aws

用户指南

AWS 故障注入服务

AWS 故障注入服务: 用户指南

Table of Contents

什么 AWS 是 FIS？ 1
概念 1
操作 2
目标 2
停止条件 2
支持的 AWS 服务 2
访问 AWS FIS 3
定价 4
规划实验 5
基本原则和指南 5
实验规划指南 6
教程 8
测试实例停止和启动 8
先决条件 8
步骤 1：创建实验模板 8
步骤 2：开始实验 11
步骤 3：跟踪实验进度 11
步骤 4：验证实验结果 12
步骤 5：清除 12
在实例上运行 CPU 压力测试 13
先决条件 13
步骤 1：为停止条件创建 CloudWatch 警报 14
步骤 2：创建实验模板 14
步骤 3：开始实验 16
步骤 4：跟踪实验进度 17
步骤 5：验证实验结果 17
步骤 6：清理 12
测试竞价型实例中断情况 19
先决条件 19
步骤 1：创建实验模板 21
步骤 2：开始实验 23
步骤 3：跟踪实验进度 23
步骤 4：验证实验结果 24
第 5 步：清理 24

模拟连接事件 25
先决条件 26
步骤 1：创建 AWS FIS 实验模板 26
步骤 2：对 Amazon S3 端点执行 Ping 操作 27
第 3 步：开始你的 AWS FIS 实验 28
第 4 步：追踪您 AWS 的 FIS 实验进度 29
步骤 5：验证 Amazon S3 网络中断 29
步骤 5：清除 29
安排定期实验 30
先决条件 30
步骤 1：创建 IAM 角色和策略 30
步骤 2：创建 Amazon EventBridge 调度器 32
步骤 3：验证实验 33
步骤 4：清除 33
操作 35
操作标识符 35
操作参数 35
操作目标 36
操作参考 37
故障注入操作 37
等待动作 39
亚马逊的 CloudWatch 行动 40
Amazon DynamoDB 操作 40
Amazon EBS 操作 42
Amazon EC2 操作 43
Amazon ECS 操作 48
Amazon EKS 操作 54
亚马逊的 ElastiCache 行动 63
网络操作 63
Amazon RDS 操作 67
Amazon S3 操作 68
Systems Manager 操作 69
使用 SSM 文档 71
执行 aws:ssm:send-command 操作。 72
预先配置的 AWS FIS SSM 文档 73
示例 80

故障排除 80
执行 ECS 任务操作 81
操作 81
限制 81
要求 81
脚本参考版本 84
实验模板示例 87
执行 EKS 容器组（pod）操作 88
操作 88
限制 88
要求 89
为 Kubernetes 服务账户创建服务角色。 89
配置 Kubernetes 服务账户 89
将实验角色映射到 Kubernetes 用户 91
容器组（pod）容器映像 91
实验模板示例 93
列出动作 94
实验模板 96
模板组件 96
模板语法 96
开始使用 97
操作集 97
操作语法 97
操作持续时间 98
操作示例 99
目标 101
目标语法 101
资源类型 103
标识目标资源 103
选择模式 106
示例目标 107
示例筛选条件： 108
停止条件 112
停止条件语法 112
了解更多信息 112
实验角色 113

先决条件 113
选项 1：创建实验角色并附加 AWS 托管策略 115
选项 2：创建实验角色并添加内联策略文档 115
实验选项 117
账户定位 118
空目标解析模式 119
动作模式 119
使用实验模板 120
创建实验模板 120
查看实验模板 122
根据实验模板生成目标预览 123
通过模板开始实验 124
更新实验模板 124
标记实验模板 125
删除实验模板 125
示例模板 127
根据筛选条件停止 EC2 实例 127
停止运行指定数量的 EC2 实例 128
运行预先配置的 AWS FIS SSM 文档 129
运行预定义的自动化运行手册 130
使用目标 IAM 角色限制 EC2 实例上的 API 操作 131
对 Kubernetes 集群中的容器组（pod）CPU 进行压力测试 133
多账户实验 136
概念 136
Orchestrator 账户 136
目标账户 136
目标账户配置 137
先决条件 137
权限 137
停止条件（可选） 140
使用多账户实验 140
最佳实践 140
创建多账户实验模板 141
更新目标账户配置 142
删除目标账户配置 142
场景库 144

使用场景 144
查看场景 144
使用场景 145
导出场景 145
场景参考 146
AZ Availability: Power Interruption 148
操作 148
限制 150
要求 151
权限 151
场景内容 155
Cross-Region: Connectivity 160
操作 161
限制 162
要求 162
权限 162
场景内容 170
实验 173
开始实验 173
查看实验 174
实验状态 174
操作状态 175
标记实验 175
停止实验 176
列出已解析的目标 176
实验调度器 177
开始使用 177
安排 FIS 实验 180
使用控制台更新计划 181
更新实验计划 182
使用控制台禁用或删除实验执行 182
监控 183
使用 CloudWatch 进行监控 184
监测 AWS FIS 实验 184
AWS FIS 使用情况指标 184
监视器使用 EventBridge 185

实验日志记录 187
权限 187
日志架构 187
日志目的地 189
日志记录示例 189
启用实验日志记录 194
禁用实验日志记录 194
使用 AWS CloudTrail 记录 API 调用 195
使用 CloudTrail 195
了解 AWS FIS 日志文件条目 196
安全性 201
数据保护 201
静态加密 202
传输中加密 202
Identity and Access Management 202
受众 203
使用身份进行身份验证 203
使用策略管理访问 206
AWS 故障注入服务如何与 IAM 配合使用 208
策略示例 213
使用服务相关角色 222
AWS 托管策略 224
基础设施安全性 228
AWS PrivateLink 228
注意事项 229
创建接口 VPC 终端节点 229
创建 VPC 端点策略 229
标记资源 231
添加标签限制 231
使用标签 231
配额和限制 233
文档历史记录 241
..... ccxlv

什么是 AWS 故障注入服务？

AWS 故障注入服务 (AWS FIS) 是一项托管服务，可让您对 AWS 工作负载执行故障注入实验。基于混沌工程原理执行故障注入操作。这些实验通过创建破坏性事件来对应用程序施加压力，以便您可以观察应用程序的响应情况。这些信息可用于提高应用程序的性能和弹性，确保其按预期运行。

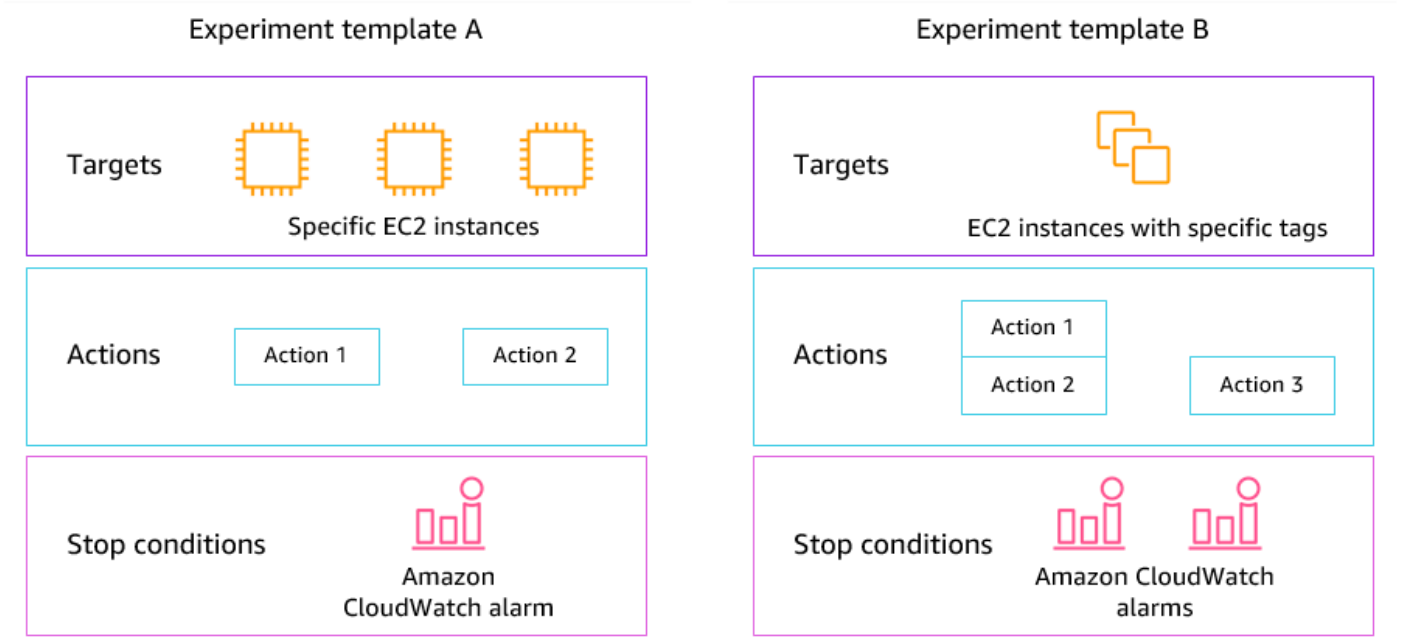
要使用 AWS FIS，您需要设置并运行实验，这些实验可以帮助您创建所需的真实条件，以发现原本很难发现的应用程序问题。 AWS FIS 提供了生成中断的模板，以及在生产中运行实验所需的控制和护栏，例如在满足特定条件时自动回滚或停止实验。

> **Important**
>
> AWS FIS 对系统中的真实 AWS 资源执行实际操作。因此，在使用 AWS FIS 在生产环境中运行实验之前，我们强烈建议您完成计划阶段并在预生产环境中运行实验。

有关实验规划的更多信息，请参阅测试可靠性和 规划 AWS FIS 实验。有关 AWS FIS 的更多信息，请参阅AWS 故障注入服务。

AWS FIS 概念

要使用 AWS FIS，您需要对自己的 AWS 资源进行实验，以测试应用程序或系统在故障条件下将如何运行的理论。要运行实验，则首先要创建实验模板。实验模板是指导实验的蓝图。其中包含实验的操作、目标和停止条件。您可以使用创建的实验模板运行实验。也可以在实验运行期间跟踪进度并查看状态。当所有实验操作都运行完毕后，即为完成实验。



操作

操作 AWS 是 FIS 在实验期间对 AWS 资源执行的活动。 AWS FIS 根据资源类型提供了一组预配置的操作。 AWS 实验期间，各项操作会运行指定时长，或者运行到您停止实验。这些操作可以按顺序运行，也可以同时运行（并行）。

目标

目标 AWS 是 FIS 在实验期间对其执行操作的一个或多个 AWS 资源。您可以选择特定资源，也可以根据特定标准（如标签或状态）选择一组资源。

停止条件

AWS FIS 提供了在工作负载上安全运行实验所需的控件和护栏。 AWS 停止条件是一种在实验达到您定义为 Amazon CloudWatch 警报的阈值时停止实验的机制。如果在实验运行时触发了停止条件，AWS FIS 将停止实验。

支持的 AWS 服务

AWS FIS 为跨 AWS 服务的特定类型的目标提供预配置的操作。 AWS FIS 支持针对以下 AWS 服务目标资源的操作：

- Amazon CloudWatch

- Amazon DynamoDB
- Amazon EBS
- Amazon EC2
- Amazon ECS
- Amazon EKS
- Amazon ElastiCache
- Amazon RDS
- Amazon S3
- AWS Systems Manager
- Amazon VPC

对于单账户实验，目标资源必须与实验 AWS 账户 相同。您可以使用 AWS FIS 多账户实验运行针对不同 AWS 账户 账户资源 AWS 的 FIS 实验。

有关更多信息，请参阅 的操作 AWS FIS。

访问 AWS FIS

您可以通过以下任何 AWS 一种方式与 FIS 合作：

- AWS Management Console— 提供可用于访问 AWS FIS 的 Web 界面。有关更多信息，请参阅使用 AWS Management Console。
- AWS Command Line Interface (AWS CLI) — 为包括 AWS FIS 在内的各种 AWS 服务提供命令，并在 Windows、macOS 和 Linux 上受支持。有关更多信息，请参阅 AWS Command Line Interface。有关 AWS FIS 命令的更多信息，请参阅《AWS CLI 命令参考》中的 fis。
- AWS CloudFormation— 创建描述您的 AWS 资源的模板。借助模板，您可以将这些资源作为一个单位进行预置和管理。有关更多信息，请参阅 AWS Fault Injection Service 资源类型参考。
- AWS 软件开发工具包 — 提供特定语言的 API 并处理许多连接细节，例如计算签名、处理请求重试和处理错误。有关更多信息，请参阅 AWS 软件开发工具包。
- HTTPS API：提供可通过 HTTPS 请求调用的低级别 API 操作。有关更多信息，请参阅 AWS Fault Injection Service API 参考。

AWS FIS 的定价

根据实验的目标账户数量，从开始到结束，操作按运行的分钟数收费。有关更多信息，请参阅 AWS FIS 的定价。

规划 AWS FIS 实验

故障注入通过创建中断事件（如服务器故障或 API 限制）在测试或生产环境中对应用程序施加压力的过程。通过观察系统的响应，你可以进行改进。当您在系统上运行实验时，会帮助您以可控方式标识系统漏洞，以免影响依赖系统运行的客户。然后，您可以主动解决问题，预防不可预测的结果。

亚马逊建议您先熟悉以下原则和指南，再使用 AWS FIS 运行故障注入实验。

> **Important**
>
> AWS FIS 对系统中的真实 AWS 资源执行实际操作。因此，亚马逊强烈建议您先在预生产或测试环境中完成规划阶段和测试，然后再开始使用 AWS FIS 运行实验。

内容

- 基本原则和指南
- 实验规划指南

基本原则和指南

请先采取以下步骤，再开始 AWS FIS 实验：

1. 标识实验的目标部署：首先标识目标部署。亚马逊建议您在首次实验时从预生产或测试环境开始。
2. 查看应用程序架构：必须确保已标识了每个组件的所有应用程序组件、依赖项和恢复过程。首先查看应用程序架构。根据应用程序的不同，请参阅 AWS Well-Architected Framework。
3. 定义稳定状态行为：根据技术和业务方面的重要指标（如延迟、CPU 负载、每分钟登录失败次数、重试次数或页面加载速度），定义系统的稳定状态行为。
4. 提出假设：假设在实验过程中系统行为会如何变化。请按照以下格式提出假设：

   如果执行######，则##########不应超过#。

   例如，对于身份验证服务可以假设：“如果网络延迟提高 10%，则登录失败次数增加不到 1%。”实验结束后，您需要评估应用程序的弹性是否符合业务和技术预期。

此外，亚马逊建议您根据以下指南使用 AWS FIS：

- 始终在测试环境中开始 AWS FIS 实验。切勿从生产环境开始。随着故障注入实验取得进展，您可以在除测试环境以外的其他可控环境中进行实验。
- 从简单的小型实验开始，例如在某个目标上执行 aws:ec2:stop-instances 操作，从而增强团队对应用程序弹性的信心。
- 故障注入会引发实际问题。请谨慎操作，同时确保在测试实例上进行首次故障注入，以免影响客户。
- 测试，测试，再测试。故障注入旨在通过精心规划的实验在受控环境中实现。这有助于您对应用程序和工具承受动荡条件的能力建立信心。
- 亚马逊强烈建议您先设置好性能出众的监控和警报程序，再开始实验。否则，您将无法理解或衡量实验造成的影响，这对于故障注入实践的持续开展至关重要。

实验规划指南

通过 AWS FIS，您可以对 AWS 资源运行实验，以测试应用程序或系统在故障条件下如何运行的理论。

以下是规划 AWS FIS 实验的推荐指南。

- 查看中断历史记录：查看系统之前发生的中断和事件。这有助于您了解系统的整体健康状况和弹性。您应该先解决系统中的已知问题和漏洞，再在系统上运行实验。
- 标识影响最大的服务：查看服务，并标识在停机或无法正常运行时对最终用户或客户影响最大的服务。
- 标识目标系统：目标系统就是您要运行实验的系统。如果您从未使用过 AWS FIS 或此前从未进行过故障注入实验，亚马逊建议您先在预生产或测试系统上运行实验。
- 咨询团队：询问团队顾虑。您可以提出假设以证明或反驳他们的担忧。也可以询问团队放心的方面。此问题可以揭示两种常见谬误：沉没成本谬误和确认偏见谬误。基于团队答案形成假设可提供有关系统状态现实的更多信息。
- 查看应用程序架构：查看系统或应用程序，并确保已标识了每个组件的所有应用程序组件、依赖项和恢复过程。

  亚马逊建议您查看 AWS Well-Architected Framework。该框架可以帮助您为应用程序和工作负载构建安全高效、高性能和高弹性的基础设施。有关更多信息，请参阅 AWS Well-Architected。
- 确定适用的指标 — 您可以使用 Amazon CloudWatch 指标监控实验对您的AWS资源的影响。当您的应用程序处于最佳性能时，您可以使用这些指标来确定基准或“稳定状态”。然后可以在实验期间或实验结束后监控指标以确定影响。有关更多信息，请参见 使用 Amazon CloudWatch 监控 AWS FIS 的使用情况指标。

- 为系统定义可接受的性能阈值：标识代表系统可接受的稳定状态的指标。您将使用此指标来创建代表实验停止条件的一个或多个 CloudWatch 警报。如果触发警报，实验将自动停止。有关更多信息，请参阅 AWS FIS 的停止条件。

AWS 故障注入服务教程

以下教程向您展示如何使用 AWS 故障注入服务 (AWS FIS) 创建和运行实验。

教程

- 教程：使用 AWS FIS 测试实例停止和启动
- 教程：使用 AWS FIS 在实例上运行 CPU 压力测试
- 教程：使用 AWS FIS 测试竞价型实例中断情况
- 教程：模拟连接事件
- 教程：安排定期实验

教程：使用 AWS FIS 测试实例停止和启动

您可以使用 AWS Fault Injection Service (AWS FIS)，测试应用程序如何处理实例停止和启动。按教程创建实验模板，其通过 AWS FIS `aws:ec2:stop-instances` 操作逐个停止实例。

先决条件

要完成本教程，请确保您已做好以下准备：

- 在账户中启动两个 EC2 测试实例。然后，记下两个实例的 ID。
- 创建 IAM 角色，以便 AWS FIS 服务代表您执行 `aws:ec2:stop-instances` 操作。有关更多信息，请参见 适用于 AWS FIS 实验的 IAM 角色。
- 确保您有权访问 AWS FIS。有关更多信息，请参阅 AWS FIS 策略示例。

步骤 1：创建实验模板

使用 AWS FIS 控制台创建实验模板。您可以在模板中指定两项操作，分别按顺序运行三分钟。第一项操作会停止 AWS FIS 随机选择的一个测试实例。第二项操作会停止所有测试实例。

创建实验模板

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在导航窗格中，选择实验模板。

3. 选择创建实验模板。
4. 对于描述和名称，输入模板的描述和名称。
5. 对于操作，请执行以下操作：
   a. 选择添加操作。
   b. 输入操作名称。例如，输入 **stopOneInstance**。
   c. 对于操作类型，选择 aws:ec2:stop-instances。
   d. 对于目标，保留 AWS FIS 为您创建的目标。
   e. 对于操作参数，在持续时间后启动实例，指定 3 分钟 (PT3M)。
   f. 选择保存。
6. 对于目标，请执行以下操作：
   a. 对于 AWS FIS 在上一步中自动创建的目标，选择编辑。
   b. 将默认名称替换为更具描述性的名称。例如，输入 **oneRandomInstance**。
   c. 验证资源类型是否为 aws:ec2:instance。
   d. 对于目标方法，选择资源 ID，然后选择两个测试实例的 ID。
   e. 对于选择模式，选择计数。对于资源数量，输入 **1**。
   f. 选择保存。
7. 选择添加目标，然后执行以下操作：
   a. 输入目标名称。例如，输入 **bothInstances**。
   b. 对于资源类型，选择 aws:ec2:instance。
   c. 对于目标方法，选择资源 ID，然后选择两个测试实例的 ID。
   d. 对于选择模式，选择全部。
   e. 选择保存。
8. 通过操作部分，选择添加操作。执行以下操作：
   a. 对于名称，输入操作名称。例如，输入 **stopBothInstances**。
   b. 对于操作类型，选择 aws:ec2:stop-instances。
   c. 对于稍后开始，选择您添加的第一项操作 (**stopOneInstance**)。
   d. 对于目标，选择您添加的第二个目标 (**bothInstances**)。
   e. 对于操作参数，在持续时间后启动实例，指定 3 分钟 (PT3M)。
   f. 选择保存。

9. 对于服务访问权限，选择使用现有 IAM 角色，然后选择您按照本教程先决条件中所述创建的 IAM 角色。如未显示此角色，请验证其是否具有必要的信任关系。有关更多信息，请参见 the section called “实验角色”。
10. （可选）对于标签，选择添加新标签，然后指定标签键和标签值。您添加的标签将应用于实验模板，而不是应用于使用此模板运行的实验。
11. 选择创建实验模板。当系统提示您确认时，输入 **create**，然后选择创建实验模板。

