
Amazon CloudWatch Events

用户指南



Amazon CloudWatch Events: 用户指南

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆或者贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Amazon CloudWatch Events ?	1
Concepts	1
相关 AWS 服务	2
设置	3
注册 Amazon Web Services (AWS)	3
要登录 Amazon CloudWatch 控制台	3
账户凭证	3
设置命令行界面	4
区域终端节点	4
入门	5
创建对事件触发的规则	5
通过 CloudTrail 创建对 AWS API 调用触发的规则	6
创建按计划触发的规则	7
删除或禁用规则	8
教程	9
教程：将事件中继到 Systems Manager Run Command	9
教程：记录 EC2 实例状态	10
步骤 1：创建 AWS Lambda 函数	10
步骤 2：创建规则	11
步骤 3：测试规则	11
教程：记录 Auto Scaling 组状态	12
步骤 1：创建 AWS Lambda 函数	12
步骤 2：创建规则	13
步骤 3：测试规则	13
教程：记录 S3 对象级别操作	14
步骤 1：配置您的 AWS CloudTrail 跟踪	14
步骤 2：创建 AWS Lambda 函数	14
步骤 3：创建规则	15
步骤 4：测试规则	15
教程：使用输入转换器自定义要传递给事件目标的内容	16
创建规则	16
教程：记录 AWS API 调用	17
Prerequisite	17
步骤 1：创建 AWS Lambda 函数	17
步骤 2：创建规则	18
步骤 3：测试规则	18
教程：计划自动化 EBS 快照	19
步骤 1：创建规则	19
步骤 2：测试规则	19
教程：计划 Lambda 函数	20
步骤 1：创建 AWS Lambda 函数	20
步骤 2：创建规则	21
步骤 3：验证规则	22
教程：将 Systems Manager 设置为目标	22
教程：将事件中继到 Kinesis 流	23
Prerequisite	23
步骤 1：创建 Amazon Kinesis 流	23
步骤 2：创建规则	24
步骤 3：测试规则	24
步骤 4：验证事件是否已中继	24
教程：当文件上传至 Amazon S3 存储桶时运行 Amazon ECS 任务	25
教程：使用 CodeBuild 安排自动构建	26
教程：记录 Amazon EC2 实例的状态更改	27
规则的计划表达式	28

Cron 表达式	28
Rate 表达式	30
事件模式	32
事件模式	33
在事件模式中匹配 Null 值和空字符串。	34
事件模式下的数组	35
受支持服务的事件	36
Amazon Augmented AI 事件	37
Application Auto Scaling 事件	37
AWS Batch 事件	37
Amazon CloudWatch Events 预定事件	37
Amazon Chime 事件	37
来自 CloudWatch 的事件	38
CodeBuild 事件	38
CodeCommit 事件	38
AWS CodeDeploy 事件	38
CodePipeline 事件	39
AWS Config 事件	40
Amazon EBS 事件	40
Amazon EC2 Auto Scaling 事件	40
Amazon EC2 Instance 再平衡建议事件	41
Amazon EC2 Spot 实例中断事件	41
Amazon EC2 状态更改事件	41
Amazon ECR Events	41
Amazon ECS Events	41
AWS Elemental MediaConvert 事件	42
AWS Elemental MediaPackage 事件	42
AWS Elemental MediaStore 事件	42
Amazon EMR 事件	42
Amazon GameLift 事件	44
AWS Glue 事件	51
AWS Ground Station 事件	56
Amazon GuardDuty 事件	56
AWS Health 事件	56
AWS KMS 事件	58
Amazon Macie Classic 事件	59
Amazon Macie 事件	63
AWS Management Console 登录事件	63
AWS OpsWorks Stacks 事件	64
SageMaker 事件	66
AWS Security Hub 事件	67
AWS Server Migration Service 事件	67
AWS Systems Manager 事件	67
AWS Systems Manager 自动化事件	68
AWS Systems Manager 更改日历事件	69
AWS Systems Manager 合规性事件	69
AWS Systems Manager 维护 Windows 事件	71
AWS Systems Manager Parameter Store 事件	73
AWS Systems Manager 运行命令事件	74
AWS Systems Manager 状态管理器事件	75
AWS Step Functions 事件	76
AWS 资源上的标签更改事件	76
AWS Trusted Advisor 事件	76
WorkSpaces 事件	78
通过 CloudTrail 交付的事件	78
在 AWS 账户之间发送和接收事件	80
允许您的 AWS 账户从其他 AWS 账户接收事件	80

将事件发送到另一个 AWS 账户	81
编写与来自其他 AWS 账户的事件进行匹配的规则	83
将发送方-接收方关系迁移为使用 AWS Organizations	84
使用 PutEvents 添加事件	86
处理使用 PutEvents 时出现的失败情况	86
使用 AWS CLI 发送事件	87
计算 PutEvents 事件条目大小	88
将 CloudWatch Events 与接口 VPC 端点结合使用	90
Availability	90
为 CloudWatch Events 创建 VPC 端点	91
控制对 CloudWatch Events VPC 端点的访问	91
使用 CloudWatch 指标监控使用情况	92
CloudWatch Events 指标	92
CloudWatch Events 指标的维度	92
托管式规则	94
安全性	95
标记您的 CloudWatch Events 资源	96
CloudWatch Events 中支持的资源	96
管理标签	96
标签命名和使用约定	97
记录 API 调用	98
CloudTrail 中的 CloudWatch Events 信息	98
示例：CloudWatch Events 日志文件条目	99
Service Quotas	101
问题排查	102
我的规则已触发，但未调用我的 Lambda 函数	102
我刚刚创建/修改了规则，但规则未匹配测试事件	103
我的规则未在 ScheduleExpression 中指定的时间自触发	104
我的规则时未在我期望的时间自触发	104
我的规则匹配 IAM API 调用但未触发	104
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	104
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	105
向目标传输我的事件时存在延迟	105
某些事件从未传送到我的目标	105
我的规则在回应一个事件时被多次触发。CloudWatch Events 提供了什么有关触发规则或传输事件到目标的保证？	105
防止无限循环	105
我的事件没有传送到目标 Amazon SQS 队列	105
正在触发我的规则，但我发现没有任何消息发布到我的 Amazon SNS 主题	106
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	107
我可以对 CloudWatch Events 使用哪种 IAM 条件密钥	107
我如何在违反 CloudWatch Events 规则发出通知	107
文档历史记录	109
AWS 词汇表	111

什么是 Amazon CloudWatch Events ?

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

Amazon CloudWatch Events 提供几乎实时的系统事件流，这些事件描述 Amazon Web Services (AWS) 资源中的更改。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 随着运营变化的发生而发现。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 \(p. 28\)](#)。

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 中的交付流
- 在 Amazon CloudWatch Logs 中记录组
- Amazon ECS 任务
- Systems Manager 运行命令
- Systems Manager Automation
- AWS Batch 个作业
- Step Functions 状态机
- CodePipeline 中的管道
- 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

Concepts

在您开始使用 CloudWatch Events 之前，应了解以下概念：

- 事件 – 事件指示 AWS 环境中的更改。AWS 资源可以在状态发生变化时生成事件。例如，Amazon EC2 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 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 [受支持服务的 CloudWatch Events 事件示例 \(p. 36\)](#)。

- 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。
- 目标 – 目标负责处理事件。目标可以包含 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。

规则的目标必须与规则位于同一区域中。

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 Management Console、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 [使用 AWS CloudTrail 记录 Amazon CloudWatch Events API 调用 \(p. 98\)](#)。
- 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 则可为您预置和配置这些资源。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 可用于记录 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，以检查资源是否符合企业或组织的策略。有关更多信息，请参见 [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。通过 IAM 可以控制哪些人可以使用您的 AWS 资源（身份验证）、他们可以使用哪些资源以及如何使用这些资源（授权）。有关更多信息，请参阅 [IAM 用户指南](#)。
- Amazon Kinesis Data Streams 可实现快速而近乎持续的数据接收和聚合。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 可用于构建快速响应新信息的应用程序。将您的应用程序代码作为 Lambda 函数上载，Lambda 在高可用性计算基础架构上运行代码。Lambda 执行计算资源的所有管理工作，包括服务器和操作系统维护、容量预置、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 开发人员指南](#)。

设置 Amazon CloudWatch Events

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

要使用 Amazon CloudWatch Events，您需要一个 AWS 账户。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 AWS 账户，请跳到下一步。如果您还没有 AWS 账户，请使用以下步骤创建。

如何注册 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

要登录 Amazon CloudWatch 控制台

登录 Amazon CloudWatch 控制台

1. 登录到 AWS Management Console 并通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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

虽然可以使用根用户凭证访问 CloudWatch Events，但是建议您使用 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": "*"    
  }  
]  
}
```

设置命令行界面

您可以使用 AWS CLI 来执行 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。有关更多信息，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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

使用本节中的过程创建和删除 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](#)。

每个规则

目录

- [创建在事件上触发的 CloudWatch Events 规则 \(p. 5\)](#)
- [使用 AWS CloudTrail 创建对 AWS API 调用触发的 CloudWatch Events 规则 \(p. 6\)](#)
- [创建按计划触发的 CloudWatch Events 规则 \(p. 7\)](#)
- [删除或禁用 CloudWatch Events 规则 \(p. 8\)](#)

Restrictions

- 与规则关联的目标必须与规则位于同一区域中。
- 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 Management Console 中创建带内置目标的规则。
- 如果您创建的规则使用加密的 Amazon SQS 队列作为目标，则必须在 KMS 密钥策略中包含以下部分。它允许该事件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": "*"
}
```

