



用户指南

EventBridge 调度器



EventBridge 调度器: 用户指南

Copyright © 2026 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 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 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
报名参加 AWS	2
注册获取 AWS 账户	2
使用托管策略	2
设置执行角色	3
设置目标	7
接下来做什么？	9
入门	10
先决条件	10
使用控制台	11
使用 AWS CLI	14
使用 SDK	14
接下来做什么？	16
计划类型	17
基于速率的计划	17
语法	18
示例	18
基于 Cron 的计划	18
语法	19
示例	20
一次性计划	20
语法	21
示例	21
时区	21
夏令时	21
管理计划	23
更改计划状态	23
配置灵活的时间窗口	25
配置 DLQ	26
创建 Amazon SQS 队列	26
设置执行角色权限	27

指定死信队列	28
检索死信事件	29
删除计划	32
计划完成后删除	32
手动删除	33
接下来做什么？	33
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	43
Lambda ##	45
Step Functions StartExecution	47
使用通用目标	49
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
AWS PrivateLink	54
注意事项	54
创建接口端点	54
创建端点策略	54
安全性	56
管理访问权限	56
受众	57
使用身份进行身份验证	57
使用策略管理访问	58
集成 IAM	59
使用基于身份的策略	64
防范混淆代理问题	74
问题排查	76
数据保护	78

静态加密	78
传输中加密	84
合规性验证	85
恢复能力	85
基础设施安全性	85
监控和指标	86
使用监控 CloudWatch	86
术语	87
Dimensions	87
访问指标	88
指标列表	88
使用情况指标	92
使用 CloudTrail 日志进行监控	94
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
配额	97
排查限额问题	106
ServiceQuotaExceededException	106
问题排查	108
目标错误	108
常见原因 :	108
故障排除步骤	108
角色权限	110
常见原因	110
症状	110
故障排除步骤	110
服务配额	112
确定配额问题	113
解决配额问题	113
模式和触发时机	113
常见原因	113
故障排除步骤	113
创建模式	115
常见问题	115
故障排除步骤	115
我的目标是否被触发?	115

模板化目标与通用目标	116
通用目标输入无效	116
症状	116
示例	116
解决方法	117
调度更新触发意外调用	118
禁用或启用一次性调度	118
文档历史记录	119
.....	cxxii

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结构，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 控制台、EventBridge 调度器开发工具包、AWS CLI，或者通过直接使用 EventBridge 调度器 API 来使用 EventBridge 调度器。

设置 Amazon EventBridge 日程安排

在使用 EventBridge 日程安排器之前，必须完成以下步骤。

主题

- [报名参加 AWS](#)
- [使用托管策略](#)
- [设置执行角色](#)
- [设置目标](#)
- [接下来做什么？](#)

报名参加 AWS

注册获取 AWS 账户

要开始使用 AWS，你需要一个 AWS 账户。有关创建的信息 AWS 账户，请参阅《AWS 账户管理 参考指南》AWS 账户中的[入门](#)指南。

使用托管策略

设置 AWS 账户和管理用户后，我们建议您创建单独的用户、群组或角色，这些用户、群组或角色仅具有使用 EventBridge 日程安排程序所需的权限。EventBridge 对于常见用例，计划程序支持以下托管策略。

- AmazonEventBridgeSchedulerFullAccess — 使用控制台和 API 授予对 EventBridge 调度程序的完全访问权限。
- AmazonEventBridgeSchedulerReadOnlyAccess — 授予对 EventBridge 日程安排器的只读访问权限。

您可以将这些托管策略附加到您的 IAM 委托人。有关使用基于身份的 IAM 策略管理对 EventBridge 调度程序的访问权限的更多信息，请参阅。[the section called “使用基于身份的策略”](#)

设置执行角色

执行角色是一个 IAM 角色，由 EventBridge 计划程序代替您与其他 AWS 服务角色进行交互。您可以将权限策略附加到此角色以授予 EventBridge 调度程序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，S EventBridge scheduler 会代表您创建一个角色，其权限基于您选择的目标。当 EventBridge Scheduler 为您创建角色时，该角色的信任策略包括限制哪些委托人可以代表您担任该角色的[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”](#)。

要创建执行角色，请使用 AWS CLI

1. 复制以下 JSON 格式的角色代入策略，并作为 Scheduler-Execution-Role.json 保存到本地。此信任策略允许 EventBridge 调度员代表您担任该角色。

JSON

```
{
  "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 “防范混淆代理问题”](#)。

- 在 AWS Command Line Interface (AWS CLI) 中，输入以下命令以创建新角色。将 *SchedulerExecutionRole* 替换为您想授予此角色的名称。

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

成功替换后，您将看到以下输出内容：

```
{
  "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"
        }
      ]
    }
  }
}
```

- 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 日程安排器对您账户中的所有 Amazon SQS 队列进行 `sqs:SendMessage` 操作。

JSON

```
{
```

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

Amazon SNS – Publish

以下内容允许 EventBridge 日程安排器对您账户中的所有 Amazon SNS 主题进行sns:Publish操作。

JSON

```
{
  "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。

JSON

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

4. 运行以下命令，新建权限策略。将 *PolicyName* 替换为您想授予此策略的名称。

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

如果成功，您将会看到以下输出。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": "2022-03-015T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/PolicyName",
    "UpdateDate": "2022-03-015T19:31:18.620Z"
  }
}
```

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* 替换为在上一步中创建的策略 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 调度程序计划之前，至少需要一个目标供调度调用。您可以使用现有 AWS 资源，也可以创建新资源。以下步骤说明如何使用创建新的标准 Amazon SQS 队列。CloudFormation

创建新的 Amazon SQS 队列

1. 复制以下 JSON CloudFormation 模板并将其另存为本地 `Scheduler-Target-SQS.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"
        ]
      }
    }
  }
}
```

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

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

成功替换后，您将看到以下输出内容：

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

3. 运行以下命令以查看 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，您将使用值QueueARN将队列设置为 EventBridge 调度程序的目标。

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 调度器调度程序并调用目标。

开始使用 EventBridge 调度器

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 管理控制台，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 管理控制台的区域选择器切换 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，使用 cron 表达式配置计划。要使用 rate 表达式，请选择基于 rate 的计划，并在值中输入一个正数，然后从下拉选项中选择单位。

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. 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. 对于事件的最长保留时间 - 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，然后从下拉列表中选择 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](#)。

日程安排 EventBridge 器中的计划类型

以下主题介绍了 Amazon S EventBridge scheduler 支持的不同计划类型，以及日 EventBridge 程安排器如何处理夏令时和在不同时区进行调度。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 调度器计算该表达式的时区。

一次性计划仅调用一次目标。您可以通过指定调度 EventBridge 器评估日程安排的时间、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（假设没有设置灵活的时间窗口）。

使用以下部分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 S EventBridge scheduler 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](#)
- [日 EventBridge 程安排器上的时区](#)
- [日 EventBridge 程安排器上的夏令时](#)

