使用 CloudWatch Events 规则调用您的 AWS Lambda 函数，可将以下权限添加到您的 Lambda 函数的策略中。

```
{
  "Effect": "Allow",
  "Action": "lambda:InvokeFunction",
  "Resource": "arn:aws:lambda:region:account-id:function:function-name",
  "Principal": {
    "Service": "events.amazonaws.com"
```

```
  },
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
    }
  },
  "Sid": "TrustCWEToInvokeMyLambdaFunction"
}
```

添加允许 CloudWatch Events 调用 Lambda 函数的权限

- 在命令提示符处，输入以下命令：

```
aws lambda add-permission --statement-id "TrustCWEToInvokeMyLambdaFunction" \
--action "lambda:InvokeFunction" \
--principal "events.amazonaws.com" \
--function-name "arn:aws:lambda:region:account-id:function:function-name" \
--source-arn "arn:aws:events:region:account-id:rule/rule-name"
```

有关设置允许 CloudWatch Events 调用 Lambda 函数的权限的更多信息，请参阅 AWS Lambda Developer Guide中的 AddPermission 和使用 Lambda 处理计划的事件。

Amazon SNS 权限

要允许 CloudWatch Events 发布 Amazon SNS 主题，请使用 `aws sns get-topic-attributes` 和 `aws sns set-topic-attributes` 命令。

添加允许 CloudWatch Events 发布 SNS 主题的权限

1. 首先，列出 SNS 主题属性。在命令提示符处输入以下命令：

```
aws sns get-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
```

该命令返回 SNS 主题的所有属性。以下示例显示新创建的 SNS 主题的结果。

```
{
    "Attributes": {
        "SubscriptionsConfirmed": "0",
        "DisplayName": "",
        "SubscriptionsDeleted": "0",
        "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":
{\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries
\":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear\"},
\"disableSubscriptionOverrides\":false}}",
        "Owner": "account-id",
        "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
\",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",\"SNS:Subscribe
\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\",\"SNS:Receive\"],\"Resource
\":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals\":
{\"AWS:SourceOwner\":\"account-id\"}}}]}",
        "TopicArn": "arn:aws:sns:region:account-id:topic-name",
        "SubscriptionsPending": "0"
    }
}
```

2. 下一步，将以下语句转换为字符串并将其添加到“Policy”属性内部的“Statement”集合中。

```
{
  "Sid": "TrustCWEToPublishEventsToMyTopic",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:region:account-id:topic-name"
}
```

在将语句转换为字符串后，它应如下所示：

```
{\"Sid\":\"TrustCWEToPublishEventsToMyTopic\",\"Effect\":\"Allow\",\"Principal\":
{\"Service\":\"events.amazonaws.com\"},\"Action\":\"sns:Publish\",\"Resource\":
\"arn:aws:sns:region:account-id:topic-name\"}
```

3. 在您将语句字符串添加到语句集合后，使用 `aws sns set-topic-attributes` 命令设置新的策略。

```
aws sns set-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name" \
--attribute-name Policy \
--attribute-value "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
\",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",\"SNS:Subscribe
\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\",\"SNS:Receive\"],\"Resource
\":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals\":
{\"AWS:SourceOwner\":\"account-id\"}}}, {\"Sid\":\"TrustCWEToPublishEventsToMyTopic\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},\"Action\":
\"sns:Publish\",\"Resource\":\"arn:aws:sns:region:account-id:topic-name\"}]}"
```

有关更多信息，请参阅 Amazon Simple Notification Service API Reference中的 SetTopicAttributes 操作。

Amazon SQS 权限

要允许 CloudWatch Events 规则调用 Amazon SQS 队列，请使用 `aws sqs get-queue-attributes` 和 `aws sqs set-queue-attributes` 命令。

添加允许 CloudWatch Events 规则调用 SQS 队列的权限

1. 首先，列出 SQS 队列属性。在命令提示符处输入以下命令：

```
aws sqs get-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attribute-names Policy
```

对于新创建的 SQS 队列，默认情况下其策略为空。除了添加语句之外，您还需要创建包含此语句的策略。

2. 以下语句允许 CloudWatch Events 向 SQS 队列发送消息：

```
{
  "Sid": "TrustCWEToSendEventsToMyQueue",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
  "Action": "sqs:SendMessage",
```

```
  "Resource": "arn:aws:sqs:region:account-id:queue-name",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
    }
  }
}
```

3. 然后，将上述语句转换成字符串。在将策略转换为字符串后，它应如下所示：

```
{\"Sid\": \"TrustCWEToSendEventsToMyQueue\", \"Effect\": \"Allow\", \"Principal\":
 {\"AWS\": \"events.amazonaws.com\"}, \"Action\": \"sqs:SendMessage\", \"Resource
\": \"arn:aws:sqs:region:account-id:queue-name\", \"Condition\": {\"ArnEquals\":
 {\"aws:SourceArn\": \"arn:aws:events:region:account-id:rule/rule-name\"}}
```