创建在事件上触发的 CloudWatch Events 规则

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

可以按照以下步骤创建对 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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 依次选择 Event Pattern 和 Build event pattern to match events by service。
 - b. 对于服务名称，选择发出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。如果只有通过 CloudTrail 进行 AWS API 调用这一个选项，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 \(p. 6\)](#)。
 - 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. 对于目标，选择添加目标，并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 需要权限以便将事件发送到目标。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，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。
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。

规则名称在此区域中必须是唯一的。
9. 选择 Create rule (创建规则)。

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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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。AWS CloudTrail 会记录这些 API 调用。有关可用作规则触发器的 API 调用的更多信息，请参阅 [CloudTrail 事件历史记录所支持的服务](#)。

CloudWatch Events 中的规则仅适用于创建它们的区域。如果您在多个区域中配置 CloudTrail 跟踪 API 调用，并且希望基于 CloudTrail 的规则在每个区域中触发，则必须在您想跟踪的每个区域中创建单独的规则。

通过 CloudTrail 传递的所有事件都具有 AWS API Call via CloudTrail 作为 detail-type 的值。

Note

在 CloudWatch Events 中，可能创建导致无限循环的规则，即反复触发一个规则。例如，某规则可能检测到 S3 存储桶上的 ACL 已更改，然后触发软件以将 ACL 更改为所需状态。如果编写该规则时不小心，则 ACL 的后续更改将再次触发该规则，从而产生无限循环。

为防止出现这种情况，请在编写规则时使触发的操作不会重复激发同一规则。例如，您的规则可能仅在发现 ACL 处于错误状态时而不是在进行任何更改之后激发。无限循环可能快速导致费用超出预期。我们建议您使用预算功能，以便在费用超出您指定的限制时提醒您。有关更多信息，请参阅[通过预算管理成本](#)。

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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，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. 对于事件类型，选择通过 CloudTrail 进行 AWS API 调用。
 - 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，请选择特定操作，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. 对于目标，选择添加目标，并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 需要权限以便将事件发送到目标。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，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。
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。

规则名称在此区域中必须是唯一的。
9. 选择 Create rule (创建规则)。

创建按计划触发的 CloudWatch Events 规则

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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.aws.amazon.com/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 \(p. 28\)](#)。
5. 对于目标，选择添加目标，并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 需要权限以便将事件发送到目标。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，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。
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。

规则名称在此区域中必须是唯一的。
10. 选择 Create rule (创建规则)。

删除或禁用 CloudWatch Events 规则

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

使用以下步骤删除或禁用 CloudWatch Events 规则。

删除或禁用规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Rules (规则)。

托管规则在其名称旁具有框图标。有关更多信息，请参阅 [Amazon CloudWatch Events 托管式规则 \(p. 94\)](#)。

3. 请执行下列操作之一：
 - 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。

如果该规则是托管规则，您必须键入规则的名称以确认它是托管规则，并且删除它可能会在创建规则的服务中停止功能。要继续，请键入规则名称并选择 Force delete (强制删除)。
 - 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。

您不能禁用托管规则。

CloudWatch Events 教程

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

以下教程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 AWS Systems Manager Run Command \(p. 9\)](#)
- [教程：使用 CloudWatch Events 记录 Amazon EC2 Instance 的状态 \(p. 10\)](#)
- [教程：使用 CloudWatch Events 记录 Auto Scaling 组的状态 \(p. 12\)](#)
- [教程：使用 CloudWatch Events 记录 Amazon S3 对象级别操作 \(p. 14\)](#)
- [教程：使用输入转换器自定义要传递给事件目标的内容 \(p. 16\)](#)
- [教程：使用 CloudWatch Events 记录 AWS API 调用 \(p. 17\)](#)
- [教程：使用 CloudWatch Events 计划自动化 Amazon EBS 快照 \(p. 19\)](#)
- [教程：使用 CloudWatch Events 计划 AWS Lambda 函数 \(p. 20\)](#)
- [教程：将 AWS Systems Manager Automation 设置为 CloudWatch Events 目标 \(p. 22\)](#)
- [教程：使用 CloudWatch Events 将事件中继到 Amazon Kinesis 流 \(p. 23\)](#)
- [教程：当文件上传至 Amazon S3 存储桶时运行 Amazon ECS 任务 \(p. 25\)](#)
- [教程：使用 CodeBuild 安排自动构建 \(p. 26\)](#)
- [教程：记录 Amazon EC2 实例的状态更改 \(p. 27\)](#)

教程：使用 CloudWatch Events 将事件中继到 AWS Systems Manager Run Command

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 实例执行操作。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。

要创建 CloudWatch Events 规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。

3. 对于 Event source (事件源)，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 实例的文档选项。) 对于目标键，键入 **tag:environment**。对于目标值，键入 **production** 并选择添加。
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. 在这些情况下，CloudWatch Events 可以创建运行任务所需的 IAM 角色：
 - 若要自动创建 IAM 角色，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。
10. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
11. 选择 Create rule (创建规则)。

教程：使用 CloudWatch Events 记录 Amazon EC2 Instance 的状态

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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. 打开 AWS Lambda 控制台，地址：<https://console.aws.amazon.com/lambda/>。
2. 如果您是首次接触 Lambda，您将看到欢迎页面。选择 Get Started Now。否则，请选择 Create a Lambda function (创建 Lambda 函数)。
3. 在选择蓝图页面上，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. 对于角色，选择选择现有角色。对于现有角色，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 (下一步)。
6. 在 Review 页面上，选择 Create function。

步骤 2：创建规则

创建一个规则，以便启动 Lambda 实例时运行 Amazon EC2 函数。

要创建 CloudWatch Events 规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 依次选择 EC2 和 EC2 实例状态更改通知
 - d. 依次选择特定状态和正在运行。
 - 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) (特定实例)，然后选择一个或多个实例。
4. 对于目标，依次选择添加目标和 Lambda 函数。
5. 对于函数，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 (创建规则)。

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，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. 通过以下网址打开 Amazon EC2 控制台：<https://console.aws.amazon.com/ec2/>。
2. 启动一个实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 [启动您的实例](#)。
3. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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. 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 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。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南 中的 [终止实例](#)。

教程：使用 CloudWatch Events 记录 Auto Scaling 组的状态

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 场景信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的 [当您的 Auto Scaling Group 缩放时获取 CloudWatch 事件](#)。

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。在创建规则时指定此函数。

创建 Lambda 函数

1. 打开 AWS Lambda 控制台，地址：<https://console.aws.amazon.com/lambda/>。
2. 如果您是首次接触 Lambda，您将看到欢迎页面。选择 Get Started Now。否则，请选择 Create a Lambda function (创建 Lambda 函数)。
3. 在选择蓝图页面上，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. 对于角色，选择选择现有角色。对于现有角色，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 (下一步)。

6. 选择创建函数。

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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 选择 Auto Scaling、实例启动和终止。
 - d. 选择任何实例事件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。若要使该规则与特定 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. 对于目标，依次选择添加目标和 Lambda 函数。
6. 对于函数，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 (创建规则)。

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，以便其启动一个实例。等待几分钟，在扩展事件发生之后，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. 通过以下网址打开 Amazon EC2 控制台：<https://console.aws.amazon.com/ec2/>。
 - b. 在导航窗格上，依次选择 Auto Scaling 和 Auto Scaling Groups。
 - c. 选中您的 Auto Scaling 组对应的复选框。
 - 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。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.aws.amazon.com/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. 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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

您可以在 S3 存储桶上记录对象级别 API 操作。必须使用 AWS CloudTrail 设置跟踪并配置为接收这些事件，Amazon CloudWatch Events 才能匹配这些事件。

步骤 1：配置您的 AWS CloudTrail 跟踪

要将 S3 存储桶的数据事件记录到 AWS CloudTrail 和 CloudWatch Events，需创建一个跟踪。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. 访问 <https://console.aws.amazon.com/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，请选择添加 S3 存储桶，然后指定 S3 存储桶，并可选指定对象前缀。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，指定是要记录 Read (读取) 事件、Write (写入) 事件，还是同时记录这两类事件。
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. 选择创建。

有关更多信息，请参阅 AWS CloudTrail 用户指南中的 [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. 打开 AWS Lambda 控制台，地址：<https://console.aws.amazon.com/lambda/>。
2. 如果您是首次接触 Lambda，您将看到欢迎页面。选择 Create a function。否则，选择创建函数。
3. 选择 Author from scratch。
4. 在从 scratch 开始创作下，执行以下操作：
 - a. 为 Lambda 函数键入名称。例如，将函数命名为“LogS3DataEvents”。

- b. 对于 Role，请选择 Create a custom role。
 - 此时会打开一个新窗口。根据需要更改角色名称，然后选择允许。
 - c. 回到 Lambda 控制台中，选择创建函数。
5. 将 Lambda 函数的代码编辑为以下内容，然后选择保存。

```
'use strict';

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

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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，依次选择规则、创建规则。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 依次选择 Simple Storage Service (S3) 和 Object Level Operations。
 - d. 依次选择特定操作和 PutObject (放置对象)。
 - 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. 对于目标，依次选择添加目标和 Lambda 函数。
5. 对于函数，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 (创建规则)。

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。

要查看您的 Lambda 函数的日志

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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 用户指南中的[获取和查看 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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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。不能对字符 `<` 和 `>` 进行转义。

如果您指定一个变量以匹配在事件中不存在的 JSON 路径，则不会创建该变量，并且不会在输出中显示该变量。

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，下面的 Amazon EC2 实例状态更改通知事件会生成 Amazon SNS 消息：The EC2 instance i-1234567890abcdef0 has changed state to stopped.