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其目标的计划，则可以使用基于速率的计划来实现此目的。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)
```

值

一个正数。

unit

您希望您的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费率表达式和 AWS CLI `create-schedule` 命令来配置基于费率的时间表。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重复计划，该计划在您选择的特定时间运行。EventBridge 计划程序支持按照通用协调时间 (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。当您需要 EventBridge 调度程序的某个速率表达式不支持

的自定义重复计划时，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 day-of-month、day-of-week、月，以及一个可选字段“年”，如下所示。

语法

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

字段	值	通配符
Minutes	0-59	, - * /
Hours	0-23	, - * /
Day-of-month	1-31	, - * ? / L W
Month	1-12 或 JAN-DEC	, - * /
Day-of-week	1-7 或 SUN-SAT	, - * ? L #
Year	1970-2199	, - * /

通配符

- , (逗号) 通配符包含其他值。在“Month”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“Day”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”字段中，* 包括每个小时。不能在 Day-of-month 和 Day-of-week 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“Minutes”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。在该 Day-of-month 字段中，您可以输入 7，如果一周中的任何一天可以接受，则可以输入？ 在 Day-of-week 野外。
- 或 字段中的 Day-of-monthL Day-of-week 通配符用于指定月或周的最后一天。

- 该 Day-of-month 字段中的 W 通配符指定工作日。在该 Day-of-month 字段中，**3W** 指定最接近该月第三天的工作日。
- Day-of-week 字段中的 # 通配符指定一个月内一周中指定某一天的特定实例。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 day-of-week 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 的时间表。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

一次性计划只能在您使用有效日期和时间戳指定的日期和时间戳调用目标一次。EventBridge 调度器支持按通用协调时间 (UTC) 或您在创建计划时指定的时区进行调度。

Note

一次性计划在完成运行并调用其目标后，仍会计入您的账户配额。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 S EventBridge scheduler 调用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，日程 EventBridge 安排器会忽略EndDate您为计划指定的StartDate和。

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" }'
```

日 EventBridge 程安排器上的时区

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 调度程序使用由互联网号码分配机构 (IANA) 维护的[时区数据库](#)。

借助 AWS CLI，您可以使用 --schedule-expression-timezone 参数设置希望日 EventBridge 程安排器在哪个时区评估您的日程安排。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" }'
```

日 EventBridge 程安排器上的夏令时

EventBridge 日程安排器会自动调整您的日程安排以适应夏令时。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 日程安排器会根据您在创建计划时指定的时区来调整您的日程安排。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到 23 小时或延长到 25 小时时，EventBridge 调度器仍会在上次调用 24 小时后计算速率表达式。

Note

根据当地规章制度，某些时区不实行夏令时。如果您在不遵守夏令时的时区创建计划，则日 EventBridge 程安排器不会调整您的日程安排。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。

- Spring- forward — 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调度调用，并在第二天恢复正常运行日程安排。
- fall-back — 秋季时间从凌晨 2:59 向后移动到凌晨 2:00 时，EventBridge 调度器只在轮班发生前的凌晨 2:30 运行一次计划，但不会在时移后的凌晨 2:30 再次重复调度。

在日程安排器中管理日程 EventBridge 安排

计划是您使用 Amazon EventBridge 计划程序创建、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：费率、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。目标是一个 API 操作，EventBridge 计划程序每次运行时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见 API 操作，以及通用目标参数 (UTP)，可用于在 270 多个服务中调用 6,000 多个操作。有关配置目标的更多信息，请参阅 [管理目标](#)。

使用两种主要机制：重试策略和死信队列 (DLQ)，您可以配置计划在 EventBridge 调度程序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 是一种标准 Amazon SQS EventBridge 队列调度程序，用于在重试策略用尽后向其传送失败的事件。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 DLQ”](#)。

在本节中，您可以找到使用控制台、AWS CLI 和 EventBridge 日程安排器管理日 EventBridge 程安排 SDKs 的示例。

主题

- [在“日程安排 EventBridge 器”中更改日程表状态](#)
- [在 EventBridge 调度器中配置灵活的时间窗口](#)
- [在调度器中配置日程安排的死信队列 EventBridge](#)
- [在“日程安排”中删除日 EventBridge 程安排](#)
- [接下来做什么？](#)

在“日程安排 EventBridge 器”中更改日程表状态

日 EventBridge 程安排有两种状态：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。

使用时 UpdateSchedule，必须提供所有必需的参数。EventBridge 日程安排器会用您提供的信息替换您的日程安排。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 SDK

```
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')
```

在 EventBridge 调度器中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，S EventBridge scheduler 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 S EventBridge scheduler SDK 示例用于 UpdateSchedule 为每小时运行一次的计划设置 15 分钟的弹性时间窗口。

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 SDK

```
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)
```

在调度器中配置日程安排的死信队列 EventBridge

Amazon EventBridge Scheduler 支持使用亚马逊简单队列服务的死信队列 (DLQ)。当计划无法调用其目标时，EventBridge 计划程序会将包含调用详细信息和从目标收到的任何响应的 JSON 有效负载传送到您指定的 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 调度程序会传送已耗尽您设置的最大重试次数的死信事件。

以下主题介绍如何将 Amazon SQS 队列配置为日程安排的 DLQ，如何设置计划 EventBridge 程序向 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 调度程序不支持使用 FIFO 队列作为日程安排的 DLQ。

使用以下 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，请记住队列 ARN。当你为日程安排指定 DLQ 时，你需要 ARN。EventBridge 您可以在 Amazon SQS 控制台中找到您的队列 ARN，也可以使用命令来查找。[get-queue-attributes](#)
AWS CLI

```
$ 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，您将向计划执行角色添加所需的权限，以允许计划 EventBridge 程序向 Amazon SQS 传送死信事件。

设置执行角色权限

要让 S EventBridge scheduler 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](#)。

JSON

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

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

Note

如果您使用计划程序调用 Amazon SQS API 目标，则您的 EventBridge 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 日程安排控制台并为您的日程安排指定 DLQ。

指定死信队列

要指定 DLQ，请使用 EventBridge 调度程序控制台或更新现有计划，或者创建一个新的计划。AWS CLI

Console

使用控制台指定 DLQ

1. 登录 AWS 管理控制台，然后选择以下链接以打开 EventBridge 控制台的“EventBridge 日程安排”部分：<https://console.aws.amazon.com/scheduler/home>
2. 在 EventBridge 日程安排控制台上，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 Dead-letter 队列 (DLQ)，执行以下任一操作：
 - 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你的 DLQ 选择队列 ARN。
 - 选择将其他 AWS 账户中的 Amazon SQS 队列指定为 DLQ，然后输入您的 DLQ 队列 ARN。如果您选择其他 AWS 账户中的队列，则 EventBridge 日程安排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 Dead-letter 队列选项卡。

AWS CLI

要更新现有时间表，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"}
```

要使用 DLQ 创建新计划，请使用 AWS CLI

- 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"}
```

在下一节中，您将使用接收 AWS CLI 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/
w3htX6r3dXpXvvFNpGoQb8ihY37+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，以帮助您识别和排除目标调用失败的可能原因。

- **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`。如果由于调 EventBridge 度器问题而导致计划无法调用目标，则会改为看到以下错误代码：`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`。如果由于计划 EventBridge 程序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。`MaximumRetryAttempts` 如果 EventBridge 调度程序在超过您为计划配置的最大重试尝试次数后将其发送到 `DLQMaximumEventAgeInSeconds`，或者事件已超过您在计划中配置的最大持续时间但仍无法传送，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。

Note

如果您使用[通用目标](#)，请注意 EventBridge 调度器不会在创建计划时验证 Input 字段的内容。带有无效输入参数的计划将成功创建，但每次调用都会失败。DLQ 消息将包含来自目标服务的错误代码和消息，这可以帮助您识别无效的参数。有关更多信息，请参阅[通用目标输入配置无效](#)。

在“日程安排”中删除日 EventBridge 程安排

您可以通过配置自动删除或手动删除单个计划来删除计划。使用以下主题来了解如何使用这两种方法删除计划，以及为什么可以选择一种方法而不是另一种方法。

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 S EventBridge scheduler 上单独管理您的日程安排资源，请配置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。