4. 创建包含以下内容的、名为 set-queue-attributes.json 的文件：

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"arn:aws:sqs:region:account-
id:queue-name/SQSDefaultPolicy\",\"Statement\":[{\"Sid\":
 \"TrustCWEToSendEventsToMyQueue\", \"Effect\": \"Allow\", \"Principal\": {\"AWS
\": \"events.amazonaws.com\"}, \"Action\": \"sqs:SendMessage\", \"Resource\":
 \"arn:aws:sqs:region:account-id:queue-name\", \"Condition\": {\"ArnEquals\":
 {\"aws:SourceArn\": \"arn:aws:events:region:account-id:rule/rule-name\"}}}]}"
}
```

5. 使用 set-queue-attributes.json 文件作为输入来设置策略属性。在命令提示符下，输入：

```
aws sqs set-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attributes file://set-queue-attributes.json
```

   如果 SQS 队列已有一个策略，您需要复制原始策略，并将其与 set-queue-attributes.json 文件中的新语句组合，然后运行上述命令来更新策略。

有关更多信息，请参阅 Amazon Simple Queue Service 开发人员指南中的 Amazon SQS 策略示例。

CloudWatch Events 权限参考

在设置 访问控制 (p. 90) 和编写您可挂载到 IAM 身份的权限策略 (基于身份的策略) 时，可以使用下表作为参考。此表列出每个 CloudWatch Events API 操作及您可授予执行该操作的权限的对应操作。可在策略的 `Action` 字段中指定操作，在策略的 `Resource` 字段中指定通配符 (*) 作为资源值。

您可以在 CloudWatch Events 策略中使用 AWS 范围的条件键来表达条件。有关 AWS 范围内的密钥的完整列表，请参阅 IAM 用户指南 中的可用密钥。

> **Note**
>
> 要指定操作，请在 API 操作名称之前使用 `events:` 前缀。例如：`events:PutRule`、`events:EnableRule` 或 `events:*` (针对所有 CloudWatch Events 操作)。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": ["events:action1", "events:action2"]
```

您也可以使用通配符指定多项操作。例如，您可以指定名称以单词“Put”开头的所有操作，如下所示：

```
"Action": "events:Put*"
```

要指定所有 CloudWatch Events API 操作，请使用 * 通配符，如下所示：

```
"Action": "events:*"
```

下面列出了 IAM 策略中可指定用于 CloudWatch Events 的操作。

CloudWatch Events API 操作和必需的操作权限

CloudWatch Events API 操作	所需权限 (API 操作)
DeleteRule	`events:DeleteRule` 删除规则所必需的。
DescribeEventBus	`events:DescribeEventBus` 列出可以将事件写入当前账户的事件总线的 AWS 账户所必需的。
DescribeRule	`events:DescribeRule` 列出有关规则的详细信息所必需的。
DisableRule	`events:DisableRule` 禁用规则所必需的。
EnableRule	`events:EnableRule` 启用规则所必需的。
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` 列出与目标关联的规则所必需的。
ListRules	`events:ListRules` 列出您账户中的所有规则所必需的。
ListTargetsByRule	`events:ListTargetsByRule` 列出与规则关联的所有目标所必需的。
PutEvents	`events:PutEvents` 添加可匹配到规则的自定义活动所必需的。
PutPermission	`events:PutPermission` 向另一个账户授予将事件写入此账户的默认事件总线的权限所必需的。
PutRule	`events:PutRule` 创建或更新规则所必需的。
PutTargets	`events:PutTargets` 将目标添加到规则所必需的。

CloudWatch Events API 操作	所需权限 (API 操作)
RemovePermission	`events:RemovePermission` 撤销另一个账户拥有的将事件写入此账户的默认事件总线的权限所必需的。
RemoveTargets	`events:RemoveTargets` 从规则中删除目标所必需的。
TestEventPattern	`events:TestEventPattern` 针对给定事件测试事件模式所必需的。

使用 IAM 策略条件实现精细访问控制

当您授予权限时，可使用 IAM 策略语言来指定规定策略何时生效的条件。在策略语句中，您可以选择性指定控制策略生效时间的条件。每个条件都包含一个或多个键值对。条件键不区分大小写。例如，您可能希望策略仅在特定日期后应用。

如果您指定了多个条件或在单一条件中指定了多个密钥，则将使用逻辑 AND 操作对其进行评估。如果您在单一条件中指定了一个具有多个值的密钥，则将使用逻辑 OR 操作对其进行评估。必须匹配所有条件才能授予权限。

在指定条件时，您也可使用占位符。有关更多信息，请参阅 IAM 用户指南 中的策略变量。有关使用 IAM 策略语言指定条件的更多信息，请参阅 IAM 用户指南 中的条件。

默认情况下，IAM 用户和角色无法访问您的账户中的事件。要使用事件，用户必须获得 `PutRule` API 操作的授权。如果您允许 IAM 用户或角色对其策略执行 `events:PutRule` 操作，则他们将能够创建匹配特定事件的规则。您必须向规则添加目标，否则，没有目标的规则除了在匹配传入事件时发布 CloudWatch 指标之外，不会执行任何操作。您的 IAM 用户或角色必须拥有 `events:PutTargets` 操作的权限。

