



用户指南

EventBridge 调度器



EventBridge 调度器: 用户指南

Copyright © 2024 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、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其它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 EventBridge 调度器？	1
EventBridge 调度器的主要特点	1
访问 EventBridge 调度器	1
设置	2
注册AWS	2
创建 IAM 用户	2
使用托管策略	3
设置执行角色	4
设置目标	7
接下来做什么？	10
开始使用	11
先决条件	11
使用控制台	12
使用 AWS CLI	15
使用 软件开发工具包	15
接下来做什么？	17
计划类型	18
基于速率的计划	18
语法	19
示例	19
基于 Cron 的计划	19
语法	20
示例	21
一次性计划	21
语法	21
示例	22
时区	22
夏令时	22
管理计划	24
更改计划状态	24
配置灵活的时间窗口	26
配置死信队列	27
创建 Amazon SQS 队列	27
设置执行角色权限	28

指定死信队列	29
检索死信事件	30
删除计划	33
计划完成后删除	33
手动删除	34
接下来做什么？	34
管理计划组	35
创建计划组	35
第一步：创建新计划组	36
关联计划	37
删除计划组	38
相关资源	40
管理目标	41
使用模板化目标	41
Amazon SQS SendMessage	42
Lambda Invoke	44
Step Functions StartExecution	46
使用通用目标	48
不支持的操作	49
示例	50
添加上下文属性	52
接下来做什么？	53
安全性	54
管理访问权限	54
受众	55
使用身份进行身份验证	55
使用策略管理访问	58
EventBridge 调度器如何与 IAM 配合使用	60
使用基于身份的策略	66
混淆代理问题防范	76
故障排除	77
数据保护	79
静态加密	80
传输中加密	87
合规性验证	87
弹性	88

基础设施安全性	88
监控和指标	89
使用 CloudWatch 进行监控	89
术语	90
维度	90
访问指标	90
指标的列表	91
使用 CloudTrail 日志进行监控	95
CloudTrail 中的 EventBridge 调度器信息	95
了解如何使用 EventBridge 调度器日志文件条目	96
限额	97
文档历史记录	101
.....	ciii

什么是 Amazon EventBridge 调度器？

Amazon EventBridge 调度器是一个无服务器调度器，使您能够从一个中央托管服务创建、运行和管理任务。EventBridge 调度器高度可扩展，允许您调度数百万个任务，这些任务可以调用超过 270 个 AWS 服务和超过 6000 个 API 操作。EventBridge 调度器让您无需配置和管理基础设施，也无需与多种服务集成，即可大规模交付计划并降低维护成本。

EventBridge 调度器通过内置机制根据下游目标的可用性调整计划，可靠地交付您的任务。借助 EventBridge 调度器，您可以使用 cron 和 rate 表达式为定期模式创建计划，也可以配置一次性调用。您可以设置灵活的传输时间窗口、定义重试限制，并为失败的触发器设置最大保留时间。

主题

- [EventBridge 调度器的主要特点](#)
- [访问 EventBridge 调度器](#)

EventBridge 调度器的主要特点

EventBridge 调度器提供以下关键功能，可用于配置目标和扩展计划。

- **模板化目标：**EventBridge 调度器支持模板化目标使用 Amazon SQS、Amazon SNS、Lambda 和 EventBridge 执行常见的 API 操作。通过预定义的目标，您可以使用 EventBridge 调度器控制台、EventBridge 调度器开发工具包或 AWS CLI，快速配置您的计划。
- **通用目标：**EventBridge 调度器提供了一个通用目标参数 (UTP)，可用于创建自定义触发器，该触发器在一个计划上将超过 270 个 AWS 服务和超过 6,000 个 API 操作设为目标。使用 UTP，您可以使用事件 EventBridge 调度器控制台、EventBridge 调度器开发工具包或 AWS CLI。
- **灵活的时间窗口：**EventBridge 调度器支持灵活的时间窗口，允许您分散计划，提高触发器的可靠性，适用于不需要通过精确计划来调用目标的用例。
- **重试：**EventBridge 调度器向目标提供至少一次的事件传输，这意味着至少有一次传输成功并得到目标的响应。EventBridge 调度器允许您为失败的任务设置计划的重试次数。EventBridge 调度器会重试有延迟尝试的失败任务，以提高计划的可靠性并确保目标可用。

访问 EventBridge 调度器

您可以通过 EventBridge 调度器控制台、EventBridge 调度器开发工具包或 AWS CLI 直接使用 EventBridge 调度器 API 来使用 EventBridge 调度器。

设置 Amazon EventBridge 调度器

必须先完成以下步骤，然后才能使用 EventBridge 调度器。

主题

- [注册AWS](#)
- [创建 IAM 用户](#)
- [使用托管策略](#)
- [设置执行角色](#)
- [设置目标](#)
- [接下来做什么？](#)

注册AWS

如果您还没有 AWS 账户，请完成以下步骤创建一个。

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。

当您注册 AWS 账户时，系统将会创建一个 AWS 账户根用户。根用户有权访问该账户中的所有 AWS 服务和资源。作为安全最佳实践，请 [为管理用户分配管理访问权限](#)，并且只使用根用户执行 [需要根用户访问权限的任务](#)。

创建 IAM 用户

要创建管理员用户，请选择以下选项之一。

选择一种方法来管理您的管理员	目的	方式	您也可以
在 IAM Identity Center 中 (建议)	使用短期凭证访问 AWS。 这符合安全最佳实践。有关最佳实践的信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践 。	有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 入门 。	按照《AWS Command Line Interface 用户指南》中的 配置 AWS CLI 以使用 AWS IAM Identity Center ，配置程式访问。
在 IAM 中 (不推荐使用)	使用长期凭证访问 AWS。	按照《IAM 用户指南》中的 创建您的首个 IAM 管理员用户和组 的说明操作。	按照《IAM 用户指南》中的 管理 IAM 用户的访问密钥 ，配置程式访问。

使用托管策略

在上一步中，您设置了一个 IAM 用户，该用户拥有访问您的 AWS 资源的凭证。在大多数情况下，为了安全地使用 EventBridge 调度器，我们建议您创建单独的用户、组或角色，这些用户、组或角色仅具有使用 EventBridge 调度器所需的权限。EventBridge 调度器支持适用于常见使用案例的以下托管策略。

- [the section called “AmazonEventBridgeSchedulerFullAccess”](#)：使用控制台和 API 授予对 EventBridge 调度器的完全访问权限。
- [the section called “AmazonEventBridgeSchedulerReadOnlyAccess”](#)：授予对 EventBridge 调度器的只读访问权限。

您可以像在上一步中附加 AdministratorAccess 策略一样将这些托管策略附加到您的 IAM 主体。有关使用基于身份的 IAM 策略管理对 EventBridge 调度器的访问的更多信息，请参阅[the section called “使用基于身份的策略”](#)。

设置执行角色

执行角色是 EventBridge 调度器代入的 IAM 角色，用于代表您与其他 AWS 服务 进行交互。您可以为该角色附加权限策略，以授予 EventBridge 调度器调用目标的访问权限。

在使用控制台[创建新计划](#)时，也可以创建新的执行角色。如果您使用控制台，EventBridge 调度器会根据您选择的目标代表您创建一个具有权限的角色。当 EventBridge 调度器为您创建角色时，该角色的信任策略包括限制哪些主体可以代表您代入该角色的[条件键](#)。这样可以防范潜在的[混淆代理安全问题](#)。

以下步骤介绍如何创建新的执行角色以及如何授予 EventBridge 调度器调用目标的访问权限。本主题介绍常用模板化目标的权限。有关为其他目标添加权限的信息，请参阅 [the section called “使用模板化目标”](#)。

使用 AWS CLI 创建执行角色

1. 复制以下代入角色 JSON 策略并将其另存为本地 Scheduler-Execution-Role.json。此信任策略允许 EventBridge 调度器代表您代入角色。

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

Important

要在生产环境中设置执行角色，我们建议实施额外的保护措施，以防止出现混淆代理问题。有关更多信息和示例策略，请参阅 [the section called “混淆代理问题防范”](#)。

2. 从 AWS Command Line Interface (AWS CLI) 键入以下命令以创建新角色。将 *SchedulerExecutionRole* 替换为您要赋予此角色的名称。

```
$ aws iam create-role --role-name SchedulerExecutionRole --assume-role-policy-document file://Scheduler-Execution-Role.json
```

如果成功，将会看到以下输出：

```
{
  "Role": {
    "Path": "/",
    "RoleName": "Scheduler-Execution-Role",
    "RoleId": "BR1L2DZK3K4CTL5ZF9EIL",
    "Arn": "arn:aws:iam::123456789012:role/SchedulerExecutionRole",
    "CreateDate": "2022-03-10T18:45:01+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "scheduler.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  }
}
```

3. 要创建允许 EventBridge 调度器调用目标的新策略，请选择以下常用目标之一。复制 JSON 权限策略并将其作为 .json 文件保存在本地。

Amazon SQS – SendMessage

以下内容允许 EventBridge 调度器对您账户中的所有 Amazon SQS 队列调用 `sqs:SendMessage` 操作。

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

```
        "Action": [
            "sqs:SendMessage"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
]
```

Amazon SNS – Publish

以下内容允许 EventBridge 调度器对您账户中的所有 Amazon SNS 主题调用 `sns:Publish` 操作。

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

Lambda – Invoke

以下内容允许 EventBridge 调度器对您账户中的所有 Lambda 函数调用 `lambda:InvokeFunction` 操作。

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

```
]
}
```

4. 要创建新的权限策略，请运行以下命令。将 *PolicyName* 替换为您要赋予此策略的名称。

```
$ aws iam create-policy --policy-name PolicyName --policy-document file://
PermissionPolicy.json
```

如果成功，将会看到以下输出。记下策略 ARN。您将在下一步中使用此 ARN 来将策略关联到我们的执行角色。

```
{
  "Policy": {
    "PolicyName": "PolicyName",
    "CreateDate": "2022-03-01T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/PolicyName",
    "UpdateDate": "2022-03-01T19:31:18.620Z"
  }
}
```

5. 要将该策略附加到您的执行角色，请运行以下命令。将 *your-policy-arn* 替换为您在上一步中创建的策略的 ARN。将 *SchedulerExecutionRole* 替换为您执行角色的名称。

```
$ aws iam attach-role-policy --policy-arn your-policy-arn --role-
name SchedulerExecutionRole
```

该 `attach-role-policy` 操作不会在命令行上返回响应。

设置目标

在创建 EventBridge 调度器计划之前，您需要至少有一个目标供计划调用。您可以使用现有的 AWS 资源或创建一个新的资源。以下步骤显示如何使用 AWS CloudFormation 创建新的标准 Amazon SQS 队列。

创建新的 Amazon SQS 队列

1. 复制以下 JSON AWS CloudFormation 模板并将其另存为本地 SchedulerTargetSQS.json。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyQueue": {
      "Type": "AWS::SQS::Queue",
      "Properties": {
        "QueueName": "MyQueue"
      }
    }
  },
  "Outputs": {
    "QueueName": {
      "Description": "The name of the queue",
      "Value": {
        "Fn::GetAtt": [
          "MyQueue",
          "QueueName"
        ]
      }
    },
    "QueueURL": {
      "Description": "The URL of the queue",
      "Value": {
        "Ref": "MyQueue"
      }
    },
    "QueueARN": {
      "Description": "The ARN of the queue",
      "Value": {
        "Fn::GetAtt": [
          "MyQueue",
          "Arn"
        ]
      }
    }
  }
}
```

- 在 AWS CLI 中，运行以下命令以根据 Scheduler-Target-SQS.json 模板创建 AWS CloudFormation 堆栈。

```
$ aws cloudformation create-stack --stack-name Scheduler-Target-SQS --template-body file://Scheduler-Target-SQS.json
```

如果成功，将会看到以下输出：

```
{
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/Scheduler-Target-SQS/1d2af345-a121-12eb-abc1-012e34567890"
}
```

- 运行以下命令以查看您 AWS CloudFormation 堆栈的摘要信息。此信息包括堆栈的状态和模板中指定的输出。

```
$ aws cloudformation describe-stacks --stack-name Scheduler-Target-SQS
```

如果成功，该命令会创建 Amazon SQS 队列并返回以下输出：