为计划配置自动删除时，计划程序会在上次 EventBridge 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。如果您为计划配置了自动删除功能，但未为其指定值 EndDate，则日 EventBridge 程安排程序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 管理控制台

1. 打开 EventBridge 日程安排器控制台，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. 在计划完成后的操作部分，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 `--action-after-completion` 设置为 `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 调度程序数据类型和 API 操作的更多信息，请参阅[EventBridge 调度器 API 参考](#)。

在 EventBridge 调度器中管理计划组

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。还可以将计划与组相关联。最后，可以删除组。

主题

- [在 EventBridge 调度器中创建计划组](#)
- [在 EventBridge 调度器中删除计划组](#)
- [相关资源](#)

在 EventBridge 调度器中创建计划组

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 管理控制台

使用 AWS 管理控制台 创建新组

1. 登录 AWS 管理控制台并打开 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 管理控制台

要将计划与组关联，请使用 AWS 管理控制台

1. 登录 AWS 管理控制台并打开 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[入门](#)。
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。

在 EventBridge 调度器中删除计划组

在下文中，您可以学习如何使用 AWS 管理控制台 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 管理控制台

使用 AWS 管理控制台 删除组

1. 登录 AWS 管理控制台并打开 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。如果您没有看到要查找的组，请选择其他 AWS 区域。

 Note

您不能删除或编辑默认组。

4. 选中您要删除的组对应的复选框。
5. 选择删除。
6. 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。
7. 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。
8. 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

1. 打开新的命令提示符窗口。
2. 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。

3. 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。

在 EventBridge 调度器中管理目标

以下主题描述了如何在 EventBridge 调度器中使用模板化目标和通用目标，并提供了支持 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 计划程序的通用目标参数 (UTP) 通过操作创建新的 Amazon SQS 队列。[CreateQueue](#)

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的 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

- [在调度器中 EventBridge 使用模板化目标](#)
- [在 EventBridge 调度器中使用通用目标](#)
- [在 EventBridge 调度器中添加上下文属性](#)
- [接下来做什么？](#)

在调度器中 EventBridge 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](#)

 Important

EventBridge 当 EventBridgePutEvents 用作模板化目标时，调度器不支持跨区域交付。只有当事件总线与调度相同 AWS 区域时，您 AWS 账户 才可以使用此目标将事件发送到不同位置的事件总线。要将事件发送到不同区域的事件总线，请在目标事件总线所在的区域中创建时间表。

- Amazon Inspector – [StartAssessmentRun](#)
- Kinesis – [PutRecord](#)
 - 参数：[KinesisParameters](#)
- Firehose – [PutRecord](#)
- Lambda – [Invoke](#)
- SageMaker 人工智能 — [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

JSON

```
{
  "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 SDK

```
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

Example 执行角色的权限策略

JSON

```
{  
  "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 SDK

```
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

JSON

```
{
  "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 SDK

```
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");
}
}
```

在 EventBridge 调度器中使用通用目标

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。

要使用或其中一个计划程序 AWS CLI为您的 EventBridge 计划配置通用目标 SDKs，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`。

Note

通用目标 ARN 中的 *service* 值必须与目标服务的 AWS SDK 服务标识符匹配。此标识符可能与服务的终端节点前缀不同。例如，对于亚马逊 Cognito 身份提供商，请使用 `cognitoidentityprovider`（不是 `cognito-idp`）。要查找正确的服务标识符，请参阅要定位的服务的 AWS SDK 文档。

- 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 调度器不支持以下前缀列表开头的只读 API GET 操作，例如常见操作：

```
get
```

```
describe
list
poll
receive
search
scan
query
select
read
lookup
discover
validate
batchGet
batchDescribe
batchRead
transactGet
adminGet
adminList
testMigration
retrieve
testConnection
translateDocument
isAuthorized
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 Scheduler 不支持此目标。同样，不支持将 Amazon MQ 操作 [ListBrokers](#) 作为目标，因为该操作以前缀 `list` 开始。

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](#)。

要确定可以为不同服务设置的参数 APIs，请参阅该服务的 API 参考。与 Lambda 类似 `Invoke`，有些 APIs 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");
    }
}
```

在 EventBridge 调度器中添加上下文属性

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

使用接口端点 (AWS PrivateLink) 访问 Amazon EventBridge 调度器

您可以使用 AWS PrivateLink 在您的 VPC 和 Amazon EventBridge 调度器之间创建私有连接。您可以像在 VPC 中一样访问 EventBridge 调度器，而无需使用互联网网关、NAT 设备、VPN 连接或 Direct Connect 连接。VPC 中的实例不需要公有 IP 地址即可访问 EventBridge 调度器。

您可以通过创建由 AWS PrivateLink 提供支持的接口端点来建立此私有连接。我们将在您为接口端点启用的每个子网中创建一个端点网络接口。这些是请求者托管的网络接口，用作发往 EventBridge 调度器的流量的入口点。

有关更多信息，请参阅《AWS PrivateLink 指南》中的[通过 AWS PrivateLink 访问 AWS 服务](#)。

EventBridge 调度器的注意事项

在为 EventBridge 调度器设置接口端点之前，请首先查看《AWS PrivateLink 指南》中的[注意事项](#)。

EventBridge 调度器支持通过接口端点调用其所有 API 操作。

为 EventBridge 调度器创建接口端点

您可以使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI) 为 EventBridge 调度器创建接口端点。有关更多信息，请参阅《AWS PrivateLink 指南》中的[创建接口端点](#)。

使用以下服务名称为 EventBridge 调度器创建接口端点：