可通过以下方式来限制对事件的访问：将授权的范围限定为事件的特定源和类型 (使用 `events:source` 和 `events:detail-type` 条件键)。可以在 IAM 用户或角色的策略语句中提供条件，允许其创建仅匹配一组特定的源和详细类型的规则。有关所有条件键值及其适用的 CloudWatch Events 操作和资源的列表，请参阅使用 IAM 策略条件实现精细访问控制 (p. 105)。

同样，通过在策略语句中设置条件，您可以决定您账户中的哪些特定资源可由 IAM 用户或角色添加到规则中 (使用 `events:TargetArn` 条件键)。例如，如果您在账户中打开 CloudTrail 并且您有 CloudTrail 流，则您的账户中的用户也可通过 CloudWatch Events 使用 CloudTrail 事件。如果您希望用户使用 CloudWatch Events 并访问除 CloudTrail 事件之外的所有其他事件，则可添加有关 `PutRule` API 操作的拒绝语句以及一个条件，使该用户或角色创建的任何规则都无法匹配 CloudTrail 事件类型。

对于 CloudTrail 事件，可限制对原始 API 调用源自的特定委托人的访问 (使用 `events:detail.userIdentity.principalId` 条件键)。例如，您可以允许用户查看所有 CloudTrail 事件，但您的账户中用于审计或取证的某个特定 IAM 角色创建的事件除外。

条件键	键值对	评估类型
`events:source`	`"events:source":"`*`source`*` "` 其中，*`source`* 为事件的 source 字段的文字字符串，例如 `"aws.ec2"` 和 `"aws.s3"`。要查看 *`source`* 的更多可能的值，请参阅每个支持服	Source，Null

条件键	键值对	评估类型
	务的 CloudWatch Events 事件示例 (p. 34)中的示例事件。	
events:detail-type	"events:detail-type":"*detail-type* " 其中，*detail-type* 为事件的 detail-type 字段的文字字符串，例如 "AWS API Call via CloudTrail" 和 "EC2 Instance State-change Notification"。要查看 *detail-type* 的更多可能的值，请参阅每个支持服务的 CloudWatch Events 事件示例 (p. 34)中的示例事件。	Detail Type，Null
events:detail.userIdentity.principalId	"events:detail.userIdentity.principalId":"*principal-id*" 其中，*principal-id* 为事件的 detail.userIdentity.principalId 字段的文字字符串，而 detail-type 为“AWS API Call via CloudTrail”，例如 "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName.".	Principal Id，Null
events:TargetArn	"events:TargetArn":"*target-arn* " 其中，*target-arn* 为可放入规则的目标的 ARN，例如 "arn:aws:lambda:*:*:function:*"。	ARN，Null

有关适用于 CloudWatch Events 的策略语句示例，请参阅 管理您的 CloudWatch Events 资源的访问权限概述 (p. 90)。

主题

- 示例 1：限制对特定源的访问 (p. 106)
- 示例 2：定义可在事件模式中单独使用的多个源 (p. 108)
- 示例 3：定义可在事件模式中使用的 Source 和 DetailType (p. 109)
- 示例 4：确保在事件模式中定义源 (p. 110)
- 示例 5：在包含多个源的事件模式中定义允许的源的列表 (p. 111)
- 示例 6：确保使用来自特定 PrincipalId 的 API 调用的 AWS CloudTrail 事件 (p. 112)
- 示例 7：限制对目标的访问 (p. 113)

示例 1：限制对特定源的访问

以下示例策略可附加到 IAM 用户。策略 A 允许所有事件的 `PutRule` API 操作，而策略 B 仅在要创建的规则的事件模式与 Amazon EC2 事件匹配时允许 `PutRule` 操作。

策略 A：允许所有事件

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEvents",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*"
        }
    ]
    }
```

策略 B：仅允许 Amazon EC2 中的事件

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEC2Events",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

EventPattern 是 `PutRule` 的必需参数。因此，如果具有策略 B 的用户使用类似于下面的事件模式调用 `PutRule`：

```
{
    "source": [ "aws.ec2" ]
}
```

将创建规则，因为策略允许此特定源，即“aws.ec2”。不过，如果具有策略 B 的用户使用类似于下面的事件模式调用 `PutRule`：