```
{
  "Stacks": [
    {
      "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/Scheduler-Target-SQS/1d2af345-a121-12eb-abc1-012e34567890",
      "StackName": "Scheduler-Target-SQS",
      "CreationTime": "2022-03-17T16:21:29.442000+00:00",
      "RollbackConfiguration": {},
      "StackStatus": "CREATE_COMPLETE",
      "DisableRollback": false,
      "NotificationARNs": [],
      "Outputs": [
        {
          "OutputKey": "QueueName",
          "OutputValue": "MyQueue",
          "Description": "The name of the queue"
        },
        {
          "OutputKey": "QueueARN",
          "OutputValue": "arn:aws:sqs:us-west-2:123456789012:MyQueue",
          "Description": "The ARN of the queue"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "OutputKey": "QueueURL",
      "OutputValue": "https://sqs.us-
west-2.amazonaws.com/123456789012/MyQueue",
      "Description": "The URL of the queue"
    }
  ],
  "Tags": [],
  "EnableTerminationProtection": false,
  "DriftInformation": {
    "StackDriftStatus": "NOT_CHECKED"
  }
}
]
```

在本指南的后面部分，您将使用 QueueARN 的值将队列设置为 EventBridge 调度器的目标。

接下来做什么？

完成设置步骤后，使用[入门](#)指南创建您的第一个 EventBridge 调度器计划并调用目标。

EventBridge 调度器入门

本主题介绍如何创建新的 EventBridge 调度器计划。您可以使用 EventBridge 调度器控制台、AWS Command Line Interface (AWS CLI) 或 AWS 软件开发工具包来创建带有模板化 Amazon SQS 目标的计划。然后，您将设置日志记录，配置重试次数，并为失败的任务设置最长保留时间。创建计划后，您将验证您的计划是否成功调用了目标并向目标队列发送了消息。

Note

要遵循本指南，我们建议您使用 [the section called “使用基于身份的策略”](#) 中所述的最低要求权限设置 IAM 用户。创建和配置用户后，运行以下命令来设置您的访问凭证。要配置 AWS CLI，您需要访问密钥 ID 和秘密访问密钥。

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

有关设置凭证的不同方式的更多信息，请参阅《版本 2 AWS Command Line Interface 用户指南》中的 [配置设置和优先级](#)。

主题

- [先决条件](#)
- [使用 EventBridge 调度器控制台创建计划](#)
- [使用 AWS CLI 创建计划](#)
- [使用 EventBridge 调度器开发工具包创建计划](#)
- [接下来做什么？](#)

先决条件

在尝试本部分中的步骤之前，您必须执行以下操作：

- 完成 [设置](#) 中所述的任务

使用 EventBridge 调度器控制台创建计划

要使用控制台创建新计划

1. 登录 AWS Management Console，然后选择以下链接打开 EventBridge 控制台的 EventBridge 调度器部分：<https://us-west-2.console.aws.amazon.com/scheduler/home?region=us-west-2#home>

Note

您可以使用 AWS Management Console 的区域选择器切换 AWS 区域。

2. 在计划页面，选择创建计划。
3. 在指定计划详细信息页面，在计划名称和描述部分中，执行以下操作：
 - a. 对于计划名称，输入计划的名称。例如 **MyTestSchedule**
 - b. 对于描述（可选），输入对计划的描述。例如，**My first schedule**。
 - c. 对于计划组，从下拉选项选择一个计划组。如果您之前没有创建过任何计划组，则可以为您的计划选择 default 组。要创建新的计划组，请在控制台描述中选择创建自己的计划链接。您可以使用计划组将标签添加到计划组。
4. 在计划模式部分执行以下操作：
 - a. 在出现中，请选择以下模式选项之一。配置选项会根据您选择的模式而变化。

- 一次性计划：一次性计划仅在您指定的日期和时间调用一次目标。

在日期和时间中，以 YYYY/MM/DD 格式输入有效的日期。然后，指定 24 小时 hh:mm 格式的时间戳。最后，从下拉选项选择一个时区。

- 定期计划：定期计划按照您使用 cron 表达式或 rate 表达式指定的速率调用目标。

选择基于 Cron 的计划以使用 cron 表达式配置计划。要使用 rate 表达式，请选择基于速率的计划并在值中输入正数，然后从下拉选项选择一个单位。

有关使用 cron 和 rate 表达式的更多信息，请参阅 [计划类型](#)。

- b. 对于灵活的时间窗口，选择关闭以关闭该选项，或者从下拉列表选择一个预定义的时间窗口。例如，如果您选择 15 分钟并且将定期计划设置为每小时调用一次其目标，则该计划将在每小时开始后的 15 分钟内运行。

5.

Note

弹性时间窗功能不适用于一次性计划。

如果您在上一步中选择了定期计划，则在时间范围部分中，指定时区，也可以设置计划的开始日期和时间以及结束日期和时间。没有开始日期的定期计划将在创建并可用后立即开始。没有结束日期的定期计划将继续无限期地调用其目标。

6. 选择下一步。

7. 在选择目标页面上，执行以下操作：

- a. 选择模板化目标并选择目标 API。在本示例中，我们将选择 Amazon SQS **SendMessage** 模板化目标。
- b. 在发送消息部分，对于 SQS 队列，从下拉列表中选择一个现有的 Amazon SQS 队列 ARN，例如 `arn:aws:sqs:us-west-2:123456789012:TestQueue`。要创建新队列，请选择创建新 SQS 队列以导航到 Amazon SQS 控制台。队列创建完毕后，返回 EventBridge 调度器控制台并刷新下拉列表。您的新队列 ARN 将显示并可供选择。
- c. 在目标中，输入您希望 EventBridge 调度器传送到目标的有效负载。在本示例中，将向目标队列发送以下消息：**Hello, it's EventBridge Scheduler.**

8. 选择下一步，然后在设置（可选）页面上，执行以下操作：

9.

- a. 在计划状态部分中，对于启用计划，使用开关开启或关闭功能。默认情况下，EventBridge 调度器会启用您的计划。
- b. 在计划完成后的操作部分，配置 EventBridge 调度器在计划完成后采取的操作：
 - 如果要自动删除计划，请选择删除。对于一次性计划，这种情况发生在计划调用一次目标之后。对于定期计划，这发生在计划的最后一次计划调用之后。有关自动删除的更多信息，请参阅 [the section called “计划完成后删除”](#)。
 - 如果您不想让 EventBridge 调度器在计划完成后执行任何操作，请选择无或不选择任何值。
- c. 在重试策略和死信队列 (DLQ) 部分中，在重试策略中，打开重试，为计划配置重试策略。配置重试策略后，如果计划未能调用其目标，EventBridge 调度器将重新运行该计划。如果已配置，则必须为计划设置最长保留时间和最大重试次数。
- d. 对于事件的最长期限（可选），输入 EventBridge 调度器必须保留未处理事件的最长小时数和分钟数。

Note

最大值为 24 小时。

- e. 对于最大重试次数，输入在目标返回错误的情况下，EventBridge 调度器重试计划的最大次数。

Note

最大值为 185 次重试。

- f. 对于死信队列 (DLQ)，请从以下选项中选择：
- 无：如果您不想配置 DLQ，请选择此选项。
 - 在我的 AWS 账户中选择一个 Amazon SQS 队列作为 DLQ：选择此选项，然后从下拉列表选择一个队列 ARN，使用与您创建计划时相同的 AWS 账户配置 DLQ。
 - 将其他 AWS 账户中的 Amazon SQS 队列指定为 DLQ：选择此选项，然后输入配置为 DLQ 的队列的 ARN（如果队列位于另一个 AWS 账户中）。要使用此选项，您必须为队列输入确切的 ARN。
- g. 要使用客户托管 KMS 密钥加密目标输入，在加密部分中，选择自定义加密设置（高级）。如果选择此选项，请输入现有的 KMS 密钥 ARN 或选择创建一个 AWS KMS 密钥以导航到 AWS KMS 控制台。有关 EventBridge 调度器如何加密静态数据的更多信息，请参阅 [the section called “静态加密”](#)。
- h. 对于权限，选择使用现有角色，然后从下拉列表中选择您在[设置](#)过程中创建的角色。您也可以选择前往 IAM 控制台来创建新角色。

如果要让 EventBridge 调度器为您创建新的执行角色，请选择为此计划创建新角色。然后，在角色名称中输入名称。如果您选择此选项，EventBridge 调度器会将模板化目标所需的必要权限添加到该角色。

10. 选择下一步。
11. 在查看并创建计划页面上，查看计划的详细信息。在每个部分中，选择编辑返回到该步骤并编辑其详细信息。
12. 选择创建计划以完成新计划的创建。您可以在计划页面上查看新的和现有的计划列表。在状态列下，验证新计划是否已启用。
13. 要验证您的计划是否调用了 Amazon SQS 目标，请打开 Amazon SQS 控制台并执行以下操作：

- a. 从队列列表中选择目标队列。
- b. 选择 Send and receive messages (发送和接收消息) 。
- c. 在发送和接收消息页面的接收消息下，选择轮询消息以检索您的计划发送到目标队列的测试消息。

使用 AWS CLI 创建计划

以下示例说明如何使用 AWS CLI 命令 [create-schedule](#) 创建带有模板化的 Amazon SQS 目标的 EventBridge 调度器计划。在下面的命令中，将占位符的值替换为您自己的信息：

- `--name`：为计划输入一个名称。
- `RoleArn`：输入要与计划关联的执行角色的 ARN。
- `Arn`：输入目标的 ARN。在本例中，目标为 Amazon SQS 队列。
- `Input`：输入 EventBridge 调度器传送到目标队列的消息。

```
$ aws scheduler create-schedule --name sqs-templated-schedule --schedule-expression  
'rate(5 minutes)' \  
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \  
--flexible-time-window '{ "Mode": "OFF" }'
```

使用 EventBridge 调度器开发工具包创建计划

在以下示例中，您将使用 EventBridge 调度器开发工具包创建带有模板化 Amazon SQS 目标的 EventBridge 调度器计划。

Example Python 开发工具包

```
import boto3  
scheduler = boto3.client('scheduler')  
  
flex_window = { "Mode": "OFF" }  
  
sqs_templated = {  
    "RoleArn": "<ROLE_ARN>",  
    "Arn": "<QUEUE_ARN>",
```

```
"Input": "Message for scheduleArn: '<aws.scheduler.schedule-arn>', scheduledTime:
'<aws.scheduler.scheduled-time>'
}

scheduler.create_schedule(
    Name="sqs-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

Example Java 软件开发工具包

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;

public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
            .region(Region.US_WEST_2)
            .build();

        Target sqsTarget = Target.builder()
            .roleArn("<ROLE_ARN>")
            .arn("<QUEUE_ARN>")
            .input("Message for scheduleArn: '<aws.scheduler.schedule-arn>',
scheduledTime: '<aws.scheduler.scheduled-time>'")
            .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
            .name("<SCHEDULE_NAME>")
            .scheduleExpression("rate(10 minutes)")
            .target(sqsTarget)
            .flexibleTimeWindow(FlexibleTimeWindow.builder()
                .mode(FlexibleTimeWindowMode.OFF)
                .build())
            .build();
```

```
    client.createSchedule(createScheduleRequest);
    System.out.println("Created schedule with rate expression and an Amazon SQS
templated target");
  }
}
```

接下来做什么？

- 有关使用控制台、AWS CLI 或 EventBridge 调度器开发工具包管理计划的更多信息，请参阅 [管理计划](#)。
- 有关如何配置模板化目标和了解如何使用通用目标参数的更多信息，请参阅 [管理目标](#)。
- 有关 EventBridge 调度器数据类型和 API 操作的更多信息，请参阅 [EventBridge 调度器 API 参考](#)。

EventBridge 调度器上的计划类型

以下主题介绍了 Amazon EventBridge 调度器支持的不同计划类型，以及 EventBridge 调度器如何处理夏令时和不同时区的计划。配置计划时，您可以从三种计划类型中进行选择：基于速率的计划、基于 cron 的计划和一次性的计划。

基于速率的计划和基于 cron 的计划都是定期计划。您可以为要配置的计划类型使用计划表达式来配置每种定期计划类型，并指定 EventBridge 调度器评估该表达式的时区。

一次性计划仅调用一次目标。您可以通过指定 EventBridge 调度器评估计划的时间、日期和时区来配置一次性计划。

Note

EventBridge 调度器上的所有计划类型都以 60 秒的精度调用其目标。这意味着，如果您将计划设置为在 1:00 运行，它将在 1:00:00 和 1:00:59 之间调用目标 API。

使用以下各节来了解如何为每种定期计划类型配置计划表达式，以及如何在 EventBridge 调度器上设置一次性计划。

主题

- [基于速率的计划](#)
- [基于 Cron 的计划](#)
- [一次性计划](#)
- [EventBridge 调度器上的时区](#)
- [EventBridge 调度器上的夏令时](#)

基于速率的计划

基于速率的计划在您为计划指定的开始日期之后开始，并以您定义的固定速率运行，直到计划的结束日期。您可以使用基于速率的计划来设置最常见的定期计划用例。例如，如果您想要一个每 15 分钟、每两小时或每五天调用一次目标的计划，则可以使用基于速率的计划来实现这一点。您可以使用 rate 表达式配置基于速率的计划。

对于基于速率的计划，您可以使用 [StartDate](#) 属性来设置该计划的首次出现时间。如果您没有为基于速率的计划提供 StartDate，则您的计划会立即开始调用目标。

rate 表达式有两个必填字段，中间用空格分隔，如下所示。

语法

```
rate(value unit)
```

value

正数。

单位

您希望计划调用其目标的时间单位。

有效输入：minutes | hours | days

示例

以下示例说明如何使用 rate 表达式和 AWS CLI create-schedule 命令来配置基于速率的计划。此示例创建了一个计划，该计划每五分钟运行一次，并使用模板化的 SqsParameters 目标类型向 Amazon SQS 队列传送一条消息。

由于此示例未为 --start-date 参数设置值，因此计划会在您创建并激活计划后立即开始调用其目标。

```
$ aws scheduler create-schedule --schedule-expression 'rate(5 minutes)' --  
name schedule-name \  
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \  
--flexible-time-window '{ "Mode": "OFF" }'
```

基于 Cron 的计划

cron 表达式会创建精细的定期计划，该计划在您选择的特定时间运行。EventBridge 调度器支持按协调世界时 (UTC) 或您在创建计划时指定的时区配置基于 cron 的计划。使用基于 cron 的计划，您可以更好地控制计划运行的时间和频率。如果您需要自定义的定期计划，但 EventBridge 调度器的某个 rate 表达式不支持，请使用基于 cron 的计划。例如，您可以创建早上 8:00 运行的基于 cron 的计划。PST，每个月的第一个星期一。您可以使用 cron 表达式配置基于 cron 的计划。

cron 表达式由五个用空格分隔的必填字段组成：分钟、小时、月份、星期和一个可选字段“年”，如下所示。

语法

```
cron(minutes hours day-of-month month day-of-week year)
```

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

通配符

- , (逗号) 通配符包含其他值。在“月份”字段中，JAN、FEB 和 MAR 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 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“Hours (小时)”字段中，* 包括每个小时。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 ?。
- / (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) 通配符用于指定任何内容。在“日期”字段中，您可以输入 7，如果一周的任何日期都是可接受的，则可以在“日期”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。

示例

以下示例说明如何使用 cron 表达式和 AWS CLI `create-schedule` 命令来配置基于 cron 的计划。此示例创建了一个计划，该计划在 2022 年至 2023 年期间每个月的最后一个星期五上午 10:15 (UTC+0) 运行，并使用模板化的 SqsParameters 目标类型向 Amazon SQS 队列发送消息。

```
$ aws scheduler create-schedule --schedule-expression "cron(15 10 ? * 6L 2022-2023)" --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--flexible-time-window '{ "Mode": "OFF" }'
```

一次性计划

一次性计划将仅在您使用有效日期和时间戳指定的日期和时间调用一次目标。EventBridge 调度器支持按世界协调时 (UTC) 或您在创建计划时指定的时区安排计划。

Note

一次性计划在完成运行和调用目标后，仍会计入您的账户配额。我们建议在一次性计划完成运行后将其[删除](#)。

您可以使用 `at` 表达式配置一次性计划。`at` 表达式由您希望 EventBridge 调度器调用您的计划的日期和时间组成，如下所示。

语法

```
at(yyyy-mm-ddThh:mm:ss)
```

配置一次性计划时，EventBridge 调度器会忽略您为计划指定的 `StartDate` 和 `EndDate`。

示例

以下示例说明如何使用 `at` 表达式和 AWS CLI `create-schedule` 命令来配置一次性计划。此示例创建了一个计划，该计划在 2022 年 11 月 20 日下午 1 点 (UTC-8) 运行一次，并使用模板化的 `SqsParameters` 目标类型向 Amazon SQS 队列发送消息。

```
$ aws scheduler create-schedule --schedule-expression "at(2022-11-20T13:00:00)" --
name schedule-name \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }' \
--schedule-expression-timezone "America/Los_Angeles"
--flexible-time-window '{ "Mode": "OFF" }'
```

EventBridge 调度器上的时区

EventBridge 调度器支持在您指定的任何时区配置基于 cron 的计划和一次性计划。EventBridge 调度器使用由互联网编号分配局 (IANA) 维护的[时区数据库](#)。

借助 AWS CLI，您可以使用 `--schedule-expression-timezone` 参数设置希望 EventBridge 调度器用于评估您的计划的时区。例如，以下命令创建一个基于 cron 的计划，该计划每天上午 8:30 调用 `America/New_York` 的模板化 Amazon SQS `SendMessage` 目标。