```
{
  "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“The EC2 instance `<instance>` has changed state to `<state>`.”

创建规则

使用输入转换器自定义发送到目标的实例状态更改信息

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 依次选择 EC2 和 EC2 实例状态更改通知

- d. 依次选择任何状态和任何实例。
4. 对于目标，依次选择添加目标和 SNS 主题。
5. 对于主题，选择要在 Amazon EC2 实例更改状态时获得通知的 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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

可以使用简单 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 实例停止时就进行记录。

Prerequisite

必须使用 AWS CloudTrail 设置跟踪才能匹配这些事件。如果您没有跟踪，请完成以下步骤。

创建跟踪

1. 访问 <https://console.aws.amazon.com/cloudtrail/>，打开 CloudTrail 控制台。
2. 依次选择 Trails (跟踪)、Create trail (创建跟踪)。
3. 对于 Trail name，键入跟踪的名称。
4. 对于存储位置，在 创建新的 S3 存储桶 中键入 CloudTrail 应将日志传输到其中的新存储桶的名称。
5. 选择创建。

步骤 1：创建 AWS Lambda 函数

创建 Lambda 函数以记录 API 调用事件。在创建规则时指定此函数。

创建 Lambda 函数

1. 打开 AWS Lambda 控制台，地址：<https://console.aws.amazon.com/lambda/>。
2. 如果您是首次接触 Lambda，您将看到欢迎页面。选择 Get Started Now。否则，请选择 Create a Lambda function (创建 Lambda 函数)。
3. 在选择蓝图页面上，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. 对于角色，选择选择现有角色。对于现有角色，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 (下一步)。
6. 在 Review 页面上，选择 Create function。

步骤 2：创建规则

创建一个规则，以便停用 Amazon EC2 实例时运行 Lambda 函数。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 选择 EC2、AWS API Call via CloudTrail。
 - d. 选择 Specific operation(s)，然后在下面的框中键入 StopInstances。
4. 对于目标，依次选择添加目标和 Lambda 函数。
5. 对于函数，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 (创建规则)。

步骤 3：测试规则

可以使用 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. 通过以下网址打开 Amazon EC2 控制台：<https://console.aws.amazon.com/ec2/>。
2. 启动一个实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 [启动您的实例](#)。
3. 停止实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的 [停止和启动实例](#)。
4. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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. 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。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的[终止实例](#)。

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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 Management Console 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 \(p. 28\)](#)。

创建规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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 (创建规则)。

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. 通过以下网址打开 Amazon EC2 控制台：<https://console.aws.amazon.com/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.aws.amazon.com/cloudwatch/>。
 - b. 在导航窗格中，依次选择 Events 和 Rules。
 - c. 选择规则，然后依次选择操作和禁用。
 - d. 当系统提示确认时，选择 Disable。

教程：使用 CloudWatch Events 计划 AWS Lambda 函数

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 Management Console 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。在创建规则时指定此函数。

创建 Lambda 函数

1. 打开 AWS Lambda 控制台，地址：<https://console.aws.amazon.com/lambda/>。
2. 如果您是首次接触 Lambda，您将看到欢迎页面。选择 Get Started Now。否则，请选择 Create a Lambda function (创建 Lambda 函数)。
3. 在选择蓝图页面上，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. 对于角色，选择选择现有角色。对于现有角色，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 (下一步)。
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.aws.amazon.com/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. 对于目标，依次选择添加目标和 Lambda 函数。
5. 对于函数，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 (创建规则)。

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 服务委托人 (events.amazonaws.com) 并使用指定的 Amazon Resource Name (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：验证规则

完成步骤 2 后至少五分钟，您可以验证是否已调用 Lambda 函数。

测试您的规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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. 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函数数据。
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.aws.amazon.com/cloudwatch/>。
 - b. 在导航窗格中，依次选择 Events 和 Rules。
 - c. 选择规则，然后依次选择操作和禁用。
 - 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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

您可以使用 CloudWatch Events 定期在安排好的时间调用 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。

要创建 CloudWatch Events 规则

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，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. 在这些情况下，CloudWatch Events 可以创建运行任务所需的 IAM 角色：
 - 若要自动创建 IAM 角色，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。
9. 选择 Configure details。对于 Rule definition，键入规则的名称和描述。
10. 选择 Create rule (创建规则)。

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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

可以中继 AWS CloudWatch Events 中的 API 调用事件到 Amazon Kinesis 中的流。

Prerequisite

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，以便在您停止 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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 依次选择 EC2 和实例状态更改通知
 - d. 依次选择特定状态和正在运行。
4. 对于目标，依次选择添加目标和 Kinesis 流。
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 (创建规则)。

步骤 3：测试规则

为了测试规则，停用一个 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. 通过以下网址打开 Amazon EC2 控制台：<https://console.aws.amazon.com/ec2/>。
2. 启动一个实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的[启动您的实例](#)。
3. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的[终止实例](#)。

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。分片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，它将从指定分片的流中请求记录。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：当文件上传至 Amazon S3 存储桶时运行 Amazon ECS 任务

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

如果某些 AWS 事件发生，您可以使用 CloudWatch Events 来运行 Amazon ECS 任务。本教程中，您将设置一个 CloudWatch Events 规则，使得每当使用 Amazon S3 PUT 操作将文件上传至某个 Amazon S3 存储桶时运行一个 Amazon ECS 任务。

本教程假定您已在 Amazon ECS 中创建任务定义。

每当使用 PUT 操作将文件上传至某个 S3 存储桶时运行一个 Amazon ECS 任务。

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create rule。
3. 对于 Event source (事件源)，执行以下操作：
 - a. 选择 Event Pattern (事件模式)。
 - b. 选择 Build event pattern to match events by service。
 - c. 对于服务名称，选择 Simple Storage Service (S3)。
 - d. 对于事件类型，选择 Object Level Operations。
 - e. 依次选择特定操作和 Put Object (放置对象)。
 - f. 选择 Specific bucket(s) by name，然后键入存储桶名称。
4. 对于目标，请执行以下操作：
 - a. 依次选择添加目标和 ECS 任务。
 - b. 对于集群和任务定义，选择您创建的资源。

- c. 对于 Launch Type (启动类型)，选择 FARGATE 或 EC2。FARGATE 仅在支持 AWS Fargate 的区域中显示。
- d. (可选) 为 Task Group (任务组) 指定一个值。如果 Launch Type (启动类型) 为 FARGATE，可指定一个平台版本 (可选)。仅指定平台版本的数字部分，如 1.1.0。
- e. (可选) 指定任务定义修订版和任务计数。如果未指定任务定义修订版，则使用最新版本。
- f. 如果任务定义使用 awsvpc 网络模式，则必须指定子网和安全组。所有子网和安全组必须位于同一 VPC。

如果指定了多个安全组和子网，则用逗号分隔它们 (而非空格)。

对于子网，为每个子网指定完整的 subnet-id 值，如下例所示：

```
subnet-123abcd,subnet-789abcd
```

- g. 选择是否允许自动分配公有 IP 地址。
- h. CloudWatch Events 可以创建运行任务所需的 IAM 角色：
 - 若要自动创建 IAM 角色，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。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。
5. 选择 Configure details。
6. 对于 Rule definition，键入规则的名称和描述。
7. 选择 Create rule (创建规则)。

教程：使用 CodeBuild 安排自动构建

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

在本教程的示例中，您安排 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 CodeBuild 以用于该计划构建。

要创建规则，安排每晚 8 点构建 CodeBuild 项目

1. 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/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 \(p. 28\)](#)。
4. 对于目标，依次选择添加目标和 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 CodeBuild 以覆盖默认值。在将 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。

- CloudWatch Events 可以创建运行构建程序所需的 IAM 角色：
 - 若要自动创建 IAM 角色，请选择为此特定资源创建新角色。
 - 要使用您之前创建的 IAM 角色，请选择使用现有角色。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。
- 选择配置详细信息。
- 对于 Rule definition，键入规则的名称和描述。
- 选择 Create rule (创建规则)。

教程：记录 Amazon EC2 实例的状态更改

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

在本教程的示例中，您创建了一个规则，该规则使得在 CloudWatch Logs 中记录 Amazon EC2 中的状态更改通知。

要创建规则以在 CloudWatch Logs 中记录 Amazon EC2 状态更改通知

- 通过以下网址打开 CloudWatch 控制台：<https://console.aws.amazon.com/cloudwatch/>。
- 在导航窗格中，选择事件，然后选择创建规则。
- 对于 Event Source，执行以下操作：
 - 选择 Event Pattern (事件模式)。
 - 对于 Service Name (服务名称)，请选择 EC2。
 - 对于 Event Type (事件类型)，请选择 EC2 Instance State-change Notification (EC2 实例状态更改通知)。
- 对于 Targets，选择 Add target。在服务列表中，选择 CloudWatch log group (CloudWatch 日志组)。
- 对于日志组，输入要接收状态更改通知的日志组的名称。
- 选择 Configure details。
- 对于规则定义，输入规则的名称和描述。
- 选择 Create rule (创建规则)。