（可选）查看 JSON 格式的实验模板

选择导出选项卡。以下是通过前述控制台程序创建的 JSON 示例。

```
{
    "description": "Test instance stop and start",
    "targets": {
        "bothInstances": {
            "resourceType": "aws:ec2:instance",
            "resourceArns": [
                "arn:aws:ec2:region:123456789012:instance/instance_id_1",
                "arn:aws:ec2:region:123456789012:instance/instance_id_2"
            ],
            "selectionMode": "ALL"
        },
        "oneRandomInstance": {
            "resourceType": "aws:ec2:instance",
            "resourceArns": [
                "arn:aws:ec2:region:123456789012:instance/instance_id_1",
                "arn:aws:ec2:region:123456789012:instance/instance_id_2"
            ],
            "selectionMode": "COUNT(1)"
        }
    },
    "actions": {
        "stopBothInstances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "startInstancesAfterDuration": "PT3M"
            },
            "targets": {
                "Instances": "bothInstances"
            },
            "startAfter": [
```

```
                "stopOneInstance"
            ]
        },
        "stopOneInstance": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "startInstancesAfterDuration": "PT3M"
            },
            "targets": {
                "Instances": "oneRandomInstance"
            }
        }
    },
    "stopConditions": [
        {
            "source": "none"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/AllowFISEC2Actions",
    "tags": {}
}
```

步骤 2：开始实验

您可以使用创建好的实验模板开始实验。

开始实验

1. 您应该位于刚刚创建的实验模板的详细信息页面。否则，请选择实验模板，然后选择实验模板 ID，打开详细信息页面。
2. 请选择开始实验。
3. （可选）要为实验添加标签，请选择添加新标签，然后输入标签键和标签值。
4. 请选择开始实验。当系统提示您确认时，输入 **start**，然后选择开始实验。

步骤 3：跟踪实验进度

您可以跟踪正在运行的实验进度，直到实验完成、停止或失败。

跟踪实验进度

1. 您应该位于刚开始的实验的详细信息页面。否则，请选择实验，然后选择实验 ID，打开详细信息页面。
2. 要查看实验状态，请在详细信息窗格中选择状态。有关更多信息，请参阅实验状态。
3. 当实验状态为正在运行时，转到下一步。

步骤 4：验证实验结果

您可以验证实验是否如预期般停止并启动实例。

验证实验结果

1. 您可以在新的浏览器选项卡或窗口中访问 https://console.aws.amazon.com/ec2/，打开 Amazon EC2 控制台。您可以继续在 AWS FIS 控制台中跟踪实验进度，同时在 Amazon EC2 控制台中查看实验结果。
2. 在导航窗格中，选择 实例。
3. 当第一项操作的状态从待处理更改为正在运行（AWS FIS 控制台）时，则其中一个目标实例的状态将从正在运行更改为已停止（Amazon EC2 控制台）。
4. 三分钟后，第一项操作和第二项操作的状态会分别更改为已完成和正在运行，同时另一个目标实例的状态会更改为已停止。
5. 三分钟后，第二项操作的状态更改为已完成，目标实例的状态更改为正在运行，实验状态更改为已完成。

步骤 5：清除

如果不再需要为本教程创建的 EC2 测试实例，可以将其删除。

终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择两个测试实例，然后依次选择 Instance state（实例状态）、Terminate instance（终止实例）。
4. 当系统提示您确认时，选择终止。

如果您不再需要实验模板，可以将其删除。

使用 AWS FIS 控制台删除实验模板

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和删除实验模板。
4. 当系统提示您确认时，输入 **delete**，然后选择删除实验模板。

教程：使用 AWS FIS 在实例上运行 CPU 压力测试

您可以使用 AWS Fault Injection Service (AWS FIS)，测试应用程序如何处理 CPU 压力。按教程创建实验模板，其通过 AWS FIS 实施在实例上运行 CPU 压力测试的预配置 SSM 文档。此教程会在实例的 CPU 利用率超过配置阈值时，使用停止条件以停止实验。

有关更多信息，请参见 the section called “预先配置的 AWS FIS SSM 文档”。

先决条件

请先满足以下先决条件，再使用 AWS FIS 运行 CPU 压力测试。

创建 IAM 角色

创建角色并附加策略，使 AWS FIS 代表您执行 `aws:ssm:send-command` 操作。有关更多信息，请参见 适用于 AWS FIS 实验的 IAM 角色。

验证对 AWS FIS 的访问权限

确保您有权访问 AWS FIS。有关更多信息，请参阅 AWS FIS 策略示例。

准备 EC2 测试实例

- 按照预配置 SSM 文档的要求，使用 Amazon Linux 2 或 Ubuntu 启动 EC2 实例。
- 此实例必须由 SSM 托管。要验证实例是否由 SSM 托管，则打开 Fleet Manager 控制台。如果实例不是由 SSM 管理的，请验证是否已安装 SSM 代理，以及该实例是否已附加具有 A mazon ManagedInstanceCore SSM 策略的 IAM 角色。要验证已安装的 SSM 代理，则连接实例并运行以下命令。

  Amazon Linux 2

```
yum info amazon-ssm-agent
```

Ubuntu

```
apt list amazon-ssm-agent
```

- 对实例启用详细监控。此操作会提供时长 1 分钟的数据，需要额外付费。选择所需实例，然后依次选择操作、监控和问题排查，以及管理详细监控。

步骤 1：为停止条件创建 CloudWatch 警报

配置 CloudWatch 警报，以便在 CPU 使用率超过您指定的阈值时可以停止实验。以下过程将目标实例的 CPU 利用率阈值设置为 50%。有关更多信息，请参见 停止条件。

创建警报，指明 CPU 利用率何时超过阈值

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择目标实例，然后选择操作、监控和故障排除、管理 CloudWatch 警报。
4. 对于警报通知，使用切换按钮关闭 Amazon SNS 通知。
5. 对于警报阈值，请使用以下设置：
   - 样本分组依据：最大值
   - 要采样的数据类型：CPU 利用率
   - 百分比：**50**
   - 周期：**1 Minute**
6. 配置完警报后，选择创建。

步骤 2：创建实验模板

使用 AWS FIS 控制台创建实验模板。在模板中，您可以指定要运行的以下操作：aws: ssm: s AWSFIS end-command/-run-cpu-Stress。

创建实验模板

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。

2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 对于描述和名称，输入模板的描述和名称。
5. 对于操作，请执行以下操作：
   a. 选择添加操作。
   b. 输入操作名称。例如，输入 **runCpuStress**。
   c. 对于操作类型，选择 aws: ssm: s AWSFIS end-command/-run-cpu-stress。此操作会自动向 Document ARN 添加 SSM 文档的 ARN。
   d. 对于目标，保留 AWS FIS 为您创建的目标。
   e. 对于操作参数，文档参数，输入以下内容：

      ```
      {"DurationSeconds":"120"}
      ```

   f. 对于操作参数，持续时间，指定为 5 分钟 (PT5M)。
   g. 选择保存。
6. 对于目标，请执行以下操作：
   a. 对于 AWS FIS 在上一步中自动创建的目标，选择编辑。
   b. 将默认名称替换为更具描述性的名称。例如，输入 **testInstance**。
   c. 验证资源类型是否为 aws:ec2:instance。
   d. 对于目标方法，选择资源 ID，然后选择测试实例 ID。
   e. 对于选择模式，选择全部。
   f. 选择保存。
7. 对于服务访问权限，选择使用现有 IAM 角色，然后选择您按照本教程先决条件中所述创建的 IAM 角色。如未显示此角色，请验证其是否具有必要的信任关系。有关更多信息，请参见 the section called “实验角色”。
8. 对于停止条件，请选择您在步骤 1 中创建的 CloudWatch 警报。
9. （可选）对于标签，选择添加新标签，然后指定标签键和标签值。您添加的标签将应用于实验模板，而不是应用于使用此模板运行的实验。
10. 选择创建实验模板。

（可选）查看 JSON 格式的实验模板

选择导出选项卡。以下是通过前述控制台程序创建的 JSON 示例。

```
{
    "description": "Test CPU stress predefined SSM document",
    "targets": {
        "testInstance": {
            "resourceType": "aws:ec2:instance",
            "resourceArns": [
                "arn:aws:ec2:region:123456789012:instance/instance_id"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "runCpuStress": {
            "actionId": "aws:ssm:send-command",
            "parameters": {
                "documentArn": "arn:aws:ssm:region::document/AWSFIS-Run-CPU-Stress",
                "documentParameters": "{\"DurationSeconds\":\"120\"}",
                "duration": "PT5M"
            },
            "targets": {
                "Instances": "testInstance"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:region:123456789012:alarm:awsec2-instance_id-
GreaterThanOrEqualToThreshold-CPUUtilization"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/AllowFISSSMActions",
    "tags": {}
}
```

步骤 3：开始实验

您可以使用创建好的实验模板开始实验。

开始实验

1. 您应该位于刚刚创建的实验模板的详细信息页面。否则，请选择实验模板，然后选择实验模板 ID，打开详细信息页面。
2. 请选择开始实验。
3. （可选）要为实验添加标签，请选择添加新标签，然后输入标签键和标签值。
4. 请选择开始实验。当系统提示进行确认时，输入 **start**。请选择开始实验。

步骤 4：跟踪实验进度

您可以跟踪正在运行的实验进度，直到实验完成、停止或失败。

跟踪实验进度

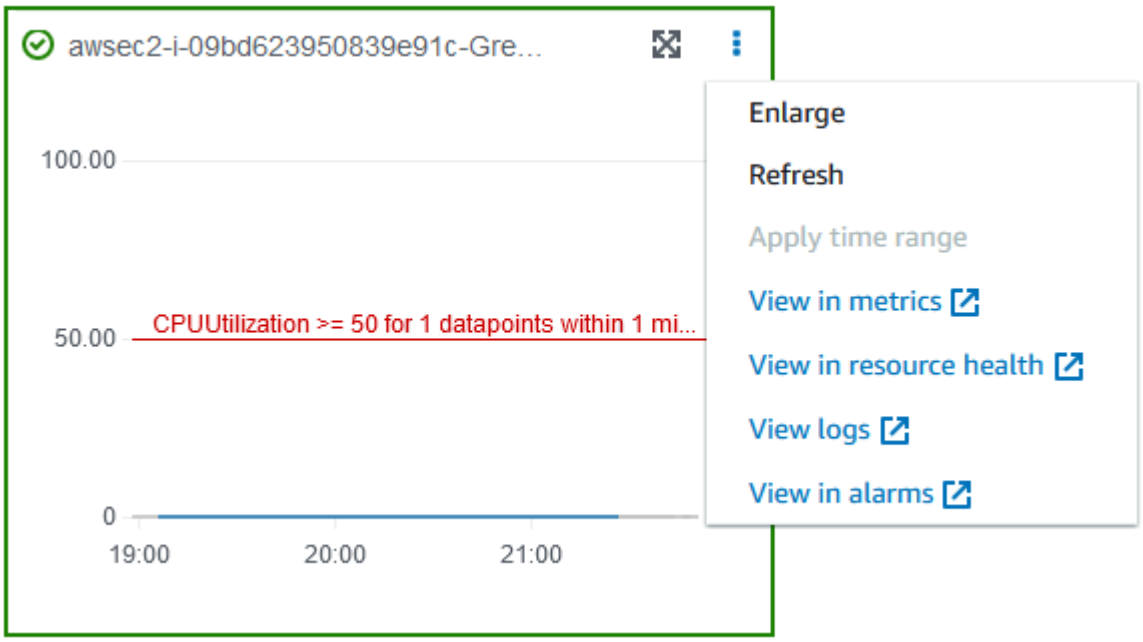
1. 您应该位于刚开始的实验的详细信息页面。否则，请选择实验，然后选择对应 ID 打开详细信息页面。
2. 要查看实验状态，请在详细信息窗格中选择状态。有关更多信息，请参阅实验状态。
3. 当实验状态为正在运行时，转到下一步。

步骤 5：验证实验结果

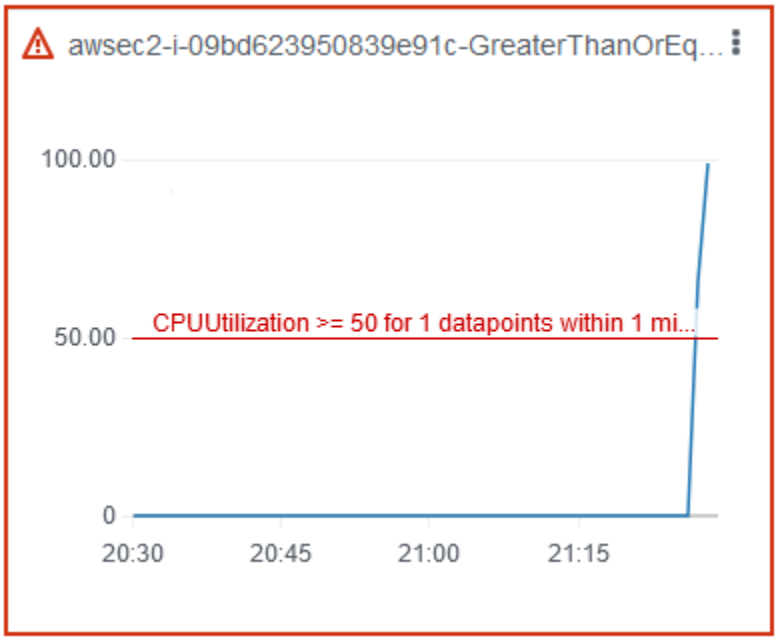
您可以在实验运行期间监控实例的 CPU 利用率。CPU 利用率达到阈值时会触发警报，实验也会因停止条件而停止。

验证实验结果

1. 选择停止条件选项卡。绿色边框和绿色勾选图标表示警报的初始状态为 OK。红线则表示警报阈值。如需更详细的图表，请在小组件菜单中选择放大。



2. 当 CPU 利用率超过阈值时，停止条件选项卡中会显示红色的边框和感叹号图标，表示警报状态已更改为 ALARM。详细信息窗格中的实验状态显示为已停止。如果选择此状态，则会显示“实验因停止条件而停止”的消息。



3. 当 CPU 利用率降低到阈值以下时，会显示绿色的边框和勾选图标，表示警报状态已更改为 OK。
4. （可选）在小组件菜单中选择在警报中查看。这将打开 CloudWatch 控制台中的警报详细信息页面，您可以在其中获取有关警报的更多详细信息或编辑警报设置。

步骤 6：清理

如果您不再需要为实验创建的 EC2 测试实例，可以将其删除。

终止实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择测试实例，然后依次选择实例状态和终止实例。
4. 当系统提示您确认时，选择终止。

如果您不再需要实验模板，可以将其删除。

使用 AWS FIS 控制台删除实验模板

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和删除实验模板。
4. 当系统提示您确认时，输入 **delete**，然后选择删除实验模板。

教程：使用 AWS FIS 测试竞价型实例中断情况

竞价型实例使用的是可用的 EC2 备用容量，与按需定价相比，可享受高达 90% 的折扣。但 Amazon EC2 可以在需要收回时中断竞价型实例。在使用竞价型实例时，您必须为可能发生的中断情况做好准备。有关更多信息，请参阅 Amazon EC2 用户指南中的竞价型实例中断情况。

您可以使用 AWS Fault Injection Service (AWS FIS)，测试应用程序如何处理竞价型实例中断情况。按教程创建实验模板，其通过 AWS FIS `aws:ec2:send-spot-instance-interruptions` 操作中断某个竞价型实例。

或者，要使用 Amazon EC2 控制台启动实验，请参阅 Amazon EC2 用户指南中的 启动竞价型实例中断。

先决条件

请先满足以下先决条件，再使用 AWS FIS 中断竞价型实例。

1. 创建 IAM 角色

创建角色并附加策略，使 AWS FIS 代表您执行 `aws:ec2:send-spot-instance-interruptions` 操作。有关更多信息，请参见 适用于 AWS FIS 实验的 IAM 角色。

2. 验证对 AWS FIS 的访问权限

确保您有权访问 AWS FIS。有关更多信息，请参阅 AWS FIS 策略示例。

3. （可选）创建竞价型实例请求

如果要为实验使用新竞价型实例，请运行请求竞价型实例的 run-instances 命令。默认在竞价型实例中断时将其终止。如果将中断行为设置为 `stop`，则必须将类型设置为 `persistent`。请勿在本教程中将中断行为设置为 `hibernate`，这会立即启动休眠程序。

```
aws ec2 run-instances \
    --image-id ami-0ab193018fEXAMPLE \
    --instance-type "t2.micro" \
    --count 1 \
    --subnet-id subnet-1234567890abcdef0 \
    --security-group-ids sg-111222333444aaab \
    --instance-market-options file://spot-options.json \
    --query Instances[*].InstanceId
```

以下是 `spot-options.json` 文件的示例。

```
{
    "MarketType": "spot",
    "SpotOptions": {
        "SpotInstanceType": "persistent",
        "InstanceInterruptionBehavior": "stop"
    }
}
```

示例命令中的 `--query` 选项使此命令仅返回竞价型实例 ID。下面是示例输出。

```
[
    "i-0abcdef1234567890"
]
```

4. 添加标签，以便 AWS FIS 可以标识目标竞价型实例

运行 create-tags 命令，为您的目标竞价型实例添加 Name=interruptMe 标签。

```
aws ec2 create-tags \
    --resources i-0abcdef1234567890 \
    --tags Key=Name,Value=interruptMe
```

步骤 1：创建实验模板

使用 AWS FIS 控制台创建实验模板。您可以在模板中指定要运行的操作。此操作会中断带有指定标签的竞价型实例。如果有多个带有该标签的竞价型实例，AWS FIS 会随机中断其中一个。

创建实验模板

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 对于描述和名称，输入模板的描述和名称。
5. 对于操作，请执行以下操作：
    a. 选择添加操作。
    b. 输入操作名称。例如，输入 **interruptSpotInstance**。
    c. 对于操作类型，请选择 aws: ec2:。send-spot-instance-interruptions
    d. 对于目标，保留 AWS FIS 为您创建的目标。
    e. 对于操作参数，中断前持续时间，指定为 2 分钟 (PT2M)。
    f. 选择保存。
6. 对于目标，请执行以下操作：
    a. 对于 AWS FIS 在上一步中自动创建的目标，选择编辑。
    b. 将默认名称替换为更具描述性的名称。例如，输入 **oneSpotInstance**。
    c. 验证资源类型是否为 aws:ec2:spot-instance。
    d. 对于目标方法，选择资源标签、筛选条件和参数。
    e. 对于资源标签，选择添加新标签，然后输入标签键和标签值。使用您为竞价型实例添加的标签进行中断，如本教程在先决条件中所述。
    f. 对于资源筛选条件，选择添加新筛选条件，然后输入 **State.Name** 作为路径并输入 **running** 作为值。

g. 对于选择模式，选择计数。对于资源数量，输入 **1**。

h. 选择保存。

7. 对于服务访问权限，选择使用现有 IAM 角色，然后选择您按照本教程先决条件中所述创建的 IAM 角色。如未显示此角色，请验证其是否具有必要的信任关系。有关更多信息，请参见 the section called “实验角色”。

8. （可选）对于标签，选择添加新标签，然后指定标签键和标签值。您添加的标签将应用于实验模板，而不是应用于使用此模板运行的实验。

9. 选择创建实验模板。当系统提示您确认时，输入 **`create`**，然后选择创建实验模板。

（可选）查看 JSON 格式的实验模板

选择导出选项卡。以下是通过前述控制台程序创建的 JSON 示例。

```
{
    "description": "Test Spot Instance interruptions",
    "targets": {
        "oneSpotInstance": {
            "resourceType": "aws:ec2:spot-instance",
            "resourceTags": {
                "Name": "interruptMe"
            },
            "filters": [
                {
                    "path": "State.Name",
                    "values": [
                        "running"
                    ]
                }
            ],
            "selectionMode": "COUNT(1)"
        }
    },
    "actions": {
        "interruptSpotInstance": {
            "actionId": "aws:ec2:send-spot-instance-interruptions",
            "parameters": {
                "durationBeforeInterruption": "PT2M"
            },
            "targets": {
                "SpotInstances": "oneSpotInstance"
```

```
            }
        }
    },
    "stopConditions": [
        {
            "source": "none"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/AllowFISSpotInterruptionActions",
    "tags": {
        "Name": "my-template"
    }
}
```

步骤 2：开始实验

您可以使用创建好的实验模板开始实验。

开始实验

1. 您应该位于刚刚创建的实验模板的详细信息页面。否则，请选择实验模板，然后选择实验模板 ID，打开详细信息页面。
2. 请选择开始实验。
3. （可选）要为实验添加标签，请选择添加新标签，然后输入标签键和标签值。
4. 请选择开始实验。当系统提示您确认时，输入 **start**，然后选择开始实验。