```
$ aws scheduler create-schedule --schedule-expression "cron(30 8 * * ? *)" --name
schedule-in-est \
--target '{"RoleArn": "role-arn", "Arn": "QUEUE_ARN", "Input": "This schedule runs
in the America/New_York time zone." }' \
--schedule-expression-timezone "America/New_York"
--flexible-time-window '{ "Mode": "OFF" }'
```

EventBridge 调度器上的夏令时

EventBridge 调度器会根据夏令时自动调整您的计划。当春季时间向前移动时，如果 cron 表达式落在不存在的日期和时间上，则会跳过您的计划调用。当秋季时间向后移动时，您的计划只运行一次，并且不会重复调用。以下调用通常发生在指定的日期和时间。

EventBridge 调度器会根据您在创建计划时指定的时区来调整您的计划。如果您在 `America/New_York` 中配置计划，则当该时区的时间变化时，您的计划会调整，而 `America/Los_Angeles` 的计划将在三小时后当西海岸的时间发生变化时进行调整。

对于以 days 为单位的基于速率的计划，例如 `rate(1 days)`，days 表示 24 小时的持续时间。这意味着，当夏令时导致一天缩短至 23 小时或延长至 25 小时时，EventBridge 调度器仍会在计划最后一次调用发生后 24 小时后评估 rate 表达式。

Note

根据当地规章制度，某些时区不实行夏令时。如果您在不实行夏令时的时区创建计划，则 EventBridge 调度器不会调整您的计划。夏令时调整不适用于世界协调时 (UTC) 的计划。

示例

考虑一个场景，即在 `America/Los_Angeles` 中使用以下 cron 表达式创建计划：`cron(30 2 * * ? *)`。此计划每天凌晨 2:30 在指定时区运行。

- 春季向前调整：当春季的时间从凌晨 1:59 向前移动到凌晨 3:00 时，EventBridge 调度器会跳过当天的计划调用，并在第二天恢复并正常运行计划。
- 秋季向后调整：当秋季时间从凌晨 2:59 向后移动到凌晨 2:00 时，EventBridge 调度器在时间制度变化前的凌晨 2:30 只运行一次计划，但不会在时间制度变化后的凌晨 2:30 再次重复调用计划。

管理计划

计划是您使用 Amazon EventBridge 调度器创建、配置和管理的主要资源。

每个计划都有一个计划表达式，用于确定计划运行的时间和频率。EventBridge 调度器支持三种类型的计划：基于速率、基于 cron 和一次性计划。有关不同计划类型的更多信息，请参阅 [计划类型](#)。

创建计划时，需要配置计划要调用的目标。目标是 API 操作，每当您的计划运行时，EventBridge 调度器都会代表您调用该操作。EventBridge 调度器支持两种类型的目标：模板化目标在核心服务组中调用常用 API 操作，以及通用目标参数 (UTP) 可用于在超过 270 个服务中调用 6,000 多个操作。有关配置目标的更多信息，请参阅 [管理目标](#)。

使用重试策略和死信队列 (DLQ) 这两种主要机制，您可以配置计划在 EventBridge 调度器无法成功将事件传送到目标时如何处理故障。重试策略决定了 EventBridge 调度器必须重试失败事件的次数，以及未处理的事件保留多长时间。DLQ 是标准的 Amazon SQS 队列，在重试策略用尽后，EventBridge 调度器用来传送失败的事件。您可以使用 DLQ 来解决您的计划或其下游目标的问题。有关更多信息，请参阅 [the section called “配置死信队列”](#)。

在本节中，您可以找到使用控制台、AWS CLI 和 EventBridge 调度器开发工具包管理 EventBridge 调度器计划的示例。

主题

- [更改计划状态](#)
- [配置灵活的时间窗口](#)
- [为计划配置死信队列](#)
- [删除计划](#)
- [接下来做什么？](#)

更改计划状态

EventBridge 调度器计划有两种状态：启用和禁用。以下示例使用 UpdateSchedule 禁用每五分钟触发一次并调用 Lambda 目标的计划。

使用 UpdateSchedule 时，必须提供所有必需的参数。EventBridge 调度器会用您提供的信息替换您的计划。如果不指定之前已设置的参数，则默认为 null。

Example AWS CLI

```
$ aws scheduler update-schedule --name lambda-universal --schedule-expression 'rate(5
minutes)' \
--target '{"RoleArn": "<ROLE_ARN>", "Arn": "arn:aws:scheduler::aws-sdk:lambda:invoke"
"Input": "{\\"FunctionName\\":\\"arn:aws:lambda:<REGION>:123456789012:function:HelloWorld
\\",\\"InvocationType\\":\\"Event\\",\\"Payload\\":\\"{\\\\"message\\\\":\\"testing function\\
\\"}\\"}" }' \
--flexible-time-window '{ "Mode": "OFF" }' \
--state DISABLED
```

```
{
  "ScheduleArn": "arn:aws:scheduler:us-west-2:123456789012:schedule/default/lambda-
universal"
}
```

以下示例使用 Python 软件开发工具包和 UpdateSchedule 操作来禁用使用模板化目标将 Amazon SQS 设为目标的计划。

Example Python 开发工具包

```
import boto3
scheduler = boto3.client('scheduler')

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "{}"}

flex_window = { "Mode": "OFF" }

scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window,
    State='DISABLED')
```

配置灵活的时间窗口

当您使用灵活的时间窗口配置计划时，EventBridge 调度器会在您设置的时间窗口内调用目标。这在不需要通过精确计划来调用目标的情况下很有用。设置灵活的时间窗口可分散目标调用，从而提高计划的可靠性。

例如，如果您为每小时运行一次的计划配置了 15 分钟的灵活时间窗口，则它会在计划时间之后的 15 分钟内调用目标。以下 AWS CLI 和 EventBridge 调度器开发工具包示例使用 UpdateSchedule 为每小时运行一次的计划设置 15 分钟的灵活时间窗口。

Note

您必须指定是否要设置灵活的时间窗口。如果不希望设置此选项，请指定 OFF。如果将该值设置为 FLEXIBLE，则必须指定计划运行的最大时间段。

Example AWS CLI

```
$ aws scheduler update-schedule --name lambda-universal --schedule-expression 'rate(1
hour)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "arn:aws:scheduler::aws-sdk:lambda:invoke"
"Input": "{\"FunctionName\": \"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\", \"InvocationType\": \"Event\", \"Payload\": \"{\\\"message\\\": \\\"testing function\\
\\\"}\" }' \
--flexible-time-window '{ "Mode": "FLEXIBLE", "MaximumWindowInMinutes": 15} \
```

```
{
  "ScheduleArn": "arn:aws:scheduler:us-west-2:123456789012:schedule/lambda-universal"
}
```

Example Python 开发工具包

```
import boto3
scheduler = boto3.client('scheduler')

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "{}"
```

```
flex_window = { "Mode": "FLEXIBLE", "MaximumWindowInMinutes": 15}

scheduler.update_schedule(Name="your-schedule",
    ScheduleExpression="rate(1 hour)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)
```

为计划配置死信队列

Amazon EventBridge 调度器支持使用 Amazon Simple Queue Service 的死信队列 (DLQ)。当计划无法调用其目标时，EventBridge 调度器会将包含调用详细信息的 JSON 有效负载和从目标收到的任何响应传送到您指定的 Amazon SQS 标准队列。

以下主题将此 JSON 称为死信事件。死信事件可让您对计划或目标的问题进行故障排除。如果您为计划配置了重试策略，则 EventBridge 调度器会传送已耗尽您设置的最大重试次数的死信事件。

以下主题描述了如何将 Amazon SQS 队列配置为计划的 DLQ，如何设置 EventBridge 调度器向 Amazon SQS 发送消息所需的权限以及如何从 DLQ 接收死信事件。

主题

- [创建 Amazon SQS 队列](#)
- [设置执行角色权限](#)
- [指定死信队列](#)
- [检索死信事件](#)

创建 Amazon SQS 队列

在为计划配置 DLQ 之前，您必须创建一个标准的 Amazon SQS 队列。有关使用 Amazon SQS 控制台创建队列的说明，请参阅《Amazon Simple Queue Service 开发人员指南》中的[创建 Amazon SQS 队列](#)。

Note

EventBridge 调度器不支持使用 FIFO 队列作为计划的 DLQ。

使用下面的 AWS CLI 命令创建标准队列。


```
$ aws sqs create-queue --queue-name queue-name
```

如果成功，将会在输出中看到 QueueURL。

```
{
  "QueueUrl": "https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-dlq-test"
}
```

创建队列后，请记下队列 ARN。当为 EventBridge 调度器计划指定 DLQ 时，需要这个 ARN。您可以在 Amazon SQS 控制台中找到您的队列 ARN，也可以使用 [get-queue-attributes](#) AWS CLI 命令来查找。

```
$ aws sqs get-queue-attributes --queue-url your-dlq-url --attribute-names QueueArn
```

如果成功，您将在输出中看到队列 ARN。

```
{
  "Attributes": {
    "QueueArn": "arn:aws:sqs:us-west-2:123456789012:scheduler-dlq-test"
  }
}
```

在下一节中，您将向您的计划执行角色添加所需的权限，以允许 EventBridge 调度器向 Amazon SQS 传送死信事件。

设置执行角色权限

要让 EventBridge 调度器向 Amazon SQS 传送死信事件，您的计划执行角色需要以下权限策略。有关为计划执行角色附加新权限策略的更多信息，请参阅[设置执行角色](#)。

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

```
    }  
  ]  
}
```

Note

如果您使用 EventBridge 调度器调用 Amazon SQS API 目标，则您的计划执行角色可能已经附加了所需的权限。

在下一节中，您将使用 EventBridge 调度器控制台为您的计划指定一个 DLQ。

指定死信队列

要指定 DLQ，请使用 EventBridge 调度器控制台或 AWS CLI 更新现有计划，或创建新计划。

Console

使用控制台指定 DLQ

1. 登录 AWS Management Console，然后选择以下链接打开 EventBridge 控制台的 EventBridge 调度器部分：<https://console.aws.amazon.com/scheduler/home>
2. 在 EventBridge 调度器控制台上，创建新计划，或从计划列表中选择现有计划进行编辑。
3. 在设置页面上，对于死信队列 (DLQ)，请执行以下操作之一：
 - 选择在我的 AWS 账户中选择一个 Amazon SQS 队列作为 DLQ，然后从下拉列表中为您的 DLQ 选择队列 ARN。
 - 选择将其他 AWS 账户中的 Amazon SQS 队列指定为 DLQ，然后输入您 DLQ 的队列 ARN。如果您选择其他 AWS 账户中的队列，则 EventBridge 调度器控制台将无法在下拉列表中显示队列 ARN。
4. 查看您的选择，然后选择创建计划或保存计划以完成 DLQ 的配置。
5. (可选) 要查看计划的 DLQ 详细信息，请从列表中选择计划名称，然后在计划详细信息页面上选择死信队列选项卡。

AWS CLI

要更新现有计划，请使用 AWS CLI

- 使用 [update-schedule](#) 命令更新您的计划。将您之前创建的 Amazon SQS 队列指定为 DLQ。将您对其附加所需 Amazon SQS 权限的 IAM 角色 ARN 指定为执行角色。将占位符的值替换为您自己的信息。

```
$ aws scheduler update-schedule --name existing-schedule \  
  --schedule-expression 'rate(5 minutes)' \  
  --target '{"DeadLetterConfig": {"Arn": "DLQ_ARN"}, "RoleArn": "ROLE_ARN",  
  "Arn": "QUEUE_ARN", "Input": "Hello world!" }' \  
  --flexible-time-window '{ "Mode": "OFF"}
```

通过 AWS CLI 使用 DLQ 创建新计划

- 要创建计划，请使用 [create-schedule](#) 命令。将占位符值替换为您自己的信息。

```
$ aws scheduler create-schedule --name new-schedule \  
  --schedule-expression 'rate(5 minutes)' \  
  --target '{"DeadLetterConfig": {"Arn": "DLQ_ARN"}, "RoleArn": "ROLE_ARN",  
  "Arn": "QUEUE_ARN", "Input": "Hello world!" }' \  
  --flexible-time-window '{ "Mode": "OFF"}
```

在下一个部分中，您将使用 AWS CLI 接收来自 DLQ 的死信事件。

检索死信事件

使用 [receive-message](#) 命令从 DLQ 中检索死信事件，如下所示。您可以使用 `--max-number-of-messages` 属性设置要检索的消息数量。

```
$ aws sqs receive-message --queue-url your-dlq-url --attribute-names All --message-  
attribute-names All --max-number-of-messages 1
```

如果成功，您将看到与以下内容类似的输出。

```
{  
  "Messages": [  
    {
```

```

    "MessageId": "2aeg3510-fe3a-4f5a-ab6a-6906560eaf7e",
    "ReceiptHandle": "AQEBkNKTD0MrWgHKPoITRBwrPoK3eCSZICzWVqCY0BZ
+FFtC0RFpopJbtCqj36VbBT1HreM8+qM/m5jcwqS1A1GmIJ0/hYmMgn/
+dwIty9izE7HnpvRhhEyHxbE7Z5V05RbeasYaBdNyi9WLCnAHviDh6MebLXXNWoFyYnsxdwJuG0f/
w3htX6r3dpxVvFNPGoQb8ihY37+u0gtsbuIwhLtUSmE8rbldEEwiUfi3IJ1zEZpUS77n/k1GwrMrnYg0Gx/
BuaLz0rFi2F738XI/
Hnh45uv3ca60YwS1ojPQ1LtX2URg1haV5884FY1aRvY8jR1pCZabTkYRTZKSXG5KNGyZnHpmsspii6JNkjitYVFKPo0H91w
    "MD50fBody": "07adc3fc889d6107d8bb8fda42fe0573",
    "Body": "{\"MessageBody\": \"Hello, world!\", \"QueueUrl\": \"https://sqs.us-
west-2.amazonaws.com/123456789012/does-not-exist\"}",
    "Attributes": {
        "SenderId": "ARO2DZE3W4CTL5ZR7EIN:ff00212d8c453aaaae644bc6846d4723",
        "ApproximateFirstReceiveTimestamp": "1652499058144",
        "ApproximateReceiveCount": "2",
        "SentTimestamp": "1652490733042"
    },
    "MD50fMessageAttributes": "f72c1d78100860e00403d849831d4895",
    "MessageAttributes": {
        "ERROR_CODE": {
            "StringValue": "AWS.SimpleQueueService.NonExistentQueue",
            "DataType": "String"
        },
        "ERROR_MESSAGE": {
            "StringValue": "The specified queue does not exist for this wsdl
version.",
            "DataType": "String"
        },
        "EXECUTION_ID": {
            "StringValue": "ad06616e51cdf74a",
            "DataType": "String"
        },
        "EXHAUSTED_RETRY_CONDITION": {
            "StringValue": "MaximumEventAgeInSeconds",
            "DataType": "String"
        }
    },
    "IS_PAYLOAD_TRUNCATED": {
        "StringValue": "false",
        "DataType": "String"
    },
    "RETRY_ATTEMPTS": {
        "StringValue": "0",
        "DataType": "String"
    },
    "SCHEDULED_TIME": {

```