```
com.amazonaws.region.scheduler
```

如果为接口端点启用私有 DNS，则可使用其默认区域 DNS 名称向 EventBridge 调度器发出 API 请求。例如 scheduler.us-east-1.amazonaws.com。

为 VPC 端点创建端点策略

端点策略是一种 IAM 资源，您可以将其附加到接口端点。默认端点策略允许通过接口端点完全访问 EventBridge 调度器。要控制允许从 VPC 访问 EventBridge 调度器的权限，请将自定义端点策略附加到接口端点。

端点策略指定以下信息：

- 可执行操作的主体 (AWS 账户、IAM 用户和 IAM 角色)。
- 可执行的操作。
- 可对其执行操作的资源。

有关更多信息，请参阅《AWS PrivateLink 指南》中的[使用端点策略控制对服务的访问权限](#)。

示例：EventBridge 调度器操作的 VPC 端点策略

以下是自定义端点策略的示例。将此策略附加到接口端点时，它会向所有资源上的所有主体授予对列出的 EventBridge 调度器操作的访问权限。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "scheduler:GetSchedule",
        "scheduler:ListSchedules",
        "scheduler:GetScheduleGroup",
        "scheduler:ListScheduleGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon EventBridge 计划程序中的安全

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。Third-party 作为[AWS 合规计划](#)的一部分，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](#)的范围服务。
- 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](#)
- [在调度程序中使用基于身份的策略 EventBridge](#)
- [EventBridge 调度器中的副手预防混乱](#)
- [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) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（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：确定用户访问权限并提交权限请求（请参阅[EventBridge 调度器如何与 IAM 配合使用](#)）
- IAM 管理员：编写用于管理访问权限的策略（请参阅[在调度程序中使用基于身份的策略 EventBridge](#)）

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 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center））、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的[如何登录您的 AWS 账户](#)。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的[适用于 API 请求的 AWS 签名版本 4](#)。

AWS 账户 根用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，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，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。AWS IAM Identity Center有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center ?](#)。

IAM 用户和群组

[IAM 用户](#)是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的[要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问](#)。

[IAM 组](#)指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的[IAM 用户使用案例](#)。

IAM 角色

[IAM 角色](#)是具有特定权限的身份，可提供临时凭证。您可以通过[从用户切换到 IAM 角色 \(控制台\)](#) 或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的[担任角色的方法](#)。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

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。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概述](#)。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

Identity-based 政策

Identity-based 策略是您附加到身份 (用户、组或角色) 的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

Identity-based 策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。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](#)。

Resource-based 政策

Resource-based 策略是您附加到资源的 JSON 策略文档。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。您必须在基于资源的策略中[指定主体](#)。

Resource-based 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。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，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) – 指定 AWS Organizations 中组织或组织单元的最大权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。
- 资源控制策略 (RCP) – 设置对账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[资源控制策略 \(RCP\)](#)。
- 会话策略 – 在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。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 功能	EventBridge 调度器支持
Identity-based 政策	是
Resource-based 政策	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
Service-linked 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](#)。

Identity-based EventBridge 日程安排器的策略

支持基于身份的策略：是

Identity-based 策略是您可以附加到身份（例如 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[使用客户管理型策略定义自定义 IAM 权限](#)。

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。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](#)。

Identity-based EventBridge 调度程序的策略示例

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

Resource-based EventBridge 日程安排器中的策略

支持基于资源的策略：否

Resource-based 策略是您附加到资源的 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。您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。

要启用跨账户访问，您可以将整个账户或其它账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 中的跨账户资源访问](#)。

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。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。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于不支持资源级权限的操作，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 [在调度程序中使用基于身份的策略](#) [EventBridge](#)

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。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用 [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。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 计划程序条件键列表，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 [在调度程序中使用基于身份的策略](#)
[EventBridge](#)

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，但它们不使用 JSON 策略文档格式。

带调度器的 ABA EventBridge C

支持 ABAC (策略中的标签)：部分支持

Attribute-based 访问控制 (ABAC) 是一种授权策略，它根据称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `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 [使用 ABAC 授权定义权限](#)。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

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。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 用户指南》中的 [IAM 中的临时安全凭证](#) 和 [使用 IAM 的 AWS 服务](#)

Cross-service EventBridge 日程安排器的委托人权限

支持转发访问会话 (FAS)：是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。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担任、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 scheduler 的功能。只有在 S EventBridge scheduler 提供相关指导时才编辑服务角色。

Service-linked EventBridge 日程安排器的角色

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。该服务可以代替您执行操作。Service-linked 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关联角色的权限。

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 Service-linked 角色列 Yes 中包含的服务。选择是链接以查看该服务的服务相关角色文档。

在调度程序中使用基于身份的策略 EventBridge

默认情况下，用户和角色无权创建或修改 EventBridge 日程安排资源。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。

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 \(控制台\)](#)。

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

Identity-based 策略决定是否有人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，请遵循以下指南和建议：

- 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 策略设置权限时，请仅授予执行任务所需的权限。为此，您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 策略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，您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 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 策略，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，以帮助您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 [使用 IAM Access Analyzer 验证策略](#)。
- 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 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [使用 MFA 保护 API 访问](#)。

有关 IAM 中的最佳实操的更多信息，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 日程安排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：

- AmazonEventBridgeSchedulerFullAccess

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。

要查看此策略的权限，请参阅《AWS 托管策略参考》中的 [AmazonEventBridgeSchedulerFullAccess](#)。

授予只读权限，以查看有关您的调度和调度组的详细信息。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 [AmazonEventBridgeSchedulerReadOnlyAccess](#)。

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

- [示例：CreateSchedule](#)
- [示例：GetSchedule](#)
- [示例：UpdateSchedule](#)
- [示例：DeleteScheduleGroup](#)

示例：**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。

JSON

```
{
  "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。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "scheduler:CreateSchedule"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:scheduler:us-east-1:123456789012: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"
      ],
      "Condition": {
        "StringLike": {
          "kms:ViaService": "scheduler.us-east-1.amazonaws.com",
          "kms:EncryptionContext:aws:scheduler:schedule:arn":
            "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"
        }
      }
    }
  ]
}

```

示例：GetSchedule

使用以下策略允许主体获取有关计划的信息。

JSON

```

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

```

```

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

```

示例：UpdateSchedule

使用以下策略允许主体通过调用 scheduler:UpdateSchedule 操作来更新计划。与之类似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：

JSON

```

{
  "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：

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "scheduler:UpdateSchedule"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:scheduler:us-east-1:123456789012: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"
      ],
    "Condition": {
      "StringLike": {
        "kms:ViaService": "scheduler.us-east-1.amazonaws.com",
        "kms:EncryptionContext:aws:scheduler:schedule:arn":
          "arn:aws:scheduler:us-east-1:123456789012:schedule/my-group/my-schedule-name"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/*",
    "Condition": {
      "StringLike": {
        "iam:PassedToService": "scheduler.amazonaws.com"
      }
    }
  }
]
}

```

示例：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"
```

JSON

```

{
  "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
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

更改	描述	日期
AmazonEventBridgeSchedulerReadOnlyAccess : 新托管策略	EventBridge Scheduler 增加了对新托管策略的支持，该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。	2022 年 11 月 10 日
AmazonEventBridgeSchedulerReadOnlyAccess —更新了托管策略	EventBridge 计划程序更新了 AmazonEventBridgeSchedulerReadOnlyAccess 托管策略，使用通配符操作（scheduler:List* 和 scheduler:Get*），以确保在添加新的只读操作后，策略保持最新状态。	2026年3月25日
EventBridge 日程安排器已开始跟踪更改	EventBridge 计划程序开始跟踪其 AWS 托管策略的更改。	2022 年 11 月 10 日

EventBridge 调度器中的副手预防混乱

混淆代理问题是一个安全性问题，即不具有某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。Cross-service 当一个服务（调用服务）调用另一个服务（被调用的服务）时，可能会发生模仿行为。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：

JSON

```
{
  "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"
        }
      }
    }
  ]
}
```

如果在的值中使用通配符 `aws:SourceArn`，则必须使用 `ArnLike` 而不是 `StringEquals` 作为条件运算符。例如，以下信任策略 `ArnLike` 用于匹配账户中的任何计划组：

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

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进入 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，指明您无权执行某个操作，则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进入 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。您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
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的 [为经过外部身份验证的用户 \(身份联合验证 \) 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访问之间的差别，请参阅《IAM 用户指南》中的 [IAM 中的跨账户资源访问](#)。

Amazon EventBridge 计划程序中的数据保护

责任 AWS [共担模式分](#)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结构上的内容的控制。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AWS](#)条款。有关欧洲数据保护的信息，请参阅[通用数据保护条例 \(GDPR\) 中心](#)。

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。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)。
- 用于 SSL/TLS 与 AWS 资源通信。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。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的[使用跟 CloudTrail 踪](#)。
- 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 140-3 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 140-3 版》<https://aws.amazon.com/compliance/fips/>。

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 URL，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。

EventBridge 调度器中的静态加密

本节介绍 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 调度器和服务的底层组件中的数据。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 服务拥有并管理的 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	说明	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

数据类型	说明	AWS 拥有的密钥	客户托管密钥
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 scheduler 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 scheduler 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 相关权限附加到调用 EventBridge 调度程序 API 的用户或角色。此外，EventBridge 计划程序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。

Identity-based 客户托管密钥的权限

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。

- **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**— 生成 EventBridge 调度器用于执行客户端加密的数据密钥时必需的。
- **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。

除以下操作外，还包括这些内容：

- **scheduler:***
- **iam:PassRole**：需要传递执行角色。

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。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEventBridgeSchedulerToDecryptDataUsingCMKMS",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-east-1:123456789012: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。

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 调度程序资源可以使用您的密钥。

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": "ABCDEFGHIIJKLMNOP2Q3",
    "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"
  }
}
```