```
{
    "source": [ "aws.s3" ]
}
```

规则创建操作将被拒绝，因为策略不允许此特定源，即“aws.s3”。实质上，仅允许具有策略 B 的用户创建与源自 Amazon EC2 的事件匹配的规则，因此他们只能访问 Amazon EC2 中的事件。

有关策略 A 和策略 B 的比较，请参见下表：

事件模式	策略 A 允许的	策略 B 允许的
`{ "source": [ "aws.ec2" ] }`	是	是
`{`	是	否 (不允许源 aws.s3)

事件模式	策略 A 允许的	策略 B 允许的

<pre>"source":
["aws.ec2",
"aws.s3"]
}</pre>		

<pre>{
"source":
["aws.ec2"],
"detail-type":
["EC2 Instance State-change Notification"]
}</pre>
	是	是

<pre>{
"detail-type":
["EC2 Instance State-change Notification"]
}</pre>
	是	否 (必须指定源)

示例 2：定义可在事件模式中单独使用的多个源

以下策略允许来自 Amazon EC2 或 CloudWatch Events 的事件。换言之，它允许 IAM 用户或角色创建一个规则，其中将 EventPattern 中的源指定为“aws.ec2”或“aws.ecs”。不定义源会导致“deny”。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsEC2OrECS",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": [ "aws.ec2", "aws.ecs" ]
                }
            }
        }
    ]
}
```

有关此策略将允许或拒绝的事件模式的示例，请参见下表：

事件模式	策略允许的

<pre>{
"source": ["aws.ec2"]
}</pre>
	是

<pre>{
"source": ["aws.ecs"]
}</pre>
	是

事件模式	策略允许的

<pre>{
    "source": ["aws.s3"]
}</pre>
	否

<pre>{
    "source": ["aws.ec2",
 "aws.ecs"]
}</pre>
	否

<pre>{
    "detail-type": ["AWS API Call
 via CloudTrail"]
}</pre>
	否

示例 3：定义可在事件模式中使用的 Source 和 DetailType

以下策略仅允许来自 `aws.ec2` 源且 DetailType 等于 `EC2 instance state change notification` 的事件。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":
 "AllowPutRuleIfSourceIsEC2AndDetailTypeIsInstanceStateChangeNotification",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2",
                    "events:detail-type": "EC2 Instance State-change Notification"
                }
            }
        }
    ]
}
```

有关此策略将允许或拒绝的事件模式的示例，请参见下表：

事件模式	策略允许的

<pre>{
    "source": ["aws.ec2"]
}</pre>
	否

<pre>{
    "source": ["aws.ecs"]
}</pre>
	否

事件模式	策略允许的

<pre>{
    "source": ["aws.ec2"],
    "detail-type": ["EC2 Instance
 State-change Notification"]
}</pre>
	是

<pre>{
    "source": ["aws.ec2"],
    "detail-type": ["EC2 Instance
 Health Failed"]
}</pre>
	否

<pre>{
    "detail-type": ["EC2 Instance
 State-change Notification"]
}</pre>
	否

示例 4：确保在事件模式中定义源

以下策略允许创建包含必须具有 source 字段的 `EventPatterns` 的规则。换而言之，IAM 用户或角色不能创建包含不提供特定源的 `EventPattern` 的规则。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecified",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

有关此策略将允许或拒绝的事件模式的示例，请参见下表：

事件模式	策略允许的

<pre>{
    "source": ["aws.ec2"],
    "detail-type": ["EC2 Instance
 State-change Notification"]
}</pre>
	是

<pre>{
    "source": ["aws.ecs",
 "aws.ec2"]</pre>
	是

事件模式	策略允许的
<pre>}</pre>	

<pre>{
    "detail-type": ["EC2 Instance
 State-change Notification"]
}</pre>
	否

示例 5：在包含多个源的事件模式中定义允许的源的列表

以下策略允许创建包含可具有多个源的 `EventPatterns` 的规则。事件模式中列出的每个源必须是条件中提供的列表的成员。在使用 ForAllValues 条件时，请确保定义条件列表中的至少一个项。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecifiedAndIsEitherS3OrEC2OrBoth",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "events:source": [ "aws.ec2", "aws.s3" ]
                },
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

有关此策略将允许或拒绝的事件模式的示例，请参见下表：

事件模式	策略允许的

<pre>{
    "source": ["aws.ec2"]
}</pre>
	是

<pre>{
    "source": ["aws.ec2", "aws.s3"]
}</pre>
	是

<pre>{
    "source": ["aws.ec2",
 "aws.autoscaling"]
}</pre>
	否

<pre>{</pre>	否

事件模式	策略允许的

<pre>"detail-type": ["EC2 Instance State-change Notification"]
}</pre>	

示例 6：确保使用来自特定 PrincipalId 的 API 调用的 AWS CloudTrail 事件

所有 AWS CloudTrail 事件的 `detail.userIdentity.principalId` 路径中都具有执行 API 调用的用户的 ID (PrincipalId)。借助 `events:detail.userIdentity.principalId` 条件键，您可以仅允许 IAM 用户或角色访问来自特定账户的 CloudTrail 事件。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleOnlyForCloudTrailEventsWhereUserIsASpecificIAMUser",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail-type": [ "AWS API Call via CloudTrail" ],
                    "events:detail.userIdentity.principalId": [ "AIDAJ45Q7YFFAREXAMPLE" ]
                }
            }
        }
    ]
}
```