```

        "StringValue": "2022-05-14T01:12:00Z",
        "DataType": "String"
    },
    "SCHEDULE_ARN": {
        "StringValue": "arn:aws:scheduler:us-west-2:123456789012:schedule/
DLQ-test",
        "DataType": "String"
    },
    "TARGET_ARN": {
        "StringValue": "arn:aws:scheduler::aws-sdk:sqs:sendMessage",
        "DataType": "String"
    }
}
]
}

```

请注意死信事件中的以下属性，以帮助您确定目标调用失败的可能原因并对其进行故障排除。

- **ERROR_CODE**：包含 EventBridge 调度器从目标的服务 API 收到的错误代码。在前面的示例中，Amazon SQS 返回的错误代码是 `AWS.SimpleQueueService.NonExistentQueue`。如果由于 EventBridge 调度器出现问题而导致计划无法调用目标，则会看到以下错误代码：`AWS.Scheduler.InternalServerError`。
- **ERROR_MESSAGE**：包含 EventBridge 调度器从目标的服务 API 收到的错误消息。在前面的示例中，Amazon SQS 返回的错误信息是 `The specified queue does not exist for this wsdl version`。如果由于 EventBridge 调度器出现问题而导致计划失败，您将看到以下错误消息：`Unexpected error occurred while processing the request`。
- **TARGET_ARN**：您的计划调用的目标的 ARN，采用以下服务 ARN 格式：`arn:aws:scheduler::aws-sdk:service:apiAction`。
- **EXHAUSTED_RETRY_CONDITION**：表示将事件传送到 DLQ 的原因。如果来自目标 API 的错误是可重试的错误，而不是永久性错误，则会出现此属性。如果 EventBridge 调度器在超过您为计划配置的最大重试尝试次数 (`MaximumRetryAttempts`) 后将此属性发送到 DLQ，或者如果事件已超过您在计划中配置的最大持续时间 (`MaximumEventAgeInSeconds`) 但仍无法传送，则该属性可以包含这些值。

在前面的示例中，我们可以根据错误代码和错误消息确定为计划指定的目标队列不存在。

删除计划

您可以通过配置自动删除或手动单个删除来删除计划。使用以下主题来了解如何使用这两种方法删除计划，以及为什么在某些情况下应仅选择其中一种方法。

主题

- [计划完成后删除](#)
- [手动删除](#)

计划完成后删除

如果您想避免在 EventBridge 调度器上单独管理您的计划资源，请配置在计划完成后自动删除。在您一次创建数千个计划并且需要灵活按需纵向扩展计划数量的应用程序中，自动删除可以确保您不会达到指定区域[计划数量](#)的账户配额。

当您为计划配置自动删除时，EventBridge 调度器会在上次调用目标后删除该计划。对于一次性计划，这种情况发生在计划调用一次目标之后。对于使用 rate 或 cron 表达式设置的定期计划，您的计划将在上次调用后删除。定期计划的最后一次调用是最接近您指定的 [EndDate](#) 发生的调用。如果您将计划配置为自动删除，但未为其 EndDate 指定值，则 EventBridge 调度器不会自动删除该计划。

您可以在首次创建计划时设置自动删除，或者更新现有计划的首选项。以下步骤介绍如何为现有计划配置自动删除。

AWS Management Console

1. 打开 EventBridge 调度器控制台，网址为：<https://console.aws.amazon.com/scheduler/>。
2. 从计划列表中，选择要编辑的计划，然后选择编辑。
3. 从左侧导航列表中，选择设置。
4. 在计划完成后操作部分中，从下拉列表中选择删除，然后保存您的更改。

AWS CLI

1. 打开新的提示窗口。
2. 使用 [update-schedule](#) AWS CLI 命令更新现有计划，如下所示。该命令将设置 `--action-after-completion` 为 DELETE。此示例假定您已在 JSON 文件中本地定义目标配置。要更新计划，必须提供目标以及要为现有计划配置的任何其他计划参数。

这是一个定期计划，速率为每小时调用一次。因此，您可以在设置 `--action-after-completion` 参数时指定结束日期。

```
$ aws scheduler update-schedule --name schedule-name \
  \
  --action-after-completion 'DELETE' \
  --schedule-expression 'rate(1 hour)' \
  --end-date '2024-01-01T00:00:00' \
  --target file://target-configuration.json \
  --flexible-time-window '{ "Mode": "OFF" }' \
```

手动删除

当您不再需要某个计划，可使用 [DeleteSchedule](#) 操作删除。

Example AWS CLI

```
$ aws scheduler delete-schedule --name your-schedule
```

Example Python 开发工具包

```
import boto3
scheduler = boto3.client('scheduler')

scheduler.delete_schedule(Name="your-schedule")
```

接下来做什么？

- 有关如何为 Lambda 和 Step Functions 配置模板化目标的更多信息，以及要了解如何使用通用目标参数，请参阅 [管理目标](#)。
- 有关 EventBridge 调度器数据类型和 API 操作的更多信息，请参阅 [EventBridge 调度器 API 参考](#)。

管理计划组

计划组是 Amazon EventBridge 调度器资源，用于组织计划。

您的 AWS 账户 附带了一个 default 计划组。您可以将新计划与 default 组或您创建和管理的计划组相关联。您最多可以在 AWS 账户 中创建 [500 个计划组](#)。使用 EventBridge 调度器，您可以通过应用 [标签](#) 来组织计划组，而不是单个计划。

标签是由您定义的区分大小写的密钥和区分大小写的值组成的标签。您可以创建标签，按用途、所有者、环境等标准对资源进行分类。例如，您可以使用以下标签来标识计划所属的环境：`environment:production`。

Important

请勿在标签中添加个人身份信息 (PII) 或其他机密或敏感信息。标签可供许多 AWS 服务访问，包括计费。标签不适合用于私有或敏感数据。

计划组有两种可能的 [状态](#)：活动和正在删除。

首次创建组后，默认处于 ACTIVE 状态。您可以向 ACTIVE 组中添加计划。当您删除组时，状态将更改为 DELETING，直到 EventBridge 调度器删除关联的计划。EventBridge 调度器删除组中的计划后，您的账户中将不再有该组。

使用以下主题创建计划组并对其应用标签。您还可以将计划与组关联。最后，您将删除组。

主题

- [创建计划组](#)
- [删除计划组](#)
- [相关资源](#)

创建计划组

使用计划组和标签来组织具有共同目的或属于相同环境的计划。在以下步骤中，您将创建一个新的计划组并使用标签对其进行标记。然后，您将一个新的计划与该组相关联。

Note

创建组后，您无法从该组中移除计划，也无法将该计划与其他组关联。只有在首次创建计划时，才能将计划与组关联。

第一步：创建新计划组

以下主题介绍了如何创建新计划组，并使用以下标签对其进行标记：`environment:development`。

AWS Management Console

使用 AWS Management Console 创建新组

1. 登录 AWS Management Console 并打开 Amazon EventBridge 控制台 (<https://console.aws.amazon.com/events/>)。
2. 在左侧导航窗格中，选择计划组。
3. 在“计划组”页面，选择创建计划组。
4. 在计划组详细信息部分的名称中，输入该组的名称。例如，**TestGroup**。
5. 请在标签部分执行以下操作：
 - a. 选择添加新标签。
 - b. 在密钥中，输入您要分配给此密钥的名称。在本教程中，要标记此计划组所属的环境，请输入 **environment**。
 - c. 在值（可选）中，输入要分配给此密钥的值。在本教程中，输入您的环境密钥的值 **development**。

Note

创建组后，可以对其添加其他标签。

6. 选择创建计划组以完成操作。您的新组将显示在计划组列表中。
7. （可选）要编辑组或管理其标签，请选中新组的复选框并选择编辑。

Note

您无法编辑 default 计划组。

AWS CLI

使用 AWS CLI 创建新组

1. 打开新的命令提示符窗口。
2. 在 AWS Command Line Interface (AWS CLI) 中，输入以下 `create-schedule-group` 命令以创建新组。此命令使用一个标签创建组：`environment:development`。您可以使用此标签或类似的标记系统根据您的计划组所属的环境对其进行标记。

将计划名称以及标签键和值替换为您的信息。

```
$ aws scheduler create-schedule-group --name TestGroup --tags  
Key=environment,Value=development
```

默认情况下，您的新组处于 ACTIVE 状态。现在，您可以将新计划与您创建的新组相关联。

第二步：将计划表与组关联

使用以下步骤将新计划与您在[上一步](#)中创建的组相关联。

AWS Management Console

要将计划与组关联，请使用 AWS Management Console

1. 登录 AWS Management Console 并打开 Amazon EventBridge 控制台 (<https://console.aws.amazon.com/events/>)。
2. 在左侧导航窗格中，选择计划。
3. 从计划表中，选择创建计划以创建新计划。
4. 在指定计划详细信息页面上，在计划组中，从下拉列表中选择新组的名称。例如，选择 `TestGroup`。
5. 指定计划模式、目标和设置，然后在查看并保存计划页面上查看您的选择。有关配置新计划的信息，请参阅 [开始使用](#)。
6. 要完成并保存您的计划，请选择保存计划。

AWS CLI

要将计划与组关联，请使用 AWS CLI

1. 打开新的命令提示符窗口。
2. 在 AWS Command Line Interface (AWS CLI) 中，输入下面的 [create-schedule](#) 命令。这将创建一个计划并将其与[上一步](#)中名为 `sqs-test-schedule` 的组相关联。此计划使用模板化的 [Amazon SQS](#) 目标类型来调用 `SendMessage` 操作。用您的信息替换计划名称、目标和组名称。

```
$ aws scheduler create-schedule --name sqs-test-schedule --schedule-expression  
'rate(5 minutes)' \  
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "TEST_PAYLOAD" }'  
\  
--group-name TestGroup  
--flexible-time-window '{ "Mode": "OFF" }'
```

现在，您的新计划已与 `TestGroup` 计划组关联。

删除计划组

在下文中，您可以学习如何使用 AWS Management Console 和 AWS Command Line Interface 删除计划组。删除组时，该组一直处于 `DELETING` 状态，直到 EventBridge 调度器删除该组中的所有计划。EventBridge 调度器删除组中的计划后，您的账户中将不再有该组。

Note


创建组后，您无法从该组中移除计划，也无法将该计划与其他组关联。只有在首次创建计划时，才能将计划与组关联。

AWS Management Console

使用 AWS Management Console 删除组

1. 登录 AWS Management Console 并打开 Amazon EventBridge 控制台 (<https://console.aws.amazon.com/events/>)。
2. 在左侧导航窗格中，选择计划组。

- 在计划组页面上，从当前 AWS 区域的现有组列表中找到要删除的组。如果您没有看到要找的组，请选择其他 AWS 区域。

 Note

您不能删除或编辑默认组。

- 选中您要删除的组对应的复选框。
- 选择 Delete (删除)。
- 在删除计划组对话框中，输入该组的名称以确认您的选择，然后选择删除。
- 在计划组列表中，状态列会发生变化，表示您的组现在正在删除。在 EventBridge 调度器删除与该组关联的所有计划之前，该组将保持此状态。
- 要刷新列表并确认组已删除，请选择刷新图标。

AWS CLI

使用 AWS CLI 删除组

- 打开新的命令提示符窗口。
- 从 AWS Command Line Interface (AWS CLI) 中输入以下 [delete-schedule-group](#) 命令以删除计划组。用您的信息替换 `--name` 的值。

```
$ aws scheduler delete-schedule-group --name TestGroup
```

如果成功，此 AWS CLI 操作不返回响应。

- 要验证该组是否处于 DELETING 状态，请运行以下 [get-schedule-group](#) 命令。

```
$ aws scheduler get-schedule-group --name TestGroup
```

如果成功运行，则您将收到类似于以下内容的输出：

```
{
  "Arn": "arn:aws::scheduler:us-west-2:123456789012:schedule-group/TestGroup",
  "CreationDate": "2023-01-01T09:00:00.000000-07:00",
  "LastModificationDate": "2023-01-01T09:00:00.000000-07:00",
  "Name": "TestGroup",
  "State": "DELETING"
```

```
}
```

EventBridge 调度器在删除与组关联的计划后删除该组。如果您再次运行 `get-schedule-group`，则会收到以下 `ResourceNotFoundException` 响应：

```
An error occurred (ResourceNotFoundException) when calling the GetScheduleGroup operation: Schedule group TestGroup does not exist.
```

相关资源

有关计划组的更多信息，请参阅以下资源：

- EventBridge 调度器 API 参考中的 [CreateScheduleGroup](#) 操作。
- EventBridge 调度器 API 参考中的 [DeleteScheduleGroup](#) 操作。

管理目标

以下主题介绍如何将模板化目标和通用目标与 EventBridge 调度器配合使用，并提供了支持的 AWS 服务的列表，您可以使用 EventBridge 调度器的通用目标参数配置这些服务。

模板化目标是一组核心 AWS 服务（例如 Amazon SQS、Lambda 和 Step Functions）中的一组常见 API 操作。例如，您可以通过提供函数 ARN 来将 Lambda 的 [调用](#) API 操作设为目标，或者使用目标的队列 ARN 来将 Amazon SQS 的 [SendMessage](#) 操作设为目标。

通用目标是一组可自定义的参数，允许您为许多 AWS 服务调用更广泛的 API 操作。例如，您可以使用 EventBridge 调度器的通用目标参数 (UTP) 通过 [CreateQueue](#) 操作创建新的 Amazon SQS 队列。

要配置模板化目标或通用目标，您的计划必须有权调用您配置为目标 API 操作。您可以通过附加计划的执行角色所需权限来完成此操作。例如，要将 Amazon SQS 的 [SendMessage](#) 操作设为目标，应向执行角色授予执行 `sqs:SendMessage` 操作的权限。在大多数情况下，您可以使用目标服务支持的 [AWS 托管策略](#) 来添加必要的权限。但是，您也可以创建自己的 [客户管理型策略](#)，或者为附加到执行角色的现有策略添加 [内联权限](#)。以下主题演示了为模板化目标类型和通用目标类型添加权限的示例。

有关为计划设置执行角色的更多信息，请参阅 [the section called “设置执行角色”](#)。

主题

- [使用模板化目标](#)
- [使用通用目标](#)
- [添加上下文属性](#)
- [接下来做什么？](#)

使用模板化目标

模板化目标是一组核心 AWS 服务（例如 Amazon SQS、Lambda 和 Step Functions）中的一组常见 API 操作。例如，您可以通过提供函数 ARN 来将 Lambda 的 [Invoke](#) 操作设为目标，或者使用队列 ARN 来将 Amazon SQS 的 [SendMessage](#) 操作设为目标。要配置模板化目标，您还必须向计划的执行角色授予执行目标 API 操作的权限。

要使用 AWS CLI 或其中一个 EventBridge 调度器 SDK 以编程方式配置模板化目标，您需要指定执行角色的 ARN、目标资源的 ARN、您希望 EventBridge 调度器传送到目标的可选输入，对于某些模板化目标，还需要为该目标指定一组带有附加配置选项的唯一参数。当您为模板化目标资源指定 ARN

时，EventBridge 调度器会自动假设您要为该服务调用支持的 API 操作。如果您想让 EventBridge Scheduler 针对服务使用不同的 API 操作，则必须将该目标配置为[通用目标](#)。

以下是 S EventBridge scheduler 支持的所有模板化目标的完整列表，以及每个目标的唯一关联参数集（如果适用）。为每个参数集选择链接，以查看《EventBridge 调度器 API 参考》中的必填字段和可选字段。

- CodeBuild – [StartBuild](#)
- CodePipeline – [StartPipelineExecution](#)
- Amazon ECS – [RunTask](#)
 - 参数：[EcsParameters](#)
- EventBridge – [PutEvents](#)
 - 参数：[EventBridgeParameters](#)
- Amazon Inspector – [StartAssessmentRun](#)
- Kinesis – [PutRecord](#)
 - 参数：[KinesisParameters](#)
- Firehose — [PutRecord](#)
- Lambda – [Invoke](#)
- SageMaker – [StartPipelineExecution](#)
 - 参数：[SageMakerPipelineParameters](#)
- Amazon SNS – [Publish](#)
- Amazon SQS - [SendMessage](#)
 - 参数：[SqsParameters](#)
- Step Functions – [StartExecution](#)

使用以下示例来学习如何配置不同的模板化目标，以及如何为每个描述的目标配置所需的 IAM 权限。

Amazon SQS `SendMessage`

Example 执行角色的权限策略

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