EventBridge 调度程序中传输中的加密

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 时对您的数据进行加密。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 调度器还会加密您从我们的 API 收到的整个响应对象。

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。有关您在使用时的合规责任的更多信息AWS服务，请参阅[AWS 安全文档](#)。

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区域提供多个物理隔离和隔离的可用区，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设施相比，可用区具有更高的可用性、容错能力和可扩展性。

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 (短暂的) 或 ECDHE (椭圆曲线短暂的 Diffie-Hellman)。Diffie-Hellman大多数现代系统 (如 Java 7 及更高版本) 都支持这些模式。

Amazon EventBridge 调度器监控及相关指标

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 可实时监控您的 AWS 资源以及您在 AWS 上运行的应用程序。您可以收集和跟踪指标，创建自定义的控制平面，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、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

- [使用亚马逊监控亚马逊 EventBridge 调度器 CloudWatch](#)
- [使用 AWS CloudTrail 记录 Amazon EventBridge 调度器 API 调用](#)

使用亚马逊监控亚马逊 EventBridge 调度器 CloudWatch

您可以使用监控 Amazon S EventBridge scheduler CloudWatch，它会收集原始数据并将其处理为可读的近乎实时的指标。EventBridge Scheduler 会为所有计划发出一组指标，为具有关联死信队列 (DLQ) 的计划发出一组额外的指标。如果您为计划[配置 DLQ](#)，则当您的 EventBridge 计划用尽其重试策略时，计划程序会发布其他指标。

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。还可以设置特定阈值监视警报，在达到对应阈值时发送通知或采取行动。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。

Best-effort CloudWatch 指标交付

CloudWatch 指标是在尽力而为的基础上提供的。大多数 EventBridge 调度器操作都会将数据点发送到。CloudWatch 无法保证指标的完整性和及时性。给定分钟内的数据点在通过之前可能会延迟 CloudWatch，或者可能根本无法传送。CloudWatch 指标可以让你近乎实时地了解活动的性质。它们并不意味着对所有操作的完整核算。

主题

- [术语](#)
- [Dimensions](#)
- [访问指标](#)
- [指标列表](#)
- [EventBridge 调度程序使用率指标](#)

术语

命名空间

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

CloudWatch 指标

CloudWatch 指标表示特定于一组按时间顺序排列的数据点。CloudWatch

维度

维度是作为指标标识一部分的 name/value 配对。

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。

Dimensions

本节介绍中计划程序指标的 CloudWatch 维 EventBridge 度分组。CloudWatch

维度	说明
ScheduleGroup	您要使用其查看指标的计划组 CloudWatch。如果您尚未创建任何群组，则 EventBridge 日程安排器会将您的日程安排与该default群组相关联。

访问指标

本节介绍如何访问特定 EventBridge 调 CloudWatch 度程序计划中的性能指标。

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 区域选择器为您的日程安排选择区域
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. 为维 CloudWatch 度选择一个指标。例如，InvocationAttemptCount 或 InvocationDroppedCount，然后选择图表搜索。
6. 选择“图表化指标”选项卡可查看 EventBridge 计划程序指标的性能统计信息。

指标列表

下表列出了所有 EventBridge 调度程序计划的指标，以及您为其配置了 DLQ 的计划的指标。

所有计划的指标

命名空间	指标	单位	说明
AWS/Scheduler	InvocationAttemptCount	计数	每次尝试调用时都会发出。使用此指标来检查 S EventBridge cheduler 是否正在尝试调用您的计划，并查看调用何时接近您的账户配额。
AWS/Scheduler	TargetErrorCount	计数	在 EventBridge 调度器调用目标 API 后目标返回异常时发出。使用此指标可检查

命名空间	指标	单位	说明
			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。当根本原因是调度器发出的目标 API 限制调用时，使用它来诊断交付失败 EventBridge
AWS/Scheduler	InvocationThrottleCount	计数	当 EventBridge 调度器因目标调用超出调度器设置的服务配额而限制目标调用时发出。EventBridge 使用此指标可确定何时超过调用限额。有关服务限额的更多信息，请参阅 配额 。
AWS/Scheduler	InvocationDroppedCount	计数	当 EventBridge 调度的重试策略用尽后，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 日程安排器 API 参考》 RetryPolicy 中的。

带有 DLQ 的计划的指标

命名空间	指标	单位	说明
AWS/Scheduler	InvocationsSentToDeadLetterCount	计数	每次成功传输到计划的 DLQ 时都会发出。使用此指标可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。
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount	计数	当 EventBridge 调度器无法向 DLQ 传送事件时发出。
AWS/Scheduler	InvocationsFailedToBeSentToDeadLetterCount_<error_code>	计数	使用这两个指标来确定 EventBridge 调度程序无法向 DLQ 发送事件的原因，并修改您的 DLQ 配置以解决问题。 以下是您指定为 DLQ 的 Amazon SQS 队列不存在时的

命名空间	指标	单位	说明
			<p>InvocationsFailedToBeSentToDeadLetterCount_<error_code> 指标示例：InvocationsFailedToBeSentToDeadLetterCount_ AWS.SimpleQueueService.NonExistentQueue</p>
AWS/Scheduler	InvocationsSentToDeadLetterCount_Truncated_MessageSize Exceeded	计数	<p>当发送到 DLQ 的事件的有效负载超过 Amazon SQS 允许的最大大小时发出，EventBridge 并且计划程序会截断您在计划属性中指定的有效负载。Input</p>

EventBridge 调度程序使用率指标

CloudWatch 收集跟踪某些 AWS 资源使用情况的指标。这些指标对应于 AWS 服务配额。跟踪这些指标可帮助您主动管理限额。有关服务限额的更多信息，请参阅 [配额](#)。

这些指标包含在AWS/Usage命名空间中，而不是AWS/Scheduler，并且每分钟收集一次。CloudWatch 在此命名空间中发布带有维度Service、ClassType、和的指标Resource。

使用量指标的常用维度

维度	说明
Service	包含资源的 AWS 服务的名称。对于 EventBridge 调度器 使用量指标，该值为Scheduler。
Class	所跟踪的资源的类。对于 EventBridge 调度器 使用量指标，该值为None。
Type	所跟踪的资源的类型。对于 API 使用情况指标，该值为API。对于资源计数指标，该值为Resource。
Resource	正在跟踪的特定资源。对于 API 使用情况指标，这是 API 操作名称。对于资源计数指标，这是要计算的资源类型。

API 使用情况指标

API 使用量指标跟踪您的账户中执行的 API 操作数量。使用这些指标来监控 API 调用量并管理 API 速率配额。

指标名称是CallCount。此指标最有用的统计数据是SUM，它表示 1 分钟内的总操作数。

API 使用情况指标

指标	资源	说明
CallCount	CreateSchedule	在您的账户中执行CreateSchedule 的 API 操作数量。
CallCount	CreateScheduleGroup	在您的账户中执行CreateScheduleGroup 的 API 操作数量。

指标	资源	说明
CallCount	DeleteSchedule	在您的账户中执行DeleteSchedule 的 API 操作数量。
CallCount	DeleteScheduleGroup	在您的账户中执行DeleteScheduleGroup 的 API 操作数量。
CallCount	GetSchedule	在您的账户中执行GetSchedule 的 API 操作数量。
CallCount	GetScheduleGroup	在您的账户中执行GetScheduleGroup 的 API 操作数量。
CallCount	ListScheduleGroups	在您的账户中执行ListScheduleGroups 的 API 操作数量。
CallCount	ListSchedules	在您的账户中执行ListSchedules 的 API 操作数量。
CallCount	ListTagsForResource	在您的账户中执行ListTagsForResource 的 API 操作数量。
CallCount	TagResource	在您的账户中执行TagResource 的 API 操作数量。
CallCount	UntagResource	在您的账户中执行UntagResource 的 API 操作数量。
CallCount	UpdateSchedule	在您的账户中执行UpdateSchedule 的 API 操作数量。