规则的计划表达式

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 支持 cron 表达式和 rate 表达式。Rate 表达式更容易定义，但不提供 cron 表达式支持的精细安排控制。例如，使用 cron 表达式，您可以定义在每周或每月的某一天的指定时间触发的规则。相反，rate 表达式以常规速率触发规则，例如每小时一次或每天一次。

Note

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。

格式

- [Cron 表达式 \(p. 28\)](#)
- [Rate 表达式 \(p. 30\)](#)

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 #

字段	值	通配符
年	1970-2199	, - * /

Wildcards

- , (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。您不能在“日期”和“星期几”字段中同时使用 *。如果您在一个中使用它，则必须在另一个中使用？。
- / (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。

Note

如果使用“#”字符，则只能在星期字段中定义一个表达式。例如，“3#1,6#3”是无效的，因为它被解释为两个表达式。

Restrictions

- 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 (或一个 *)，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。

Examples

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	年份	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

分钟	小时	日期	月	星期几	年份	意义
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)
```

值

正数。

unit

时间单位。需要不同的单位，例如，对于值 1 为 minute；对于大于 1 的值 1 为 minutes。

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

Restrictions

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。

Examples

以下示例说明如何将 Rate 表达式与 AWS CLI `put-rule` 命令结合使用。第一个示例每分钟触发一次规则，第二个示例每 5 分钟触发一次规则，第三个示例每小时触发一次规则，最后一个示例每天触发一次规则。

```
aws events put-rule --schedule-expression "rate(1 minute)" --name MyRule2
```

```
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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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。

版本

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。

通过 CloudTrail 传递的所有事件都具有 AWS API Call via CloudTrail 作为 detail-type 的值。有关更多信息，请参阅[通过 CloudTrail 交付的事件 \(p. 78\)](#)。

source

标识发起事件的服务。在 AWS 中的所有事件都以 "AWS" 为开头。只要不以 "AWS" 为开头，在此客户生成的事件可能有任何值。我们建议使用 Java 包名称样式的反向域名字符串。

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`。

账户

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。

区域

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。AWSAPI 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 模式中的数组

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"
]
```


受支持服务的 CloudWatch Events 事件示例

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

以下列表中的 AWS 服务会发出 CloudWatch Events 可检测到的事件。

此外，您还可以通过观察通过 CloudTrail 传送的事件，将 CloudWatch Events 与不发出事件且未在此页面上列出的服务一起使用。有关更多信息，请参阅[通过 CloudTrail 交付的事件 \(p. 78\)](#)。

事件类型

- [Amazon Augmented AI 事件 \(p. 37\)](#)
- [Application Auto Scaling 事件 \(p. 37\)](#)
- [AWS Batch 事件 \(p. 37\)](#)
- [Amazon CloudWatch Events 预定事件 \(p. 37\)](#)
- [Amazon Chime 事件 \(p. 37\)](#)
- [来自 CloudWatch 的事件 \(p. 38\)](#)
- [CodeBuild 事件 \(p. 38\)](#)
- [CodeCommit 事件 \(p. 38\)](#)
- [AWS CodeDeploy 事件 \(p. 38\)](#)
- [CodePipeline 事件 \(p. 39\)](#)
- [AWS Config 事件 \(p. 40\)](#)
- [Amazon EBS 事件 \(p. 40\)](#)
- [Amazon EC2 Auto Scaling 事件 \(p. 40\)](#)
- [Amazon EC2 Instance 再平衡建议事件 \(p. 41\)](#)
- [Amazon EC2 Spot 实例中断事件 \(p. 41\)](#)
- [Amazon EC2 状态更改事件 \(p. 41\)](#)
- [Amazon Elastic Container Registry 事件 \(p. 41\)](#)
- [Amazon Elastic Container Service 事件 \(p. 41\)](#)
- [AWS Elemental MediaConvert 事件 \(p. 42\)](#)
- [AWS Elemental MediaPackage 事件 \(p. 42\)](#)
- [AWS Elemental MediaStore 事件 \(p. 42\)](#)
- [Amazon EMR 事件 \(p. 42\)](#)
- [Amazon GameLift 事件 \(p. 44\)](#)
- [AWS Glue 事件 \(p. 51\)](#)
- [AWS Ground Station 事件 \(p. 56\)](#)
- [Amazon GuardDuty 事件 \(p. 56\)](#)
- [AWS Health 事件 \(p. 56\)](#)
- [AWS KMS 事件 \(p. 58\)](#)

- [Amazon Macie Classic 事件](#) (p. 59)
- [Amazon Macie 事件](#) (p. 63)
- [AWS Management Console 登录事件](#) (p. 63)
- [AWS OpsWorks Stacks 事件](#) (p. 64)
- [SageMaker 事件](#) (p. 66)
- [AWS Security Hub 事件](#) (p. 67)
- [AWS Server Migration Service 事件](#) (p. 67)
- [AWS Systems Manager 事件](#) (p. 67)
- [AWS Step Functions 事件](#) (p. 76)
- [AWS 资源上的标签更改事件](#) (p. 76)
- [AWS Trusted Advisor 事件](#) (p. 76)
- [WorkSpaces 事件](#) (p. 78)
- [通过 CloudTrail 交付的事件](#) (p. 78)

Amazon Augmented AI 事件

有关 Amazon Augmented AI 生成的事件示例，请参阅 [Amazon Augmented AI 中使用事件](#)。

Application Auto Scaling 事件

有关由 Application Auto Scaling 生成的事件的示例，请参阅 [Application Auto Scaling Events](#) 和 [EventBridge](#)。

AWS Batch 事件

有关 AWS Batch 生成的事件示例，请参阅 [AWS Batch 事件](#)。

Amazon CloudWatch Events 预定事件

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": "2019-10-08T16:53:06Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:events:us-east-1:123456789012:rule/MyScheduledRule" ],
  "detail": {}
}
```

Amazon Chime 事件

有关 Amazon Chime 生成的事件示例，请参阅 [EventBridge 自动执行 Amazon Chime](#)。

来自 CloudWatch 的事件

有关 CloudWatch 中的示例事件，请参阅 AWS CodeBuild 用户指南中的[警报事件](#)和[EventBridge](#)。

CodeBuild 事件

有关 CodeBuild 示例事件，请参阅 AWS CodeBuild 用户指南中的[构建通知输入格式参考](#)。

CodeCommit 事件

有关 CodeCommit 示例事件，请参阅 AWS CodeCommit 用户指南中的[监控 EventBridge 和 CloudWatch Events 中的 CodeCommit 事件](#)。

AWS CodeDeploy 事件

以下是 CodeDeploy 事件的示例。有关更多信息，请参阅 AWS CodeDeploy 用户指南中的[使用 CloudWatch Events 来监控部署](#)。

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-0000000aaaaaaaa",
  "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup",
  "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication"
],
"detail": {
  "instanceId": "i-0000000aaaaaaaa",
  "region": "us-east-1",
  "state": "SUCCESS",
  "application": "myApplication",
  "deploymentId": "d-123456789",
  "instanceGroupId": "8cd3bfa8-9e72-4cbe-a1e5-da4efc7efd49",
  "deploymentGroup": "myDeploymentGroup"
}
}
```

CodePipeline 事件

以下是 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

在此示例中有两个 `region` 字段。顶部的一个字段是在其中执行目标管道中的操作的 AWS 区域的名称。在本例中，它是 `us-east-1`。`region` 部分中的 `detail` 是在其中创建事件的 AWS 区域。这与在其中创建管道的区域相同。在本例中，它是 `us-west-2`。

```
{
  "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",
    "region": "us-west-2",
    "type": {
      "owner": "AWS",
      "category": "Deploy",
      "provider": "CodeDeploy",
      "version": 1
    }
  }
}
```

AWS Config 事件

有关 AWS Config 事件的信息，请参阅 AWS Config 开发人员指南中的[使用 Amazon CloudWatch Events 监控 AWS Config](#)。

Amazon EBS 事件

有关 Amazon EBS 事件的信息，请参阅适用于 Linux Instances 的 Amazon EC2 用户指南中的[适用于 Amazon EBS 的 Amazon CloudWatch Events](#)。

Amazon EC2 Auto Scaling 事件

有关 Auto Scaling 事件的信息，请参阅 Amazon EC2 Auto Scaling 用户指南中的[当您的 Auto Scaling Group 缩放时获取 CloudWatch Events](#)。

Amazon EC2 Instance 再平衡建议事件

有关 EC2 实例重新平衡建议的事件的信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的[监控再平衡建议信号](#)。

Amazon EC2 Spot 实例中断事件

有关竞价型实例中断事件的信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的[Spot 实例中断通知](#)。

Amazon EC2 状态更改事件

以下是实例状态更改时 Amazon EC2 实例的事件示例。

EC2 实例状态更改通知

此示例适用于 pending 状态中的实例。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": "2019-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
  ],
  "detail": {
    "instance-id": "i-abcd1111",
    "state": "pending"
  }
}
```