```

        "Action": [
            "sqs:SendMessage"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
]
}

```

Example AWS CLI

```

$ aws scheduler create-schedule --name sqs-templated --schedule-expression 'rate(5
minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "QUEUE_ARN", "Input": "Message for scheduleArn:
'<aws.scheduler.schedule-arn>', scheduledTime: '<aws.scheduler.scheduled-time>' }' \
--flexible-time-window '{ "Mode": "OFF"}'

```

Example Python 开发工具包

```

import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<QUEUE_ARN>",
    "Input": "Message for scheduleArn: '<aws.scheduler.schedule-arn>', scheduledTime:
'<aws.scheduler.scheduled-time>'"}
}

scheduler.create_schedule(
    Name="sqs-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_templated,
    FlexibleTimeWindow=flex_window)

```

Example Java 软件开发工具包

```

package com.example;

import software.amazon.awssdk.regions.Region;

```



```
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;

public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
            .region(Region.US_WEST_2)
            .build();

        Target sqsTarget = Target.builder()
            .roleArn("<ROLE_ARN>")
            .arn("<QUEUE_ARN>")
            .input("Message for scheduleArn: '<aws.scheduler.schedule-arn>',
scheduledTime: '<aws.scheduler.scheduled-time>'")
            .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
            .name("<SCHEDULE_NAME>")
            .scheduleExpression("rate(10 minutes)")
            .target(sqsTarget)
            .flexibleTimeWindow(FlexibleTimeWindow.builder()
                .mode(FlexibleTimeWindowMode.OFF)
                .build())
            .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and an Amazon SQS
templated target");
    }
}
```

Lambda Invoke

Example 执行角色的权限策略

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

```

        "lambda:InvokeFunction"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
]
}

```

Example AWS CLI

```

$ aws scheduler create-schedule --name lambda-templated-schedule --schedule-expression
'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "FUNCTION_ARN", "Input": "{ \"Payload\":
\"TEST_PAYLOAD\" }" }' \
--flexible-time-window '{ "Mode": "OFF"}'

```

Example Python 开发工具包

```

import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

lambda_templated = {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<LAMBDA_ARN>",
    "Input": "{ 'Payload': 'TEST_PAYLOAD' }"}
}

scheduler.create_schedule(
    Name="lambda-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=lambda_templated,
    FlexibleTimeWindow=flex_window)

```

Example Java 软件开发工具包

```

package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;

```

```
public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
            .region(Region.US_WEST_2)
            .build();

        Target lambdaTarget = Target.builder()
            .roleArn("<ROLE_ARN>")
            .arn("<Lambda ARN>")
            .input("{ 'Payload': 'TEST_PAYLOAD' }")
            .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
            .name("<SCHEDULE_NAME>")
            .scheduleExpression("rate(10 minutes)")
            .target(lambdaTarget)
            .flexibleTimeWindow(FlexibleTimeWindow.builder()
                .mode(FlexibleTimeWindowMode.OFF)
                .build())
            .clientToken("<Token GUID>")
            .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Lambda templated
target");
    }
}
```

Step Functions **StartExecution**

Example 执行角色的权限策略

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "states:StartExecution"
      ],

```

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

Example AWS CLI

```
$ aws scheduler create-schedule --name sfn-templated-schedule --schedule-expression
'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn": "STATE_MACHINE_ARN", "Input": "{ \"Payload\":
\"TEST_PAYLOAD\" }" }' \
--flexible-time-window '{ "Mode": "OFF" }'
```

Example Python 开发工具包

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sfn_templated= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "<STATE_MACHINE_ARN>",
    "Input": "{ 'Payload': 'TEST_PAYLOAD' }"
}

scheduler.create_schedule(Name="sfn-python-templated",
    ScheduleExpression="rate(5 minutes)",
    Target=sfn_templated,
    FlexibleTimeWindow=flex_window)
```

Example Java 软件开发工具包

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;

public class MySchedulerApp {
```

```
public static void main(String[] args) {

    final SchedulerClient client = SchedulerClient.builder()
        .region(Region.US_WEST_2)
        .build();

    Target stepFunctionsTarget = Target.builder()
        .roleArn("<ROLE_ARN>")
        .arn("<STATE_MACHINE_ARN>")
        .input("{ 'Payload': 'TEST_PAYLOAD' }")
        .build();

    CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
        .name("<SCHEDULE_NAME>")
        .scheduleExpression("rate(10 minutes)")
        .target(stepFunctionsTarget)
        .flexibleTimeWindow(FlexibleTimeWindow.builder()
            .mode(FlexibleTimeWindowMode.OFF)
            .build())
        .clientToken("<Token GUID>")
        .build();

    client.createSchedule(createScheduleRequest);
    System.out.println("Created schedule with rate expression and Step Function
templated target");
}
}
```

使用通用目标

通用目标是一组可自定义的参数，允许您为许多 AWS 服务调用更广泛的 API 操作。例如，您可以使用通用目标参数 (UTP) 通过 [CreateQueue](#) 操作创建新的 Amazon SQS 队列。

要使用 AWS CLI 或其中一个 EventBridge 调度器开发工具包为您的计划配置通用目标，您需要指定以下信息：

- **RoleArn**：要用于目标的执行角色的 ARN。您指定的执行角色必须具有调用作为您计划的目标的 API 操作的权限。
- **Arn**：完整的服务 ARN，包括您要设为目标的 API 操作，格式如下：
下：`arn:aws:scheduler:::aws-sdk:service:apiAction`。

例如，对于 Amazon SQS，您指定的服务名称为 `arn:aws:scheduler:::aws-sdk:sqs:sendMessage`。

- 输入：您指定的格式正确的 JSON，其中包含 EventBridge 调度器发送到目标 API 的请求参数。您在 Input 设置的 JSON 的参数和形状由您的计划调用的服务 API 决定。有关此信息，请参阅您想要设为目标的服务的 API 参考文档。

不支持的操作

EventBridge 调度器不支持以下列前缀开头的只读 API 操作，例如常见 GET 操作：

```
get
describe
list
poll
receive
search
scan
query
select
read
lookup
discover
validate
batchGet
batchDescribe
batchRead
transactGet
adminGet
adminList
testMigration
retrieve
testConnection
translateDocument
isAuthorized
isAuthorizedWithToken
invokeModel
```

例如，[GetQueueUrl](#) API 操作的服务 ARN 将如下所示：`arn:aws:scheduler:::aws-sdk:sqs:getQueueURL`。由于 API 操作以 `get` 前缀开头，因此 EventBridge 调度器不支持此目标。同样，不支持 Amazon MQ 操作 [ListBrokers](#) 作为目标，因为该操作使用前缀 `list`。

使用通用目标的示例

您在计划 Input 字段中传递的参数取决于您要调用的服务 API 接受的请求参数。例如，要以 Lambda [Invoke](#) 为目标，您可以设置 [AWS Lambda API 参考](#) 中列出的参数。这包括您可以传递给 Lambda 函数的可选 JSON [有效负载](#)。

要确定可以为不同 API 设置的参数，请参阅该服务的 API 参考。与 Lambda Invoke 类似，某些 API 接受 URI 参数以及请求正文有效负载。在这种情况下，您可以在计划 Input 中指定 URI 路径参数以及 JSON 有效负载。

以下示例展示了如何使用通用目标通过 Lambda、Amazon SQS 和 Step Functions 调用常见 API 操作。

Example Lambda

```
$ aws scheduler create-schedule --name lambda-universal-schedule --schedule-expression
'rate(5 minutes)' \
--target '{"RoleArn": "ROLE_ARN", "Arn":"arn:aws:scheduler::aws-sdk:lambda:invoke"
"Input": "{\"FunctionName\":\"arn:aws:lambda:REGION:123456789012:function:HelloWorld
\", \"InvocationType\":\"Event\", \"Payload\":\"{\\\"message\\\":\\\"testing function\\
\\\"}\" }' \
--flexible-time-window '{ "Mode": "OFF" }'
```

Example Amazon SQS

```
import boto3
scheduler = boto3.client('scheduler')

flex_window = { "Mode": "OFF" }

sqs_universal= {
    "RoleArn": "<ROLE_ARN>",
    "Arn": "arn:aws:scheduler::aws-sdk:sqs:sendMessage",
    "Input": "{\"MessageBody\":\"My message\", \"QueueUrl\":\"<QUEUE_URL>\"}"
}

scheduler.create_schedule(
    Name="sqs-sdk-test",
    ScheduleExpression="rate(5 minutes)",
    Target=sqs_universal,
    FlexibleTimeWindow=flex_window)
```

Example Step Functions

```
package com.example;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.scheduler.SchedulerClient;
import software.amazon.awssdk.services.scheduler.model.*;

public class MySchedulerApp {

    public static void main(String[] args) {

        final SchedulerClient client = SchedulerClient.builder()
            .region(Region.US_WEST_2)
            .build();

        Target stepFunctionsUniversalTarget = Target.builder()
            .roleArn("<ROLE_ARN>")
            .arn("arn:aws:scheduler::aws-sdk:sfn:startExecution")
            .input("{\"Input\":\"{}\"\",\"StateMachineArn\":\"<STATE_MACHINE_ARN>\"}")
            .build();

        CreateScheduleRequest createScheduleRequest = CreateScheduleRequest.builder()
            .name("<SCHEDULE_NAME>")
            .scheduleExpression("rate(10 minutes)")
            .target(stepFunctionsUniversalTarget)
            .flexibleTimeWindow(FlexibleTimeWindow.builder()
                .mode(FlexibleTimeWindowMode.OFF)
                .build())
            .clientToken("<Token GUID>")
            .build();

        client.createSchedule(createScheduleRequest);
        System.out.println("Created schedule with rate expression and Step Function
universal target");
    }
}
```


添加上下文属性

在传递给目标的有效负载中使用以下关键字来收集有关计划的元数据。当您的计划调用目标时，EventBridge 调度器会将每个关键字替换为其各自的值。

- **<aws.scheduler.schedule-arn>**：计划的 ARN。
- **<aws.scheduler.scheduled-time>**：您为计划指定的调用其目标的时间，例如 2022-03-22T18:59:43Z。
- **<aws.scheduler.execution-id>**：EventBridge 调度器为每次尝试调用目标分配的唯一 ID，例如 d32c5kddcf5bb8c3。
- **<aws.scheduler.attempt-number>**：用于标识当前调用的尝试次数的计数器，例如 1。

此示例演示如何创建一个计划，该计划每五分钟触发一次，并调用 Amazon SQS SendMessage 操作作为通用目标。消息正文包含 schedule-time 的值。

Example AWS CLI

```
$ aws scheduler create-schedule --name your-schedule \  
  --schedule-expression 'rate(5 minutes)' \  
  --target '{"RoleArn": "ROLE_ARN", \  
    "Arn": "arn:aws:scheduler::aws-sdk:sqs:sendMessage", \  
    "Input": "{\\"MessageBody\\":\\"<aws.scheduler.scheduled-time>\\"",\\"QueueUrl\\":  
\\"https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-cli-test\\"}"}' \  
  --flexible-time-window '{"Mode": "OFF"}'
```

Example Python 开发工具包

```
import boto3  
scheduler = boto3.client('scheduler')  
  
sqs_universal= {  
    "RoleArn": "<ROLE_ARN>",  
    "Arn": "arn:aws:scheduler::aws-sdk:sqs:sendMessage",  
    "Input": "{\\"MessageBody\\":\\"<aws.scheduler.scheduled-time>\\"",\\"QueueUrl\\":  
\\"https://sqs.us-west-2.amazonaws.com/123456789012/scheduler-cli-test\\"}"  
}  
  
flex_window = { "Mode": "OFF" }
```

```
scheduler.update_schedule(Name="your-schedule",  
    ScheduleExpression="rate(5 minutes)",  
    Target=sqs_universal,  
    FlexibleTimeWindow=flex_window)
```

接下来做什么？

有关 EventBridge 调度器数据类型和 API 操作的更多信息，请参阅 [EventBridge 调度器 API 参考](#)。

Amazon EventBridge 日程安排程序中的安全

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。AWS 还为您提供可以安全使用的服务。作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon S EventBridge scheduler 的合规计划，请参阅[合规计划范围内的AWSAWS 服务按合规计划](#)分的范围内服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 S EventBridge scheduler 时如何应用责任共担模型。以下主题向您介绍如何配置 EventBridge 计划程序以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 EventBridge 日程安排器资源。

主题

- [管理对 Amazon EventBridge 日程安排程序的访问权限](#)
- [Amazon EventBridge 计划程序中的数据保护](#)
- [Amazon EventBridge 计划程序的合规性验证](#)
- [Amazon EventBridge 调度器中的弹性](#)
- [Amazon EventBridge 计划程序中的基础设施安全](#)

管理对 Amazon EventBridge 日程安排程序的访问权限

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 EventBridge 计划程序资源。您可以使用 IAM AWS 服务，无需支付额外费用。

主题

- [受众](#)
- [使用身份进行身份验证](#)

- [使用策略管理访问](#)
- [EventBridge 调度器如何与 IAM 配合使用](#)
- [使用基于身份的策略](#)
- [混淆代理问题防范](#)
- [对 Amazon EventBridge 日程安排程序的身份和访问进行故障排除](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 EventBridge 计划程序中所做的工作。