例如，具有以下维度的CallCount指标表示您的账户中调CreateSchedule用 API 操作的次数：

- "Service": "Scheduler"
- “班级”：“无”
- "Type": "API"
- “资源”：“CreateSchedule”

资源计数指标

资源计数指标可跟踪您账户中的资源的大致数量。使用这些指标来监控您何时接近服务配额限制，从而允许您在容量用完之前请求增加配额。

指标名称是ResourceCount。此指标最有用的统计数据是 Maximum。

资源计数指标

指标	资源	说明
ResourceCount	ApproximateSchedule	<p>您账户中的日程安排的大致数量。使用此指标来监控您何时接近计划配额限制，以帮助您避免在拨打电话时ServiceQuotaExceededException 出错CreateSchedule。</p> <p>当计划数少于 100 万时，此指标可能显示为零。由于默认的计划配额为 1000 万，因此我们建议使用阈值为 500 万或更高的Maximum统计数据，以便在接近计划配额时发出通知。</p>
ResourceCount	ApproximateScheduleGroup	<p>您账户中的日程组的大致数量。使用此指标来监控您何时接近日程组配额限制，以帮助您避免在拨打电话时ServiceQuotaExceededException 出错CreateScheduleGroup。</p>

例如，具有以下维度的ResourceCount指标和Maximum统计数据表示您账户中的大致计划数量：

- "Service": "Scheduler"
- “班级”：“无”
- “类型”：“资源”
- “资源”：“ApproximateSchedule”

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》<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html>。

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，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- [创建跟踪记录概述](#)
- [CloudTrail 支持的服务和集成](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [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 EventBridge 调度器 API 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用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

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

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记账条目。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，否则每个配额都是 Region-specific。您可以请求增加大多数限额，但一些限额无法增加。

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。

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

Note

EventBridge 调度器最常引用的配额是调度数（每个区域默认为 10,000,000，可调整为十亿）、CreateSchedule 请求速率（主区域默认 5,000 TPS，可调整为数万）和调用限制限制（主区域默认 1,000 TPS，可调整为数万）。这三者都是可调整的，详情请参阅《[Service Quotas 用户指南](#)》中的“[申请增加配额](#)”。

您的 AWS 账户具有以下与 EventBridge 日程安排相关的配额。

Name	默认值	可调整	说明
CreateSchedule 请求速率	us-east-1 : 5,000 个	是	每秒的最大 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。这可以调整为每秒处理成千上万个请求。
	us-east-2 : 5,000 个		
	us-west-2 : 5,000 个		
	ap-northeast-1 : 5,000 个		
	ap-south-1 : 5,000 个		

Name	默认值	可调整	说明
	ap-southeast-1 : 5,000 ap-southeast-2 : 5,000 eu-central-1 : 5,000 eu-west-1 : 5,000 个 eu-west-2 : 5,000 sa-east-1 : 5,000 每个其他支持的区域 : 250 个		
CreateScheduleGroup 请求速率	每个受支持的区域 : 10 个	<u>是</u>	每秒的最大 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。

Name	默认值	可调整	说明
DeleteSchedule 请求速率	us-east-1 : 1000 个 us-east-2 : 1000 个 us-west-2 : 1000 个 ap-northeast-1 : 1000 个 ap-south-1 : 1000 个 ap-southeast-1 : 1000 个 ap-southeast-2 : 1000 个 eu-central-1 : 1000 个 eu-west-1 : 1000 个 eu-west-2 : 1000 个 sa-east-1 : 1000 个 每个其他支持的区域 : 250 个	<u>是</u>	每秒的最大 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。这可以调整为每秒处理成千上万个请求。

Name	默认值	可调整	说明
DeleteScheduleGroup 请求速率	每个受支持的区域：10 个	<u>是</u>	每秒的最大 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。

Name	默认值	可调整	说明
GetSchedule 请求速率	us-east-1 : 1000 个 us-east-2 : 1000 个 us-west-2 : 1000 个 ap-northeast-1 : 1000 个 ap-south-1 : 1000 个 ap-southeast-1 : 1000 个 ap-southeast-2 : 1000 个 eu-central-1 : 1000 个 eu-west-1 : 1000 个 eu-west-2 : 1000 个 sa-east-1 : 1000 个 每个其他支持的区域 : 250 个	<u>是</u>	每秒的最大 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。这可以调整为每秒处理成千上万个请求。

Name	默认值	可调整	说明
GetScheduleGroup 请求速率	每个受支持的区域：10 个	<u>是</u>	每秒的最大 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。

Name	默认值	可调整	说明
调用每秒事务数节流限制	us-east-1 : 1000 个 us-east-2 : 1000 个 us-west-2 : 1000 个 ap-northeast-1 : 1000 个 ap-south-1 : 1000 个 ap-southeast-1 : 1000 个 ap-southeast-2 : 1000 个 eu-central-1 : 1000 个 eu-west-1 : 1000 个 eu-west-2 : 1000 个 sa-east-1 : 1000 个 每个其他支持的区域 : 500 个	<u>是</u>	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。这可以调整至每秒数万个事务。

Name	默认值	可调整	说明
ListScheduleGroups 请求速率	每个受支持的区域：10 个	是	每秒的最大 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	每秒的最大 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	每个区域计划组的最大数
计划数	每个受支持的区域：1000 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。我们建议使用该 ActionAfterCompletion 功能将您的计划配置为在完成后自动删除。这可以调整至数十亿个计划。
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。

Name	默认值	可调整	说明
UpdateSchedule 请求速率	us-east-1 : 1000 个 us-east-2 : 1000 个 us-west-2 : 1000 个 ap-northeast-1 : 1000 个 ap-south-1 : 1000 个 ap-southeast-1 : 1000 个 ap-southeast-2 : 1000 个 eu-central-1 : 1000 个 eu-west-1 : 1000 个 eu-west-2 : 1000 个 sa-east-1 : 1000 个 每个其他支持的区域 : 250 个	<u>是</u>	每秒的最大 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。这可以调整为每秒处理成千上万个请求。

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](#)。

排查 EventBridge 调度器中的限额问题

使用以下信息可帮助您诊断和修复可能遇到的与 EventBridge 调度器限额有关的常见问题。

ServiceQuotaExceededException

尽管我的速率低于默认速率限制，但我还是收到了有关 CreateSchedule、DeleteSchedule、GetSchedule 或 UpdateSchedule 请求速率的限制错误。

常见原因

2023 年 9 月 7 日，EventBridge 调度器开始在执行角色信任策略中支持 ScheduleGroup ARN (Amazon 资源名称)，而不是 Schedule ARN。对允许在其信任策略中继续使用 Schedule ARN 的客户的限制可能为 50 TPS，而不是默认的 250 到 1000 TPS (视地区而定)。

解决方案

请联系[支持部门](#)申请提高最大限制。

预防措施

通过以下方式之一修改您现有的信任策略：

- 从角色中移除所有作用范围。
- 确定角色的范围，以便可以使用 Schedule ARN 或 ScheduleGroup ARN 来代入该角色。

例如，假设您拥有以下现有信任策略：

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "scheduler.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
```