步骤 3：跟踪实验进度

您可以跟踪正在运行的实验进度，直到实验完成、停止或失败。

跟踪实验进度

1. 您应该位于刚开始的实验的详细信息页面。否则，请选择实验，然后选择实验 ID，打开详细信息页面。
2. 要查看实验状态，请在详细信息窗格中选择状态。有关更多信息，请参阅实验状态。
3. 当实验状态为正在运行时，转到下一步。

步骤 4：验证实验结果

实验操作完成后，将出现以下情况：

- 目标竞价型实例会收到一条实例重新平衡建议。
- 在 Amazon EC2 终止或竞价型实例停止前两分钟，发送竞价型实例中断通知。
- 两分钟后终止或停止竞价型实例。
- 在重新启动前，由 AWS FIS 停止的竞价型实例将一直处于停止状态。

验证实验是否已经中断实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 从导航窗格中，在单独的浏览器选项卡或窗口中打开 Spot Requests（竞价型实例请求）和 Instances（实例）。
3. 对于 Spot Requests（竞价型实例请求），选择该竞价型实例请求。初始状态为 `fulfilled`。实验完成后的状态变化如下：
   - `terminate`：状态变为 `instance-terminated-by-experiment`。
   - `stop`：状态先变为 `marked-for-stop-by-experiment`，然后变为 `instance-stopped-by-experiment`。
4. 对于实例，选择竞价型实例。初始状态为 `Running`。在您收到竞价型实例中断通知后两分钟，状态会根据中断行为发生以下变化：
   - `stop`：状态先变为 `Stopping`，然后变为 `Stopped`。
   - `terminate`：状态先变为 `Shutting-down`，然后变为 `Terminated`。

第 5 步：清理

如果您不再需要通过 `stop` 中断行为为实验创建的竞价型测试实例，则可以取消竞价型实例请求并终止竞价型实例。

使用 AWS CLI 取消请求并终止实例

1. 使用cancel-spot-instance-requests命令取消竞价型实例请求。

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-ksie869j
```

2. 运行 terminate-instances 命令，终止实例。

```
aws ec2 terminate-instances --instance-ids i-0abcdef1234567890
```

如果您不再需要实验模板，可以将其删除。

使用 AWS FIS 控制台删除实验模板

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和删除实验模板。
4. 当系统提示您确认时，输入 **delete**，然后选择删除实验模板。

教程：模拟连接事件

您可以使用 AWS 故障注入服务 (AWS FIS) 来模拟各种连接事件。 AWS FIS 通过以下方式之一阻止网络连接来模拟连接事件：

- `all`：拒绝所有往返于子网的流量。请注意，此选项允许子网内流量，包括往返于子网中网络接口的流量。
- `availability-zone`：拒绝往返于其他可用区子网的 VPC 内流量。
- `dynamodb`：拒绝往返于当前区域中 DynamoDB 区域端点的流量。
- `prefix-list`：拒绝往返于指定前缀列表的流量。
- `s3`：拒绝往返于当前区域中 Amazon S3 区域端点的流量。
- `vpc`：拒绝往返于 VPC 的流量。

使用本教程创建实验模板，该模板使用 AWS FIS `aws:network:disrupt-connectivity` 操作在目标子网中引入与 Amazon S3 的连接中断。

主题

- 先决条件
- 步骤 1：创建 AWS FIS 实验模板
- 步骤 2：对 Amazon S3 端点执行 Ping 操作
- 第 3 步：开始你的 AWS FIS 实验
- 第 4 步：追踪您 AWS 的 FIS 实验进度

- 步骤 5：验证 Amazon S3 网络中断
- 步骤 5：清除

先决条件

在开始本教程之前，您需要在您的角色中拥有相应权限 AWS 账户，并需要一个测试 Amazon EC2 实例：

在你中拥有权限的角色 AWS 账户

创建一个角色并附加一个策略，使 AWS FIS 能够代表您执行`aws:network:disrupt-connectivity`操作。

IAM 角色需要以下策略：

- AWSFaultInjectionSimulatorNetworkAccess— 授予 AWS FIS 服务在 Amazon EC2 联网和其他必需服务中的权限，以执行与网络基础设施相关 AWS 的 FIS 操作。

> **Note**
>
> 为简单起见，本教程使用 AWS 托管策略。亚马逊建议您仅授予用例所需的最低生产权限。有关如何创建 IAM 角色的更多信息，请参阅 I AM 用户指南中的用 AWS 于 FIS 实验的 IAM 角色 (AWS CLI) 或创建 IAM 角色（控制台）。

Amazon EC2 测试实例

启动并连接 Amazon EC2 测试实例。您可以使用以下教程启动和连接亚马逊 EC2 实例：教程：亚马逊 EC2 用户指南中的亚马逊 EC2 Linux 实例入门。

步骤 1：创建 AWS FIS 实验模板

使用 AWS FI AWS Management Console S 创建实验模板。 AWS FIS 模板由动作、目标、停止条件和实验角色组成。有关模板工作原理的更多信息，请参阅适用于 AWS FIS 的实验模板。

在开始之前，请确保您做好以下准备：

- 具有适当权限的 IAM 角色。
- 一个 Amazon EC2 实例。

- Amazon EC2 实例的子网 ID。

创建实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在左侧导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 输入模板描述，如 `Amazon S3 Network Disrupt Connectivity`。
5. 在操作下，选择添加操作。
   a. 对于名称，输入 `disruptConnectivity`。
   b. 对于操作类型，选择 aws:network:disrupt-connectivity。
   c. 在操作参数下，将持续时间设置为 `2 minutes`。
   d. 在范围下，选择 s3。
   e. 选择顶部的保存。
6. 在目标下，您会看到已自动创建的目标。选择编辑。
   a. 验证资源类型是否为 `aws:ec2:subnet`。
   b. 在目标方法下，选择资源 ID，然后选择您在先决条件步骤中创建 Amazon EC2 实例时使用的子网。
   c. 验证选择模式为全部。
   d. 选择保存。
7. 在服务访问权限下，选择您按照本教程先决条件中所述创建的 IAM 角色。如未显示此角色，请验证其是否具有必要的信任关系。有关更多信息，请参阅 the section called “实验角色”。
8. （可选）在“停止”条件下，您可以选择一个 CloudWatch警报，以便在条件出现时停止实验。有关更多信息，请参阅 AWS FIS 停止条件。
9. （可选）在“日志”下，您可以选择一个 Amazon S3 存储桶，或者将日志发送到 CloudWatch 进行实验。
10. 选择创建实验模板，当系统提示您确认时，输入 `create`。选择创建实验模板。

步骤 2：对 Amazon S3 端点执行 Ping 操作

验证您的 Amazon EC2 实例能否访问 Amazon S3 端点。

1. 连接到您在先决条件步骤中创建的 Amazon EC2 实例。

   要进行故障排除，请参阅 Amazon EC2 用户指南中的实例连接故障排除。

2. 查看您的实例 AWS 区域 所在的位置。您可以在 Amazon EC2 控制台中或通过运行以下命令完成此操作。

   ```
   hostname
   ```

   例如，当您在 us-west-2 中启动 Amazon EC2 实例，就会看到以下输出。

   ```
   [ec2-user@ip-172.16.0.0 ~]$ hostname
   ip-172.16.0.0.us-west-2.compute.internal
   ```

3. Ping 您的 Amazon S3 终端节点 AWS 区域。将 *AWS ##* 替换为您的区域。

   ```
   ping -c 1 s3.AWS ##.amazonaws.com
   ```

   您会在输出结果中看到丢包率为 0% 的成功 Ping 操作，如以下示例所示。

   ```
   PING s3.us-west-2.amazonaws.com (x.x.x.x) 56(84) bytes of data.
   64 bytes from s3-us-west-2.amazonaws.com (x.x.x.x: icmp_seq=1 ttl=249 time=1.30 ms

   --- s3.us-west-2.amazonaws.com ping statistics ---
   1 packets transmitted, 1 received, 0% packet loss, time 0ms
   rtt min/avg/max/mdev = 1.306/1.306/1.306/0.000 ms
   ```

第 3 步：开始你的 AWS FIS 实验

使用刚刚创建的实验模板开始实验。

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在左侧导航窗格中，选择实验模板。
3. 选择创建的实验模板 ID，打开详细信息页面。
4. 请选择开始实验。
5. （可选）在确认页面中为实验添加标签。
6. 在确认页面中，选择开始实验。

第 4 步：追踪您 AWS 的 FIS 实验进度

您可以跟踪正在运行的实验进度，直到实验完成、停止或失败。

1. 您应该位于刚开始的实验的详细信息页面。否则，请选择实验，然后选择实验 ID，打开其详细信息页面。
2. 要查看实验状态，请在详细信息窗格中选择状态。有关更多信息，请参阅实验状态。
3. 当实验状态为正在运行时，移至下一步。

步骤 5：验证 Amazon S3 网络中断

您可以对 Amazon S3 端点执行 Ping 操作，以验证实验进度。

- 您可以通过 Amazon EC2 实例，对 AWS 区域中的 Amazon S3 端点执行 Ping 操作。将 *AWS ##* 替换为您的区域。

```
ping -c 1 s3.AWS ##.amazonaws.com
```

您会在输出结果中看到丢包率为 100% 的失败 Ping 操作，如以下示例所示。

```
ping -c 1 s3.us-west-2.amazonaws.com
PING s3.us-west-2.amazonaws.com (x.x.x.x) 56(84) bytes of data.

--- s3.us-west-2.amazonaws.com ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

步骤 5：清除

如果无需再使用为此实验创建的 Amazon EC2 实例或 AWS FIS 模板，可以将其删除。

删除 Amazon EC2 实例

1. 通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Instances (实例)。
3. 选择测试实例，然后依次选择实例状态和终止实例。
4. 当系统提示您确认时，选择终止。

使用 AWS FIS 控制台删除实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和删除实验模板。
4. 当系统提示您确认时，输入 `delete`，然后选择删除实验模板。

教程：安排定期实验

您可以使用 AWS Fault Injection Service (AWS FIS)，对 AWS 工作负载执行故障注入实验。这些实验运行在模板上，其中包含要在指定目标上运行的一项或多项操作。如果还使用 Amazon EventBridge，则可以安排实验为一次性任务或循环任务。

使用本教程创建每隔 5 分钟运行一个 AWS FIS 实验模板的 EventBridge 计划。

任务

- 先决条件
- 步骤 1：创建 IAM 角色和策略
- 步骤 2：创建 Amazon EventBridge 调度器
- 步骤 3：验证实验
- 步骤 4：清除

先决条件

请务必先创建按计划运行的 AWS FIS 实验模板，再开始使用本教程。如果已有可用的实验模板，请记下模板 ID 和 AWS 区域。否则，您可以按 the section called “测试实例停止和启动” 中的说明创建模板，然后返回本教程。

步骤 1：创建 IAM 角色和策略

创建 IAM 角色和策略

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 在左侧的导航窗格中，选择角色，然后选择创建角色。
3. 选择自定义信任策略，然后插入以下代码段，允许 Amazon EventBridge 调度器代表您代入角色。

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

请选择 Next（下一步）。

4. 在添加权限下，选择创建策略。
5. 选择 JSON 格式，然后插入以下策略。将该*your-experiment-template-id*值替换为先决条件步骤中实验的模板 ID。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": [
                "arn:aws:fis:*:*:experiment-template/your-experiment-template-id",
                "arn:aws:fis:*:*:experiment/*"
            ]
        }
    ]
}
```

您可以限制调度器，使其只运行具有特定标签值的 AWS FIS 实验。例如，以下策略向所有 AWS FIS 实验模板授予 `StartExperiment` 权限，但限制调度器只运行带有 `Purpose=Schedule` 标签的实验。

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

```
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment/*"
        },
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment-template/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Schedule"
                }
            }
        }
    ]
}
```

选择下一步：标签。

6. 选择下一步：审核。
7. 在查看策略下，命名策略 `FIS_RecurringExperiment`，然后选择创建策略。
8. 在添加权限下，为您的角色添加 `FIS_RecurringExperiment` 新策略，然后选择下一步。
9. 在命名、检查并创建下，命名角色 `FIS_RecurringExperiment_role`，然后选择创建角色。

步骤 2：创建 Amazon EventBridge 调度器

创建 Amazon EventBridge 调度器

1. 打开亚马逊 EventBridge 控制台，网址为 https://console.aws.amazon.com/events/。
2. 在左侧导航窗格中，选择计划。
3. 验证您是否与 AWS FIS 实验模板使用了相同的 AWS 区域。
4. 选择创建计划，然后填写以下内容：

   - 在计划名称下，插入 `FIS_recurring_experiment_tutorial`。
   - 在计划模式下，选择定期计划。
   - 在计划类型下，选择基于速率的计划。
   - 在 Rate 表达式下，选择 5 分钟。
   - 在灵活时间窗口下，选择关闭。

- （可选）在时间范围下，选择您的时区。
- 请选择 Next（下一步）。

5. 在选择目标下，选择所有 API，然后搜索 AWS FIS。
6. 选择 AWSFIS，然后选择StartExperiment。
7. 在输入下，插入以下 JSON 格式的有效负载。将该*your-experiment-template-id*值替换为实验的模板 ID。`ClientToken` 是调度器的唯一标识符。本教程使用的是 Amazon EventBridge 调度器所允许的上下文关键字。有关更多信息，请参阅 Amazon EventBridge 用户指南中的添加上下文属性。

```
{
    "ClientToken": "<aws.scheduler.execution-id>",
    "ExperimentTemplateId": "your-experiment-template-id"
}
```

   请选择 Next（下一步）。
8. （可选）在设置下，设置重试策略、死信队列 (DLQ) 和加密设置。您也可以保留默认值。
9. 在权限下，选择使用现有角色，然后搜索 `FIS_RecurringExperiment_role`。
10. 请选择 Next（下一步）。
11. 在查看并创建计划下，查看调度器的详细信息，然后选择创建计划。

步骤 3：验证实验

验证 AWS FIS 实验是否按计划运行

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在左侧导航窗格中，选择实验。
3. 创建计划五分钟后，您会看到正在运行的实验。

步骤 4：清除

禁用 Amazon EventBridge 调度器

1. 打开亚马逊 EventBridge 控制台，网址为 https://console.aws.amazon.com/events/。
2. 在左侧导航窗格中，选择计划。

3. 选择新创建的调度器，然后选择禁用。

的操作 AWS FIS

操作是您使用 AWS Fault Injection Service (AWS FIS) 在目标上运行的错误注入活动。 AWS FIS 为跨 AWS 服务的特定类型的目标提供预配置的操作。您可以为实验模板添加操作，然后使用此模板运行实验。

内容

- 操作标识符
- 操作参数
- 操作目标
- AWS FIS 动作参考
- 将 Systems Manager SSM 文档与 FIS 一起使用 AWS
- 使用 AWS FIS aws: ecs: task 操作
- 使用 FIS aw AWS s: eks: pod 操作
- 使用列出 AWS FIS 操作 AWS CLI

操作标识符

每个 AWS FIS 操作都有一个标识符，其格式如下：

```
aws:service-name:action-type
```

例如，以下操作停止运行 Amazon EC2 目标实例：

```
aws:ec2:stop-instances
```

有关操作的完整列表，请参阅 AWS FIS 动作参考。要使用获取列表 AWS CLI，请参阅列出动作。

操作参数

有些 AWS FIS 操作具有特定于该操作的其他参数。这些参数用于在操作运行 AWS FIS 时向其传递信息。

AWS FIS 支持使用操作自定义故障类型，该aws:ssm:send-command操作使用 SSM 代理和 SSM 命令文档在目标实例上创建故障条件。aws:ssm:send-command 操作包含 documentArn 参数，将 SSM 文档的 Amazon 资源名称 (ARN) 作为值。当为实验模板添加操作时，您可以指定参数值。

有关为 aws:ssm:send-command 操作指定参数的更多信息，请参阅 执行 aws:ssm:send-command 操作。。

您可以尽可能向操作参数输入回滚配置（也称为后置操作）。后置操作可将目标返回到操作运行之前的状态。此操作会在操作持续时间的指定时段后运行。并非所有操作都支持后置操作。例如，您无法恢复由操作终止的 Amazon EC2 实例。

操作目标

运行在您指定的目标资源上的操作。定义目标后，您可以在定义操作时指定其名称。

```
"targets": {
    "resource_type": "resource_name"
}
```

AWS FIS 操作支持操作目标的以下资源类型：

- 自动扩缩组：Amazon EC2 自动扩缩组
- 存储桶：Amazon S3 存储桶
- 集群：Amazon EKS 集群
- 集群：Amazon ECS 集群或 Amazon Aurora 数据库集群
- 数据库实例：Amazon RDS 数据库实例
- 加密的全局表：Amazon DynamoDB；使用客户托管密钥进行加密的全局表
- 全局表 — 亚马逊 DynamoDB；全局表
- 实例：Amazon EC2 实例
- 节点组：Amazon EKS 节点组
- 容器组（pod）：Amazon EKS 上的 Kubernetes 容器组（pod）
- ReplicationGroups— ElastiCache Redis 复制组
- 角色：IAM 角色
- SpotInstances— 亚马逊 EC2 竞价型实例
- 子网：VPC 子网

- 任务：Amazon ECS 任务
- TransitGateways— 公交网关
- 卷：Amazon EBS 卷

有关示例，请参阅the section called “操作示例”。

AWS FIS 动作参考

本参考描述了中的常见操作 AWS FIS，包括有关操作参数和所需的 IAM 权限的信息。您也可以使用 AWS FIS 控制台或 AWS Command Line Interface ()AWS CLI中的 list-actions 命令列出支持的 AWS FIS 操作。

有关更多信息，请参阅 的操作 AWS FIS 和 AWS 故障注入服务如何与 IAM 配合使用。

操作

- 故障注入操作
- 等待动作
- 亚马逊的 CloudWatch 行动
- Amazon DynamoDB 操作
- Amazon EBS 操作
- Amazon EC2 操作
- Amazon ECS 操作
- Amazon EKS 操作
- 亚马逊的 ElastiCache 行动
- 网络操作
- Amazon RDS 操作
- Amazon S3 操作
- Systems Manager 操作

故障注入操作

AWS FIS 支持以下故障注入操作。

操作

- aws:fis:inject-api-internal-error
- aws:fis:inject-api-throttle-error
- aws:fis:inject-api-unavailable-error

aws:fis:inject-api-internal-error

在目标 IAM 角色发出的请求中注入内部错误。

资源类型

- aws:iam:role

参数

- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。
- service— 目标 AWS API 命名空间。支持的值为 `ec2`。
- percentage：注入故障信息的调用百分比 (1 - 100)。
- operations：注入故障信息的操作，用逗号分隔。有关 `ec2` 命名空间的 API 操作列表，请参阅 Amazon EC2 API 参考中的操作。

权限

- `fis:InjectApiInternalError`

aws:fis:inject-api-throttle-error

在目标 IAM 角色发出的请求中注入节流错误。

资源类型

- aws:iam:role

参数

- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

- service— 目标 AWS API 命名空间。支持的值为 `ec2`。
- percentage：注入故障信息的调用百分比 (1 - 100)。
- operations：注入故障信息的操作，用逗号分隔。有关 `ec2` 命名空间的 API 操作列表，请参阅 Amazon EC2 API 参考中的操作。

权限

- `fis:InjectApiThrottleError`

aws:fis:inject-api-unavailable-error

在目标 IAM 角色发出的请求中注入不可用错误。

资源类型

- aws:iam:role

参数

- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。
- service— 目标 AWS API 命名空间。支持的值为 `ec2`。
- percentage：注入故障信息的调用百分比 (1 - 100)。
- operations：注入故障信息的操作，用逗号分隔。有关 `ec2` 命名空间的 API 操作列表，请参阅 Amazon EC2 API 参考中的操作。

权限

- `fis:InjectApiUnavailableError`

等待动作

AWS FIS 支持以下等待操作。

aws:fis:wait

运行 AWS FIS 等待操作。

参数

- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- 无

亚马逊的 CloudWatch 行动

AWS FIS 支持以下 Amazon CloudWatch 操作。

aws:cloudwatch:assert-alarm-state

验证指定警报是否处于指定警报状态之一。

资源类型

- 无

参数

- alarmArns：警报 ARN，用逗号分隔。您最多可以指定五个警报。
- alarmStates：警报状态，用逗号分隔。警报状态可能是 `OK`、`ALARM` 和 `INSUFFICIENT_DATA`。

权限

- `cloudwatch:DescribeAlarms`

Amazon DynamoDB 操作

AWS FIS 支持以下亚马逊 DynamoDB 操作。

aws:dynamodb:global-table-pause-replication

暂停向任何副本表的 Amazon DynamoDB 全局表复制。操作开始后，表可能会继续复制最多 5 分钟。

以下语句将动态附加到目标 DynamoDB 全局表的策略中：

```
{
   "Statement":[
      {
         "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxx"
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
         },
         "Action":[
            "dynamodb:GetItem",
            "dynamodb:PutItem",
            "dynamodb:UpdateItem",
            "dynamodb:DeleteItem",
            "dynamodb:DescribeTable",
            "dynamodb:UpdateTable",
            "dynamodb:Scan",
            "dynamodb:DescribeTimeToLive",
            "dynamodb:UpdateTimeToLive"
         ],
         "Resource":"arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
         "Condition": {
            "DateLessThan": {
            "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
         }
       }
      }
   ]
}
```

以下语句将动态附加到目标 DynamoDB 全局表的直播策略中：