服务用户-如果您使用 EventBridge 计划程序服务完成工作，则您的管理员会为您提供所需的凭证和权限。当你使用更多的 EventBridge 日程安排器功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法在“EventBridge 日程安排”中访问某项功能，请参阅[对 Amazon EventBridge 日程安排程序的身份和访问进行故障排除](#)。

服务管理员-如果您负责公司的 EventBridge 日程安排器资源，则可能拥有对 EventBridge 日程安排器的完全访问权限。您的工作是确定您的服务用户应访问哪些 EventBridge 计划程序功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解贵公司如何将 IAM 与 EventBridge 计划程序一起使用，请参阅[EventBridge 调度器如何与 IAM 配合使用](#)。

IAM 管理员 — 如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理对 EventBridge 计划程序的访问权限。要查看您可以在 IAM 中使用的基于身份的 EventBridge 计划程序策略示例，请参阅[使用基于身份的策略](#)

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担任 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。AWS IAM Identity Center（IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》[中的如何登录到您 AWS 账户的](#)。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的[签署 AWS API 请求](#)。

无论使用何种身份验证方法，您可能需要提供其它安全信息。例如，AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[多重身份验证](#)和《IAM 用户指南》中的[在 AWS 中使用多重身份验证 \(MFA\)](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity Center 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。AWS Directory Service 当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center ?](#)

IAM 用户和组

[IAM 用户](#)是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果有一些特定的使用场景需要长期凭证以及 IAM 用户，我们建议轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[对于需要长期凭证的使用场景定期轮换访问密钥](#)。

[IAM 组](#)是一个用于指定一组 IAM 用户的身份。您不能使用组的身份登录。可以使用群组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，可能具有一个名为 IAMAdmins 的群组，并为该群组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人或应用程序关联，而角色旨在让需要它的任何人担任。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的[何时创建 IAM 用户（而不是角色）](#)。

IAM 角色

[IAM 角色](#)是您内部具有特定权限 AWS 账户的身份。它类似于 IAM 用户，但与特定人员不关联。您可以通过 AWS Management Console 通过[切换角色在中临时担任 IAM 角色](#)。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色](#)。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的[为第三方身份提供商创建角色](#)。如果使用 IAM Identity Center，则需要配置权限集。为控制身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的[权限集](#)。
- 临时 IAM 用户权限 – IAM 用户或角色可代入 IAM 角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取：您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户存取权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户存取的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的[IAM 角色与基于资源的策略有何不同](#)。
- 跨服务访问 — 有些 AWS 服务使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service（Amazon S3）中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
- 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。
- 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。

- 服务相关角色-服务相关角色是一种链接到的服务角色。AWS 服务服务可以担任代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的[何时创建 IAM 角色（而不是用户）](#)。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS，当与身份或资源关联时，它会定义其权限。AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概览](#)。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。然后，管理员可以向角色添加 IAM 策略，并且用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略](#)。

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS

托管策略和客户托管策略。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管策略与内联策略之间进行选择](#)。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACL)

访问控制列表 (ACL) 控制哪些主体 (账户成员、用户或角色) 有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。AWS WAF 要了解有关 ACL 的更多信息，请参阅 Amazon Simple Storage Service 开发人员指南 中的[访问控制列表 \(ACL \) 概览](#)。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型所授予的最大权限。

- **权限边界**：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体 (IAM 用户或角色) 授予的最大权限。可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 Principal 字段中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- **服务控制策略 (SCP)**-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果在组织内启用了所有特征，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体 (包括每个 AWS 账户根用户实体) 的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的[SCP 的工作原理](#)。
- **会话策略**：会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策

略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的[策略评估逻辑](#)。

EventBridge 调度器如何与 IAM 配合使用

在使用 IAM 管理对 EventBridge 调度程序的访问权限之前，请先了解计划 EventBridge 程序可以使用哪些 IAM 功能。

您可以在 Amazon EventBridge 计划程序中使用的 IAM 功能

IAM 特征	EventBridge 调度器支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是
ACL	否
ABAC（策略中的标签）	部分
临时凭证	是
主体权限	是
服务角色	是
服务相关角色	否

要全面了解 EventBridge 调度器和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的[AWS 服务](#)。

日程安排器的基于身份的策略 EventBridge

支持基于身份的策略 是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[创建 IAM policy](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

日程安排器的基于身份的策略示例 EventBridge

要查看基于 EventBridge 调度程序身份的策略的示例，请参阅[使用基于身份的策略](#)

调度程序中 EventBridge 基于资源的策略

支持基于资源的策略 否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户存取，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅 IAM 用户指南中的[IAM 角色与基于资源的策略有何不同](#)。

EventBridge 日程安排程序的策略操作

支持策略操作 是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 EventBridge 计划程序操作列表，请参阅《[服务授权参考](#)》中的 [Amazon EventBridge 计划程序定义的操作](#)。

EventBridge 计划程序中的策略操作在操作前使用以下前缀：

```
scheduler
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

您也可以使用通配符 (*) 指定多个操作。例如，要指定以单词 `List` 开头的所有操作，包括以下操作：

```
"Action": [  
  "scheduler:List*"  
]
```

EventBridge 日程安排器的策略资源

支持策略资源 是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

ResourceJSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

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

要查看 EventBridge 计划程序资源类型及其 ARN 的列表，请参阅《服务授权参考》中的 [Amazon S EventBridge cheduler 定义的资源](#)。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅 [Amazon S EventBridge zon Scheduler 定义的操作](#)。

要查看基于 EventBridge 调度程序身份的策略的示例，请参阅 [使用基于身份的策略](#)

EventBridge 日程安排器的策略条件密钥

支持特定于服务的策略条件键 **是**

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。可以创建使用 [条件运算符](#)（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评测它们。如果您为单个条件键指定多个值，则使用逻辑 OR 运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 策略元素：变量和标签](#)。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的 [AWS 全局条件上下文密钥](#)。

要查看 EventBridge 计划程序条件密钥列表，请参阅《服务授权参考》中的 [Amazon EventBridge 计划程序条件密钥](#)。要了解您可以使用条件键的操作和资源，请参阅 [Amazon S EventBridge cheduler 定义的操作](#)。

要查看基于 EventBridge 调度程序身份的策略的示例，请参阅 [使用基于身份的策略](#)

调度程序中的 EventBridge ACL

支持 ACL	否
--------	---

访问控制列表 (ACL) 控制哪些主体 (账户成员、用户或角色) 有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

带调度器的 ABA EventBridge C

支持 ABAC (策略中的标签)	部分
--------------------	----

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以向 IAM 实体 (用户或角色) 和许多 AWS 资源附加标签。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的 [条件元素](#) 中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的 [什么是 ABAC ?](#)。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的 [使用基于属性的访问权限控制 \(ABAC \)](#)。

在 EventBridge 日程安排器中使用临时证书

支持临时凭证	是
--------	---

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的 [AWS 服务与 IAM 配合使用的信息](#)。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以

用户身份登录控制台，然后切换角色时，还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的[切换到角色 \(控制台\)](#)。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 [IAM 中的临时安全凭证](#)。

日程安排器的跨服务主体 EventBridge 权限

支持转发访问会话 (FAS)	是
----------------	---

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。

EventBridge 日程安排器的服务角色

支持服务角色	是
--------	---

服务角色是由一项服务代入、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的 [创建向 AWS 服务 委派权限的角色](#)。

Warning

更改服务角色的权限可能会中断 S EventBridge cheduler 的功能。只有在 S EventBridge cheduler 提供相关指导时才编辑服务角色。

日程安排器的服务相关角色 EventBridge

支持服务相关角色	否
----------	---

服务相关角色是一种与服务相关联的 AWS 服务角色。服务可以担任代表您执行操作的角色。服务相关角色出现在您的 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅[能够与 IAM 搭配使用的 AWS 服务](#)。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

使用基于身份的策略

默认情况下，用户和角色无权创建或修改 EventBridge 日程安排器资源。他们也无法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM policy。然后，管理员可以向角色添加 IAM policy，并且用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略](#)。

有关 EventBridge 计划程序定义的操作和资源类型（包括每种资源类型的 ARN 格式）的详细信息，请参阅《服务授权参考》中的[Amazon S EventBridge scheduler 的操作、资源和条件密钥](#)。

主题

- [策略最佳实践](#)
- [EventBridge 日程安排器权限](#)
- [AWS EventBridge 日程安排程序的托管策略](#)
- [客户托管的 EventBridge 日程安排器策略](#)
- [AWS 托管策略更新](#)

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 EventBridge 日程安排资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的[AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。

- 应用最低权限：在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM policy 中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM Access Analyzer 策略验证](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到策略中。有关更多信息，请参阅《IAM 用户指南》中的 [配置受 MFA 保护的 API 访问](#)。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

EventBridge 日程安排器权限

为了让 IAM 委托人 (用户、群组或角色) 在 EventBridge 调度器中创建计划并通过控制台或 API 访问 EventBridge 计划程序资源，委托人必须在其权限策略中添加一组权限。您可以根据主体的工作职能配置这些权限。例如，仅使用 EventBridge 调度程序控制台查看现有计划列表的用户或角色无需拥有调用 CreateSchedule API 操作所需的权限。我们建议您调整基于身份的权限，以便仅提供最低访问权限。

以下列表显示了 EventBridge 调度程序的资源及其相应的支持的操作。

- 计划
 - scheduler:ListSchedules
 - scheduler:GetSchedule
 - scheduler>CreateSchedule
 - scheduler:UpdateSchedule
 - scheduler>DeleteSchedule
- 计划组
 - scheduler:ListScheduleGroups

- scheduler:GetScheduleGroup
- scheduler:CreateScheduleGroup
- scheduler>DeleteScheduleGroup
- scheduler:ListTagsForResource
- scheduler:TagResource
- scheduler:UntagResource

您可以使用 EventBridge 计划程序权限来创建自己的客户托管策略，以便与 EventBridge 日程安排器一起使用。您还可以使用下一节中描述的 AWS 托管策略为常见用例授予必要的权限，而无需管理自己的策略。

AWS EventBridge 日程安排程序的托管策略

AWS 通过提供用于 AWS 创建和管理的独立 IAM 策略来解决许多常见用例。托管策略也称为预定义策略，可针对常见使用场景授予必要的权限，让您不必调查需要哪些权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管式策略](#)。您可以将以下 AWS 托管策略附加到账户中的用户，这些托管策略特定于 S EventBridge scheduler：

- [the section called “AmazonEventBridgeSchedulerFullAccess”](#)— 使用控制台和 API 授予对 EventBridge 调度程序的完全访问权限。
- [the section called “AmazonEventBridgeSchedulerReadOnlyAccess”](#)— 授予对 EventBridge 日程安排器的只读访问权限。

AmazonEventBridgeSchedulerFullAccess

AmazonEventBridgeSchedulerFullAccess 托管策略授予对计划和 EventBridge 计划组使用所有日程安排程序操作的权限。

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

```

        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::*:role/*",
        "Condition": {
            "StringLike": {
                "iam:PassedToService": "scheduler.amazonaws.com"
            }
        }
    ]
}

```

AmazonEventBridgeSchedulerReadOnlyAccess

AmazonEventBridgeSchedulerReadOnlyAccess 托管策略授予只读权限，以查看有关您的计划和计划组的详细信息。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "scheduler:ListSchedules",
        "scheduler:ListScheduleGroups",
        "scheduler:GetSchedule",
        "scheduler:GetScheduleGroup",
        "scheduler:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}

```

客户托管的 EventBridge 日程安排器策略

使用以下示例为 EventBridge 计划程序创建您自己的客户托管策略。[客户管理型策略](#)允许您根据主体的工作职能，仅授予团队中应用程序和用户所需的操作和资源的权限。

主题

- [例如：CreateSchedule](#)
- [例如：GetSchedule](#)

- [例如 : UpdateSchedule](#)
- [例如 : DeleteScheduleGroup](#)

例如 : CreateSchedule

创建新计划时，您可以选择是使用客户管理的密钥还是使用[客户管理的密钥](#)对 EventBridge [AWS 拥有的密钥](#)计划程序上的数据进行加密。

以下策略允许主体使用 AWS 拥有的密钥创建计划并应用加密。使用时 AWS 拥有的密钥，AWS 可以为您管理 AWS Key Management Service (AWS KMS) 上的资源，因此您无需其他权限即可与之交互 AWS KMS。

```
{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "scheduler:CreateSchedule"
      ],
      "Effect": "Allow",
      "Resource":
      [
        "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "scheduler.amazonaws.com"
        }
      }
    }
  ]
}
```

使用以下策略允许委托人创建计划并使用 AWS KMS 客户托管密钥进行加密。要使用客户托管密钥，委托人必须有权访问您账户中的 AWS KMS 资源。此策略授予对单个指定 KMS 密钥的访问权限，该密钥用于加密 EventBridge 计划程序上的数据。或者，您可以使用通配符 (*) 来授予对账户中所有密钥或与给定名称模式匹配的子集的访问权限。

```
{
  "Version": "2012-10-17"
  "Statement":
  [
    {
      "Action":
      [
        "scheduler:CreateSchedule"
      ],
      "Effect": "Allow",
      "Resource":
      [
        "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
      ]
    },
    {
      "Action":
      [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Effect": "Allow",
      "Resource":
      [
        "arn:aws:kms:us-west-2:123456789012:key/my-key-id"
      ],
      "Conditions": {
        "StringLike": {
          "kms:ViaService": "scheduler.amazonaws.com",
          "kms:EncryptionContext:aws:scheduler:schedule:arn":
            "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
        }
      }
    }
  ]
}
```

```

    "Resource": "arn:aws:iam::123456789012:role/*",
    "Condition": {
      "StringLike": {
        "iam:PassedToService": "scheduler.amazonaws.com"
      }
    }
  }
]
}

```

例如：GetSchedule

使用以下策略允许主体获取有关计划的信息。

```

{
  "Version": "2012-10-17",
  "Statement":
  [
    {
      "Action":
      [
        "scheduler:GetSchedule"
      ],
      "Effect": "Allow",
      "Resource":
      [
        "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-
schedule-name"
      ]
    }
  ]
}

```

例如：UpdateSchedule

使用以下策略允许主体通过调用 scheduler:UpdateSchedule 操作来更新计划。与此类似CreateSchedule，该策略取决于计划是使用客户托管密钥还是客户托管密钥进行加密。AWS KMS AWS 拥有的密钥 对于配置了的计划 AWS 拥有的密钥，请使用以下策略：

```

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

```

```
[
  {
    "Action":
      [
        "scheduler:UpdateSchedule"
      ],
    "Effect": "Allow",
    "Resource":
      [
        "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
      ]
  },
  {
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/*",
    "Condition": {
      "StringLike": {
        "iam:PassedToService": "scheduler.amazonaws.com"
      }
    }
  }
]
```