```
    "aws:SourceArn":
      "arn:aws:scheduler:region:account:schedule/schedule_group/schedule"
    }
  }
}
```

您可以按照如下所示更新该信任策略：

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "scheduler.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:SourceArn": [
        "arn:aws:scheduler:region:account:schedule/schedule_group/schedule",
        "arn:aws:scheduler:region:account:schedule-group/schedule_group"
      ]
    }
  }
}
```

对 Amazon EventBridge 计划程序进行故障排除

您可以使用本节中的主题来解决常见的 Amazon EventBridge 计划程序问题。

主题

- [我的调度因目标错误而失败](#)
- [调度执行角色权限问题](#)
- [了解和管理服务配额](#)
- [调度模式和触发时机问题](#)
- [创建调度模式和 cron 表达式](#)
- [我的目标是否被触发？](#)
- [模板化目标与通用目标](#)
- [通用目标输入配置无效](#)
- [调度更新触发意外调用](#)
- [禁用或启用一次性调度](#)

我的调度因目标错误而失败

目标调用失败是 EventBridge 调度器最常见的问题之一。这些失败可能是由于以下几个原因引起的：

常见原因：

- 目标参数缺失或不正确。
- 网络连接问题。
- API 节流。
- 目标配置不正确。

故障排除步骤

1. 设置 Dead-Letter 队列 (DLQ)
 - DLQ 可帮助您捕获和分析失败的调用。
 - 失败的调用将发送到 DLQ，并附有详细的错误消息。

- 要[配置 DLQ](#)，请将其添加到您的调度配置中：

```
{
  "DeadLetterConfig": {
    "Arn": "arn:aws:sqs:region:account-id:MyDLQ"
  }
}
```

注意：如果您的 DLQ 使用 KMS 密钥加密，请确保密钥策略允许 EventBridge 调度程序使用它：

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

2. 验证 API 参数

- 确保目标 API 调用的所有必需参数均存在且格式正确。
- 检查参数值是否在允许的范围内。
- 如果使用 VPC 端点，请验证 API 端点是否可从您的 VPC 访问。

3. 查看网络配置

- 如果由于临时网络问题导致调用失败，请实现[重试](#)逻辑。
- 重试策略示例：

```
{
  "RetryPolicy": {
    "MaximumRetryAttempts": 3,
    "MaximumEventAgeInSeconds": 3600
  }
}
```

```
}  
}
```

4. 检查特定于目标的配置

- 对于模板化目标（如 ECS 任务），请确保通过调度创建 API 的 `Target.Input` 参数提供覆盖值。
- 确认您的目标服务 [受支持](#) 且配置正确。

调度执行角色权限问题

IAM 角色权限问题是调度执行失败的常见原因。以下是排查和解决这些问题的方法：

常见原因

- 缺少目标服务所需的权限
- 调度中的角色配置不正确
- 与 EventBridge 日程安排器服务缺少信任关系
- 访问加密资源的权限不足

症状

- 增加了 `TargetErrorCount` 指标 CloudWatch
- 调度无法执行，调度配置中没有明显问题

故障排除步骤

1. 监控 CloudWatch 指标

- 在中查看 `TargetErrorCount` 指标 CloudWatch。

2. 使用 Dead-Letter 队列 (DLQ) 确认权限问题

- 为您的调度配置 DLQ。
- 如果您的目标存在权限问题，并且 DLQ 配置正确，您将在 DLQ 中看到包含权限相关错误消息的失败调用记录。
- 如果尽管 CloudWatch 指标中显示执行失败，但 DLQ 仍为空，则这可能表示存在权限问题，导致 EventBridge 调度程序无法写入 DLQ 本身。

Note

请确保 DLQ 本身具有正确的权限。如果已加密，请确保 EventBridge 调度程序有权使用 KMS 密钥。

3. 验证信任关系

- 确保您的 IAM 角色与 EventBridge 调度程序有正确的信任关系：

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

4. 检查调度执行角色权限

- 调度的执行角色需要特定的权限才能调用不同的目标类型。
- 调度执行角色策略中应包含的权限示例：

```
// For Lambda function targets - add to schedule execution role
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:region:account-id:function:function-name"
  }]
}

// For SQS queue targets - add to schedule execution role
{
```

```
"Version": "2012-10-17",
"Statement": [{
  "Effect": "Allow",
  "Action": [
    "sqs:SendMessage"
  ],
  "Resource": "arn:aws:sqs:region:account-id:queue-name"
}]
}
```