有关此策略将允许或拒绝的事件模式的示例，请参见下表：

事件模式	策略允许的

<pre>{
"detail-type": ["AWS API Call via CloudTrail"]
}</pre>
	否

<pre>{
"detail-type": ["AWS API Call via CloudTrail"],
 	
"detail.userIdentity.principalId": ["AIDAJ45Q7YFFAREXAMPLE"]
}</pre>
	是

<pre>{
"detail-type": ["AWS API Call via CloudTrail"],
 	
"detail.userIdentity.principalId": ["AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName"]
}</pre>
	否

示例 7：限制对目标的访问

如果 IAM 用户或角色具有 `events:PutTargets` 权限，他们就可以在相同账户下将任何目标添加到他们有权访问的规则。例如，以下策略仅限将目标添加到特定规则 (账户 123456789012 下的 MyRule)。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRule",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule"
        }
    ]
}
```

为了限制可以添加到规则的目标，请使用 `events:TargetArn` 条件密钥。例如，您可以限制仅将目标添加到 Lambda 函数，如以下示例中所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRuleAndOnlyLambdaFunctions",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule",
            "Condition": {
                "ArnLike": {
                    "events:TargetArn": "arn:aws:lambda:*:*:function:*"
                }
            }
        }
    ]
}
```

在 AWS CloudTrail 中记录 Amazon CloudWatch Events API 调用

AWS CloudTrail 是一项服务，可用于捕获由您的 AWS 账户或代表您的 AWS 账户发出的 API 调用。这种信息经过收集后，写入存储在您指定的 Amazon S3 存储桶中的日志文件中。每当您使用 API、控制台或 AWS CLI 时就会记录 API 调用。通过使用由 CloudTrail 收集的信息，您可以确定发出了什么请求、发出请求的源 IP 地址、发出请求的人员以及发出请求的时间等。

要了解有关 CloudTrail 的更多信息 (包括如何对其进行配置和启用)，请参阅 AWS CloudTrail User Guide中的什么是 AWS CloudTrail。

主题

CloudTrail 中的 CloudWatch Events 信息

如果启用了 CloudTrail 日志记录，则会在日志文件中捕获对 API 操作的调用。每个日志文件条目都包含有关生成请求的人员的信息。例如，如果发出请求以创建 CloudWatch Events 规则 (`PutRule`)，则 CloudTrail 会记录发出请求的人员或服务的身份。

日志条目中的身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 IAM 用户凭证发出的
- 请求是使用角色还是联合身份用户的临时安全凭证发出的
- 请求是否由其他 AWS 服务发出

有关更多信息，请参阅 AWS CloudTrail User Guide 中的 CloudTrail userIdentity 元素。

日志文件可以在存储桶中存储任意长时间，不过您也可以定义 Amazon S3 生命周期规则以自动存档或删除日志文件。默认情况下，将使用 Amazon S3 服务器端加密 (SSE) 对日志文件进行加密。

要获得日志文件传输的通知，可以将 CloudTrail 配置为在传输新日志文件时发布 Amazon SNS 通知。有关更多信息，请参阅 AWS CloudTrail User Guide 中的为 CloudTrail 配置 Amazon SNS 通知。

您还可以将多个 AWS 区域和多个 AWS 账户中的 Amazon CloudWatch Logs 日志文件汇总到单个 S3 存储桶中。有关更多信息，请参阅 AWS CloudTrail User Guide 中的从多个区域接收 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件。

当日志记录开启时，以下 API 操作将会写入到 CloudTrail：

- DeleteRule
- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule

- PutRule
- PutTargets
- RemoveTargets
- TestEventPattern

有关这些操作的更多信息，请参阅 Amazon CloudWatch Events API 参考。

了解日志文件条目

CloudTrail 日志文件包含一个或多个日志条目。每个条目列出了多个 JSON 格式的事件。一个日志条目表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。日志条目不是公用 API 调用的有序堆栈跟踪，因此它们不会以任何特定顺序显示。所有 API 操作的日志文件条目都类似于下面的示例。

以下日志文件条目显示某个用户调用了 CloudWatch Events PutRule 操作。


```
{
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext":{
               "attributes":{
                  "mfaAuthenticated":"false",
                  "creationDate":"2015-11-17T23:56:15Z"
               }
            }
         },
         "eventTime":"2015-11-18T00:11:28Z",
         "eventSource":"events.amazonaws.com",
         "eventName":"PutRule",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"AWS Internal",
         "userAgent":"AWS CloudWatch Console",
         "requestParameters":{
            "description":"",
            "name":"cttest2",
            "state":"ENABLED",
            "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
            "scheduleExpression":""
         },
         "responseElements":{
            "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
         },
         "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
         "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
         "eventType":"AwsApiCall",
         "apiVersion":"2015-10-07",
         "recipientAccountId":"123456789012"
}
```