Amazon Elastic Container Registry 事件

Amazon ECR 将映像操作事件发送到 EventBridge。当推送、扫描或删除映像时，会发送事件。

有关 Amazon ECS 示例事件，请参阅 Amazon Elastic Container Registry 用户指南中的[Amazon ECR Events](#)。

Amazon Elastic Container Service 事件

Amazon ECS 将两种类型的事件发送到 EventBridge：容器实例事件和任务事件。仅当您对任务使用 EC2 启动类型时，才发送容器实例事件。对于使用 Fargate 启动类型的任务，您将只收到任务状态事件。Amazon ECS 跟踪容器实例和任务的状态。如果任一资源发生变更，将触发事件。这些事件分类为容器实例状态更改事件或任务状态更改事件。

有关 Amazon ECS 示例事件，请参阅 Amazon Elastic 容器服务开发人员指南中的 [Amazon ECR 事件](#)。

AWS Elemental MediaConvert 事件

有关 MediaConvert 示例事件，请参阅 AWS Elemental MediaConvert 用户指南中的 [使用 CloudWatch Events 监控 AWS Elemental MediaConvert 作业](#)。

AWS Elemental MediaPackage 事件

有关 MediaConvert 示例事件，请参阅 AWS Elemental MediaConvert 用户指南中的 [使用 Amazon CloudWatch Events 监控 AWS Elemental MediaPackage](#)。

AWS Elemental MediaStore 事件

有关 MediaStore 示例事件，请参阅 AWS Elemental MediaStore 用户指南中的 [使用 Amazon CloudWatch Events 自动化 AWS Elemental MediaStore](#)。

Amazon EMR 事件

Amazon EMR 报告的事件以 `aws.emr` 作为 `source` 的值，而通过 CloudTrail 报告的 Amazon EMR API 事件以 `aws.elasticmapreduce` 作为 `source` 的值。

以下是 Amazon EMR 报告的事件的示例。

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-aaaa-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 Events 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-ae8a-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."
}
}
```

Timeout

```
{
  "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": "WARN",
    "state": "TIMEOUT",
    "jobRunId": "jr_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
    "message": "Job run timed out"
  }
}
```

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-3508bc9695",
  "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_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbb3f7a86",
    "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_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbb3f7a86",
    "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_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbb3f7a86",
  "message": "Job is in STOPPING state",
  "startedOn": "2018-04-24T20:55:47.941Z"
}
}
```

AWS Glue 数据目录表状态更改

```
{
  "version": "0",
  "id": "2617428d-715f-edef-70b8-d210da0317a0",
  "detail-type": "Glue Data Catalog Table State Change",
  "source": "aws.glue",
  "account": "123456789012",
  "time": "2019-01-16T18:16:01Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:glue:eu-west-1:123456789012:table/d1/t1"
  ],
  "detail": {
    "databaseName": "d1",
    "changedPartitions": [
      "[C.pdf, dir3]",
      "[D.doc, dir4]"
    ],
    "typeOfChange": "BatchCreatePartition",
    "tableName": "t1"
  }
}
```

AWS Glue 数据目录数据库状态更改

在下面的示例中，`typeOfChange` 为 `CreateTable`。此字段的其他可能值为 `CreateDatabase` 和 `UpdateTable`。

```
{
  "version": "0",
  "id": "60e7ddc2-a588-5328-220a-21c060f6c3f4",
  "detail-type": "Glue Data Catalog Database State Change",
  "source": "aws.glue",
  "account": "123456789012",
  "time": "2019-01-16T18:08:48Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:glue:eu-west-1:123456789012:table/d1/t1"
  ],
  "detail": {
    "databaseName": "d1",
    "typeOfChange": "CreateTable",
    "changedTables": [

```

```
    "t1"  
  ]  
}  
}
```

AWS Ground Station 事件

有关示例 AWS Ground Station 事件的信息，请参阅 AWS Ground Station 用户指南中的[使用 CloudWatch Events 实现自动化 AWS Ground Station](#)。

Amazon GuardDuty 事件

有关 Amazon GuardDuty 事件的信息，请参阅 Amazon GuardDuty 用户指南中的[通过 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 来管理 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 `MAINTENANCE_SCHEDULED` 之前两周左右推送出包含 `startTime` 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 开发人员指南中的 [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 Classic 事件

以下是 Amazon Macie Classic 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/11122223333/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/11122223333/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
reconnaissance or exfiltration campaign",
      "anomalous": true,
      "multiplier": 12.017441860465116,
      "excession_times": [
        "2017-04-21T18:00:00Z"
      ],
      "risk": 2
    }
  }
}
```

Amazon Macie 事件

有关 Amazon Macie 生成的事件示例，请参阅 [Amazon Macie 调查结果的事件架构](#)。

AWS Management Console 登录事件

CloudWatch Watch Events 仅在美国东部（弗吉尼亚北部）区域检测 AWS Management Console 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"
}
```

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 `initiated_by`、`requested` 或 `terminating` 状态时，才会填充 `stopping` 字段。`initiated_by` 字段可以包含以下值之一。

- `user` - 用户使用 API 或 请求的实例状态更改。AWS Management Console
- `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，因此跳过了命令。
- `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-aafe-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-b632-841c1cfeb22ff",
  "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 提醒

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."
  }
}
```

SageMaker 事件

有关示例 SageMaker 事件的信息，请参阅 SageMaker 开发人员指南中的[使用 Amazon EventBridge 自动执行 SageMaker](#)。

AWS Security Hub 事件

有关 Security Hub 事件的信息，请参阅 AWS Security Hub 用户指南中的[使用 Amazon CloudWatch Events 监控 AWS Security Hub](#)。

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 Systems Manager 事件

以下是 AWS Systems Manager 事件的示例。有关更多信息，请参阅 AWS Systems Manager 用户指南中的[使用 Amazon EventBridge 监控 Systems Manager 事件](#)。

Systems Manager 事件类型

- [AWS Systems Manager 自动化事件](#) (p. 68)
- [AWS Systems Manager 更改日历事件](#) (p. 69)
- [AWS Systems Manager 合规性事件](#) (p. 69)
- [AWS Systems Manager 维护 Windows 事件](#) (p. 71)
- [AWS Systems Manager Parameter Store 事件](#) (p. 73)
- [AWS Systems Manager 运行命令事件](#) (p. 74)
- [AWS Systems Manager 状态管理器事件](#) (p. 75)

AWS Systems Manager 自动化事件

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"
  }
}
```

```
}  
}
```

AWS Systems Manager 更改日历事件

以下是 AWS Systems Manager Change Calendar 事件的示例。

Note

目前不支持从其他 AWS 帐户日历的状态更改。

日历开放

```
{  
  "version": "0",  
  "id": "47a3f03a-f30d-1011-ac9a-du3bdEXAMPLE",  
  "detail-type": "Calendar State Change",  
  "source": "aws.ssm",  
  "account": "111222333444",  
  "time": "2020-09-19T18:00:07Z",  
  "region": "us-east-2",  
  "resources": [  
    "arn:aws:ssm:us-east-2:111222333444:document/MyCalendar"  
  ],  
  "detail": {  
    "state": "OPEN",  
    "atTime": "2020-09-19T18:00:07Z",  
    "nextTransitionTime": "2020-10-11T18:00:07Z"  
  }  
}
```

日历关闭

```
{  
  "version": "0",  
  "id": "f30df03a-1011-ac9a-47a3-f761eEXAMPLE",  
  "detail-type": "Calendar State Change",  
  "source": "aws.ssm",  
  "account": "111222333444",  
  "time": "2020-09-17T21:40:02Z",  
  "region": "us-east-2",  
  "resources": [  
    "arn:aws:ssm:us-east-2:111222333444:document/MyCalendar"  
  ],  
  "detail": {  
    "state": "CLOSED",  
    "atTime": "2020-08-17T21:40:00Z",  
    "nextTransitionTime": "2020-09-19T18:00:07Z"  
  }  
}
```

AWS Systems Manager 合规性事件

以下是 AWS Systems Manager 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"
  }
}
```

AWS Systems Manager 维护 Windows 事件

以下是 Systems Manager 维护 Windows 事件的示例。

注册目标

另一个有效的状态值是 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"
  }
}
```

时间段执行类型

其他有效状态值为 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

其他有效状态值为 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

其他有效状态值为 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"
  }
}
```

```
}
```

时间段状态更改

其他有效状态值为 `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"
  }
}
```

AWS Systems Manager Parameter Store 事件

以下是 Systems Manager 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 运行命令事件

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"
}
}
```

AWS Systems Manager 状态管理器事件

State Manager 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 Step Functions 事件

有关 Step Functions 示例事件，请参阅 [AWS Step Functions 开发人员指南](#)和 [Step Functions 事件示例](#)。

AWS 资源上的标签更改事件

下面是一个标签事件示例。

```
{
  "version": "0",
  "id": "ffd8a6fe-32f8-ef66-c85c-111111111111",
  "detail-type": "Tag Change on Resource",
  "source": "aws.tag",
  "account": "123456789012",
  "time": "2018-09-18T20:41:06Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaa"
  ],
  "detail": {
    "changed-tag-keys": [
      "key2",
      "key3"
    ],
    "service": "ec2",
    "resource-type": "instance",
    "version": 5,
    "tags": {
      "key4": "value4",
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```

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 用户指南](#)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"  
}
```

Exposed Access Keys