对于配置了客户托管的密钥的计划，请使用以下策略。此策略包括允许委托人访问您账户中的 AWS KMS 资源的额外权限：

```
{
  "Version": "2012-10-17",
  "Statement":
    [
      {
        "Action":
          [
            "scheduler:UpdateSchedule"
          ],
        "Effect": "Allow",
        "Resource":
          [
            "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
          ]
      }
    ]
}
```

```

    },
    {
      "Action":
      [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Effect": "Allow",
      "Resource":
      [
        "arn:aws:kms:us-west-2:123456789012:key/my-key-id"
      ],
      "Conditions": {
        "StringLike": {
          "kms:ViaService": "scheduler.amazonaws.com",
          "kms:EncryptionContext:aws:scheduler:schedule:arn":
            "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
        }
      }
    }
  ]
}

```

例如：DeleteScheduleGroup

使用以下策略允许主体删除计划组。删除组时，还会删除与该组相关联的计划。删除该组的主体还必须拥有删除与该组关联的计划的权限。此策略授予主体对指定的计划组以及该组中的所有计划调用 `scheduler:DeleteScheduleGroup` 操作的权限：

Note

EventBridge 计划程序不支持为单个计划指定资源级别权限。例如，以下语句无效，不应包含在您的策略中：

```
"Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/my-schedule-name"
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "scheduler:DeleteSchedule",
      "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group/*"
    },
    {
      "Effect": "Allow",
      "Action": "scheduler:DeleteScheduleGroup",
      "Resource": "arn:aws:scheduler:us-west-2:123456789012:schedule/my-group"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "scheduler.amazonaws.com"
        }
      }
    }
  ]
}
```

AWS 托管策略更新

更改	描述	日期
the section called “AmazonEventBridgeSchedulerFullAccess” : 新托管策略	EventBridge Scheduler 增加了对新托管策略的支持，该策略允许用户完全访问所有资源，包括计划和计划组。	2022 年 11 月 10 日
the section called “AmazonEventBridgeS	EventBridge Scheduler 增加了对新托管策略的支持，该策略	2022 年 11 月 10 日

更改	描述	日期
chedulerReadOnlyAccess : 新托管策略	向用户授予对所有资源（包括计划和计划组）的只读访问权限。	
EventBridge 日程安排器已开始跟踪更改	EventBridge 调度程序开始跟踪其 AWS 托管策略的更改。	2022 年 11 月 10 日

混淆代理问题防范

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的“服务”）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况，AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在计划执行角色中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全局条件上下文密钥来限制 EventBridge 调度器授予其他服务访问资源的权限。如果您只希望将一个资源与跨服务访问相关联，请使用 `aws:SourceArn`。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 `aws:SourceAccount`。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。以下条件仅限于单个计划组：`arn:aws:scheduler:*:123456789012:schedule-group/your-schedule-group`

如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符字符 (*) 的 `aws:SourceArn` 全局上下文条件键。例如：`arn:aws:scheduler:*:123456789012:schedule-group/*`。

的值 `aws:SourceArn` 必须是您要将此条件限定到的 EventBridge 计划程序计划组 ARN。

Important

不要将该 `aws:SourceArn` 语句的范围限定为特定的计划或计划名称前缀。您指定的 ARN 必须是计划组。

以下示例演示如何使用执行角色信任策略中的 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键来防范混淆代理问题：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "scheduler.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012",
          "aws:SourceArn": "arn:aws:scheduler:us-west-2:123456789012:schedule-group/your-schedule-group"
        }
      }
    }
  ]
}
```

对 Amazon EventBridge 日程安排程序的身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 EventBridge 计划程序和 IAM 时可能遇到的常见问题。

主题

- [我无权在“EventBridge 日程安排”中执行操作](#)
- [我无权执行 iam : PassRole](#)
- [我想允许我以外的人 AWS 账户 访问我的 EventBridge 日程安排资源](#)

我无权在“EventBridge 日程安排”中执行操作

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 `my-example-widget` 资源的详细信息，但不拥有虚构 scheduler:`GetWidget` 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
scheduler:GetWidget on resource: my-example-widget
```

在此情况下，Mateo 的策略必须更新以允许其使用 scheduler: *GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam : PassRole

如果您收到错误消息，提示您无权执行 iam:PassRole 操作，则必须更新您的策略以允许您将角色传递给 EventBridge 调度器。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为的 IAM 用户 marymajor 尝试使用控制台在 S EventBridge scheduler 中执行操作时，会出现以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 EventBridge 日程安排资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 EventBridge 日程安排器是否支持这些功能，请参阅 [EventBridge 调度器如何与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅 [IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。

- 要了解如何向第三方提供对您的资源的访问[权限 AWS 账户](#)，请参阅 [IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过联合身份验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户 \(联合身份验证\) 提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。

Amazon EventBridge 计划程序中的数据保护

责任 AWS [共担模式分担责任模型](#)适用于 Amazon S EventBridge scheduler 中的数据保护。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础架构上的内容的控制。您还负责您所使用的 AWS 服务的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题解答](#)。有关欧洲数据保护的信息，请参阅 AWS 安全性博客上的 [AWS 责任共担模式和 GDPR](#) 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与资源通信。AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务 (例如 Amazon Macie)，它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅 [《美国联邦信息处理标准 \(FIPS\) 第 140-2 版》](#)。

我们强烈建议您切勿将机密信息或敏感信息 (如您客户的电子邮件地址) 放入标签或自由格式文本字段 (如名称字段)。这包括您使用控制台、API 或 AWS SDK AWS 服务使用 EventBridge 调度程序或其他工具包的情况。AWS CLI 在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

主题

- [静态加密](#)
- [传输中加密](#)

静态加密

本节介绍了 Amazon S EventBridge scheduler 如何加密和解密您的静态数据。静态数据是存储在 EventBridge 调度程序和服务底层组件中的数据。EventBridge 调度器与 AWS Key Management Service (AWS KMS) 集成，可使用加密和解密您的数据。[AWS KMS key](#) EventBridge 计划程序支持两种类型的 KMS 密钥：[AWS 拥有的密钥](#)和[客户托管密钥](#)。

Note

EventBridge 调度程序仅支持使用[对称](#)加密 KMS 密钥。

AWS 拥有的密钥是 AWS 服务拥有和管理的 KMS 密钥，可在多个 AWS 账户中使用。尽管 AWS 拥有的密钥 EventBridge 日程安排器使用的未存储在您的 AWS 账户中，但 EventBridge 日程安排器使用它们来保护您的数据和资源。默认情况下，S EventBridge scheduler 使用自有密钥加密和解密您的所有数据。AWS 您无需管理自己的 AWS 拥有的密钥 或其访问策略。当 S EventBridge scheduler 使用 AWS 拥有的密钥 来保护您的数据时，您不会产生任何费用，并且这些费用的使用量不计入您账户中 AWS KMS 配额的一部分。

客户托管密钥是存储在您的 AWS 账户中的 KMS 密钥，由您创建、拥有和管理这些密钥。如果您的特定用例要求您控制和审核在 EventBridge 计划程序上保护您的数据的加密密钥，则可以使用客户托管密钥。如果您选择客户托管密钥，您必须管理密钥策略。客户托管密钥会产生月费以及超过免费套餐使用量的费用。使用客户托管密钥也算作 [AWS KMS 配额](#)的一部分。有关定价的更多信息，请参阅 [AWS Key Management Service 定价](#)。

主题

- [加密构件](#)
- [管理 KMS 密钥](#)
- [CloudTrail 事件示例](#)

加密构件

下表描述了 S EventBridge scheduler 对静态数据进行加密的不同类型，以及它为每个类别支持哪种类型的 KMS 密钥。

数据类型	描述	AWS 拥有的密钥	客户托管密钥
有效负载 (最大 256KB)	您在配置要交付到目标的计划时在计划的 TargetInput 参数中指定的数据。	支持	支持
标识符和状态	计划的唯一名称和状态 (启用、禁用) 。	支持	不支持
计划配置	计划表达式，例如重复计划的 rate 或 cron 表达式，一次性调用的时间戳，以及计划的开始日期、结束日期和时区。	支持	不支持
目标配置	目标的 Amazon 资源名称 (ARN) 以及其他与目标相关的配置详细信息。	支持	不支持
调用和失败行为配置	灵活的时间窗配置、计划的重试策略以及用于失败交付的死信队列详细信息。	支持	不支持

EventBridge 如上表所述，Scheduler 仅在加密和解密目标负载时才使用您的客户托管密钥。如果您选择使用客户托管密钥，S EventBridge cheduler 会对有效负载进行两次加密和解密：一次使用默认密钥 AWS 拥有的密钥，另一次使用您指定的客户托管密钥。对于所有其他数据类型，S EventBridge cheduler 仅使用默认值 AWS 拥有的密钥 来保护您的静态数据。

使用以下[the section called “管理 KMS 密钥”](#)部分了解您必须如何管理您的 IAM 资源和密钥策略，以便在 EventBridge 计划程序中使用客户托管密钥。

管理 KMS 密钥

您可以选择提供客户托管密钥来加密和解密您的计划交付给目标的有效负载。EventBridge 调度程序可加密和解密您的有效负载（最多 256KB 的数据）。使用客户托管密钥会产生月费以及超过免费套餐使用量的费用。使用客户托管密钥算作 [AWS KMS 配额](#) 的一部分。有关定价的更多信息，请参阅 [AWS Key Management Service 定价](#)

EventBridge 计划程序使用与创建计划的委托人关联的 IAM 权限来加密您的数据。这意味着您必须将所需的 AWS KMS 相关权限附加到 EventBridge 调用计划程序 API 的用户或角色。此外，EventBridge Scheduler 使用基于资源的策略来解密您的数据。这意味着与您的计划关联的执行角色还必须具有在解密数据时调用 AWS KMS API 所需的 AWS KMS 相关权限。

Note

EventBridge 计划程序不支持使用临时权限的 [授权](#)。

使用以下部分了解如何管理 AWS KMS [密钥策略](#) 以及在 S EventBridge scheduler 上使用客户托管密钥所需的 IAM 权限。

主题

- [添加 IAM 权限](#)
- [管理密钥策略](#)

添加 IAM 权限

要使用客户托管密钥，您必须向创建计划的基于身份的 IAM 主体以及与计划关联的执行角色添加以下权限。

客户托管密钥的基于身份的权限

在创建计划时，您必须将以下 AWS KMS 操作添加到与 EventBridge 调用 Scheduler API 的任何委托人（用户、群组或角色）关联的权限策略中。

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

```

        "scheduler:*",

        # Required to pass the execution role
        "iam:PassRole",

        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
]
}

```

- **kms:DescribeKey** : 验证您提供的密钥是否为[对称](#)加密 KMS 密钥所必需。
- **kms:GenerateDataKey**— 需要生成 EventBridge 调度器用于执行客户端加密的数据密钥。
- **kms:Decrypt**— 必需解密 EventBridge 调度器与您的加密数据一起存储的加密数据密钥。

客户托管密钥的执行角色权限

您必须在日程安排的执行角色权限策略中添加以下操作，以便在解密数据时允许 EventBridge 调度程序调用 AWS KMS API。

```

{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Sid" : "Allow EventBridge Scheduler to decrypt data using a customer managed key",
      "Effect" : "Allow",
      "Action" : [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:your-region:123456789012:key/your-key-id"
    }
  ]
}

```

- **kms:Decrypt**— 必需解密 EventBridge 调度器与您的加密数据一起存储的加密数据密钥。

如果您在创建新计划时使用 EventBridge 调度器控制台创建新的执行角色，则 EventBridge 调度器会自动将所需的权限附加到您的执行角色。但是，如果您选择现有的执行角色，则必须向该角色添加所需的权限才能使用您的客户托管密钥。

管理密钥策略

默认情况下 AWS KMS，当您使用客户托管密钥创建客户托管密钥时，您的密钥具有以下密钥策略，可提供对计划执行角色的访问权限。

```
{
  "Id": "key-policy-1",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Provide required IAM Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    }
  ]
}
```

或者，您可以将密钥策略的范围限制为仅提供对执行角色的访问权限。如果您只想将客户托管密钥与您的 EventBridge 日程安排器资源一起使用，则可以这样做。使用以下[密钥策略](#)示例来限制哪些 EventBridge 计划程序资源可以使用您的密钥。

```
{
  "Id": "key-policy-2",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Provide required IAM Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::695325144837:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
  ],
}
```

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:role/schedule-execution-role"
  },
  "Action": [
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

CloudTrail 事件示例

AWS CloudTrail 捕获所有 API 调用事件。这包括每当 S EventBridge scheduler 使用您的客户托管密钥解密您的数据时，都会调用 API。以下示例显示了一个 CloudTrail 事件条目，该条目演示 S EventBridge scheduler 使用客户托管密钥使用该 `kms:Decrypt` 操作。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDEABCD1AB12ABABAB0:70abcd123a123a12345a1aa12aa1bc12",
    "arn": "arn:aws:sts::123456789012:assumed-role/execution-role/70abcd123a123a12345a1aa12aa1bc12",
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGH1JKLMNOP2Q3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ABCDEABCD1AB12ABABAB0",
        "arn": "arn:aws:iam::123456789012:role/execution-role",
        "accountId": "123456789012",
        "userName": "execution-role"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-10-31T21:03:15Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}
```

```
    },
    "eventTime": "2022-10-31T21:03:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-north-1",
    "sourceIPAddress": "13.50.87.173",
    "userAgent": "aws-sdk-java/2.17.295 Linux/4.14.291-218.527.amzn2.x86_64 OpenJDK_64-
Bit_Server_VM/11.0.17+9-LTS Java/11.0.17 kotlin/1.3.72-release-468 (1.3.72) vendor/
Amazon.com_Inc. md/internal exec-env/AWS_ECS_FARGATE io/sync http/Apache cfg/retry-
mode/standard AwsCrypto/2.4.0",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:123456789012:key/2321abab-2110-12ab-a123-
a2b34c5abc67",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "aws:scheduler:schedule:arn": "arn:aws:scheduler:us-
west-2:123456789012:schedule/default/execution-role"
        }
    },
    "responseElements": null,
    "requestID": "request-id",
    "eventID": "event-id",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:123456789012:key/2321abab-2110-12ab-a123-
a2b34c5abc67"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

传输中加密

EventBridge 调度器会在传输中的数据在网络中传输时对其进行加密。传输层安全 (TLS) 会在您调用任何调度程序 API 操作时以及 EventBridge 调度程序在调用您的 EventBridge 计划时调用任何目标 API 时对你的数据进行加密。默认情况下，EventBridge 计划程序在加密传输中的数据时使用 TLS 1.2。使用 EventBridge 计划程序时，您无需配置传输中的加密，也无法选择不同的 TLS 版本。

使用 EventBridge 调度器 API — 当您执行 API 操作（例如）时 `CreateSchedule`，EventBridge 调度器会加密整个 HTTP 请求，包括请求正文和标头。EventBridge 调度器还会对您从我们的 API 收到的整个响应对象进行加密。

使用目标 API-当 EventBridge 调度器调用您的计划时，它会调用您在创建计划时指定的目标 API。向目标传送事件时，S EventBridge cheduler 会加密整个请求，包括请求正文和所有标头，以及它从目标收到的响应。