排除 CloudWatch Events 的故障

您可以使用此部分中的步骤排除 CloudWatch Events 的故障。

主题

我的规则已触发，但未调用我的 Lambda 函数

确保您已经为您的 Lambda 函数设置了正确的权限。使用 AWS CLI 运行以下命令 (将函数名替换为您的函数并使用函数所在的 AWS 区域)：

```
aws lambda get-policy --function-name MyFunction --region us-east-1
```

您应该可以看到类似于如下所示的输出内容：

```
{
    "Policy": "{\"Version\":\"2012-10-17\",
    \"Statement\":[
        {\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:events:us-
east-1:123456789012:rule/MyRule\"}},
        \"Action\":\"lambda:InvokeFunction\",
        \"Resource\":\"arn:aws:lambda:us-east-1:123456789012:function:MyFunction\",
        \"Effect\":\"Allow\",
        \"Principal\":{\"Service\":\"events.amazonaws.com\"},
        \"Sid\":\"MyId\"}
    ],
    \"Id\":\"default\"}"
}
```

如果您看到以下内容：

```
A client error (ResourceNotFoundException) occurred when calling the GetPolicy operation:
 The resource you requested does not exist.
```

或者，您看到输出，但无法将 events.amazonaws.com 定位为策略中的受信任实体，请运行以下命令：

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
```

Note

如果策略不正确，您还可以在 CloudWatch Events 控制台中编辑规则，方式是删除策略并将策略重新添加到规则中。CloudWatch Events 控制台将设置目标的正确权限。
如果您使用特定的 Lambda 别名或版本，则必须在 `--qualifier` 和 `aws lambda get-policy` 命令中添加 `aws lambda add-permission` 参数。

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
--qualifier alias or version
```

Lambda 函数无法触发的另一个原因是，您在运行 `get-policy` 时看到的策略包含 `SourceAccount` 字段。`SourceAccount` 设置会导致 CloudWatch Events 无法调用该函数。

我刚刚创建/修改了规则，但规则未匹配测试事件

在更改规则或其目标时，传入事件可能无法立即开始或停止与新的或更新后的规则的匹配。请稍等片刻，以便更改生效。如果在一段时间后事件仍未匹配，您也可以检查 CloudWatch 中的多个事件指标 (例如 `TriggeredRules`、`Invocations` 和 `FailedInvocations`) 以进一步调试。

您还可以执行 `TestEventPattern` 操作，以使用测试事件来测试规则的事件模式，确保规则的事件模式设置正确。有关详细信息，请参阅 Amazon CloudWatch Events API 参考 中的 TestEventPattern。

我的规则未在 ScheduleExpression 中指定的时间自触发

ScheduleExpressions 用 UTC 表示。确保已采用 UTC 时区设置使规则自触发的计划。如果 ScheduleExpression 正确，则按照 我刚刚创建/修改了规则，但规则未匹配测试事件 (p. 117) 下的步骤操作。

我的规则时未在我期望的时间自触发

当您创建每一个规定时间段都会运行的规则时，CloudWatch Events 不支持设置精确的开始时间。规则一旦创建，倒计时立即开始。

您可以使用 cron 表达式在指定时间调用目标。例如，您可以使用 cron 表达式创建每 4 小时触发一次 (整点时触发) 的规则。在 CloudWatch 控制台中，您将使用 cron 表达式 `0 0/4 * * ? *`；而在 AWS CLI 中，您将使用 cron 表达式 `cron(0 0/4 * * ? *)`。例如，要使用 AWS CLI 创建一个每 4 小时会触发一次的名为 TestRule 的规则，您应该在命令提示符窗口键入以下内容：

```
aws events put-rule --name TestRule --schedule-expression 'cron(0 0/4 * * ? *)'
```

您可以使用 `0/5 * * * ? *` cron 表达式创建一个每 5 分钟触发一次的规则。例如：

```
aws events put-rule --name TestRule --schedule-expression 'cron(0/5 * * * ? *)'
```

CloudWatch Events 不在计划表达式中提供第二级精度。使用 cron 表达式的最高解析精度是一分钟。由于 CloudWatch Events 和目标服务的分布式特性，计划规则触发时间与目标服务实际执行目标资源的时间之间的延迟可能有几秒钟。您的计划规则会在这一分钟内触发，但不会精确到在 0 秒时触发。

我的规则匹配 IAM API 调用但未触发

IAM 服务仅在 美国东部（弗吉尼亚北部）地区 中可用，因此来自 IAM 的任何 AWS API 调用事件仅在该区域内可用。有关更多信息，请参阅 每个支持服务的 CloudWatch Events 事件示例 (p. 34)。

我的规则不起作用，因为与规则关联的 IAM 角色在规则触发时被忽略

规则的 IAM 角色仅用于将事件与 Kinesis 流关联。对于 Lambda 函数和 Amazon SNS 主题，您需要提供基于资源的权限。

确保您的区域 AWS STS 终端节点已启用。在承担您提供的 IAM 角色时，CloudWatch Events 会与区域 AWS STS 终端节点进行通信。有关详细信息，请参阅 IAM 用户指南 中的在 AWS 区域中激活和停用 AWS STS。