5. 检查是否有加密的资源访问权限

- 如果您的目标使用加密资源（例如 KMS-encrypted SQS 队列），请确保您的角色有权使用 KMS 密钥：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:region:account-id:key/key-id"
    }
  ]
}
```

6. 验证角色 ARN 配置

- 确保您的调度配置中的角色 ARN 正确。
- 验证该角色是否 AWS 账户 与您的日程安排位于相同的地区。

了解和管理服务配额

如果您在创建计划或看到受限制的调用时遇到问题，则可能已达到服务配额限制。EventBridge 调度器有计划数量、计划组和调用速率的配额，这些配额可能因地区而异。

确定配额问题

要确定您是否达到配额限制，请执行以下操作：

1. 监控 CloudWatch 指标

- 检查 `InvocationThrottleCount` 指标。该指标的增长表明您的调用频率已超出限制。
- 查看 `InvocationAttemptCount` 指标以了解您当前的使用情况。

2. 留意具体的错误消息

- 创建或修改调度时，`LimitExceededException` 表示您已达到调度或调度组的最大数量。
- 返回节流错误的 API 调用表明您已超出 API 请求配额。

解决配额问题

如果您确定自己已达到配额限制：

1. 查看并优化您当前的调度。考虑整合相似的调度或删除未使用的调度。
2. 对于 API 节流，请在 API 调用中[使用回退实现重试](#)。
3. 如果需要更高的配额，可以通过服务配额控制台请求增加配额。选择 `EventBridge Scheduler`，选择需要增加的配额，然后提交包含您的业务理由的申请。

调度模式和触发时机问题

用户有时会遇到调度无法在预期时间触发的问题。这通常是由于对调度模式的理解偏差、夏令时变更或灵活时间窗口导致的。

常见原因

- 对 cron 表达式的误解。
- 夏令时切换期间出现意外行为。
- 对灵活时间窗口存在理解混淆。
- 对 rate 表达式的误解。

故障排除步骤

1. 验证 cron 表达式

- 确保您的 cron 表达式格式正确。
- 请注意，在 cron 表达式中，不能同时指定日期和星期几字段。

2. 时区注意事项

- 选择创建调度时的首选时区。
- 了解夏令时如何影响您的调度，因为此调整基于 UTC。

夏令时影响示例：如果您将调度配置为在格林威治标准时间 (GMT) 上午 7:00 运行：

- 冬季：调度在格林威治标准时间 (GMT) 时间上午 7:00 运行（因为 GMT 时间 = UTC 时间）
- 夏季：时刻表仍然在世界标准时间上午 7:00 运行，现在是上午 6:00 GMT/BST

如果您需要调度全年按相同本地时间运行，请在创建调度时选择合适的时区，并留意夏令时对该时区的影响。

3. 了解灵活时间窗口

- [灵活的时间窗口](#) 允许 EventBridge 调度器优化调用。
- 调度可能不会在窗口的起始时刻准时触发。
- 监控实际调用时间以了解相关行为。

4. 查看 rate 和 cron 表达式

- 确保 rate 表达式的格式正确（例如 `rate(5 minutes)`、`rate(1 hour)`）。
- 对于 rate 和 cron 表达式，请注意，调度调用不会强制对齐到每分钟的第 0 秒。
- 调度可以在指定的分钟内触发，但不一定在该分钟的精确起始时刻触发。

例如：

- `rate(1 hour)` 的调度可能在下午 2:00:45、下午 3:00:32、下午 4:00:18 等时间运行。
- 设置为 `0 * * * ? *`（每小时）的 cron 调度可能在下午 2:00:15、下午 3:00:07、下午 4:00:52 等时间运行。

5. 监控 CloudWatch 指标

- 使用 `InvocationAttemptCount` 指标来验证您的调度是否正在触发。
- 如果调用失败，则检查 `TargetErrorCount`。
- 如果您配置了 Dead-Letter 队列，请监视 `InvocationsSentToDeadLetterCount` 以跟踪失败的调用。

创建调度模式和 cron 表达式

用户在创建调度模式时经常遇到问题，尤其是在使用 cron 表达式时。以下是一些常见问题及其解决方法：

常见问题

- cron 语法不正确
- 正在尝试使用不支持的 cron 功能
- 对哪些字段可同时使用感到困惑

故障排除步骤

1. 查看 cron 表达式语法

- 确保您的 cron 表达式遵循正确的[格式](#)：Minutes Hours Day-of-month Month Day-of-week Year。
- 请记住，EventBridge 调度器使用带有额外年份字段的 cron 标准。

2. 了解局限性

- 您不能同时指定日期和星期几字段，详见[此处](#)。
- 不支持产生的速率快于 1 分钟的 Cron 表达式。

3. 使用调度预览功能

- 创建或编辑计划时，计划 EventBridge 程序会预览接下来的 10 个执行时间。
- 使用此预览来验证您的调度是否会在预期时间运行。
- 如果预览不符合您的预期，请查看并调整您的 cron 表达式。

我的目标是否被触发？

要确认您的目标是否被触发，请执行以下操作：

1. 检查 CloudWatch 指标：

- InvocationAttemptCount 显示尝试调用的次数
- TargetErrorCount 表示是否有任何调用失败
- TargetErrorThrottledCount 显示您的目标是否受到限制

- `InvocationDroppedCount` 表示是否丢弃了任何调用

2. [配置 Dead-Letter 队列 \(DLQ\)](#) 以捕获和分析任何失败的调用。

模板化目标与通用目标

如果您收到诸如“Invalid request provided: [service] is not a supported service for a target”之类的错误，则您可能正在尝试使用不支持的服务作为模板化目标。

要解决这一问题，请执行以下操作：

1. 检查系统是否支持将您所需的服务用作[模板化目标](#)。
2. 如果不支持，请改用[通用目标](#)并将其配置为通过适当的 API 调用您的服务。

通用目标输入配置无效

当您使用[通用目标](#)创建计划时，EventBridge 调度器会验证目标 ARN 格式，但不会根据下游服务的 API 验证该 `Input` 字段的内容。这意味着，即使调度 `Input` 包含目标服务将在调用时拒绝的值，也可以成功创建计划。

目标输入配置无效的计划将在其配置的表达式上触发，但在每次调用时都会失败。在调用计划之前，您可能不会发现配置错误，这可能是创建后的几小时或几天。

症状

- 计划创建时没有出现错误，但是 `TargetErrorCount` CloudWatch 指标在每次调用时都会增加。
- DLQ 消息包含来自目标服务的错误代码（例如，`InvalidParameterValueException` 或 `ValidationException`），不是 `AWS.Scheduler.InternalServerError`。
- DLQ 消息 `ERROR_MESSAGE` 中的引用了特定的输入参数验证失败。

示例

以下示例显示了 AWS Lambda 通用目标 (`arn:aws:scheduler::aws-sdk:lambda:invoke`) 的常见无效输入配置。

预选赛不匹配

具有以下输入的计划在Qualifier字段2中指定版本FunctionName和版本1：

```
{
  "FunctionName": "MyFunction:2",
  "Qualifier": "1"
}
```

此计划已成功创建，但每次调用都失败。DLQ 消息包含：

- ERROR_CODE: InvalidParameterValueException
- ERROR_MESSAGE: The derived qualifier from the function name does not match the specified qualifier.

函数名称无效

包含以下输入的计划为以下项指定一个仅限空格的值：FunctionName

```
{
  "FunctionName": "    "
}
```

DLQ 消息包含：

- ERROR_CODE: ValidationException
- ERROR_MESSAGE: 验证错误，表明函数名称与所需的模式不匹配。

解决方法

1. 配置 DLQ。请务必为使用通用目标的计划配置死信队列。DLQ 消息属性 (ERROR_CODE和ERROR_MESSAGE) 包含目标服务返回的特定错误，该错误标识了无效的输入参数。
2. 根据目标服务 API 验证输入参数。在创建计划之前，请直接调用目标 API 来验证Input字段中的 JSON 是否包含有效值。例如，使用 AWS Lambda Invoke API 调用具有相同参数的 AWS Lambda 函数以确认请求成功。
3. 使用一次性计划进行测试。在配置定期计划之前，创建一次性计划以验证目标调用是否成功。
4. 查看目标服务 API 参考。查看您要定位的服务的 API 参考以确认所需的参数、有效值范围和约束条件。有关信息 AWS Lambda Invoke，请参阅《AWS Lambda 开发人员指南》中的“[调用](#)”。

调度更新触发意外调用

当您对调度进行更改时，调用可能不会立即反映更新的调度。请稍等片刻，以便更改生效。例如，如果您更新接近其原始触发时间的调度，则可能会看到基于原始调度配置的调用。

禁用或启用一次性调度

当一次性调度的原始预定时间已过时，若重新启用该调度，其可能会立即调用目标对象。即使调度在最初的执行时间之前被禁用，也可能发生这种情况。

例如：

- 当前时间：13:15 UTC
- One-time 时间表创建于：世界标准时间 13:30
- 调度已于 UTC 时间 13:30 之前被禁用
- 调度已于 UTC 时间 14:00 重新启用
- 结果：目标可以在重新启用后立即被调用

《EventBridge 日程安排用户指南》的文档历史记录

下表描述了 S EventBridge scheduler 的文档版本。

变更	说明	日期
更新了 AmazonEventBridgeSchedulerReadOnlyAccess 托管策略	已将AmazonEventBridgeSchedulerReadOnlyAccess 托管策略更新为使用通配符操作 (scheduler :List* 和scheduler :Get*)。	2026年3月25日
执行角色和混淆代理预防方面的变化	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。	2023 年 9 月 7 日
计划完成后自动删除	EventBridge 调度器支持自动删除。配置自动删除时，EventBridge 调度程序会在计划的最后一次调用后删除您的计划。	2023 年 8 月 2 日
更新了有关使用通用目标的主题	更新了 S EventBridge scheduler 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	2023 年 3 月 17 日

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。

- [the section called “使用通用目标”](#)

[更新了没有起始日期、基于速率的计划的信息](#)

如果您未指定，则添加了有关 EventBridge 计划程序如何处理基于速率的计划的信息

2023 年 3 月 17 日

- [StartDate](#)
- [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调度器创建调度器组的新章节。EventBridge 使用本章学习如何创建群组、向群组添加日程安排、应用标签以更轻松地管理和监控您的 EventBridge 日程安排器资源，以及最终删除群组。

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，描述了日 EventBridge 程安排器如何处理夏令时，以及如何在不同的时区创建计划。

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](#)

添加了描述 EventBridge 计划程序发布到 CloudWatch 的指标的新主题。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 “使用监控 CloudWatch”](#)

[初始版本](#)

《EventBridge 日程安排用户指南》的初始版本。

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。