```
{  
  "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"  
  }  
}
```

WorkSpaces 事件

有关 WorkSpaces 事件的信息，请参阅 Amazon WorkSpaces 管理指南中的[使用 CloudWatch Events 监控您的 WorkSpaces](#)。

通过 CloudTrail 交付的事件

您也可以对并不发出事件且不在此页面中列出的服务使用 CloudWatch Events。AWS CloudTrail 是一个服务，可用于自动记录事件，例如 AWS 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. 6)。

通过 CloudTrail 传递的所有事件都具有 AWS API Call via CloudTrail 作为 detail-type 的值。

AWS 中的某些事件可以由服务本身和由 CloudTrail 报告给 CloudWatch Events，但是以不同的方式报告。例如，启动或终止实例的 Amazon EC2 API 调用生成可通过 CloudTrail 对 CloudWatch Events 可用的事

件。但是，举例来说，Amazon EC2 实例状态更改（从“正在运行”更改为“正在终止”）是 CloudWatch Events 事件本身。

以下是通过 CloudTrail 传送的事件的示例。该事件由 AWS API 对 Amazon S3 的调用生成，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"
}
```

大于 256 KB 的 AWS API 调用事件不受支持。有关可用作规则触发器的 API 调用的更多信息，请参阅 [CloudTrail 事件历史记录所支持的服务](#)。

在 AWS 账户之间发送和接收事件

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

您可以设置您的 AWS 账户将事件发送到其他 AWS 账户，或接收来自其他账户的事件。如果这些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，请指定可以将事件发送到哪些独立 AWS 账户或哪些独立 AWS 账户可以接收来自您账户的事件。如果您使用 AWS Organizations 功能，则可以指定一个组织并授予对该组织中的所有账户的访问权限。有关更多信息，请参阅 [AWS Organizations 用户指南](#) 中的什么是 AWS Organizations。

整体过程如下所述：

- 在接收方账户上，编辑默认事件总线上的权限以允许指定 AWS 账户、一个组织或所有 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。

如果发送方账户由于是某个具有权限的 AWS 组织的一部分而有权发送事件，则发送方账户还必须拥有一个 IAM 角色，该角色具有支持它将事件发送到接收方账户的策略。如果您使用 AWS Management Console 创建针对接收方账户的规则，这会自动完成。如果您使用 AWS CLI，则必须手动创建该角色。

- 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。有关更多信息，请参阅 [Amazon CloudWatch 定价](#)。

如果接收方账户设置了一条将从发送方账户接收的事件发送到第三个账户的规则，则这些事件不会发送到第三个账户。

允许您的 AWS 账户从其他 AWS 账户接收事件

要接收来自其他账户或组织的事件，则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上的权限以向其他 AWS 账户授予权限时，可以按账户 ID 或组织 ID 指定账户。或者您可以选择从所有 AWS 账户接收事件。

Warning

如果您选择接收来自所有 AWS 账户的事件，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都包含一个 Account 字段，其中包含要从其接

收事件的一个或多个账户的账户 ID。其事件模式包含“账户”字段的规则与从在 `Account` 字段中未列出的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 \(p. 32\)](#)。

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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，依次选择 Event Buses 和 Add Permission。
3. 选择 AWS 账户 或 Organization (组织)。

如果您选择 AWS 账户，请输入要从中接收事件的账户的 12 位 AWS 账户 ID。要接收来自所有其他 AWS 账户的事件，请选择所有人(*)。

如果您选择了 Organization (组织)，请选择 My organization (我的组织) 以便为当前账户所属组织中的所有账户授予权限。或者选择 Another organization (另一个组织)，然后输入该组织的组织 ID。键入组织 ID 时，必须包含 o- 前缀。

4. 选择 Add。
5. 您可以重复这些步骤来添加其他账户或组织。

使用 AWS 允许您的账户从其他 AWS CLI 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 AWS 组织发送事件，请运行以下命令：

```
aws events put-permission --action events:PutEvents --statement-id MySid --principal \* --condition '{"Type" : "StringEquals", "Key": "aws:PrincipalOrgID", "Value": "SenderOrganizationID"}'
```

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 \*
```

您可以运行 `aws events put-permission` 多次以便为独立 AWS 账户和组织授予权限，但您无法在单个命令中同时指定独立账户和组织。

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，请配置一个 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.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择 Events 和 Create Rule。

- 对于 Event Source (事件源), 选择 Event Pattern (事件模式), 然后选择要发送到另一个账户的服务名称和事件类型。
- 选择 Add Target。
- 对于目标, 请选择另一个 AWS 账户中的事件总线。对于 Account ID (账户 ID), 请输入要向其发送事件的 AWS 账户的 12 位账户 ID。
- 当此发送方账户有权发送事件时, 需要一个 IAM 角色, 因为接收方账户已为整个组织授予权限。
 - 若要自动创建 IAM 角色, 请选择为此特定资源创建新角色。
 - 否则, 请选择 Use existing role (使用现有角色)。选择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。
- 在页面底部, 选择配置详细信息。
- 键入规则的名称和描述, 然后选择创建规则。

使用 AWS 将事件发送到另一个 AWS CLI 账户

- 如果发送方账户由于是接收方账户已授予权限的 AWS 组织的一部分而有权发送事件, 则发送方账户还必须拥有一个角色, 该角色具有支持它将事件发送到接收方账户的策略。此步骤介绍了如何创建该角色。

如果已为发送方账户授予通过其 AWS 账户 ID 而不是通过组织发送事件的权限, 则此步骤为可选步骤。您可以跳至步骤 2。

- 如果通过组织为发送方账户授予了权限, 请创建所需的 IAM 角色。首先, 使用以下内容创建名为 `assume-role-policy-document.json` 的文件:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- 要创建角色, 请输入以下命令:

```
$ aws iam create-role \
--profile sender \
--role-name event-delivery-role \
--assume-role-policy-document file://assume-role-policy-document.json
```

- 使用以下内容创建名为 `permission-policy.json` 的文件:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:${receiver_account_id}:event-bus/default"
      ]
    }
  ]
}
```

```
}  
  ]  
}
```

- d. 输入以下命令以将此策略附加到角色：

```
$ aws iam put-role-policy \  
--profile sender \  
--role-name event-delivery-role \  
--policy-name EventBusDeliveryRolePolicy \  
--policy-document file://permission-policy.json
```

2. 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。
3. 将其他账户的默认事件总线作为规则的目标添加。

如果已为发送方账户授权按其账户 ID 发送事件，则无需指定角色。运行以下命令：

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

如果已为发送方账户授权按其组织发送事件，请按以下示例所示指定角色：

```
aws events put-targets --rule NameOfRuleMatchingEventsToSend --targets  
"Id"="MyId", "Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/  
default", "RoleArn"="arn:aws:iam:${sender_account_id}:role/event-delivery-role"
```

编写与来自其他 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，则可以编写与这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。

如果将您的 AWS 账户设置为接受来自所有 AWS 账户的事件，强烈建议您将 `account` 添加到您账户的每一个 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。

要使规则触发来自您已授予权限的任何 AWS 账户的匹配事件，请不要在规则的 `account` 字段中指定 `*`。这样做不会匹配任何事件，因为 `*` 从不显示在事件的 `account` 字段中。相反，只需忽略规则的 `account` 字段即可。

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.aws.amazon.com/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"  
  ]  
}
```

添加以下内容以使规则与 AWS 账户 123456789012 和 111122223333 发送的 EBS 卷通知匹配：

```
{  
  "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 编写与来自另一个 AWS CLI 账户的事件进行匹配的规则

- 使用 `put-rule` 命令。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"
```

将发送方-接收方关系迁移为使用 AWS Organizations

如果您具有一个已直接将权限授予其账户 ID 的发送方账户，您现在想要撤消这些权限，并通过将权限授予组织以向发送账户授予访问权限，则必须执行一些额外步骤。这些步骤确保来自发送方账户的事件仍可抵达接收方账户。这是因为通过组织获得发送事件权限的账户还必须使用 IAM 角色来执行此操作。

添加迁移发送方-接收方关系所需的权限

1. 在发送方账户中，创建一个 IAM 角色，其策略使其能够将事件发送到接收方账户。
 - a. 使用以下内容创建名为 `assume-role-policy-document.json` 的文件：

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "events.amazonaws.com"  
      },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

```
]
}
```

- b. 要创建 IAM 角色，请输入以下命令：

```
$ aws iam create-role \
--profile sender \
--role-name event-delivery-role \
--assume-role-policy-document file://assume-role-policy-document.json
```

- c. 使用以下内容创建名为 `permission-policy.json` 的文件：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:${receiver_account_id}:event-bus/default"
      ]
    }
  ]
}
```

- d. 输入以下命令以将此策略附加到角色：

```
$ aws iam put-role-policy \
--profile sender \
--role-name event-delivery-role \
--policy-name EventBusDeliveryRolePolicy
--policy-document file://permission-policy.json
```

2. 编辑发送方账户中将接收方账户的默认事件总线作为目标的现有规则。通过将您在第 1 步中创建的角色添加到目标信息来编辑此规则。使用以下命令：

```
aws events put-targets --rule RuleName --targets
  "Id"="MyID", "Arn"="arn:aws:events:region:${ReceiverAccountID}:event-bus/
default", "RoleArn"="arn:aws:iam:${sender_account_id}:role/event-delivery-role"
```