```
{
   "Statement":[
      {
         "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxx"
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
         },
         "Action":[
            "dynamodb:GetRecords",
```

```
            "dynamodb:DescribeStream",
            "dynamodb:GetShardIterator"
         ],
         "Resource":"arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable/
stream/2023-08-31T09:50:24.025",
         "Condition": {
            "DateLessThan": {
            "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
         }
      }
   ]
}
```

如果目标表或流没有任何附加的资源策略，则会在实验期间创建资源策略，并在实验结束时自动删除。否则，错误语句将插入到现有策略中，而无需对现有策略语句进行任何其他修改。然后，在实验结束时，错误声明将从策略中删除。

资源类型

- aws:dynamodb:global-table

参数

- duration— 在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- dynamodb:PutResourcePolicy
- dynamodb:DeleteResourcePolicy
- dynamodb:GetResourcePolicy
- dynamodb:DescribeTable
- tag:GetResources

Amazon EBS 操作

AWS FIS 支持以下 Amazon EBS 操作。

aws:ebs:pause-volume-io

暂停 EBS 目标卷上的 I/O 操作。目标卷必须位于同一可用区内，并且必须附加到构建在 Nitro 系统上的实例。此卷无法附加到 Outpost 上的实例。

要使用 Amazon EC2 控制台启动实验，请参阅 Amazon EC2 用户指南中的 Amazon EBS 故障测试。

资源类型

- aws:ec2:ebs-volume

参数

- duration：持续时间，从一秒到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟，PT5S 代表五秒，PT6H 代表六小时。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。如果持续时间很短（如 PT5S），则在指定持续时间内暂停 I/O，但由于初始化实验需要时间，因此可能需要更长时间才能完成实验。

权限

- `ec2:DescribeVolumes`
- `ec2:PauseVolumeIO`
- `tag:GetResources`

Amazon EC2 操作

AWS FIS 支持以下 Amazon EC2 操作。

操作

- aws:ec2:api-insufficient-instance-capacity-error
- aws:ec2:asg-insufficient-instance-capacity-error
- aws:ec2:reboot-instances
- aws:ec2:send-spot-instance-interruptions
- aws:ec2:stop-instances
- aws:ec2:terminate-instances

AWS FIS 还支持通过 AWS Systems Manager SSM 代理进行故障注入操作。Systems Manager 使用的 SSM 文档定义了要在 EC2 实例上执行的操作。您可以使用自有文档注入自定义故障，也可以使用预配置的 SSM 文档。有关更多信息，请参阅 the section called “使用 SSM 文档”。

aws:ec2:api-insufficient-instance-capacity-error

对目标 IAM 角色发出的请求注入 `InsufficientInstanceCapacity` 错误响应。支持的操作是 RunInstances、、 CreateCapacityReservation StartInstances、 CreateFleet 调用。不支持包含在多个可用区中询问容量的请求。此操作不支持使用资源标签、筛选条件或参数定义目标。

资源类型

- aws:iam:role

参数

- duration— 在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。
- availabilityzoneIdentifiers：以逗号分隔的可用区列表。支持区域 ID（例如 `"use1-az1, use1-az2"`）和区域名称（例如 `"us-east-1a"`）。
- percentage：注入故障信息的调用百分比 (1 - 100)。

权限

- `ec2:InjectApiError`，条件键 `ec2:FisActionId` 值设置为 `aws:ec2:api-insufficient-instance-capacity-error`，`ec2:FisTargetArns` 条件键设置为目标 IAM 角色。

有关策略示例，请参阅示例：使用 `ec2:InjectApiError` 条件键。

aws:ec2:asg-insufficient-instance-capacity-error

对目标自动扩缩组发出的请求注入 `InsufficientInstanceCapacity` 错误响应。此操作仅支持使用启动模板的自动扩缩组。要了解有关实例容量不足错误的更多信息，请参阅 Amazon EC2 用户指南。

资源类型

- aws:ec2:autoscaling-group

参数

- duration— 在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。
- availabilityzoneIdentifiers：以逗号分隔的可用区列表。支持区域 ID（例如 "use1-az1, use1-az2"）和区域名称（例如 "us-east-1a"）。
- percentage：可选。目标自动扩缩组的启动请求中注入故障的百分比（1-100）。默认值为 100。

权限

- ec2:InjectApiError条件键 ec2: FisActionId 值设置为，aws:ec2:asg-insufficient-instance-capacity-errorec2:FisTargetArns条件键设置为目标 Auto Scaling 组。
- autoscaling:DescribeAutoScalingGroups

有关策略示例，请参阅示例：使用 ec2:InjectApiError 条件键。

aws:ec2:reboot-instances

在目标 EC2 实例RebootInstances上运行 Amazon EC2 API 操作。

资源类型

- aws:ec2:instance

参数

- 无

权限

- ec2:RebootInstances
- ec2:DescribeInstances

AWS 托管策略

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:send-spot-instance-interruptions

中断目标竞价型实例。在中断前两分钟，向目标竞价型实例发送竞价型实例中断通知。中断时间由指定的持续时间BeforeInterruption参数确定。在中断后两分钟，竞价型实例会终止或停止，具体取决于中断行为。由 AWS FIS 停止的竞价型实例会一直保持停止状态，直至重启。

启动操作后，目标实例会立刻收到一条 重新平衡 EC2 实例的建议。如果您指定了持续时间 BeforeInterruption，则在再平衡建议和中断通知之间可能会有延迟。

有关更多信息，请参阅 the section called “测试竞价型实例中断情况”。或者，要使用 Amazon EC2 控制台启动实验，请参阅 Amazon EC2 用户指南中的启动竞价型实例中断。

资源类型

- aws:ec2:spot-instance

参数

- durationBeforeInterruption：中断实例前的等待时间，从 2 到 15 分钟不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT2M 代表两分钟。在 AWS FIS 控制台中，您可以输入分钟数。

权限

- `ec2:SendSpotInstanceInterruptions`
- `ec2:DescribeInstances`

AWS 托管策略

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:stop-instances

在目标 EC2 实例StopInstances上运行 Amazon EC2 API 操作。

资源类型

- aws:ec2:instance

参数

- startInstancesAfterDuration：可选。启动实例前的等待时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。您可以在 AWS FIS 控制台中输入秒数、分钟数或小时数。如果实例具有加密的 EBS 卷，则必须向用于加密该卷的 KMS 密钥授予 AWS FIS 权限，或者将实验角色添加到 KMS 密钥策略中。
- completeIfInstancesTerminated：可选。如果为 true，并且 `startInstancesAfterDuration` 也为 true，则当目标 EC2 实例被 FIS 之外的单独请求终止且无法重启时，此操作不会失败。例如，在此操作完成之前，自动扩缩组可能会终止其控制下的已停止 EC2 实例。默认值为 false。

权限

- `ec2:StopInstances`
- `ec2:StartInstances`
- `ec2:DescribeInstances`：可选。需要使用 “完成IfInstances终止” 来验证操作结束时的实例状态。
- `kms:CreateGrant`：可选。需要使用启动InstancesAfter持续时间来重启带有加密卷的实例。

AWS 托管策略

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:terminate-instances

在目标 EC2 实例TerminateInstances上运行 Amazon EC2 API 操作。

资源类型

- aws:ec2:instance

参数

- 无

权限

- `ec2:TerminateInstances`

- `ec2:DescribeInstances`

AWS 托管策略

- AWSFaultInjectionSimulatorEC2Access

Amazon ECS 操作

AWS FIS 支持以下 Amazon ECS 操作。

操作

- aws:ecs:drain-container-instances
- aws:ecs:stop-task
- aws:ecs:task-cpu-stress
- aws:ecs:task-io-stress
- aws:ecs:task-kill-process
- aws:ecs:task-network-blackhole-port
- aws:ecs:task-network-latency
- aws:ecs:task-network-packet-loss

aws:ecs:drain-container-instances

运行 Amazon ECS API 操作UpdateContainerInstancesState以耗尽目标集群上指定百分比的底层 Amazon EC2 实例。

资源类型

- aws:ecs:cluster

参数

- drainagePercentage：百分比 (1 - 100)。
- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- ecs:DescribeClusters
- ecs:UpdateContainerInstancesState
- ecs:ListContainerInstances
- tag:GetResources

AWS 托管策略

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:stop-task

运行 Amazon ECS API 操作StopTask以停止目标任务。

资源类型

- aws:ecs:task

参数

- 无

权限

- ecs:DescribeTasks
- ecs:ListTasks
- ecs:StopTask
- tag:GetResources

AWS 托管策略

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:task-cpu-stress

在目标任务上运行 CPU 压力测试。使用 AWSFIS-run-CPU-stres s SSM 文档。任务必须由管理 AWS Systems Manager。有关更多信息，请参阅 执行 ECS 任务操作。

资源类型

- aws:ecs:task

参数

- duration：压力测试的持续时间，格式为 ISO 8601。
- percent：可选。目标负载百分比，从 0（空载）到 100（满载）不等。默认值为 100。
- workers：可选。要使用的压力源数量。默认为 0，使用所有压力源。
- installDependencies：可选。如果值为 `True`，则 Systems Manager 会在 sidecar 容器上为 SSM 代理安装必要依赖项（如未安装）。默认值为 `True`。依赖项为 stress-ng。

权限

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-io-stress

在目标任务上运行 I/O 压力测试。使用 AWSFIS-run-io-stress SS M 文档。任务必须由管理 AWS Systems Manager。有关更多信息，请参阅 执行 ECS 任务操作。

资源类型

- aws:ecs:task

参数

- duration：压力测试的持续时间，格式为 ISO 8601。
- percent：可选。在压力测试期间，文件系统上使用的空闲空间百分比。默认为 80%。

- workers：可选。工作程序数量。工作程序会混合执行读/写操作（顺序操作、随机操作和内存映射操作）、强制同步操作和缓存删除操作。多个子进程对同一个文件执行不同的 I/O 操作。默认 为 1。
- installDependencies：可选。如果值为 `True`，则 Systems Manager 会在 sidecar 容器上为 SSM 代理安装必要依赖项（如未安装）。默认值为 `True`。依赖项为 stress-ng。

权限

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-kill-process

运行 killall 命令，停止任务中的指定进程。使用 AWSFIS-Run-Kill-Pro cess SSM 文档。必须在任务定义中将 `pidMode` 设为 `task`。任务必须由管理 AWS Systems Manager。有关更多信息，请参阅 执行 ECS 任务操作。

资源类型

- aws:ecs:task

参数

- processName：待停止进程的名称。
- signal：可选。要与命令一起发送的信号。这些值可能是 `SIGTERM`（接收者可以忽略的值）和 `SIGKILL`（接收者不能忽略的值）。默认值为 `SIGTERM`。
- installDependencies – 可选。如果值为 `True`，则 Systems Manager 会在 sidecar 容器上为 SSM 代理安装必要依赖项（如未安装）。默认值为 `True`。依赖项为 killall。

权限

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-blackhole-port

丢弃指定协议和端口的入站或出站流量。使用 AWSFIS-Run-Network-Blackhole-Port SSM 文档。必须在任务定义中将 pidMode 设为 task。任务必须由管理 AWS Systems Manager。您不能在任务定义中将 networkMode 设为 bridge。有关更多信息，请参阅 执行 ECS 任务操作。

资源类型

- aws:ecs:task

参数

- duration：测试的持续时间，格式为 ISO 8601。
- port：端口号。
- trafficType：流量类型。可能的值为 ingress 和 egress。
- protocol：可选。协议。可能的值为 tcp 和 udp。默认为 tcp。
- installDependencies – 可选。如果值为 True，则 Systems Manager 会在 sidecar 容器上为 SSM 代理安装必要依赖项（如未安装）。默认值为 True。依赖项分别是 atd、dig 和 iptables。

权限

- ssm:SendCommand
- ssm:ListCommands
- ssm:CancelCommand

aws:ecs:task-network-latency

使用 tc 工具，为往返于特定来源的流量添加网络接口的延迟和抖动。使用 AWSFIS-Run-Network-Latency-S ources SSM 文档。必须在任务定义中将 pidMode 设为 task。任务必须由管理 AWS Systems Manager。您不能在任务定义中将 networkMode 设为 bridge。有关更多信息，请参阅 执行 ECS 任务操作。

资源类型

- aws:ecs:task

参数

- duration：测试的持续时间，格式为 ISO 8601。
- interface：可选。网络接口。默认值为 `eth0`。
- delayMilliseconds – 可选。延迟（单位：毫秒）。默认为 200。
- jitterMilliseconds：可选。抖动（单位：毫秒）。默认值为 10。
- sources：可选。来源，用逗号分隔。值可能是：IPv4 地址、IPv4 CIDR 块、域名、`DYNAMODB` 和 `S3`。如果指定值为 `DYNAMODB` 或 `S3`，则仅适用于当前区域中的区域端点。默认为 0.0.0.0/0，可匹配所有 IPv4 流量。
- installDependencies：可选。如果值为 `True`，则 Systems Manager 会在 sidecar 容器上为 SSM 代理安装必要依赖项（如未安装）。默认值为 `True`。依赖项分别是 atd、dig、jq 和 tc。

权限

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-packet-loss

使用 tc 工具，为网络接口添加丢包。使用 AWSFIS-Run-Network-Packet-Loss-Sour ces SSM 文档。必须在任务定义中将 `pidMode` 设为 `task`。任务必须由管理 AWS Systems Manager。您不能在任务定义中将 `networkMode` 设为 `bridge`。有关更多信息，请参阅 执行 ECS 任务操作。

资源类型

- aws:ecs:task

参数

- duration：测试的持续时间，格式为 ISO 8601。
- interface：可选。网络接口。默认值为 `eth0`。
- lossPercent – 可选。丢包率。默认为 7%。

- sources：可选。来源，用逗号分隔。值可能是：IPv4 地址、IPv4 CIDR 块、域名、DYNAMODB 和 S3。如果指定值为 DYNAMODB 或 S3，则仅适用于当前区域中的区域端点。默认为 0.0.0.0/0，可匹配所有 IPv4 流量。
- installDependencies：可选。如果值为 `True`，则 Systems Manager 会在 sidecar 容器上为 SSM 代理安装必要依赖项（如未安装）。默认值为 `True`。依赖项分别是 atd、dig、jq 和 tc。

权限

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

Amazon EKS 操作

AWS FIS 支持以下 Amazon EKS 操作。

操作

- aws:eks:inject-kubernetes-custom-resource
- aws:eks:pod-cpu-stress
- aws:eks:pod-delete
- aws:eks:pod-io-stress
- aws:eks:pod-memory-stress
- aws:eks:pod-network-blackhole-port
- aws:eks:pod-network-latency
- aws:eks:pod-network-packet-loss
- aws:eks:terminate-nodegroup-instances

aws:eks:inject-kubernetes-custom-resource

在单个目标集群上运行 ChaosMesh 或 Litmus 实验。您必须在目标集群上安装 ChaosMesh 或 Litmus。

在创建实验模板并定义 `aws:eks:cluster` 的目标类型时，请务必将此操作定位到单个 Amazon 资源名称 (ARN) 。此操作不支持使用资源标签、筛选条件或参数定义目标。

安装时 ChaosMesh，必须指定相应的容器运行时。从 Amazon EKS 版本 1.23 起，将默认运行时从 Docker 更改为 containerd。从版本 1.24 版本起，删除 Docker。

资源类型

- aws:eks:cluster

参数

- kubernetesApiVersion：Kubernetes 自定义资源的 API 版本。值可能是 `chaos-mesh.org/v1alpha1` | `litmuschaos.io/v1alpha1`。
- kubernetesKind：Kubernetes 自定义资源类型。此值取决于 API 版本。
  - `chaos-mesh.org/v1alpha1`：值可能是 `AWSChaos` | `DNSChaos` | `GCPChaos` | `HTTPChaos` | `IOChaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodIOChaos` | `PodNetworkChaos` | `Schedule` | `StressChaos` | `TimeChaos` |
  - `litmuschaos.io/v1alpha1`：值可能是 `ChaosEngine`。
- kubernetesNamespace：Kubernetes 命名空间
- kubernetesSpec：Kubernetes 自定义资源的 `spec` 部分，格式为 JSON。
- maxDuration：允许完成自动化执行的最长时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

此操作不需要 AWS 身份和访问管理 (IAM) 权限。通过 RBAC 授权，Kubernetes 可控制执行此操作所需的权限。有关更多信息，请参阅 Kubernetes 官方文档中的使用 RBAC 授权。有关 Chaos Mesh 的更多信息，请参阅 Chaos Mesh 官方文档。有关 Litmus 的更多信息，请参阅 Litmus 官方文档。

aws:eks:pod-cpu-stress

在目标容器组（pod）上运行 CPU 压力测试。有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- duration：压力测试的持续时间，格式为 ISO 8601。
- percent：可选。目标负载百分比，从 0（空载）到 100（满载）不等。默认值为 100。
- workers：可选。要使用的压力源数量。默认为 0，使用所有压力源。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。
- fisPodSecurityPolicy：可选。用于 FIS 和临时容器创建的故障编排容器的 Kubernetes 安全标准策略。可能的值为`privileged`、`baseline`和`restricted`。此操作与所有策略级别兼容。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-delete

删除目标容器组（pod）。有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- gracePeriodSeconds：可选。等待容器组（pod）正常终止的持续时间（单位：秒）。如果值为 0，则立即执行操作。如果值为 nil，则使用容器组（pod）的默认宽限期。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。
- fisPodSecurityPolicy：可选。用于 FIS 和临时容器创建的故障编排容器的 Kubernetes 安全标准策略。可能的值为`privileged`、`baseline`和`restricted`。此操作与所有策略级别兼容。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-io-stress

在目标容器组（pod）上运行 I/O 压力测试。有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- duration：压力测试的持续时间，格式为 ISO 8601。

- workers：可选。工作程序数量。工作程序会混合执行读/写操作（顺序操作、随机操作和内存映射操作）、强制同步操作和缓存删除操作。多个子进程对同一个文件执行不同的 I/O 操作。默认 为 1。
- percent：可选。在压力测试期间，文件系统上使用的空闲空间百分比。默认为 80%。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。
- fisPodSecurityPolicy：可选。用于 FIS 和临时容器创建的故障编排容器的 Kubernetes 安全标准策略。可能的值为`privileged`、`baseline`和`restricted`。此操作与所有策略级别兼容。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-memory-stress

在目标容器组（pod）上运行内存压力测试。有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- duration：压力测试的持续时间，格式为 ISO 8601。
- workers：可选。要使用的压力源数量。默认 为 1。

- percent：可选。压力测试期间要使用的虚拟内存百分比。默认为 80%。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。
- fisPodSecurityPolicy：可选。用于 FIS 和临时容器创建的故障编排容器的 Kubernetes 安全标准策略。可能的值为`privileged`、`baseline`和`restricted`。此操作与所有策略级别兼容。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-blackhole-port

丢弃指定协议和端口的入站或出站流量。仅与 Kubernetes 安全标准政策兼容。`privileged`有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- duration：测试的持续时间，格式为 ISO 8601。
- protocol：可选。协议。可能的值为 `tcp` 和 `udp`。默认为 `tcp`。
- trafficType：流量类型。可能的值为 `ingress` 和 `egress`。

- port：端口号。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-latency

使用 tc 工具，为往返于特定来源的流量添加网络接口的延迟和抖动。仅与 Kubernetes 安全标准政策兼容。`privileged`有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- duration：测试的持续时间，格式为 ISO 8601。
- interface：可选。网络接口。默认值为 `eth0`。
- delayMilliseconds – 可选。延迟（单位：毫秒）。默认为 200。
- jitterMilliseconds：可选。抖动（单位：毫秒）。默认值为 10。

- sources：可选。来源，用逗号分隔。值可能是：IPv4 地址、IPv4 CIDR 块、域名、DYNAMODB 和 S3。如果指定值为 DYNAMODB 或 S3，则仅适用于当前区域中的区域端点。默认为 0.0.0.0/0，可匹配所有 IPv4 流量。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-packet-loss

使用 tc 工具，为网络接口添加丢包。仅与 Kubernetes 安全标准政策兼容。`privileged`有关更多信息，请参阅 执行 EKS 容器组（pod）操作。

资源类型

- aws:eks:pod

参数

- duration：测试的持续时间，格式为 ISO 8601。
- interface：可选。网络接口。默认值为 `eth0`。
- lossPercent – 可选。丢包率。默认为 7%。

- sources：可选。来源，用逗号分隔。值可能是：IPv4 地址、IPv4 CIDR 块、域名、DYNAMODB 和 S3。如果指定值为 DYNAMODB 或 S3，则仅适用于当前区域中的区域端点。默认为 0.0.0.0/0，可匹配所有 IPv4 流量。
- kubernetesServiceAccount：Kubernetes 服务账户。有关所需权限的信息，请参阅the section called “配置 Kubernetes 服务账户”。
- fisPodContainerImage：可选。用于创建故障注入容器组（pod）的容器映像。默认使用提供的图像 AWS FIS。有关更多信息，请参阅 the section called “容器组（pod）容器映像”。
- maxErrorsPercent – 可选。可能比故障注入更早失败的目标的百分比。默认值是 0。
- fisPodLabels：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 标签。
- fisPodAnnotations：可选。附加到 FIS 创建的故障编排容器上的 Kubernetes 注释。

权限

- `eks:DescribeCluster`
- `ec2:DescribeSubnets`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:terminate-nodegroup-instances

在目标节点组TerminateInstances上运行 Amazon EC2 API 操作。

资源类型

- aws:eks:nodegroup

参数

- instanceTerminationPercentage：要终止的实例百分比 (1 - 100)。

权限

- `ec2:DescribeInstances`

- ec2:TerminateInstances
- eks:DescribeNodegroup
- tag:GetResources

AWS 托管策略

- AWSFaultInjectionSimulatorEKSAccess

亚马逊的 ElastiCache 行动

AWS FIS 支持以下 ElastiCache 操作。

aws:elasticache:interrupt-cluster-az-power

中断目标 Redis 复制组中指定可用区内节点的电力。如果将某个主节点确定为目标，则复制滞后时间最短的相应只读副本将被提升为主节点。在此操作期间，指定可用区内的只读副本替换将被阻止，这意味着目标复制组的运行容量会降低。

资源类型

- aws:elasticache:redis-replicationgroup

参数

- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- elasticache:InterruptClusterAzPower
- elasticache:DescribeReplicationGroups
- tag:GetResources

网络操作

AWS FIS 支持以下网络操作。

操作

- aws:network:disrupt-connectivity
- aws:network:route-table-disrupt-cross-region-connectivity
- aws:network:transit-gateway-disrupt-cross-region-connectivity

aws:network:disrupt-connectivity

拒绝向目标子网发送指定流量。使用网络 ACL。

资源类型

- aws:ec2:subnet

参数

- scope：要拒绝的流量类型。如果范围不是`all`，则网络 ACL 中的最大条目数为 20。可能的值包括：
  - `all`：拒绝所有往返于子网的流量。请注意，此选项允许子网内流量，包括往返于子网中网络接口的流量。
  - `availability-zone`：拒绝往返于其他可用区子网的 VPC 内流量。VPC 中可以定位的最大子网数量为 30。
  - `dynamodb`：拒绝往返于当前区域中 DynamoDB 区域端点的流量。
  - `prefix-list`：拒绝往返于指定前缀列表的流量。
  - `s3`：拒绝往返于当前区域中 Amazon S3 区域端点的流量。
  - `vpc`：拒绝往返于 VPC 的流量。
- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。
- prefixListIdentifier：如果范围为 `prefix-list`，即为客户托管前缀列表的标识符。您可以指定名称、ID 或 ARN。前缀列表最多可以有 10 个条目。

权限

- `ec2:CreateNetworkAcl`：创建带有 managedByFIS=true 标签的网络 ACL。
- `ec2:CreateNetworkAclEntry`：网络 ACL 必须带有 managedByFIS=true 标签。
- `ec2:CreateTags`

- ec2:DeleteNetworkAcl：网络 ACL 必须带有 managedByFIS=true 标签。
- ec2:DescribeManagedPrefixLists
- ec2:DescribeNetworkAcls
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- ec2:GetManagedPrefixListEntries
- ec2:ReplaceNetworkAclAssociation

AWS 托管策略

- AWSFaultInjectionSimulatorNetworkAccess

aws:network:route-table-disrupt-cross-region-connectivity

阻止源自目标子网并发往指定区域的流量。创建包含要隔离的区域的所有路由的路由表。要允许 FIS 创建这些路由表，请将 Amazon VPC 配额提高`routes per route table`到 250 加上现有路由表中的路由数量。

资源类型

- aws:ec2:subnet

参数

- region：要隔离的区域的代码（例如，eu-west-1）。
- duration：操作持续的时间长度。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- ec2:AssociateRouteTable
- ec2:CreateManagedPrefixList †
- ec2:CreateNetworkInterface †
- ec2:CreateRoute †
- ec2:CreateRouteTable †

- ec2:CreateTags †
- ec2:DeleteManagedPrefixList †
- ec2:DeleteNetworkInterface †
- ec2:DeleteRouteTable †
- ec2:DescribeManagedPrefixLists
- ec2:DescribeNetworkInterfaces
- ec2:DescribeRouteTables
- ec2:DescribeSubnets
- ec2:DescribeVpcPeeringConnections
- ec2:DescribeVpcs
- ec2:DisassociateRouteTable
- ec2:GetManagedPrefixListEntries
- ec2:ModifyManagedPrefixList †
- ec2:ModifyVpcEndpoint
- ec2:ReplaceRouteTableAssociation

† 使用标签 managedByFIS=true 限定范围。

AWS 托管策略

- AWSFaultInjectionSimulatorNetworkAccess

aws:network:transit-gateway-disrupt-cross-region-connectivity

阻止源自目标中转网关对等连接并发往指定区域的流量。

资源类型

- aws:ec2:transit-gateway

参数

- region：要隔离的区域的代码（例如，eu-west-1）。
- duration：操作持续的时间长度。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- ec2:AssociateTransitGatewayRouteTable
- ec2:DescribeTransitGatewayAttachments
- ec2:DescribeTransitGatewayPeeringAttachments
- ec2:DescribeTransitGateways
- ec2:DisassociateTransitGatewayRouteTable

AWS 托管策略

- AWSFaultInjectionSimulatorNetworkAccess

Amazon RDS 操作

AWS FIS 支持以下 Amazon RDS 操作。

操作

- aws:rds:failover-db-cluster
- aws:rds:reboot-db-instances

aws:rds:failover-db-cluster

在目标 Aurora 数据库集群上运行 Amazon RDS API 操作 FailoverDBCluster。

资源类型

- aws:rds:cluster

参数

- 无

权限

- rds:FailoverDBCluster
- rds:DescribeDBClusters

- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorRDSAccess

aws:rds:reboot-db-instances

在目标数据库实例上运行 Amazon RDS API 操作 RebootDBInstance。

资源类型

- aws:rds:db

参数

- forceFailover：可选。如果值为 True，且实例为多可用区，则强制从一个可用区故障转移到另一个可用区。默认值为 false。

权限

- `rds:RebootDBInstance`
- `rds:DescribeDBInstances`
- `tag:GetResources`

AWS 托管策略

- AWSFaultInjectionSimulatorRDSAccess

Amazon S3 操作

AWS FIS 支持以下 Amazon S3 操作。

操作

- aws:s3:bucket-pause-replication

aws:s3:bucket-pause-replication

暂停从目标源存储桶到目的地存储桶的复制。目的地存储桶可以位于不同的 AWS 区域，也可以与源存储桶位于同一区域内。操作开始后，现有对象可能会继续复制最多一个小时。此操作仅支持按标签确定目标。要了解有关 Amazon S3 复制的更多信息，请参阅 Amazon S3 用户指南。

资源类型

- aws:s3:bucket

参数

- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。
- region：目的地存储桶所在的 AWS 区域。
- destinationBuckets：可选。以逗号分隔的目的地 S3 存储桶列表。
- prefixes：可选。复制规则筛选条件中以逗号分隔的 S3 对象键前缀列表。使用基于前缀的筛选条件的目标存储桶复制规则将暂停。

权限

- `S3:PutReplicationConfiguration`，条件键 `S3:IsReplicationPauseRequest` 设置为 `True`
- `S3:GetReplicationConfiguration`，条件键 `S3:IsReplicationPauseRequest` 设置为 `True`
- `S3:PauseReplication`
- `S3:ListAllMyBuckets`
- `tag:GetResources`

有关策略示例，请参阅示例：使用 `aws:s3:bucket-pause-replication` 条件键。

Systems Manager 操作

AWS FIS 支持以下 Systems Manager 操作。

操作

- aws:ssm:send-command
- aws:ssm:start-automation-execution

aws:ssm:send-command

在目标 EC2 实例SendCommand上运行 Systems Manager API 操作。Systems Manager 文档（SSM 文档）定义其在您实例上执行的操作。有关更多信息，请参阅 执行 aws:ssm:send-command 操作。。

资源类型

- aws:ec2:instance

参数

- documentArn：文档的 Amazon 资源名称 (ARN)。在控制台中，如果您从 “操作类型” 中选择一个与预先配置的 AWS FIS SSM 文档相对应的值，则会为您完成此参数。
- documentVersion：可选。文档版本。如果为空，则运行默认版本。
- documentParameters：此操作设有条件。文档接受的必要参数和可选参数。对象格式为 JSON 格式，密钥为字符串，值为字符串值或字符串数组。
- duration：持续时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

AWS 托管策略

- AWSFaultInjectionSimulatorEC2Access

aws:ssm:start-automation-execution

运行 Systems Manager API 操作StartAutomation执行。

资源类型

- 无

参数

- documentArn：自动化文档的 Amazon 资源名称 (ARN) 。
- documentVersion：可选。文档版本。如果为空，则运行默认版本。
- documentParameters：此操作设有条件。文档接受的必要参数和可选参数。对象格式为 JSON 格式，密钥为字符串，值为字符串值或字符串数组。
- maxDuration：允许完成自动化执行的最长时间，从一分钟到 12 小时不等。在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

权限

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `iam:PassRole`：可选。如果自动化文档代入角色，则为必填项。

AWS 托管策略

- AWSFaultInjectionSimulatorSSMAccess

将 Systems Manager SSM 文档与 FIS 一起使用 AWS

AWS FIS 通过 AWS Systems Manager SSM 代理和 AWS FIS 操作支持自定义故障类型。aws:ssm:send-command可用于创建常见故障注入操作的预配置的 Systems Manager SSM 文档（SSM 文档）可作为以-前缀开头的公共 AWS 文档提供。 AWSFIS

SSM 代理是一种 Amazon 软件，可以在 Amazon EC2 实例、本地服务器或虚拟机 (VM) 上安装和配置。因此能将资源托管在 Systems Manager 上。代理负责处理 Systems Manager 中的请求，然后再按照指定方式运行。您可以加入自己的 SSM 文档，以注入自定义故障，也可以参考 Amazon 拥有的某个公共文档。

要求

对于需要使用 SSM 代理才能在目标上执行的操作，请确保满足以下条件：

- 已为目标安装代理。SSM 代理已经默认安装在亚马逊机器映像 (AMI) 上。否则，您可以为实例安装 SSM 代理。有关更多信息，请参阅 AWS Systems Manager 用户指南中的手动安装适用于 EC2 实例的 SSM 代理。
- Systems Manager 有权对您的实例执行操作。您可以通过 IAM 实例配置文件授予访问权限。有关更多信息，请参阅 AWS Systems Manager 用户指南中的创建适用于 Systems Manager 的 IAM 实例配置文件以及将 IAM 实例配置文件附加到 EC2 实例。

执行 aws:ssm:send-command 操作。

SSM 文档定义 Systems Manager 对您的托管实例执行的操作。Systems Manager 包含大量预配置文档，您也可以自行创建文档。有关自行创建 SSM 文档的更多信息，请参阅 AWS Systems Manager 用户指南中的创建 Systems Manager 文档。有关 SSM 常规文档的更多信息，请参阅 AWS Systems Manager 用户指南中的 AWS Systems Manager 文档。

AWS FIS 提供预先配置的 SSM 文档。你可以在 AWS Systems Manager 控制台的 “文档” 下查看预配置的 SSM 文档：https://console.aws.amazon.com/systems-manager/documents。您也可以从 AWS FIS 控制台中的一系列预配置文档中进行选择。有关更多信息，请参阅 预先配置的 AWS FIS SSM 文档。

要在 AWS FIS 实验中使用 SSM 文档，您可以使用操作。aws:ssm:send-command此操作会在您目标实例上获取指定 SSM 文档并运行。

在实验模板中执行 `aws:ssm:send-command` 操作时，必须为此操作指定其他参数，包括以下参数：

- documentArn – 必需。SSM 文档的 Amazon 资源名称 (ARN) 。
- documentParameters：此操作设有条件。SSM 文档接受的必要参数和可选参数。对象格式为 JSON 格式，密钥为字符串，值为字符串值或字符串数组。
- documentVersion：可选。要运行的 SSM 文档版本。

您可以通过 Systems Manager 控制台或命令行查看 SSM 文档信息（包括文档参数）。

使用控制台查看有关 SSM 文档的信息

1. 打开 AWS Systems Manager 控制台，网址为 https://console.aws.amazon.com/systems-manager/。

2. 在导航窗格中，选择文档。
3. 选择文档，然后选择详细信息选项卡。

运行命令行，查看有关 SSM 文档的信息

使用 SSM 描述的文档命令。

预先配置的 AWS FIS SSM 文档

您可以将预先配置 AWS 的 FIS SSM 文档与实验模板中的`aws:ssm:send-command`操作配合使用。

要求

- 只有以下操作系统支持 AWS FIS 提供的预配置 SSM 文档：
  - Amazon Linux 2023、Amazon Linux 2 和 Amazon Linux
  - Ubuntu
  - RHEL 7、8 和 9
  - CentOS 7、8 和 9
- 仅在 EC2 实例上支持 AWS FIS 提供的预配置 SSM 文档。其他类型的托管节点（如本地服务器）并不支持此类文档。

要在 ECS 任务实验中使用这些 SSM 文档，请使用相应的 the section called “Amazon ECS 操作”。例如，对 aws:ecs:task-cpu-stress 操作使用 AWSFIS-Run-CPU-Stress 文档。

文档

- AWSFIS-Run-CPU-Stress
- AWSFIS-Run-Disk-Fill
- AWSFIS-Run-IO-Stress
- AWSFIS-Run-Kill-Process
- AWSFIS-Run-Memory-Stress
- AWSFIS-Run-Network-Blackhole-Port
- AWSFIS-Run-Network-Latency
- AWSFIS-Run-Network-Latency-Sources
- AWSFIS-Run-Network-Packet-Loss
- AWSFIS-Run-Network-Packet-Loss-Sources

AWSFIS-Run-CPU-Stress

使用 stress-ng 工具在实例上运行 CPU 压力测试。使用 AWSFIS-run-CPU-stres s SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-CPU-Stress

ARN

arn:aws:ssm:region::document/AWSFIS-Run-CPU-Stress

文档参数

- DurationSeconds – 必需。CPU 压力测试的持续时间（单位：秒）。
- CPU：可选。要使用的 CPU 压力源数量。默认为 0，使用所有 CPU 压力源。
- LoadPercent：可选。目标 CPU 负载百分比，从 0（空载）到 100（满载）不等。默认值为 100。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 `True`。依赖项为 stress-ng。

以下字符串示例可输入控制台。