我创建了一个包含应与资源匹配的 EventPattern 的规则，但我未看到与该规则匹配的任何事件

AWS 中的大多数服务将 Amazon 资源名称 (ARN) 中的冒号 (:) 或正斜杠 (/) 视为相同的字符。不过，CloudWatch Events 在事件模式和规则中使用精确匹配。请务必在创建事件模式时使用正确的 ARN 字符，以使其与需要匹配的事件中的 ARN 语法相匹配。

此外，并非每个事件都具有已填写的 resources 字段 (例如，来自 CloudTrail 的 AWS API 调用事件)。

向目标传输我的事件时存在延迟

CloudWatch Events 会在长达 24 小时的时间内一直尝试将事件传输到目标。事件一旦到达事件流，立即会进行第一次尝试。但是，如果目标服务遇到问题或您的账户被阻止，CloudWatch Events 会自动重新计划将来的另一次传输。如果从事件到达时算起过去了 24 小时，则不再计划更多的尝试，而且 `FailedInvocations` 指标会发布在 CloudWatch 中。

我的规则在回应两次相同事件时被多次触发。CloudWatch Events 提供了什么有关触发规则或传输事件到目标的保证？

CloudWatch Events 保证在响应一个事件或计划时至少触发一次规则。在很少的情况下，同一规则可能会因一个事件或计划事件而被多次触发，或同一目标可能会因特定的已触发规则而被多次调用。

我的事件没有传送到目标 Amazon SQS 队列

Amazon SQS 队列可能已加密。如果您创建的规则使用加密的 Amazon SQS 队列作为目标，则您必须在您的 KMS 密钥策略中包含以下部分，事件才能成功传送到加密的队列。

```
{
                "Sid": "Allow CWE to use the key",
                "Effect": "Allow",
                "Principal": {
                                "Service": "events.amazonaws.com"
                },
                "Action": [
                                "kms:Decrypt",
                                "kms:GenerateDataKey"
                ],
                "Resource": "*"
}
```

我的规则正在被触发，但我发现没有任何消息发布到我的 Amazon SNS 主题

确保您已经为您的 Amazon SNS 主题设置了正确的权限。使用 AWS CLI 运行以下命令 (将主题 ARN 替换为您的主题并使用主题所在的 AWS 区域)：

```
aws sns get-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic"
```

您应该可以看到类似如下所示的策略属性：

```
"{\"Version\":\"2012-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:Receive\",
\"SNS:SetTopicAttributes\"],
```

```
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}},{\"Sid\":
\"Allow_Publish_Events\",
\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}]}"
```

如果您看到如下所示的策略，则您只设置了默认策略：

```
"{\"Version\":\"2008-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:Receive\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}}]}"
```

如果您未看到策略中具有发布权限的 `events.amazonaws.com`，请使用 AWS CLI 设置主题策略属性。

复制当前策略并将以下语句添加到语句列表中：

```
{\"Sid\":\"Allow_Publish_Events\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}
```

新策略应与前面描述的策略类似。

使用 AWS CLI 设置主题属性：

```
aws sns set-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic" --attribute-name Policy --attribute-value NEW_POLICY_STRING
```

> **Note**
>
> 如果策略不正确，您还可以在 CloudWatch Events 控制台中编辑规则，方式是删除策略并将策略重新添加到规则中。CloudWatch Events 将设置目标的正确权限。

在我删除与 Amazon SNS 主题关联的规则之后，我的 Amazon SNS 主题仍然具有针对 CloudWatch Events 的权限

当您以 Amazon SNS 为目标创建规则时，CloudWatch Events 会代表您将权限添加至您的 Amazon SNS 主题。如果您在创建规则后不久删除规则，CloudWatch Events 可能无法从您的 Amazon SNS 主题删除权限。

如果发生此情况，您可以使用 aws sns 设置主题属性命令从该主题删除权限。有关用于发送事件的基于资源权限的更多信息，请参阅 使用 CloudWatch Events 的基于资源的策略 (p. 100)。

我可以对 CloudWatch Events 使用哪种 IAM 条件键

CloudWatch Events 支持 AWS 范围内的条件键 (请参阅 IAM 用户指南 中的可用密钥) 以及以下特定于服务的条件键。有关更多信息，请参阅 使用 IAM 策略条件实现精细访问控制 (p. 105)。

我如何在违反 CloudWatch Events 规则发出通知

您可以使用以下警报来在违反 CloudWatch Events 规则时发出通知。

创建警报以在违反规则时发出通知

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 选择 Create Alarm。在 CloudWatch Metrics by Category 窗格中，选择 Events Metrics。
3. 在指标列表中，选择 FailedInvocations。
4. 在图形上方，依次选择 Statistic 和 Sum。
5. 对于 Period，选择一个值，例如 5 minutes。选择 Next。
6. 在 Alarm Threshold 下的 Name 中，为警报键入一个唯一的名称，例如：myFailedRules。对于 Description，键入警报的描述，例如：Rules are not delivering events to targets。
7. 对于 is，依次选择 >= 和 1。对于 for，输入 10。
8. 在 Actions 下，为 Whenever this alarm 选择 State is ALARM。
9. 对于 Send notification to，选择一个现有 Amazon SNS 主题或创建新的主题。要创建新 主题，请选择 New list。为新 Amazon SNS 主题键入名称，例如 myFailedRules。
10. 对于 Email list，请键入警报变为 ALARM 状态时将通知发送到的电子邮件地址列表 (以逗号分隔)。
11. 选择 Create Alarm。