3. 在接收方账户中，运行以下命令以授予组织中的账户将事件发送到接收方账户的权限：

```
aws events put-permission --action events:PutEvents --statement-id Sid-For-Organization
--principal \* --condition '{"Type" : "StringEquals", "Key": "aws:PrincipalOrgID",
"Value": "SenderOrganizationID"}'
```

(可选) 您也可以撤消最初直接授予发送方账户的权限：

```
aws events remove-permission --statement-id Sid-for-SenderAccount
```

使用 PutEvents 添加事件

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 引用中的 [PutEvents](#) 和 AWS CLI 命令引用中的 [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 配额 \(p. 101\)](#)。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 FailedRecordCount 中的 PutEventsResult 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 Entry (不为空) 的 ErrorCode 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\" }"
  }
]
```

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

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

计算 PutEvents 事件条目大小

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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

已对此条目施加大小限制。即使条目大小低于此大小限制，也不意味着 CloudWatch Events 中的事件也小于此大小。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 \(p. 32\)](#)。

`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 与接口 VPC 端点结合使用

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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，可用于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 VPC 连接到 CloudWatch Events，请为 CloudWatch Events 定义一个接口 VPC 端点。这种类型的端点使您能够将 VPC 连接到 AWS 服务。该端点提供了到 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 端点由 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 [最新 – 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](#)。

Availability

CloudWatch Events 当前在以下区域中支持 VPC 端点：

- US East (Ohio)
- 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)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- 亚太地区 (新加坡)
- 亚太地区 (悉尼)
- 亚太区域 (东京)
- Canada (Central)
- Europe (Frankfurt)
- 欧洲 (爱尔兰)
- Europe (London)
- Europe (Paris)
- 南美洲 (圣保罗)

为 CloudWatch Events 创建 VPC 端点

要开始使用 CloudWatch Events 与 VPC，请为 CloudWatch Events 创建一个接口 VPC 端点。要选择的任务名称是 `com.amazonaws.Region.events`。有关更多信息，请参阅 Amazon VPC 用户指南中的[创建接口端点](#)。

您不需要更改 CloudWatch Events 的设置。CloudWatch Events 使用公有端点或私有接口 VPC 终端节点 (二者中在使用中的那个) 调用其他 AWS 服务。例如，如果为 CloudWatch Events 创建一个接口 VPC 端点，并且已经有一个在触发后向 Amazon SNS 发送通知的 CloudWatch Events 规则，通知将开始流经接口 VPC 端点。

控制对 CloudWatch Events VPC 端点的访问

VPC 终端节点策略是一种 IAM 资源策略，您在创建或修改终端节点时可将它附加到终端节点。如果在创建终端节点时未附加策略，我们将为您附加默认策略以允许对服务进行完全访问。端点策略不会覆盖或替换 IAM 用户策略或服务特定的策略。这是一个单独的策略，用于控制从终端节点中对指定服务进行的访问。

终端节点策略必须采用 JSON 格式编写。

有关更多信息，请参阅 Amazon VPC 用户指南中的[使用 VPC 终端节点控制对服务的访问](#)。

下面是用于 CloudWatch Events 的端点策略示例。该策略允许通过 VPC 连接到 CloudWatch Events 的用户将日志发送到 CloudWatch Events，并禁止他们执行其他 CloudWatch Events 操作。

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "events:PutEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

修改 CloudWatch Events 的 VPC 端点策略

1. 通过以下网址打开 Amazon VPC 控制台：<https://console.aws.amazon.com/vpc/>。
2. 在导航窗格中，选择终端节点。
3. 如果还没有为 CloudWatch Events 创建端点，请选择 Create Endpoint (创建终端节点)。接下来，选择 `com.amazonaws.Region.events`，然后选择创建终端节点。
4. 选择 `com.amazonaws.Region.events` 终端节点，然后在屏幕下半部分中选择策略选项卡。
5. 选择编辑策略并对策略进行更改。

使用 CloudWatch 指标监控使用情况

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

CloudWatch Events 每分钟向 Amazon CloudWatch 发送指标一次。

CloudWatch Events 指标

AWS/Events 命名空间包括以下指标。

所有这些指标都使用 Count 作为单位，因此 Sum 和 SampleCount 是最有用的统计数据。

指标	描述
DeadLetterInvocations	<p>测量未为响应事件而调用规则目标的次数。这包括将导致再次触发同一规则从而引发无限循环的调用。</p> <p>有效维度：RuleName</p> <p>单位：计数</p>
Invocations	<p>测量为响应事件而针对某个规则调用目标的次数。这包括成功和失败的调用，但不包括在永久失败之前被阻止或重试的尝试。它不包含 DeadLetterInvocations。</p> <p>Note</p> <p>CloudWatch Events 仅在其具有非零值时才会将此指标发送到 CloudWatch。</p> <p>有效维度：RuleName</p> <p>单位：计数</p>
FailedInvocations	<p>测量永久失败的调用的数目。这包括重试或重试尝试后成功的调用。它也不会计算在 DeadLetterInvocations 中计数的失败调用。</p> <p>有效维度：RuleName</p> <p>单位：计数</p>
TriggeredRules	<p>测量与任何事件匹配的已触发规则的数目。</p> <p>有效维度：RuleName</p> <p>单位：计数</p>

指标	描述
<code>MatchedEvents</code>	测量与任何规则匹配的事件的数目。 有效维度：无 单位：计数
<code>ThrottledRules</code>	测量被阻止的已触发规则的数目。 有效维度：RuleName 单位：计数

CloudWatch Events 指标的维度

CloudWatch Events 指标具有一个维度，该维度在下面列出。

维度	描述
<code>RuleName</code>	按规则名称筛选可用指标。

Amazon CloudWatch Events 托管式规则

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

其他 AWS 服务可在您的 AWS 账户中创建和管理这些服务中的某些函数需要的 CloudWatch Events 规则。这些策略称为托管式规则。

当某个服务创建一个托管式规则时，它也可以创建一个 IAM 策略，以向该服务授予创建该规则的权限。以这种方式创建的 IAM 策略的作用于局限于资源级权限，以仅允许创建必需的规则。

您可以使用 Force delete (强制删除) 选项删除托管式规则。仅当您确定其他服务不再需要此规则时才这样做。否则，删除托管式规则会导致依赖它的功能停止工作。

Amazon CloudWatch Events 的安全性

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

有关的 CloudWatch Events 的安全性信息，请参阅 [Amazon EventBridge 的安全性](#)。

标记您的 Amazon CloudWatch Events 资源

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

标签是您分配的自定义属性标签，或者是 AWS 分配给 AWS 资源的标签。每个标签具有两个部分：

- 标签键（例如，CostCenter、Environment 或 Project）。标签键区分大小写。
- 一个称为标签值的可选字段（例如，111122223333 或 Production）。省略标签值与使用空字符串相同。与标签键一样，标签值区分大小写。

标签可帮助您：

- 标识和整理您的 AWS 资源。许多 AWS 服务支持标记，因此，您可以将同一标签分配给来自不同服务的资源，以指示这些资源是相关的。例如，您可以将相同的标签分配给您分配给 EC2 实例的 CloudWatch Events 规则。
- 跟踪您的 AWS 成本。您可以在 AWS Billing and Cost Management 控制面板上激活这些标签。AWS 使用标签对您的成本进行分类，并向您提供每月成本分配报告。有关更多信息，请参阅 [AWS Billing and Cost Management 用户指南](#)中的 [使用成本分配标签](#)。

以下各部分提供有关 CloudWatch Events 的标签的更多信息。

CloudWatch Events 中支持的资源

CloudWatch Events 中的以下资源支持标记：

- Rule

有关添加和管理标签的信息，请参阅 [管理标签 \(p. 96\)](#)。

管理标签

标签由资源上的 Key 和 Value 属性构成。可以使用 CloudWatch 控制台、AWS CLI 或 CloudWatch Events API 添加、编辑或删除这些属性的值。有关使用标签的信息，请参阅以下内容：

- Amazon CloudWatch Events API 引用中的 [TagResource](#)、[UntagResource](#) 和 [ListTagsForResource](#)
- Amazon CloudWatch CLI 引用中的 [tag-resource](#)、[untag-resource](#) 和 [list-tags-for-resource](#)
- Resource Groups 用户指南中的 [使用标签编辑器](#)

标签命名和使用约定

以下基本命名和使用约定适用于将标签与 CloudWatch Events 资源一起使用的情况：

- 每个资源最多可以有 50 个标签。
- 对于每个资源，每个标签键都必须是唯一的，每个标签键只能有一个值。
- 最大标签键长度为 128 个 Unicode 字符 (采用 UTF-8 格式)。
- 最大标签值长度为 256 个 Unicode 字符 (采用 UTF-8 格式)。
- 允许使用的字符包括可用 UTF-8 格式表示的字母、数字和空格，以及以下字符：`.:+=@_/-` (连字符)。
- 标签键和值区分大小写。最佳实践是，决定利用标签的策略并在所有资源类型中一致地实施该策略。例如，决定是否使用 `Costcenter`、`costcenter` 或 `CostCenter`，以及是否对所有标签使用相同的约定。避免将类似的标签用于不一致的案例处理。
- 对标签禁止使用 `aws:` 前缀，因为它是为使用 AWS 而保留的。您无法编辑或删除带此前缀的标签键或值。具有此前缀的标签不计入每个资源的标签数配额。

使用 AWS CloudTrail 记录 Amazon CloudWatch Events API 调用

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

Amazon CloudWatch Events 与 AWS CloudTrail 集成，后者是在 CloudWatch Events 中提供用户、角色或 AWS 服务所采取操作的记录的服务。CloudTrail 捕获由您的 AWS 账户发出或代表该账户发出的 API 调用。捕获的调用包括来自 CloudWatch 控制台的调用和对 CloudWatch Events API 操作的代码调用。如果您创建了一个跟踪，则可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 CloudWatch Events 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 CloudWatch Events 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

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 用户指南](#)》。

主题

- [CloudTrail 中的 CloudWatch Events 信息 \(p. 98\)](#)
- [示例：CloudWatch Events 日志文件条目 \(p. 99\)](#)

CloudTrail 中的 CloudWatch Events 信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 CloudWatch Events 中发生受支持的事件活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history（事件历史记录）中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅[使用 CloudTrail 事件历史记录查看事件](#)。

要持续记录 AWS 账户中的事件（包括 CloudWatch Events 的事件），请创建跟踪。通过跟踪，CloudTrail 可将日志文件传送到 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪在 AWS 分区中记录所有区域中的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- [创建跟踪概述](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [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](#)

CloudWatch Events 支持将以下操作记录为 CloudTrail 日志文件中的事件：

- [DeleteRule](#)
- [DescribeEventBus](#)

- [DescribeRule](#)
- [DisableRule](#)
- [EnableRule](#)
- [ListRuleNamesByTarget](#)
- [ListRules](#)
- [ListTargetsByRule](#)
- [PutPermission](#)
- [PutRule](#)
- [PutTargets](#)
- [RemoveTargets](#)
- [TestEventPattern](#)

每个事件或日志条目都包含有关生成请求的人员信息。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 AWS Identity and Access Management (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 [CloudTrail userIdentity 元素](#)。

示例：CloudWatch Events 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的 CloudTrail 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"  
}
```