```
{"DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Disk-Fill

在实例根卷上分配磁盘空间，模拟磁盘已满的故障。使用 AWSFIS-Run-Disk-Fill SSM 文档。

如果注入此故障的实验已停止，无论是手动还是通过停止条件停止， AWS FIS 都会尝试通过取消正在运行的 SSM 文档来回滚。但如果因为故障或因为故障及应用程序活动导致磁盘空间已满，则 Systems Manager 可能无法完成取消操作。此时，如果要停止实验，请确保仍有磁盘空间。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Disk-Fill

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Disk-Fill

文档参数

- DurationSeconds – 必需。磁盘填充测试的持续时间（单位：秒）。
- Percent：可选。在磁盘填充测试期间分配的磁盘百分比。默认为 95%。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 `True`。依赖项分别是 atd 和 fallocate。

以下字符串示例可输入控制台。

```
{"DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-IO-Stress

使用 stress-ng 工具在实例上运行 IO 压力测试。使用 AWSFIS-run-io-stress SS M 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-IO-Stress

ARN

arn:aws:ssm:region::document/AWSFIS-Run-IO-Stress

文档参数

- DurationSeconds – 必需。IO 压力测试的持续时间（单位：秒）。
- Workers：可选。混合执行读/写操作（顺序操作、随机操作和内存映射操作）、强制同步操作和缓存删除操作的工作程序数量。多个子进程对同一个文件执行不同的 I/O 操作。默认 为 1。
- Percent：可选。在 IO 压力测试期间，文件系统上使用的空闲空间百分比。默认为 80%。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 `True`。依赖项为 stress-ng。

以下字符串示例可输入控制台。

```
{"Workers":"1", "Percent":"80", "DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Kill-Process

运行 killall 命令，停止实例中的指定进程。使用 AWSFIS-Run-Kill-Pro cess SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Kill-Process

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Kill-Process

文档参数

- ProcessName – 必需。要停止的进程名称。
- Signal：可选。要与命令一起发送的信号。这些值可能是 SIGTERM（接收者可以忽略的值）和 SIGKILL（接收者不能忽略的值）。默认值为 SIGTERM。
- InstallDependencies – 可选。如果值为 True，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 True。依赖项为 killall。

以下字符串示例可输入控制台。

```
{"ProcessName":"myapplication", "Signal":"SIGTERM"}
```

AWSFIS-Run-Memory-Stress

使用 stress-ng 工具在实例上运行内存压力测试。使用 AWSFIS-Run-Memory-Stres s SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Memory-Stress

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Memory-Stress

文档参数

- DurationSeconds – 必需。内存压力测试的持续时间（单位：秒）。
- Workers：可选。虚拟内存压力源的数量。默认 为 1。
- Percent – 必需。在内存压力测试期间要使用的虚拟内存百分比。
- InstallDependencies：可选。如果值为 True，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 True。依赖项为 stress-ng。

以下字符串示例可输入控制台。

```
{"Percent":"80", "DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Blackhole-Port

使用 iptables 工具丢弃协议和端口的入站或出站流量。使用 AWSFIS-Run-Network-Blackhole-Port SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Network-Blackhole-Port

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Blackhole-Port

文档参数

- Protocol – 必需。协议。可能的值为 `tcp` 和 `udp`。
- Port – 必需。端口号。
- TrafficType：可选。流量的类型。可能的值为 `ingress` 和 `egress`。默认为 `ingress`。
- DurationSeconds – 必需。网络黑洞测试的持续时间（单位：秒）。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 `True`。依赖项分别是 atd、dig 和 iptables。

以下字符串示例可输入控制台。

```
{"Protocol":"tcp", "Port":"8080", "TrafficType":"egress", "DurationSeconds":"60",
 "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Latency

使用 tc 工具为网络接口增加延迟。使用 AWSFIS-运行-网络-延迟 SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Network-Latency

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Latency

文档参数

- Interface：可选。网络接口。默认值为 `eth0`。
- DelayMilliseconds – 可选。延迟（单位：毫秒）。默认为 200。
- DurationSeconds – 必需。网络延迟测试的持续时间（单位：秒）。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 `True`。依赖项分别是 atd、dig 和 tc。

以下字符串示例可输入控制台。

```
{"DelayMilliseconds":"200", "Interface":"eth0", "DurationSeconds":"60",
 "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Latency-Sources

使用 tc 工具，为往返于特定来源的流量添加网络接口的延迟和抖动。使用 AWSFIS-Run-Network-Latency-S ources SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Network-Latency-Sources

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Latency-Sources

文档参数

- Interface：可选。网络接口。默认值为 `eth0`。
- DelayMilliseconds – 可选。延迟（单位：毫秒）。默认为 200。
- JitterMilliseconds：可选。抖动（单位：毫秒）。默认值为 10。
- Sources – 必需。来源，用逗号分隔。值可能是：IPv4 地址、IPv4 CIDR 块、域名、`DYNAMODB` 和 `S3`。如果指定值为 `DYNAMODB` 或 `S3`，则仅适用于当前区域中的区域端点。
- TrafficType：可选。流量的类型。可能的值为 `ingress` 和 `egress`。默认为 `ingress`。
- DurationSeconds – 必需。网络延迟测试的持续时间（单位：秒）。

- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项（如未安装）。默认值为 `True`。依赖项分别是 atd、dig、jq 和 tc。

以下字符串示例可输入控制台。

```
{"DelayMilliseconds":"200", "JitterMilliseconds":"15",
 "Sources":"S3,www.example.com,72.21.198.67", "Interface":"eth0",
 "TrafficType":"egress", "DurationSeconds":"60", "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Packet-Loss

使用 tc 工具，为网络接口添加丢包。使用 AWSFIS-Run-Network-Packet-Los s SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Network-Packet-Loss

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Packet-Loss

文档参数

- Interface：可选。网络接口。默认值为 `eth0`。
- LossPercent – 可选。丢包率。默认为 7%。
- DurationSeconds – 必需。网络丢包测试的持续时间（单位：秒）。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项。默认值为 `True`。依赖项分别是 atd、dig 和 tc。

以下字符串示例可输入控制台。

```
{"LossPercent":"15", "Interface":"eth0", "DurationSeconds":"60",
 "InstallDependencies":"True"}
```

AWSFIS-Run-Network-Packet-Loss-Sources

使用 tc 工具，为网络接口添加丢包，以处理往返于特定来源的流量。使用 AWSFIS-Run-Network-Packet-Loss-Sour ces SSM 文档。