文档历史记录

下表描述了从 2018 年 6 月开始的每个版本的 CloudWatch Events 用户指南中的重要更改。如需对此文档更新的通知，您可以订阅 RSS 源。

update-history-change	update-history-description	update-history-date
对 Amazon VPC 终端节点的支持 (p. 122)	您现在可以在 VPC 和 CloudWatch Events 之间建立私有连接。有关更多信息，请参阅 Amazon CloudWatch Events 用户指南 中的将 CloudWatch Events 与接口 VPC 终端节点一起使用。	June 28, 2018

下表介绍了对 Amazon CloudWatch Events 用户指南 的一些重要更改。

更改	描述	发行日期
AWS CodeBuild 作为目标	添加了 AWS CodeBuild 作为事件规则的目标。有关更多信息，请参阅 教程：使用 AWS CodeBuild 安排自动构建 (p. 25)。	2017 年 12 月 13 日
AWS Batch 作为目标	添加了 AWS Batch 作为事件规则的目标。有关更多信息，请参阅 AWS Batch 事件。	2017 年 9 月 8 日
AWS CodePipeline 和 AWS Glue 事件	增加了对 AWS CodePipeline 和 AWS Glue 的事件的支持。有关更多信息，请参阅 AWS CodePipeline 事件 (p. 42) 和 AWS Glue 事件 (p. 66)。	2017 年 9 月 8 日
AWS CodeBuild 和 AWS CodeCommit 事件	增加了对 AWS CodeBuild 和 AWS CodeCommit 的事件的支持。有关更多信息，请参阅 AWS CodeBuild 事件 (p. 40)。	2017 年 8 月 3 日
支持的额外目标	AWS CodePipeline 和 Amazon Inspector 可以是事件的目标。	2017 年 6 月 29 日
支持在 AWS 账户之间发送和接收事件	一个 AWS 账户可以向另一个 AWS 账户发送事件。有关更多信息，请参阅 在 AWS 账户之间发送和接收事件 (p. 81)。	2017 年 6 月 29 日
支持的额外目标	您现在可以将两个额外的 AWS 服务设置为事件操作的目标：Amazon EC2 实例 (通过 Run Command) 和 Step Functions 状态机。有关更多信息，请参阅 Amazon CloudWatch Events 入门 (p. 6)。	2017 年 3 月 7 日
Amazon EMR 事件	增加了对 Amazon EMR 事件的支持。有关更多信息，请参阅 Amazon EMR 事件 (p. 57)。	2017 年 3 月 7 日
AWS Health 事件	增加了对 AWS Health 事件的支持。有关更多信息，请参阅 AWS Health 事件 (p. 69)。	2016 年 12 月 1 日
Amazon Elastic Container Service 事件	增加了对 Amazon ECS 事件的支持。有关更多信息，请参阅 Amazon ECS 事件 (p. 57)。	2016 年 11 月 21 日

更改	描述	发行日期
AWS Trusted Advisor 事件	增加了对 Trusted Advisor 事件的支持。有关更多信息，请参阅 AWS Trusted Advisor 事件 (p. 78)。	2016 年 11 月 18 日
Amazon Elastic Block Store 事件	增加了对 Amazon EBS 事件的支持。有关更多信息，请参阅 Amazon EBS 事件 (p. 44)。	2016 年 11 月 14 日
AWS CodeDeploy 事件	增加了对 AWS CodeDeploy 事件的支持。有关更多信息，请参阅 AWS CodeDeploy 事件 (p. 41)。	2016 年 9 月 9 日
粒度为 1 分钟的计划事件	增加了对 1 分钟粒度的计划事件的支持。有关更多信息，请参阅 Cron 表达式 (p. 27) 和 Rate 表达式 (p. 29)。	2016 年 4 月 19 日
作为目标的 Amazon Simple Queue Service 队列	增加了对作为目标的 Amazon SQS 队列的支持。有关更多信息，请参阅 什么是 Amazon CloudWatch Events？ (p. 1)。	2016 年 3 月 30 日
Auto Scaling 事件	增加了对 Auto Scaling 生命周期挂钩事件的支持。有关更多信息，请参阅 Amazon EC2 Auto Scaling 事件 (p. 35)。	2016 年 2 月 24 日
新增服务	CloudWatch Events 首次发布。	2016 年 1 月 14 日

AWS 词汇表

有关最新 AWS 术语，请参阅 AWS General Reference 中的 AWS 词汇表。