CloudWatch Events 配额

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

有关 CloudWatch Events 和 EventBridge 服务配额的信息，请参阅 [Amazon EventBridge 配额](#)。

有关更多信息，请参阅下列内容。

- [Amazon EventBridge](#)
- [EventBridge Service Quotas](#)
- [Amazon EventBridge API 引用](#)

排除 CloudWatch Events 的故障

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

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 \(p. 102\)](#)
- [我刚刚创建/修改了规则，但规则未匹配测试事件 \(p. 103\)](#)
- [我的规则未在 ScheduleExpression 中指定的时间自触发 \(p. 104\)](#)
- [我的规则时未在我期望的时间自触发 \(p. 104\)](#)
- [我的规则匹配 IAM API 调用但未触发 \(p. 104\)](#)
- [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 \(p. 104\)](#)
- [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 \(p. 105\)](#)
- [向目标传输我的事件时存在延迟 \(p. 105\)](#)
- [某些事件从未传送到我的目标 \(p. 105\)](#)
- [我的规则在回应一个事件时被多次触发。CloudWatch Events 提供了什么有关触发规则或传输事件到目标的保证？ \(p. 105\)](#)
- [防止无限循环 \(p. 105\)](#)
- [我的事件没有传送到目标 Amazon SQS 队列 \(p. 105\)](#)
- [正在触发我的规则，但我发现没有任何消息发布到我的 Amazon SNS 主题 \(p. 106\)](#)
- [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 \(p. 107\)](#)
- [我可以对 CloudWatch Events 使用哪种 IAM 条件密钥 \(p. 107\)](#)
- [我如何在违反 CloudWatch Events 规则发出通知 \(p. 107\)](#)

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 `aws lambda get-policy` 和 `aws lambda add-permission` 命令中添加 `--qualifier` 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，也可以检查您的规则（例如 `Invocations`、`TriggeredRules` 和 `FailedInvocations`）的 CloudWatch 指标以进一步调试。有关这些指标的更多信息，请参阅 Amazon CloudWatch 用户指南中的 [Amazon CloudWatch Events 指标和纬度](#)。

如果该规则由来自 AWS 服务的事件触发，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。有关更多信息，请参阅 Amazon CloudWatch Events API 引用中的 [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 \(p. 103\)](#) 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）区域内可用，因此任何来自 IAM 的 AWS API 调用事件仅在该区域内可用。有关更多信息，请参阅[受支持服务的 CloudWatch Events 事件示例 \(p. 36\)](#)。

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 终端节点处于启用状态。当您担任您提供的 IAM 角色时，CloudWatch Events 会告知区域 AWS STS 终端节点。有关更多信息，请参阅 IAM 用户指南中的[在 AWS STS 区域中激活和停用 AWS](#)。

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 Resource Names (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（如来自 CloudTrail 的 AWS API 调用事件）都已填写资源字段。

向目标传输我的事件时存在延迟

CloudWatch Events 会在长达 24 小时内一直尝试将事件传递给目标，但目标资源受限的情况除外。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。我们建议您为 FailedInvocations 指标创建 CloudWatch 警报。

某些事件从未传送到我的目标

如果 CloudWatch Events 规则的目标长时间受限，CloudWatch Events 不会重试传输。例如，如果该目标未预配置为处理传入事件流量且目标服务当前限制 CloudWatch Events 代表您发出的请求，则 CloudWatch Events 不会重试传输。

我的规则在回应一个事件时被多次触发。CloudWatch Events 提供了什么有关触发规则或传输事件到目标的保证？

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。

防止无限循环

在 CloudWatch Events 中，可能创建导致无限循环的规则，即反复触发一个规则。例如，某规则可能检测到 S3 存储桶上的 ACL 已更改，然后触发软件以将 ACL 更改为所需状态。如果编写该规则时不小心，则 ACL 的后续更改将再次触发该规则，从而产生无限循环。

为防止出现这种情况，请在编写规则时使触发的操作不会重复激发同一规则。例如，您的规则可能仅在发现 ACL 处于错误状态时而不是在进行任何更改之后激发。

无限循环可能快速导致费用超出预期。我们建议您使用预算功能，以便在费用超出您指定的配额时提醒您。有关更多信息，请参阅[通过预算管理成本](#)。

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": "*"
}
```

正在触发我的规则，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 [Amazon SNS 设置主题属性](#) 命令从该主题删除权限。

我可以对 CloudWatch Events 使用哪种 IAM 条件密 钥

CloudWatch Events 支持 AWS 范围内的条件密钥（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密钥。

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.aws.amazon.com/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 (下一步)。
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. 在操作下面，为每当此警报选择状态为“警报”。
9. 对于 Send notification to，选择一个现有 Amazon SNS 主题或创建一个新 SNS 主题。要创建新主题，请选择新建列表。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

Note

Amazon EventBridge 是管理事件的首选方式。CloudWatch Events 和 EventBridge 是相同的底层服务和 API，但 EventBridge 提供了更多功能。您在 CloudWatch 或 EventBridge 中所作的更改将显示在每个控制台中。有关更多信息，请参阅 [Amazon EventBridge](#)。

下表列出了从 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	更新-历史记录-描述	更新-历史记录-日期
对标签的支持 (p. 109)	现在您可以标记某些 CloudWatch Events 资源。有关更多信息，请参阅 Amazon CloudWatch Events 用户指南中的标记 CloudWatch Events 资源 。	2019 年 3 月 21 日
对 Amazon VPC 终端节点的支持 (p. 109)	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 Interface VPC Endpoints 结合使用 。	2018 年 6 月 28 日

下表介绍了 Amazon CloudWatch Events 用户指南的重要更改。

更改	描述	发行日期
CodeBuild 作为目标	添加了 CodeBuild 作为事件规则的目标。有关更多信息，请参阅 教程：使用 CodeBuild 安排自动构建 (p. 26) 。	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。有关事件的更多信息，请参阅 AWS Batch 事件 。	2017 年 9 月 8 日
CodePipeline 和 AWS Glue 事件	增加了对 CodePipeline 和 AWS Glue 的事件的支持。有关更多信息，请参阅 CodePipeline 事件 (p. 39) 和 AWS Glue 事件 (p. 51) 。	2017 年 9 月 8 日
CodeBuild 和 CodeCommit 事件	增加了对 CodeBuild 和 CodeCommit 的事件的支持。有关更多信息，请参阅 CodeBuild 事件 (p. 38) 。	2017 年 8 月 3 日
支持的额外目标	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	AWS 账户可以向其他 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. 80) 。	2017 年 6 月 29 日

更改	描述	发行日期
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. 5)。	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. 42)。	2017 年 3 月 7 日
AWS 运行状况事件	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. 56)。	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 Elastic Container Service 事件 (p. 41)。	2016 年 11 月 21 日
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. 76)。	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. 40)。	2016 年 11 月 14 日
AWS CodeDeploy 事件	增加了对 CodeDeploy 事件的支持。有关更多信息，请参阅 AWS CodeDeploy 事件 (p. 38)。	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. 28) 和 Rate 表达式 (p. 30)。	2016 年 4 月 19 日
Amazon Simple Queue Service 队列作为目标	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 3 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. 40)。	2016 年 2 月 24 日
新增服务	CloudWatch Events 的初始版本。	2016 年 1 月 14 日

AWS词汇表

有关最新AWS术语，请参阅AWS一般参考中的[AWS术语表](#)。