操作类型（仅限控制台）

aws:ssm:send-command/AWSFIS-Run-Network-Packet-Loss-Sources

ARN

arn:aws:ssm:region::document/AWSFIS-Run-Network-Packet-Loss-Sources

文档参数

- Interface：可选。网络接口。默认值为 `eth0`。
- LossPercent – 可选。丢包率。默认为 7%。
- Sources – 必需。来源，用逗号分隔。值可能是：IPv4 地址、IPv4 CIDR 块、域名、DYNAMODB 和 S3。如果指定值为 DYNAMODB 或 S3，则仅适用于当前区域中的区域端点。
- TrafficType：可选。流量的类型。可能的值为 `ingress` 和 `egress`。默认为 `ingress`。
- DurationSeconds – 必需。网络丢包测试的持续时间（单位：秒）。
- InstallDependencies：可选。如果值为 `True`，则 Systems Manager 会在目标实例上安装必要依赖项。默认值为 `True`。依赖项分别是 atd、dig、jq 和 tc。

以下字符串示例可输入控制台。

```
{"LossPercent":"15", "Sources":"S3,www.example.com,72.21.198.67", "Interface":"eth0",
 "TrafficType":"egress", "DurationSeconds":"60", "InstallDependencies":"True"}
```

示例

有关实验模板示例，请参阅 the section called “运行预先配置的 AWS FIS SSM 文档”。

有关示例教程，请参阅在实例上运行 CPU 压力测试。

故障排除

请采取以下步骤进行问题排查。

排查 SSM 文档问题

1. 打开 AWS Systems Manager 控制台，网址为 https://console.aws.amazon.com/systems-manager/。
2. 在导航窗格中，依次选择节点管理和运行命令。
3. 在命令历史记录选项卡上，使用筛选条件查找文档运行情况。
4. 选择命令 ID，打开详细信息页面。

5. 选择实例 ID。查看各个步骤的输出结果和错误。

使用 AWS FIS aws: ecs: task 操作

您可以执行 aws:ecs:task 操作，为 Amazon ECS 任务注入故障。

这些操作使用 SSM 代理作为 Sidecar 容器来运行 SSM 文档，这些文档将执行故障注入，并通过 Sidecar 容器将 Amazon ECS 任务注册为 SSM 托管实例。要使用这些操作，您需要更新 Amazon ECS 任务定义，将 SSM 代理添加为 Sidecar 容器，以便将其运行的任务注册为 SSM 托管实例。当您运行 AWS FIS 实验定位时`aws:ecs:task`， AWS FIS 会使用添加到托管实例的资源标签将您在 AWS FIS 实验模板上指定的目标 Amazon ECS 任务映射到一组 SSM 托管实例。`ECS_TASK_ARN`标签值是应执行 SSM 文档的关联 Amazon ECS 任务的 ARN，因此在运行实验时不应删除。

操作

- the section called “aws:ecs:task-cpu-stress”
- the section called “aws:ecs:task-io-stress”
- the section called “aws:ecs:task-kill-process”
- the section called “aws:ecs:task-network-blackhole-port”
- the section called “aws:ecs:task-network-latency”
- the section called “aws:ecs:task-network-packet-loss”

限制

- 以下操作不适用于 AWS Fargate：
  - aws:ecs:task-kill-process
  - aws:ecs:task-network-blackhole-port
  - aws:ecs:task-network-latency
  - aws:ecs:task-network-packet-loss
- 必须先禁用已启用的 ECS Exec，才能执行此类操作。

要求

- 向 AWS FIS 实验角色添加以下权限：

  - ssm:SendCommand
  - ssm:ListCommands
  - ssm:CancelCommand
- 为 Amazon ECS 的任务 IAM 角色添加以下权限：
  - ssm:CreateActivation
  - ssm:AddTagsToResource
  - iam:PassRole

  请注意，您可以将托管实例角色的 ARN 指定为 iam:PassRole 资源。
- 创建 Amazon ECS 任务执行 IAM 角色并添加 A mazonECS TaskExecution RolePolicy 托管策略。
- 为附加到注册为托管实例的任务上的托管实例角色添加以下权限：
  - ssm:DeleteActivation
  - ssm:DeregisterManagedInstance
- 将 AmazonSSM ManagedInstance Core 托管策略添加到注册为托管实例的任务所关联的托管实例角色中。
- 将环境变量 MANAGED_INSTANCE_ROLE_NAME 设置为托管实例角色的名称。
- 为 ECS 任务定义添加 SSM 代理容器。命令脚本将 ECS 任务注册为托管实例。

```
{
    "name": "amazon-ssm-agent",
    "image": "public.ecr.aws/amazon-ssm-agent/amazon-ssm-agent:latest",
    "cpu": 0,
    "links": [],
    "portMappings": [],
    "essential": false,
    "entryPoint": [],
    "command": [
        "/bin/bash",
        "-c",
        "set -e; yum upgrade -y; yum install jq procps awscli -y; term_handler()
 { echo \"Deleting SSM activation $ACTIVATION_ID\"; if ! aws ssm delete-
activation --activation-id $ACTIVATION_ID --region $ECS_TASK_REGION; then
 echo \"SSM activation $ACTIVATION_ID failed to be deleted\" 1>&2; fi;
 MANAGED_INSTANCE_ID=$(jq -e -r .ManagedInstanceID /var/lib/amazon/ssm/registration);
 echo \"Deregistering SSM Managed Instance $MANAGED_INSTANCE_ID\"; if ! aws
 ssm deregister-managed-instance --instance-id $MANAGED_INSTANCE_ID --region
 $ECS_TASK_REGION; then echo \"SSM Managed Instance $MANAGED_INSTANCE_ID
```

```
 failed to be deregistered\" 1>&2; fi; kill -SIGTERM $SSM_AGENT_PID; }; trap
 term_handler SIGTERM SIGINT; if [[ -z $MANAGED_INSTANCE_ROLE_NAME ]]; then
 echo \"Environment variable MANAGED_INSTANCE_ROLE_NAME not set, exiting\"
 1>&2; exit 1; fi; if ! ps ax | grep amazon-ssm-agent | grep -v grep > /dev/
null; then if [[ -n $ECS_CONTAINER_METADATA_URI_V4 ]] ; then echo \"Found ECS
 Container Metadata, running activation with metadata\"; TASK_METADATA=$(curl
 \"${ECS_CONTAINER_METADATA_URI_V4}/task\"); ECS_TASK_AVAILABILITY_ZONE=$(echo
 $TASK_METADATA | jq -e -r '.AvailabilityZone'); ECS_TASK_ARN=$(echo $TASK_METADATA
 | jq -e -r '.TaskARN'); ECS_TASK_REGION=$(echo $ECS_TASK_AVAILABILITY_ZONE | sed
 's/.$//'); ECS_TASK_AVAILABILITY_ZONE_REGEX='^(af|ap|ca|cn|eu|me|sa|us|us-gov)-
(central|north|(north(east|west))|south|south(east|west)|east|west)-[0-9]{1}[a-z]
{1}$'; if ! [[ $ECS_TASK_AVAILABILITY_ZONE =~ $ECS_TASK_AVAILABILITY_ZONE_REGEX ]];
 then echo \"Error extracting Availability Zone from ECS Container Metadata,
 exiting\" 1>&2; exit 1; fi; ECS_TASK_ARN_REGEX='^arn:(aws|aws-cn|aws-us-gov):ecs:
[a-z0-9-]+:[0-9]{12}:task/[a-zA-Z0-9_-]+/[a-zA-Z0-9]+$'; if ! [[ $ECS_TASK_ARN
 =~ $ECS_TASK_ARN_REGEX ]]; then echo \"Error extracting Task ARN from ECS
 Container Metadata, exiting\" 1>&2; exit 1; fi; CREATE_ACTIVATION_OUTPUT=
$(aws ssm create-activation --iam-role $MANAGED_INSTANCE_ROLE_NAME --
tags Key=ECS_TASK_AVAILABILITY_ZONE,Value=$ECS_TASK_AVAILABILITY_ZONE
 Key=ECS_TASK_ARN,Value=$ECS_TASK_ARN Key=FAULT_INJECTION_SIDECAR,Value=true --
region $ECS_TASK_REGION); ACTIVATION_CODE=$(echo $CREATE_ACTIVATION_OUTPUT | jq
 -e -r .ActivationCode); ACTIVATION_ID=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e
 -r .ActivationId); if ! amazon-ssm-agent -register -code $ACTIVATION_CODE -id
 $ACTIVATION_ID -region $ECS_TASK_REGION; then echo \"Failed to register with AWS
 Systems Manager (SSM), exiting\" 1>&2; exit 1; fi; amazon-ssm-agent & SSM_AGENT_PID=
$!; wait $SSM_AGENT_PID; else echo \"ECS Container Metadata not found, exiting\"
 1>&2; exit 1; fi; else echo \"SSM agent is already running, exiting\" 1>&2; exit 1;
 fi"
    ],
    "environment": [
        {
            "name": "MANAGED_INSTANCE_ROLE_NAME",
            "value": "SSMManagedInstanceRole"
        }
    ],
    "environmentFiles": [],
    "mountPoints": [],
    "volumesFrom": [],
    "secrets": [],
    "dnsServers": [],
    "dnsSearchDomains": [],
    "extraHosts": [],
    "dockerSecurityOptions": [],
    "dockerLabels": {},
```

```
    "ulimits": [],
    "logConfiguration": {},
    "systemControls": []
}
```

有关可读性更强的脚本版本，请参阅 the section called “脚本参考版本”。

- 执行 aws:ecs:task-network-blackhole-port、aws:ecs:task-network-latency 和 aws:ecs:task-network-packet-loss 操作时，请务必使用以下选项之一，在 ECS 任务定义中更新 SSM 代理容器。
  - 选项 1：添加特定的 Linux 功能。

```
"linuxParameters": {
    "capabilities": {
        "add": [
            "NET_ADMIN"
        ]
    }
},
```

  - 选项 2：添加所有的 Linux 功能。

```
"privileged": true,
```

- 执行 aws:ecs:task-kill-process、aws:ecs:task-network-blackhole-port、aws:ecs:task-network-latency 和 aws:ecs:task-network-packet-loss 操作时，ECS 任务定义必须将 pidMode 设置为 task。

脚本参考版本

以下是“需求”部分中更具可读性的脚本版本，供您参考。

```
#!/usr/bin/env bash

# This is the activation script used to register ECS tasks as Managed Instances in SSM
# The script retrieves information form the ECS task metadata endpoint to add three
 tags to the Managed Instance
#  - ECS_TASK_AVAILABILITY_ZONE: To allow customers to target Managed Instances / Tasks
 in a specific Availability Zone
#  - ECS_TASK_ARN: To allow customers to target Managed Instances / Tasks by using the
 Task ARN
```

```
#  - FAULT_INJECTION_SIDECAR: To make it clear that the tasks were registered as
 managed instance for fault injection purposes. Value is always 'true'.
# The script will leave the SSM Agent running in the background
# When the container running this script receives a SIGTERM or SIGINT signal, it will
 do the following cleanup:
#  - Delete SSM activation
#  - Deregister SSM managed instance

set -e # stop execution instantly as a query exits while having a non-zero

yum upgrade -y
yum install jq procps awscli -y

term_handler() {
  echo "Deleting SSM activation $ACTIVATION_ID"
  if ! aws ssm delete-activation --activation-id $ACTIVATION_ID --region
 $ECS_TASK_REGION; then
    echo "SSM activation $ACTIVATION_ID failed to be deleted" 1>&2
  fi

  MANAGED_INSTANCE_ID=$(jq -e -r .ManagedInstanceID /var/lib/amazon/ssm/registration)
  echo "Deregistering SSM Managed Instance $MANAGED_INSTANCE_ID"
  if ! aws ssm deregister-managed-instance --instance-id $MANAGED_INSTANCE_ID --region
 $ECS_TASK_REGION; then
    echo "SSM Managed Instance $MANAGED_INSTANCE_ID failed to be deregistered" 1>&2
  fi

  kill -SIGTERM $SSM_AGENT_PID
}
trap term_handler SIGTERM SIGINT

# check if the required IAM role is provided
if [[ -z $MANAGED_INSTANCE_ROLE_NAME ]] ; then
  echo "Environment variable MANAGED_INSTANCE_ROLE_NAME not set, exiting" 1>&2
  exit 1
fi

# check if the agent is already running (it will be if ECS Exec is enabled)
if ! ps ax | grep amazon-ssm-agent | grep -v grep > /dev/null; then

  # check if ECS Container Metadata is available
  if [[ -n $ECS_CONTAINER_METADATA_URI_V4 ]] ; then

    # Retrieve info from ECS task metadata endpoint
```

```
    echo "Found ECS Container Metadata, running activation with metadata"
    TASK_METADATA=$(curl "${ECS_CONTAINER_METADATA_URI_V4}/task")
    ECS_TASK_AVAILABILITY_ZONE=$(echo $TASK_METADATA | jq -e -r '.AvailabilityZone')
    ECS_TASK_ARN=$(echo $TASK_METADATA | jq -e -r '.TaskARN')
    ECS_TASK_REGION=$(echo $ECS_TASK_AVAILABILITY_ZONE | sed 's/.$//')

    # validate ECS_TASK_AVAILABILITY_ZONE
    ECS_TASK_AVAILABILITY_ZONE_REGEX='^(af|ap|ca|cn|eu|me|sa|us|us-gov)-(central|north|
(north(east|west))|south|south(east|west)|east|west)-[0-9]{1}[a-z]{1}$'
    if ! [[ $ECS_TASK_AVAILABILITY_ZONE =~ $ECS_TASK_AVAILABILITY_ZONE_REGEX ]] ; then
      echo "Error extracting Availability Zone from ECS Container Metadata, exiting"
 1>&2
      exit 1
    fi

    # validate ECS_TASK_ARN
    ECS_TASK_ARN_REGEX='^arn:(aws|aws-cn|aws-us-gov):ecs:[a-z0-9-]+:[0-9]{12}:task/[a-
zA-Z0-9_-]+/[a-zA-Z0-9]+$'
    if ! [[ $ECS_TASK_ARN =~ $ECS_TASK_ARN_REGEX ]] ; then
      echo "Error extracting Task ARN from ECS Container Metadata, exiting" 1>&2
      exit 1
    fi

    # Create activation tagging with Availability Zone and Task ARN
    CREATE_ACTIVATION_OUTPUT=$(aws ssm create-activation \
      --iam-role $MANAGED_INSTANCE_ROLE_NAME \
      --tags Key=ECS_TASK_AVAILABILITY_ZONE,Value=$ECS_TASK_AVAILABILITY_ZONE
 Key=ECS_TASK_ARN,Value=$ECS_TASK_ARN Key=FAULT_INJECTION_SIDECAR,Value=true \
      --region $ECS_TASK_REGION)

    ACTIVATION_CODE=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e -r .ActivationCode)
    ACTIVATION_ID=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e -r .ActivationId)

    # Register with AWS Systems Manager (SSM)
    if ! amazon-ssm-agent -register -code $ACTIVATION_CODE -id $ACTIVATION_ID -region
 $ECS_TASK_REGION; then
      echo "Failed to register with AWS Systems Manager (SSM), exiting" 1>&2
      exit 1
    fi

    # the agent needs to run in the background, otherwise the trapped signal
    # won't execute the attached function until this process finishes
    amazon-ssm-agent &
    SSM_AGENT_PID=$!
```

```
    # need to keep the script alive, otherwise the container will terminate
    wait $SSM_AGENT_PID

  else
    echo "ECS Container Metadata not found, exiting" 1>&2
    exit 1
  fi

else
  echo "SSM agent is already running, exiting" 1>&2
  exit 1
fi
```

实验模板示例

以下是 the section called “aws:ecs:task-cpu-stress” 操作的实验模板示例。

```
{
    "description": "Run CPU stress on the target ECS tasks",
    "targets": {
        "myTasks": {
            "resourceType": "aws:ecs:task",
            "resourceArns": [
                "arn:aws:ecs:us-east-1:111122223333:task/my-
cluster/09821742c0e24250b187dfed8EXAMPLE"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "EcsTask-cpu-stress": {
            "actionId": "aws:ecs:task-cpu-stress",
            "parameters": {
                "duration": "PT1M"
            },
            "targets": {
                "Tasks": "myTasks"
            }
        }
    },
    "stopConditions": [
        {
```

```
            "source": "none",
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/fis-experiment-role",
    "tags": {}
}
```

使用 FIS aw AWS s: eks: pod 操作

您可以执行 aws:eks:pod 操作，为 EKS 集群中运行的 Kubernetes 容器组（pod）注入故障。

操作

- the section called “aws:eks:pod-cpu-stress”
- the section called “aws:eks:pod-delete”
- the section called “aws:eks:pod-io-stress”
- the section called “aws:eks:pod-memory-stress”
- the section called “aws:eks:pod-network-blackhole-port”
- the section called “aws:eks:pod-network-latency”
- the section called “aws:eks:pod-network-packet-loss”

限制

- 以下操作不适用于 AWS Fargate：
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- 以下操作不支持 `bridge` 网络模式：
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- 您无法使用资源 ARN 或资源标签，在实验模板中标识 aws:eks:pod 类型的目标。必须使用必要资源参数来标识目标。

- 操作 aws:eks:pod-network-latency 和 aws:eks:pod-network-packet-loss 不应以同一容器组（Pod）为目标并行运行。根据您指定的 maxErrors 参数值，该操作可能以已完成或失败状态结束：
  - 如果 maxErrorsPercent 为 0（默认值），则该操作将以失败状态结束。
  - 否则，失败将增加 maxErrorsPercent 预算。如果失败的注入次数未达到提供的 maxErrors，则该操作将以已完成状态结束。
  - 您可以从目标容器组（Pod）中注入的临时容器的日志中识别出这些失败。它将失败，并显示 Exit Code: 16。
- 操作 aws:eks:pod-network-blackhole-port 不应与其他以同一容器组（Pod）为目标并使用相同 trafficType 的操作并行运行。支持使用不同流量类型的并行操作。
- 只有当目标 pod 的设置为时，FIS 才能监控故障注入securityContext的状态。readOnlyRootFilesystem: false如果没有此配置，所有 EKS 容器组（Pod）操作都将失败。

要求

- 在您的计算机 AWS CLI 上安装。此操作仅适用于您使用 AWS CLI 创建 IAM 角色的情况。有关更多信息，请参阅安装或更新 AWS CLI。
- 在计算机上安装 kubectl。此操作仅适用于通过 EKS 集群交互来配置或监控目标应用程序的情况。有关更多信息，请参阅 https://kubernetes.io/docs/tasks/tools/。
- 当前支持的 EKS 最低版本为 1.23。

为 Kubernetes 服务账户创建服务角色。

创建要用作服务角色的 IAM 角色。有关更多信息，请参阅 the section called “实验角色”。

配置 Kubernetes 服务账户

配置 Kubernetes 服务账户，使用指定 Kubernetes 命名空间中的目标运行实验。以下示例中的服务账户为 *myserviceaccount*，命名空间为*###*。请注意，default 是一种标准的 Kubernetes 命名空间。

配置 Kubernetes 服务账户

1. 创建一个名为 rbac.yaml 的文件，并添加以下内容。

```
kind: ServiceAccount
```

```
apiVersion: v1
metadata:
  namespace: default
  name: myserviceaccount

---
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: role-experiments
rules:
- apiGroups: [""]
  resources: ["configmaps"]
  verbs: [ "get", "create", "patch", "delete"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["create", "list", "get", "delete", "deletecollection"]
- apiGroups: [""]
  resources: ["pods/ephemeralcontainers"]
  verbs: ["update"]
- apiGroups: [""]
  resources: ["pods/exec"]
  verbs: ["create"]
- apiGroups: ["apps"]
  resources: ["deployments"]
  verbs: ["get"]

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: bind-role-experiments
  namespace: default
subjects:
- kind: ServiceAccount
  name: myserviceaccount
  namespace: default
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: fis-experiment
roleRef:
  kind: Role
  name: role-experiments
```

```
  apiGroup: rbac.authorization.k8s.io
```

2. 运行以下命令。

```
kubectl apply -f rbac.yaml
```

将实验角色映射到 Kubernetes 用户

运行以下命令，创建身份映射。有关更多信息，请参阅 eksctl 文档中的管理 IAM 用户和角色。

```
eksctl create iamidentitymapping \
    --arn arn:aws:iam::123456789012:role/fis-experiment-role \
    --username fis-experiment \
    --cluster my-cluster
```

容器组（pod）容器映像

AWS FIS 提供的容器镜像托管在 Amazon ECR 中。从 Amazon ECR 中引用映像时，必须使用完整的映像 URI。

AWS 区域	镜像 URI
美国东部（俄亥俄）	051821878176.dkr.ecr.us-east-2.amazonaws.com/aws-fis-pod:0.1
美国东部（弗吉尼亚州北部）	731367659002.dkr.ecr.us-east-1.amazonaws.com/aws-fis-pod:0.1
美国西部（加利福尼亚北部）	080694859247.dkr.ecr.us-west-1.amazonaws.com/aws-fis-pod:0.1
美国西部（俄勒冈州）	864386544765.dkr.ecr.us-west-2.amazonaws.com/aws-fis-pod:0.1
非洲（开普敦）	056821267933.dkr.ecr.af-south-1.amazonaws.com/aws-fis-pod:0.1
亚太地区（香港）	246405402639.dkr.ecr.ap-east-1.amazonaws.com/aws-fis-pod:0.1