Amazon EventBridge 计划程序的合规性验证

要了解是否属于特定合规计划的范围，请参阅 AWS 服务“[按合规计划划分的范围](#)”，然后选择您感兴趣的合规计划。AWS 服务 有关一般信息，请参阅[AWS 合规计划AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的“[下载报告](#)”中的“[AWS Artifact](#)”。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。AWS 提供了以下资源来帮助实现合规性：

- [安全与合规性快速入门指南](#) — 这些部署指南讨论了架构注意事项，并提供了部署以安全性和合规性为重点 AWS 的基准环境的步骤。
- 在 [Amazon Web Services 上构建 HIPAA 安全与合规性](#) — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

Note

并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅[符合 HIPAA 要求的服务参考](#)。

- [AWS 合规资源AWS](#) — 此工作簿和指南集可能适用于您所在的行业和所在地区。

- [AWS 客户合规指南](#) — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践，AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- [使用 AWS Config 开发人员指南中的规则评估资源](#) — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- [AWS Security Hub](#) — 这 AWS 服务可以全面了解您的安全状态 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实操。有关受支持服务及控制的列表，请参阅 [Security Hub 控制参考](#)。
- [AWS Audit Manager](#) — 这 AWS 服务可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

Amazon EventBridge 调度器中的弹性

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。AWS 区域 提供多个物理分隔和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础架构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅[AWS 全球基础设施](#)。

除了 AWS 全球基础架构外，S EventBridge scheduler 还提供多项功能来帮助支持您的数据弹性和备份需求。

Amazon EventBridge 计划程序中的基础设施安全

作为一项托管服务，Amazon EventBridge Scheduler 受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅[AWS 云安全](#)。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS security Pillar Well-Architected Framework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用通过网络访问 EventBridge 调度程序。客户端必须支持以下内容：

- 传输层安全性协议 (TLS) 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE (Ephemeral Diffie-Hellman) 或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用[AWS Security Token Service](#) (AWS STS) 生成临时安全凭证来对请求进行签名。

Amazon EventBridge 调度器的监控和指标

监控是保持 Amazon EventBridge 调度器和您的其他 AWS 解决方案的可靠性、可用性和性能的重要方面。AWS 提供了以下一些监控工具来监控 EventBridge 调度器、在出现错误时进行报告并适时自动采取措施：

- Amazon CloudWatch 实时监控您的 AWS 资源以及在 AWS 上运行的应用程序。您可以收集和跟踪指标，创建自定义的控制面板，以及设置警报以在指定的指标达到您指定的阈值时通知您或采取措施。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。
- AWS CloudTrail 捕获由您的 AWS 账户或代表该账户发出的 API 调用和相关事件，并将日志文件传输到您指定的 Amazon S3 存储桶。您可以标识哪些用户和账户调用了 AWS、从中发出调用的源 IP 地址以及调用的发生时间。有关更多信息，请参阅 [AWS CloudTrail 用户指南](#)。

主题

- [使用 Amazon CloudWatch 监控 Amazon EventBridge 调度器](#)
- [使用 AWS CloudTrail 记录 Amazon EventBridge 调度器 API 调用](#)

使用 Amazon CloudWatch 监控 Amazon EventBridge 调度器

您可以使用 CloudWatch 监控 Amazon EventBridge 调度器，CloudWatch 会收集原始数据并将其处理为易读且近乎实时的指标。EventBridge 调度器会为所有计划发布一组指标，为关联死信队列 (DLQ) 的计划发出一组额外的指标。如果您为计划 [配置了 DLQ](#)，则当您的计划用尽其重试策略时，EventBridge 调度器会发布其他指标。

这些统计数据会保存 15 个月，从而使您能够访问历史信息，并能够更好地了解计划失败的原因，并对潜在问题进行故障排除。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 [《Amazon CloudWatch 用户指南》](#)。

主题

- [术语](#)
- [维度](#)
- [访问指标](#)
- [指标的列表](#)

术语

命名空间

命名空间是用于 AWS 服务的 CloudWatch 指标的容器。对于 EventBridge 调度器，命名空间为 AWS/Scheduler。

CloudWatch metrics (CloudWatch 指标)

CloudWatch 指标表示 CloudWatch 特定的并且按时间排序的数据点集。

维度

维度是名称/值对，是指标身份的一部分。

单位

所有统计数据都有度量单位。对于 EventBridge 调度器，单位包括计数。

维度

本节介绍了 CloudWatch 中 EventBridge 调度器指标的 CloudWatch 维度分组。

维度	描述
计划组	您用于使用 CloudWatch 查看指标的计划组。如果您尚未创建任何组，则 EventBridge 调度器会将您的计划与 default 组关联起来。

访问指标

本节介绍如何在 CloudWatch 中访问特定 EventBridge 调度器计划的性能指标。

要查看维度的性能指标，请执行以下操作

1. 在 CloudWatch 控制台上打开[指标页面](#)。
2. 使用 AWS 区域选择器来选择您计划的区域
3. 选择调度器命名空间。
4. 在所有指标选项卡中，选择一个维度，例如，计划组指标。要查看您在所选区域创建的所有计划的指标，请选择账户指标。

5. 选择 CloudWatch 指标维度。例如，InvocationAttemptCount 或 InvocationDroppedCount，然后选择 Graph 搜索。
6. 选择图表化指标选项卡，查看 EventBridge 调度器指标的性能统计信息。

指标的列表

下表列出了所有 EventBridge 调度器计划的指标，以及您配置了 DLQ 的计划的指标。

所有计划的指标

命名空间	指标	单位	描述
AWS/Scheduler	InvocationAttemptCount	计数	每次尝试调用时都会发出。使用此指标来检查 EventBridge 调度器是否正在尝试调用您的计划，并查看调用次数何时接近您的账户配额。
AWS/Scheduler	TargetErrorCount	计数	EventBridge 调度器调用目标 API 后，目标返回异常时发出。使用它来检查向目标传输失败的时间。
AWS/Scheduler	TargetErrorThrottledCount	计数	在由于目标的 API 节流而导致目标调用失败时发出。当根本原因是

命名空间	指标	单位	描述
			EventBridge 调度器发出的目标 API 节流调用时，使用它来诊断传送失败
AWS/Scheduler	InvocationThrottleCount	计数	当 EventBridge 调度器因目标调用超出其设置的服务限额而限制目标调用时发出。使用它来确定何时超过了 EventBridge 调度器配额。有关服务限额的更多信息，请参阅 限额 。

命名空间	指标	单位	描述
AWS/Scheduler	InvocationDroppedCount	计数	当计划的重试策略用尽后，EventBridge 调度器停止尝试调用目标时发出。有关重试策略的更多信息，请参阅《EventBridge 调度器 API 参考》中的 RetryPolicy 。

带有 DLQ 的计划的指标

命名空间	指标	单位	描述
AWS/Scheduler	InvocationsSentToDeadLetterCount	计数	每次成功传输到计划的 DLQ 时都会发出。使用它来确定何时将事件发送到 DLQ，然后查看传输到计划 DLQ 的事件，了解有助于确定失败原因的其他详细信息。

命名空间	指标	单位	描述
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount	计数	在 EventBridge 调度器无法向 DLQ 传输事件时发出。
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount_<error_code>	计数	<p>使用这两个指标来确定 EventBridge 调度器无法向 DLQ 发送事件的原因，并修改您的 DLQ 配置以解决问题。</p> <p>以下是您指定为 DLQ 的 Amazon SQS 队列不存在时的 <code>InvocationsFailedToBeSentToDeadLetterCount_<error_code></code> 指标示例：<code>InvocationsFailedToBeSentToDeadLetterCount_ AWS.SimpleQueueService.NonE</code></p>

命名空间	指标	单位	描述
			xistentQueue
AWS/Scheduler	InvocationsSentToDeadLetterCount_Truncated_MessageSize Exceeded	计数	当发送到 DLQ 的事件的有效负载超过 Amazon SQS 允许的最大大小，且 EventBridge 调度器截断您在计划的 Input 属性中指定的有效负载时发出。

使用 AWS CloudTrail 记录 Amazon EventBridge 调度器 API 调用

Amazon EventBridge 调度器与 AWS CloudTrail 集成，后者是提供用户、角色或 AWS 服务在 EventBridge 调度器中所采取操作的记录的服务。CloudTrail 将 EventBridge 调度器的所有 API 调用作为事件捕获。捕获的调用包括来自 EventBridge 调度器控制台的调用和对 EventBridge 调度器 API 操作的代码调用。如果您创建了一个跟踪，则可以使 CloudTrail 事件持续传输到 Amazon S3 存储桶（包括 EventBridge 调度器的事件）。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 EventBridge 调度器发出了什么请求、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，请参阅《[AWS CloudTrail 用户指南](#)》。

CloudTrail 中的 EventBridge 调度器信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 EventBridge 调度器中发生活动时，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 EventBridge 调度器事件），请创建跟踪。通过跟踪记录，CloudTrail 可将日志文件传送至 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪记录时，

此跟踪记录应用于所有AWS 区域。此跟踪记录在 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，请参阅以下内容：

- [创建跟踪记录概览](#)
- [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](#)

所有 Amazon EventBridge 调度器操作都由 CloudTrail 记录，并记录在 [Amazon EventBridge 调度器 API 参考](#)中。例如，对 CreateSchedule、UpdateSchedule 和 DeleteSchedule 操作的调用会在 CloudTrail 日志文件中生成条目。

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。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

了解如何使用 EventBridge 调度器日志文件条目

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，包括有关所请求的操作、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。

Amazon EventBridge 计划程序的配额

您的 AWS 账户对每项 AWS 服务都有默认配额（以前称为限制）。除非另有说明，否则，每个限额都特定于区域。您可以请求增加某些配额，但一些配额无法增加。

要查看 EventBridge 计划程序的配额，请打开 [Service Quotas 控制台](#)。在导航窗格中，选择 AWS 服务，然后选择 EventBridge 日程安排。

要请求提高配额，请参阅《服务限额用户指南》中的 [请求提高限额](#)。如果配额在服务限额中尚不可用，请使用 [提高限制表格](#)。

Note

EventBridge 调度器的 CreateScheduleUpdateScheduleGetSchedule、和每秒 DeleteSchedule 交易量 (TPS) 配额最多可调整为数千个 TPS。调用节流配额可调至数万 TPS。

您的 AWS 账户具有以下与 EventBridge 日程安排相关的配额。

名称	默认值	可调整	描述
CreateSchedule 请求速率	每个受支持的区域：50 个	<u>是</u>	每秒最大 CreateSchedule 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
CreateScheduleGroup 请求速率	每个受支持的区域：10 个	<u>是</u>	每秒最大 CreateScheduleGroup 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。

名称	默认值	可调整	描述
DeleteSchedule 请求速率	每个受支持的区域：50 个	是	每秒最大 DeleteSchedule 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
DeleteScheduleGroup 请求速率	每个受支持的区域：10 个	是	每秒最大 DeleteScheduleGroup 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
GetSchedule 请求速率	每个受支持的区域：50 个	是	每秒最大 GetSchedule 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
GetScheduleGroup 请求速率	每个受支持的区域：10 个	是	每秒最大 GetScheduleGroup 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
调用每秒事务数节流限制	每个受支持的区域：500 个	是	调用是传送到指定目标的计划负载。在达到限制后，调用将被节流；即，调用仍会发生，但会延迟。

名称	默认值	可调整	描述
ListScheduleGroups 请求速率	每个受支持的区域：10 个	是	每秒最大 ListScheduleGroups 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
ListSchedules 请求速率	每个受支持的区域：50 个	是	每秒最大 ListSchedules 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。
ListTagsForResource 请求速率	每个受支持的区域：10 个	是	列出与调度器资源关联的标签。
计划组数	每个受支持的区域：500 个	是	每个区域的最大计划组数。
计划数	每个受支持的区域：100 万个	是	每个区域的最大计划数。此限额包括已完成运行的一次性计划。我们建议在一次性计划完成运行并调用目标后将其删除。
TagResource 请求速率	每个受支持的区域：1 个	是	将一个或多个标签（键值对）分配给指定调度器资源。
UntagResource 请求速率	每个受支持的区域：1 个	是	删除指定调度器资源的一个或多个标签。

名称	默认值	可调整	描述
UpdateSchedule 请求速率	每个受支持的区域：50 个	<u>是</u>	每秒最大 UpdateSchedule 请求数。当您达到此配额时，EventBridge 调度器会在剩余时间间隔内拒绝此操作的请求。

有关 EventBridge 计划程序配额和服务终端节点的更多信息，请参阅AWS 通用参考指南中的 [Amazon EventBridge 计划程序终端节点和配额](#)。

《EventBridge 调度器用户指南》文档历史记录

下表介绍 EventBridge 调度器的文档版本。

变更	说明	日期
执行角色和混淆代理预防方面的变化	<p>此更新描述了当您在角色的权限策略中实施混淆代理预防时，将执行角色应用于计划组资源的方法发生了变化。</p> <ul style="list-style-type: none">• the section called “混淆代理问题防范”	2023 年 9 月 7 日
计划完成后自动删除	<p>EventBridge 调度器支持自动删除。配置自动删除时，EventBridge 调度器会在上次计划调用后删除您的计划。</p> <ul style="list-style-type: none">• the section called “计划完成后删除”	2023 年 8 月 2 日
更新了有关使用通用目标的主题	<p>更新了 EventBridge 调度器可以定位和集成的支持服务列表。此更新还包括不支持的 GET API 操作列表，包括对通用目标示例的改进以及对整个指南的其他细微改进。</p> <ul style="list-style-type: none">• the section called “使用通用目标”	2023 年 3 月 17 日
更新了没有起始日期、基于速率的计划的信息	<p>添加的有关 EventBridge 调度器如何处理基于速率计划的信息 (如果您未指定 StartDate)。</p>	2023 年 3 月 17 日

- [the section called “基于速率的计划”](#)
- [关于管理计划组的新主题](#) 添加了有关如何使用 EventBridge 调度器创建计划组的新章节。通过本章学习如何创建组、向组添加计划、应用标签以更轻松地管理和监控您的 EventBridge 调度器资源，以及如何删除组。 2023 年 3 月 17 日
- [管理计划组](#)
- [有关夏令时和时区的新主题](#) 添加了新的章节，描述了 EventBridge 调度器如何处理夏令时，以及如何在不同时区创建计划。 2022 年 11 月 17 日
- [the section called “夏令时”](#)
 - [the section called “时区”](#)
- [关于指标的新主题](#) 添加了新主题，描述了 EventBridge 调度器向 CloudWatch 发布的指标。您可以使用这些指标来监控调用失败情况并了解如何解决计划的问题。 2022 年 11 月 15 日
- [the section called “使用 CloudWatch 进行监控”](#)
- [初始版本](#) 《EventBridge 调度器用户指南》初始版本。 2022 年 11 月 10 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。