AWS 区域	镜像 URI
亚太地区（孟买）	524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1
亚太地区（首尔）	526524659354.dkr.ecr.ap-northeast-2.amazonaws.com/aws-fis-pod:0.1
亚太地区（新加坡）	316401638346.dkr.ecr.ap-southeast-1.amazonaws.com/aws-fis-pod:0.1
亚太地区（悉尼）	488104106298.dkr.ecr.ap-southeast-2.amazonaws.com/aws-fis-pod:0.1
亚太地区（东京）	635234321696.dkr.ecr.ap-northeast-1.amazonaws.com/aws-fis-pod:0.1
加拿大（中部）	490658072207.dkr.ecr.ca-central-1.amazonaws.com/aws-fis-pod:0.1
欧洲地区（法兰克福）	713827034473.dkr.ecr.eu-central-1.amazonaws.com/aws-fis-pod:0.1
欧洲地区（爱尔兰）	205866052826.dkr.ecr.eu-west-1.amazonaws.com/aws-fis-pod:0.1
欧洲地区（伦敦）	327424803546.dkr.ecr.eu-west-2.amazonaws.com/aws-fis-pod:0.1
欧洲地区（米兰）	478809367036.dkr.ecr.eu-south-1.amazonaws.com/aws-fis-pod:0.1
欧洲地区（巴黎）	154605889247.dkr.ecr.eu-west-3.amazonaws.com/aws-fis-pod:0.1
欧洲地区（斯德哥尔摩）	263175118295.dkr.ecr.eu-north-1.amazonaws.com/aws-fis-pod:0.1
中东（巴林）	065825543785.dkr.ecr.me-south-1.amazonaws.com/aws-fis-pod:0.1

AWS 区域	镜像 URI
南美洲（圣保罗）	767113787785.dkr.ecr.sa-east-1.amazonaws.com/aws-fis-pod:0.1
AWS GovCloud（美国东部）	246533647532.dkr.ecr.us-gov-east-1.amazonaws.com/aws-fis-pod:0.1
AWS GovCloud（美国西部）	246529956514.dkr.ecr.us-gov-west-1.amazonaws.com/aws-fis-pod:0.1

实验模板示例

以下是 the section called “aws:eks:pod-network-latency” 操作的实验模板示例。

```
{
    "description": "Add latency and jitter to the network interface for the target EKS
 pods",
    "targets": {
        "myPods": {
            "resourceType": "aws:eks:pod",
            "parameters": {
                "clusterIdentifier": "mycluster",
                "namespace": "default",
                "selectorType": "labelSelector",
                "selectorValue": "mylabel=mytarget"
            },
            "selectionMode": "COUNT(3)"
        }
    },
    "actions": {
        "EksPod-latency": {
            "actionId": "aws:eks:pod-network-latency",
            "description": "Add latency",
            "parameters": {
                "kubernetesServiceAccount": "myserviceaccount",
                "duration": "PT5M",
                "delayMilliseconds": "200",
                "jitterMilliseconds": "10",
                "sources": "0.0.0.0/0"
            },
```

```
            "targets": {
                "Pods": "myPods"
            }
        }
    },
    "stopConditions": [
        {
            "source": "none",
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/fis-experiment-role",
    "tags": {
        "Name": "EksPodNetworkLatency"
    }
}
```

使用列出 AWS FIS 操作 AWS CLI

您可以使用 AWS Command Line Interface (AWS CLI) 查看有关 AWS FIS 支持的操作的信息。

先决条件

在您的计算机 AWS CLI 上安装。要开始使用，请参阅 AWS Command Line Interface 用户指南。有关命令的更多信息 AWS FIS，请参阅《AWS CLI 命令参考》中的 fis。

示例：列出所有操作的名称

您可以运行以下 list-actions 命令，列出所有操作的名称。

```
aws fis list-actions --query "actions[*].[id]" --output text | sort
```

下面是示例输出。

```
aws:cloudwatch:assert-alarm-state
aws:dynamodb:global-table-pause-replication
aws:ebs:pause-volume-io
aws:ec2:api-insufficient-instance-capacity-error
aws:ec2:asg-insufficient-instance-capacity-error
aws:ec2:reboot-instances
aws:ec2:send-spot-instance-interruptions
aws:ec2:stop-instances
aws:ec2:terminate-instances
```

```
aws:ecs:drain-container-instances
aws:ecs:stop-task
aws:eks:inject-kubernetes-custom-resource
aws:eks:terminate-nodegroup-instances
aws:elasticache:interrupt-cluster-az-power
aws:fis:inject-api-internal-error
aws:fis:inject-api-throttle-error
aws:fis:inject-api-unavailable-error
aws:fis:wait
aws:network:disrupt-connectivity
aws:network:route-table-disrupt-cross-region-connectivity
aws:network:transit-gateway-disrupt-cross-region-connectivity
aws:rds:failover-db-cluster
aws:rds:reboot-db-instances
aws:s3:bucket-pause-replication
aws:ssm:send-command
aws:ssm:start-automation-execution
```

示例：查看有关操作的信息

在得知操作名称后，您可以运行以下 get-action 命令，查看有关操作的详细信息。

```
aws fis get-action --id aws:ec2:reboot-instances
```

下面是示例输出。

```
{
    "action": {
        "id": "aws:ec2:reboot-instances",
        "description": "Reboot the specified EC2 instances.",
        "targets": {
            "Instances": {
                "resourceType": "aws:ec2:instance"
            }
        },
        "tags": {}
    }
}
```

AWS FIS 的实验模板

实验模板包含实验期间要在指定目标上运行的一项或多项操作。其中还包含阻止实验越界的停止条件。您可以使用创建的实验模板运行实验。

模板组件

以下组件将用于构造实验模板：

操作集

要运行的 AWS FIS 操作。操作可以按您指定的顺序运行，也可以同时运行。有关更多信息，请参阅 操作集。

目标

执行特定操作所依据的 AWS 资源。有关更多信息，请参阅 目标。

停止条件

定义应用程序性能不可接受的阈值的 CloudWatch 警报。如果在实验运行时触发了停止条件，AWS FIS 将停止实验。有关更多信息，请参阅 停止条件。

实验角色

一个 IAM 角色， AWS 它向 FIS 授予所需的权限，使其可以代表您运行实验。有关更多信息，请参阅 实验角色。

实验选项

实验模板的选项。有关更多信息，请参阅 实验选项。

您的账户有与 AWS FIS 相关的配额。例如，每个实验模板都设置了操作次数限额。有关更多信息，请参阅 配额和限制。

模板语法

以下是实验模板的语法。

```
{
            "description": "string",
```

```
            "targets": {},
            "actions": {},
            "stopConditions": [],
            "roleArn": "arn:aws:iam::123456789012:role/AllowFISActions",
            "experimentOptions":{},
            "tags": {}
        }
```

有关示例，请参阅示例模板。

开始使用

要使用创建实验模板 AWS Management Console，请参阅创建实验模板。

要使用创建实验模板 AWS CLI，请参阅AWS FIS 实验模板示例。

AWS FIS 的行动集

要创建实验模板，则必须定义一项或多项操作以形成操作集。有关 AWS FIS 提供的预定义操作列表，请参阅操作。

实验期间只能执行一次操作。要在同一个实验中多次运行同一 AWS FIS 操作，请使用不同的名称将其多次添加到模板中。

内容

- 操作语法
- 操作持续时间
- 操作示例

操作语法

操作集采用以下语法形式：

```
{
    "actions": {
        "action_name": {
            "actionId": "aws:service:action-type",
            "description": "string",
```

```
            "parameters": {
                "name": "value"
             },
            "startAfter": ["action_name", ...],
            "targets": {
                "resource_type": "target_name"
            }
        }
    }
}
```

您需要为操作定义提供以下内容：

action_name

操作名称。

actionId

操作标识符。

description

可选的描述。

parameters

任何操作参数。

startAfter

执行操作前必须完成的所有操作。否则，在开始实验时执行此操作。

targets

任何操作目标。

有关示例，请参阅the section called “操作示例”。

操作持续时间

如果操作包含可用于指定操作持续时间的参数，则默认情况下，只有等到此时段结束，才能视为操作已完成。如果您已将 emptyTargetResolutionMode 实验选项设置为 skip，则当未解析任何目标时，操作将立即完成，状态为“已跳过”。例如，如果您将持续时间指定为 5 分钟，则 AWS FIS 会认为操作在 5 分钟后完成。然后在此时段结束后开始下一项操作，直到完成所有操作。

持续时间可以是操作条件维持时长，也可以是指标监控时长。例如，在指定的持续时间内注入延迟。对于近乎瞬时的操作类型（如终止实例），将在指定持续时间内监控停止条件。

如果操作参数中包含后期操作，则将在操作完成后再执行后期操作。后期操作的用时可能会在指定操作的持续时间到下一项操作的开始时间之间造成延迟（如果其他操作均已完成，则改为到实验的结束时间）。

操作示例

示例操作如下所示。

示例

- 停止 EC2 实例
- 中断竞价型实例
- 中断网络流量
- 终止 EKS Worker

示例：停止 EC2 实例

以下操作将停止运行通过 *targetInstances* 目标识别的 EC2 实例。两分钟后重启目标实例。

```
"actions": {
    "stopInstances": {
        "actionId": "aws:ec2:stop-instances",
        "parameters": {
            "startInstancesAfterDuration": "PT2M"
        },
        "targets": {
            "Instances": "targetInstances"
        }
    }
}
```

示例：中断竞价型实例

以下操作将停止使用名为的目标标识的竞价型实例*targetSpotInstances*。两分钟后再中断竞价型实例。

```
"actions": {
    "interruptSpotInstances": {
        "actionId": "aws:ec2:send-spot-instance-interruptions",
        "parameters": {
            "durationBeforeInterruption": "PT2M"
        },
        "targets": {
            "SpotInstances": "targetSpotInstances"
        }
    }
}
```

示例：中断网络流量

以下操作将拒绝目标子网与其他可用区子网之间的流量。

```
"actions": {
    "disruptAZConnectivity": {
        "actionId": "aws:network:disrupt-connectivity",
        "parameters": {
            "scope": "availability-zone",
            "duration": "PT5M"
        },
        "targets": {
            "Subnets": "targetSubnets"
        }
    }
}
```

示例：终止 EKS Worker

以下操作将终止 EKS 集群中使用名*targetNodeGroups*为的目标标识的 50% 的 EC2 实例。

```
"actions": {
    "terminateWorkers": {
        "actionId": "aws:eks:terminate-nodegroup-instances",
        "parameters": {
            "instanceTerminationPercentage": "50"
        },
        "targets": {
            "Nodegroups": "targetNodeGroups"
```

```
        }
    }
}
```

AWS 金融情报机构的目标

目标是一个或多个 AWS 资源， AWS 故障注入服务 (AWS FIS) 在实验期间对这些资源执行操作。目标可以与实验位于同一 AWS 账户，也可以位于不同的账户（使用多账户实验）。要详细了解如何以不同账户中的资源为目标，请参阅多账户实验。

您可以在创建实验模板时定义目标。您可以在实验模板中对多项操作使用相同的目标。

AWS FIS 在实验开始时识别所有目标，然后再开始动作集中的任何动作。 AWS FIS 使用它为整个实验选择的目标资源。如未找到目标，则实验失败。

目录

- 目标语法
- 资源类型
- 标识目标资源
  - 资源筛选条件
  - 资源参数
- 选择模式
- 示例目标
- 示例筛选条件：

目标语法

目标语法如下所示。

```
{
    "targets": {
        "target_name": {
            "resourceType": "resource-type",
            "resourceArns": [
                "resource-arn"
            ],
            "resourceTags": {
```

```
                "tag-key": "tag-value"
            },
            "parameters": {
                "parameter-name": "parameter-value"
            },
            "filters": [
                {
                    "path": "path-string",
                    "values": ["value-string"]
                }
            ],
            "selectionMode": "value"
        }
    }
}
```

在定义目标时，您需要提供以下内容：

target_name

目标的名称。

resourceType

资源类型。

resourceArns

特定资源的 Amazon 资源名称 (ARN)。

resourceTags

应用于特定资源的标签。

parameters

标识具有特定属性的目标的参数。

filters

资源筛选条件使用特定属性限定已识别目标资源的范围。

selectionMode

已识别资源的选择模式。

有关示例，请参阅the section called “示例目标”。

资源类型

每个 AWS FIS 操作都是在特定的 AWS 资源类型上执行的。定义目标时，只能指定一种资源类型。必须指定操作支持的资源类型作为目标。

AWS FIS 支持以下资源类型：

- aws: dynamodb: global-table — 亚马逊 DynamoDB 全球表
- aws:ec2:autoscaling-group：Amazon EC2 自动扩缩组
- aws:ec2:ebs-volume：Amazon EBS 卷
- aws:ec2:instance：Amazon EC2 实例
- aws:ec2:spot-instance：Amazon EC2 竞价型实例
- aws:ec2:subnet：Amazon VPC 子网
- aws:ec2:transit-gateway：中转网关
- aws:ecs:cluster：Amazon ECS 集群
- aws:ecs:task：Amazon ECS 任务
- aws:eks:cluster：Amazon EKS 集群
- aws:eks:nodegroup：Amazon EKS 节点组
- aws:eks:pod：Kubernetes 容器组（pod）
- aws: elasticache: redis-replicationgroup — 一个 Redis 复制组 ElastiCache
- aws:iam:role：IAM 角色
- aws:rds:cluster：Amazon Aurora 数据库集群
- aws:rds:db：Amazon RDS 数据库实例
- aws:s3:bucket：Amazon S3 存储桶

标识目标资源

在 AWS FIS 控制台中定义目标时，可以选择要定位的特定 AWS 资源（特定资源类型）。或者，您可以让 AWS FIS 根据您提供的标准识别一组资源。

要标识目标资源，您可以指定以下内容：

- 资源 ID-特定资源的 AWS 资源 ID。所有资源 ID 必须代表相同类型的资源。
- 资源标签-应用于特定 AWS 资源的标签。

- 资源筛选条件：表示特定属性的资源的路径和值。有关更多信息，请参阅 资源筛选条件。
- 资源参数：表示符合特定标准的资源的参数。有关更多信息，请参阅 资源参数。

注意事项

- 您不能同时为相同目标指定资源 ID 和资源标签。
- 您不能同时为相同目标指定资源 ID 和资源筛选条件。
- 值为空的指定资源标签并不等同于通配符。而是会匹配具有指定标签键和空标签值的资源。

资源筛选条件

资源筛选器是根据特定属性识别目标资源的查询。 AWS 根据您指定的资源类型，FIS 将查询应用于包含 AWS 资源规范描述的 API 操作的输出。目标定义中包含属性与查询匹配的资源。

所有筛选条件均表示为属性路径和可能的值。路径是由句点分隔的元素序列，用于描述资源在描述操作的输出中访问属性的路径。即使资源的描述操作的输出采用驼峰式大小写，每个元素也必须采用 Pascal 拼写法。例如，应使用 `AvailabilityZone`（而不是 `availablityZone`）作为属性元素。

```
"filters": [
    {
        "path": "component.component.component",
        "values": [
            "string"
        ]
    }
],
```

下表包含可用于获取每种资源类型的规范描述的 API 操作和 AWS CLI 命令。 AWS FIS 代表您运行这些操作以应用您指定的过滤器。相应的文档描述了结果中默认包含的资源。例如，结果中可能包含近期终止实例的 DescribeInstances 状态文档。

资源类型	API 操作	AWS CLI 命令
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes
aws:ec2:instance	DescribeInstances	describe-instances

资源类型	API 操作	AWS CLI 命令
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransit网关	describe-transit-gateways
aws:ecs:cluster	DescribeClusters	describe-clusters
aws:ecs:task	DescribeTasks	describe-tasks
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	describe-nodegroup
aws:elasticache:redis-replicationgroup	DescribeReplication群组	describe-replication-groups
aws:iam:role	ListRoles	list-roles
aws:rds:cluster	DescribeDBClusters	describe-db-clusters
aws:rds:db	DescribeDBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets

以下逻辑适用于所有资源筛选条件：

- 筛选条件内的值：OR
- 各筛选条件的值：AND

有关示例，请参阅the section called “示例筛选条件：”。

资源参数

资源参数会根据特定标准标识目标资源。

以下资源类型支持参数。

aws:ec2:ebs-volume

- `availabilityZoneIdentifier`：包含目标卷的可用区代码（如 us-east-1a）。

aws:ec2:subnet

- availabilityZoneIdentifier：包含目标子网的可用区代码（如 us-east-1a）或可用区 ID（如 use1-az1）。
- vpc：包含目标子网的 VPC。每个账户支持不超过一个 VPC。

aws:ecs:task

- cluster：包含目标任务的集群。
- service：包含目标任务的服务。

aws:eks:pod

- availabilityZoneIdentifier：可选。包含目标容器组（pod）的可用区。例如，us-east-1d。亚马逊通过比较容器组（pod）的 hostIP 和集群子网的 CIDR 确定可用区。
- clusterIdentifier – 必需。EKS 目标集群的名称或 ARN。
- namespace – 必需。目标容器组（pod）的 Kubernetes 命名空间。
- selectorType – 必需。选择器的类型。可能的值为 labelSelector、deploymentName 和 podName。
- selectorValue – 必需。选择器的值。此值取决于 selectorType 的值。
- targetContainerName：可选。容器组（pod）规格中定义的目标容器名称。默认是每个目标容器组（pod）规范中定义的第一个容器。

aws:rds:cluster

- writerAvailabilityZoneIdentifiers：可选。数据库集群写入器的可用区。可能的值为：以逗号分隔的可用区标识符列表，all。

aws:rds:db

- availabilityZoneIdentifiers：可选。受影响的数据库实例的可用区。可能的值为：以逗号分隔的可用区标识符列表，all。

aws:elasticache:redis-replicationgroup

- availabilityZoneIdentifier – 必需。包含目标节点的可用区代码（如 us-east-1a）或可用区 ID（如 use1-az1）。

选择模式

您可以通过指定选择模式来限定已识别资源的范围。 AWS FIS 支持以下选择模式：

- ALL：在所有目标上执行操作。

- COUNT(n)：从已识别目标中随机选择特定数量的目标以执行操作。例如，COUNT(1) 会选择某个已识别目标。
- PERCENT(n)：从已识别目标中随机选择特定百分比的目标以执行操作。例如，PERCENT(25) 会选择 25% 已识别目标。

如果您的资源数量为奇数并指定 50%，则 AWS FIS 会向下舍入。例如，如果您添加五个 Amazon EC2 实例作为目标，范围为 50%，则 AWS FIS 会向下舍入为两个实例。无法指定小于一个资源的百分比。例如，如果您添加四个 Amazon EC2 实例，且范围为 5%，则 AWS FIS 无法选择实例。

如果您使用相同的目标资源类型定义多个目标，则 AWS FIS 可以多次选择相同的资源。

无论哪种选择模式，只要指定范围内未标识任何资源，实验都将失败。

示例目标

示例目标如下。

示例

- 指定 VPC 中带有指定标签的实例
- 具有特定参数的任务

示例：指定 VPC 中带有指定标签的实例

此示例可能以指定 VPC 中带有 env=prod 标签的 Amazon EC2 实例为目标。选择模式指定 AWS FIS 随机选择其中一个目标。

```
{
    "targets": {
        "randomInstance": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {
                "env": "prod"
            },
            "filters": [
                {
                    "path": "VpcId",
                    "values": [
                        "vpc-aabbcc11223344556"
```

```
                    ]
                }
            ],
            "selectionMode": "COUNT(1)"
        }
    }
}
```

示例：具有指定参数的任务

此示例可能以具有指定集群和服务的 Amazon ECS 任务为目标。选择模式指定 AWS FIS 随机选择其中一个目标。

```
{
    "targets": {
        "randomTask": {
            "resourceType": "aws:ecs:task",
            "parameters": {
                "cluster": "myCluster",
                "service": "myService"
            },
            "selectionMode": "COUNT(1)"
        }
    }
}
```

示例筛选条件：

示例筛选条件如下。

示例

- EC2 实例
- 数据库集群

示例：EC2 实例

当您为支持 aws: ec2: instance 资源类型的操作指定筛选条件时， AWS FIS 会使用 Amazon EC2 describe-instances 命令并应用筛选器来识别目标。

在 describe-instances 命令返回的 JSON 输出中，每个实例都是 Instances 下的一个结构。以下部分输出中包含##字段。亚马逊将提供使用这些字段从 JSON 输出结构中指定属性路径的示例。

```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "ImageId": "ami-00111111111111111",
                    "InstanceId": "i-00aaaaaaaaaaaaaa",
                    "InstanceType": "t2.micro",
                    "KeyName": "virginia-kp",
                    "LaunchTime": "2020-09-30T11:38:17.000Z",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-1-240.ec2.internal",
                    "PrivateIpAddress": "10.0.1.240",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-203-0-113-17.compute-1.amazonaws.com",
                    "PublicIpAddress": "203.0.113.17",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-aabbcc11223344556",
                    "VpcId": "vpc-00bbbbbbbbbbbbbbbb",
                    ...
                },
                ...
                {
                    ...
                }
            ],
            "OwnerId": "123456789012",
            "ReservationId": "r-aaaaaabbbbb111111"
```

```
        },
        ...
    ]
}
```

要使用资源筛选条件选择特定可用区中的实例，请指定 AvailabilityZone 的属性路径和此可用区的代码作为值。例如：

```
"filters": [
    {
        "path": "Placement.AvailabilityZone",
        "values": [ "us-east-1a" ]
    }
],
```

要使用资源筛选条件选择特定子网中的实例，请指定 SubnetId 的属性路径和子网 ID 作为值。例如：

```
"filters": [
    {
        "path": "SubnetId",
        "values": [ "subnet-aabbcc11223344556" ]
    }
],
```

要选择处于特定实例状态的实例，请指定 Name 的属性路径和以下一种状态名称作为值：pending| running | shutting-down | terminated | stopping | stopped。例如：

```
"filters": [
    {
        "path": "State.Name",
        "values": [ "running" ]
    }
],
```

示例：Amazon RDS 集群（数据库集群）

当您为支持 aws: rds: clust er 资源类型的操作指定筛选条件时，FIS 会 AWS 运行 Amazon RDS describe-db-clusters 命令并应用筛选器来识别目标。

describe-db-clusters 命令会为每个数据库集群返回类似于以下内容的 JSON 输出。以下部分输出中包含##字段。亚马逊将提供使用这些字段从 JSON 输出结构中指定属性路径的示例。

```
[
    {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-2a",
            "us-east-2b",
            "us-east-2c"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "database-1",
        "DBClusterParameterGroup": "default.aurora-postgresql11",
        "DBSubnetGroup": "default-vpc-01234567abc123456",
        "Status": "available",
        "EarliestRestorableTime": "2020-11-13T15:08:32.211Z",
        "Endpoint": "database-1.cluster-example.us-east-2.rds.amazonaws.com",
        "ReaderEndpoint": "database-1.cluster-ro-example.us-east-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "11.7",
        ...
    }
]
```

要应用仅返回使用特定数据库引擎的数据库集群的资源筛选条件，请根据以下示例，指定属性路径为 Engine 并指定值为 aurora-postgresql。

```
"filters": [
    {
        "path": "Engine",
        "values": [ "aurora-postgresql" ]
    }
],
```

要应用仅返回特定可用区中数据库集群的资源筛选条件，请根据以下示例，指定属性路径和值。

```
"filters": [
    {
        "path": "AvailabilityZones",
        "values": [ "us-east-2a" ]
    }
],
```

AWS FIS 的停止条件

AWS 故障注入服务 (AWS FIS) 为您提供控制和护栏，使您能够在工作负载上安全地运行实验。 AWS 停止条件是一种在实验达到您定义为 Amazon CloudWatch 警报的阈值时停止实验的机制。如果在实验期间触发了停止条件， AWS FIS 将停止实验。您无法恢复已停止的实验。

要创建停止条件，请先为应用程序或服务定义稳定状态。稳定状态是指应用程序达到最佳性能时的业务或技术指标。例如，延迟、CPU 负载或重试次数。您可以使用稳定状态创建 CloudWatch 警报，当您的应用程序或服务达到其性能不可接受的状态时，您可以使用该警报来停止实验。有关更多信息，请参阅亚马逊 CloudWatch 用户指南中的使用亚马逊 CloudWatch 警报。

账户设有停止条件数量限制，您可以在实验模板中指定此条件。有关更多信息，请参阅 AWS 故障注入服务的配额和限制。

停止条件语法

创建实验模板时，您可以通过指定您创建的 CloudWatch 警报来指定一个或多个停止条件。

```
{
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:region:123456789012:alarm:alarm-name"
        }
    ]
}
```

以下示例表明实验模板未指定停止条件。

```
{
    "stopConditions": [
        {
            "source": "none"
        }
    ]
}
```

了解更多信息

有关演示如何创建 CloudWatch 警报和向实验模板添加停止条件的教程，请参阅在实例上运行 CPU 压力测试。

有关 AWS FIS 支持的资源类型的可用 CloudWatch 指标的更多信息，请参阅以下内容：

- 使用监控您的实例 CloudWatch
- 亚马逊 ECS CloudWatch 指标
- 使用监控 Amazon RDS 指标 CloudWatch
- 使用监控运行命令指标 CloudWatch

适用于 AWS FIS 实验的 IAM 角色

AWS Identity and Access Management (IAM) 是一种 AWS 服务，可以帮助管理员安全地控制对 AWS 资源的访问。要使用 AWS FIS，则必须创建为 AWS FIS 授予所需权限的 IAM 角色，以便 AWS FIS 能够以您的身份开展实验。您可以在创建实验模板时指定此实验角色。对于单账户实验，适用于实验角色的 IAM 策略必须授予权限，以便修改实验模板中指定为目标的资源。对于多账户实验，实验角色必须向 Orchestrator 角色授予为每个目标账户代入 IAM 角色的权限。有关更多信息，请参见 多账户实验的权限。

亚马逊建议您遵守授予最低权限的标准安全实践。您可以在策略中指定特定资源 ARN 或标签，以实现此目的。

为帮助您快速入门 AWS FIS，亚马逊提供 AWS 托管策略，您可以在创建实验角色时指定这些策略。或者，您也可以在创建专属内联策略文档时使用这些策略作为模型。

内容

- 先决条件
- 选项 1：创建实验角色并附加 AWS 托管策略
- 选项 2：创建实验角色并添加内联策略文档

先决条件

请先安装 AWS CLI 并创建必要信任策略，然后再开始操作。

安装 AWS CLI

> 在开始之前，请安装并配置 AWS CLI。配置 AWS CLI 时，系统会提示您输入 AWS 凭证。本过程中的示例假定您已配置好默认区域。否则，请为每个命令添加 `--region` 选项。有关更多信息，请参阅安装或更新 AWS CLI和配置 AWS CLI。

创建信任关系策略

实验角色必须建立允许 AWS FIS 服务代入角色的信任关系。创建 `fis-role-trust-policy.json` 文本文件并添加以下信任关系策略。

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

建议您使用 `aws:SourceAccount` 和 `aws:SourceArn` 条件键来防止出现混淆代理人问题。源帐户是实验所有者，而源 ARN 是实验 ARN。例如，您应将以下条件块添加到信任策略。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account_id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:fis:region:account_id:experiment/*"
    }
}
```

添加代入目标账户角色的权限（仅限多账户实验）

对于多账户实验，您需要拥有允许 Orchestrator 账户代入目标账户角色的权限。您可以修改以下示例，并将其添加为内联策略文档以代入目标账户角色：

```
{
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource":[
```

```
        "arn:aws:iam::target_account_id:role/role_name"
    ]
}
```

选项 1：创建实验角色并附加 AWS 托管策略

使用 AWS FIS 的其中一项 AWS 托管策略快速入门。

创建实验角色并附加 AWS 托管策略

1. 验证实验中是否使用 AWS FIS 操作托管策略。否则，您需要改为创建专有的内联策略。有关更多信息，请参见 the section called “AWS 托管策略”。
2. 运行以下 create-role 命令，创建角色并添加先决条件中创建的信任策略。

```
aws iam create-role --role-name my-fis-role --assume-role-policy-document
 file://fis-role-trust-policy.json
```

3. 使用以下attach-role-policy命令附加托AWS管策略。

```
aws iam attach-role-policy --role-name my-fis-role --policy-arn fis-policy-arn
```

以下*fis-policy-arn*内容之一在哪里：

- arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorEC2Access
- arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorECSAccess
- arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorEKSAccess
- arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorNetworkAccess
- arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorRDSAccess
- arn:aws:iam::aws:policy/service-role/AWSFaultInjectionSimulatorSSMAccess

选项 2：创建实验角色并添加内联策略文档

此选项适用于未创建托管策略的操作，或仅包含特定实验所需的权限。

创建实验并添加内联策略文档

1. 运行以下 create-role 命令，创建角色并添加先决条件中创建的信任策略。

```
aws iam create-role --role-name my-fis-role --assume-role-policy-document
 file://fis-role-trust-policy.json
```

2. 创建 fis-role-permissions-policy.json 文本文件并添加权限策略。有关可用作起始点的策略的示例，请参阅以下内容。

   - 故障注入操作：从以下策略开始。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFISExperimentRoleFaultInjectionActions",
            "Effect": "Allow",
            "Action": [
                "fis:InjectApiInternalError",
                "fis:InjectApiThrottleError",
                "fis:InjectApiUnavailableError"
            ],
            "Resource": "arn:*:fis:*:*:experiment/*"
        }
    ]
}
```

   - Amazon EBS 操作：从以下政策开始。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:PauseVolumeIO"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*"
```

```
        }
    ]
}
```

- Amazon EC2 操作 — 从AWSFaultInjectionSimulatorEC2Access政策开始。
- Amazon ECS 操作 — 从AWSFaultInjectionSimulatorECSAccess策略开始。
- Amazon EKS 操作 — 从AWSFaultInjectionSimulatorEKSAccess政策开始。
- 网络操作-从AWSFaultInjectionSimulatorNetworkAccess策略开始。
- Amazon RDS 操作 — 从AWSFaultInjectionSimulatorRDSAccess策略开始。
- Systems Manager 操作-从AWSFaultInjectionSimulatorSSMAccess策略开始。

3. 使用以下put-role-policy命令添加您在上一步中创建的权限策略。

```
aws iam put-role-policy --role-name my-fis-role --policy-name my-fis-policy --
policy-document file://fis-role-permissions-policy.json
```

实验选项

实验选项是实验的可选设置。您可以在实验模板上定义某些实验选项。当你开始实验时，会设置其他实验选项。

以下是您在实验模板上定义的实验选项的语法。

```
{
        "experimentOptions": {
        "accountTargeting": "single-account | multi-account",
            "emptyTargetResolutionMode": "fail | skip"
    }
}
```

如果您在创建实验模板时未指定任何实验选项，则使用每个选项的默认值。

以下是您在开始实验时设置的实验选项的语法。

```
{
        "experimentOptions": {
            "actionsMode": "run-all | skip-all"
     }
}
```

如果您在开始实验时未指定任何实验选项，run-all则使用默认值。

内容

- 账户定位
- 空目标解析模式
- 动作模式

账户定位

如果您有多个 AWS 账户拥有想要在实验中定位的资源，则可以使用账户定位实验选项定义多账户实验。您可以使用一个 Orchestrator 账户 运行多账户实验，这会影响多个目标账户 中的资源。协调员账户拥有 AWS FIS 实验模板和实验。目标账户是指个人 AWS 账户，其资源可能会受到 AWS FIS 实验的影响。有关更多信息，请参阅 的多账户实验 AWS FIS。

您可以通过确定目标账户来指明目标资源的位置。您可以提供两个值来确定目标账户：

- 单账户：默认值。实验将仅针对运行 AWS FIS 实验的 AWS 账户中的资源。
- 多账户：实验能够以多个 AWS 账户中的资源为目标。

目标账户配置

要运行多账户实验，必须定义一个或多个目标账户配置。目标账户配置为实验中拥有目标资源的每个账户指定 accountId、roleArn 和描述。实验模板的目标账户配置的账户 ID 必须唯一。

创建多账户实验模板时，实验模板将返回一个只读字段 targetAccountConfigurationsCount，即实验模板中所有目标账户配置的计数。

目标账户配置的语法如下所示。

```
{
    accountId: "123456789012",
    roleArn: "arn:aws:iam::123456789012:role/AllowFISActions",
    description: "fis-ec2-test"
}
```

创建目标账户配置时，您需要提供以下内容：

accountId

目标账户的 12 位 AWS 账户 ID。

roleArn

一个 IAM 角色，授予在目标账户中执行操作的 AWS FIS 权限。

description

可选的描述。

要了解有关如何使用目标账户配置的更多信息，请参阅the section called “使用多账户实验”。

空目标解析模式

该模式可让您选择即使目标资源未解析也允许完成实验。

- 失败：默认值。如果未为目标解析任何资源，则实验将立即终止，状态为 `failed`。
- 跳过：如果没有为目标解析资源，则实验将继续进行，并跳过任何未解析目标的操作。不能跳过使用唯一标识符（如 ARN）定义目标的操作。如果未找到使用唯一标识符定义的目标，则实验将立即终止，状态为 `failed`

动作模式

操作模式是一个可选参数，您可以在开始实验时指定该参数。您可以将操作模式设置为，`skip-all`以便在向目标资源注入错误之前生成目标预览。目标预览允许您验证以下内容：

- 您已将实验模板配置为针对您期望的资源。开始本实验时所针对的实际资源可能与预览版不同，因为资源可能会被随机移除、更新或采样。
- 您的日志配置设置正确。
- 对于多账户实验，您已经为每个目标账户配置正确设置了 IAM 角色。

> **Note**
>
> 该`skip-all`模式不允许您验证自己是否具有运行 AWS FIS 实验和对资源执行操作所需的权限。

操作模式参数接受以下值：

- `run-all`-（默认）实验将对目标资源采取行动。
- `skip-all`-实验将跳过对目标资源的所有操作。

要详细了解如何在开始实验时设置动作模式参数，请参阅根据实验模板生成目标预览。

使用 AWS FIS 实验模板

您可以使用 AWS FIS 控制台或命令行创建和管理实验模板。您可以使用创建的实验模板运行实验。

任务

- 创建实验模板
- 查看实验模板
- 根据实验模板生成目标预览
- 通过模板开始实验
- 更新实验模板
- 标记实验模板
- 删除实验模板

创建实验模板

开始之前，完成以下任务：

- 计划实验。
- 创建一个 IAM 角色来授予 AWS FIS 服务代表您执行操作的权限。有关更多信息，请参阅 适用于 AWS FIS 实验的 IAM 角色。
- 确保您可以访问 AWS FIS。有关更多信息，请参阅 AWS FIS 策略示例。

使用控制台创建实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。

4. （可选）对于账户定位，请选择多个账户以配置多账户实验模板。
5. 对于账户定位，请选择确认。
6. 对于描述和名称，输入模板的描述和名称。
7. 对于操作，为模板指定一组操作。对于每项操作，选择添加操作并完成以下步骤：

   - 对于名称，输入操作名称。

     允许使用字母数字字符、连字符 (-) 和下划线 (_)。名称必须以字母开头。不允许使用空格。模板中的每个操作名称都必须是唯一的。
   - （可选）对于描述，输入操作描述。最大长度为 512 个字符。
   - （可选）对于之后开始，选择此模板中定义的另一项操作，其必须在当前操作开始前完成。否则，在开始实验时执行此操作。
   - 对于操作类型，选择 AWS FIS 操作。
   - 对于目标，选择您在目标部分中定义的目标。如果您尚未为此操作定义目标， AWS FIS 会为您创建一个新目标。
   - 对于操作参数，指定操作参数。仅当 AWS FIS 操作具有参数时，才会显示此部分。
   - 选择保存。
8. 对于目标，定义执行此操作所需的目标资源。您必须指定至少一个资源 ID 或资源标签作为目标。选择编辑编辑 AWS FIS 在上一步中为您创建的目标，或者选择添加目标。对每个目标执行以下操作：

   - 对于名称，输入目标名称。

     允许使用字母数字字符、连字符 (-) 和下划线 (_)。名称必须以字母开头。不允许使用空格。模板中的每个目标名称都必须是唯一的。
   - 对于资源类型，选择操作支持的资源类型。
   - 对于目标方法，执行以下操作之一：
     - 选择资源 ID，然后选择或添加资源 ID。
     - 选择资源标签、筛选条件和参数，然后添加所需的标签和筛选条件。有关更多信息，请参阅 the section called “标识目标资源”。
   - 对于选择模式，选择计数，对指定数量的已识别目标执行操作，或者选择百分比，对已识别目标的指定百分比执行操作。默认对所有已识别目标执行操作。
   - 选择保存。

9. 要使用您创建的目标更新操作，请在操作下找到此操作，选择编辑，然后更新目标。您可以针对多项操作使用相同目标。
10. （仅限多账户实验）对于目标账户配置，请为每个目标账户添加角色 ARN 和可选描述。要上传带有 CSV 文件的目标账户角色 ARN，请选择为所有目标账户上传角色 ARN，然后选择选择 .CSV 文件
11. 对于服务访问权限，选择使用现有 IAM 角色，然后选择您按照本教程先决条件中所述创建的 IAM 角色。如未显示此角色，请验证其是否具有必要的信任关系。有关更多信息，请参阅 the section called “实验角色”。
12. （可选）对于停止条件，请为停止条件选择 Amazon CloudWatch 警报。有关更多信息，请参阅 AWS FIS 的停止条件。
13. （可选）对于日志，配置目的地选项。要向 S3 存储桶发送日志，请选择发送到 Amazon S3 存储桶，然后输入存储桶名称和前缀。要将日志发送到 CloudWatch 日志，请选择发送到 CloudWatch 日志并输入日志组。
14. （可选）对于标签，选择添加新标签，然后指定标签键和标签值。您添加的标签将应用于实验模板，而不是应用于使用此模板运行的实验。
15. 选择创建实验模板。当系统提示您确认时，输入 **`create`**，然后选择创建实验模板。

使用 CLI 创建实验模板

使用 create-experiment-template 命令。

您可以从 JSON 文件中加载实验模板。

使用 `--cli-input-json` 参数。

```
aws fis create-experiment-template --cli-input-json fileb://<path-to-json-file>
```

有关更多信息，请参阅 AWS Command Line Interface 用户指南中的生成 CLI 骨架模板。有关示例模板，请参阅 AWS FIS 实验模板示例。

查看实验模板

您可以查看自己创建的实验模板。

使用控制台查看实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。

2. 在导航窗格中，选择实验模板。
3. 要查看有关特定模板的信息，请选择实验模板 ID。
4. 您可以在详细信息部分查看模板描述和停止条件。
5. 要查看实验模板的操作，请选择操作。
6. 要查看实验模板的目标，请选择目标。
7. 要查看实验模板的标签，请选择标签。

使用 CLI 查看实验模板

使用list-experiment-templates命令获取实验模板列表，并使用该get-experiment-template命令获取有关特定实验模板的信息。

根据实验模板生成目标预览

在开始实验之前，您可以生成目标预览，以验证您的实验模板是否已配置为针对预期资源。开始实际实验时所针对的资源可能与预览中的资源不同，因为资源可能会被随机移除、更新或采样。生成目标预览时，您会启动一个跳过所有操作的实验。

> **Note**
>
> 生成目标预览不允许您验证自己是否具有对资源执行操作所需的权限。

使用控制台启动目标预览

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. 要查看实验模板的目标，请选择目标。
4. 要验证实验模板的目标资源，请选择 “生成预览”。运行实验时，此目标预览将自动更新为最新实验中的目标。

使用 CLI 启动目标预览

- 运行以下启动实验命令。用您自己的值替换斜体值。

```
aws fis start-experiment \
```

```
    --experiment-options actionsMode=skip-all \
    --experiment-template-id EXTxxxxxxxxx
```

通过模板开始实验

您可以使用创建的实验模板开始实验。

开始实验时，亚马逊会为指定模板创建快照并将其用于开展实验。因此，在实验期间对模板进行的更新或删除不会影响正在运行的实验。

当您开始实验时， AWS FIS 会代表您创建一个与服务相关的角色。有关更多信息，请参阅 为 AWS 故障注入服务使用服务相关角色。

开始实验后，你可以随时停止实验。有关更多信息，请参阅 停止实验。

使用控制台开始实验

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. （可选）要生成预览以验证您的目标，请执行以下操作：
   - 选择 “目标”。
   - 选择 “生成预览”。
4. 选择实验模板，然后选择开始实验。
5. （可选）要为实验添加标签，请选择添加新标签，然后输入标签键和标签值。
6. 请选择开始实验。当系统提示您确认时，输入 **start**，然后选择开始实验。

使用 CLI 开始实验

运行 start-experiment 命令。

更新实验模板

您可以更新现有实验模板。更新实验模板时，所做的更改不会影响正在使用此模板运行的任何实验。

使用控制台更新实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。

2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和更新实验模板。
4. 根据需要修改模板详细信息，然后选择更新实验模板。

使用 CLI 更新实验模板

使用 update-experiment-template 命令。

标记实验模板

您可以将自己的标签应用于实验模板，以便进行整理。您还可以实施基于标签的 IAM 策略，控制对实验模板的访问权限。

使用控制台标记实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和管理标签。
4. 要添加新标签，请选择添加新标签，然后指定键和值。

   要删除标签，请选择删除。
5. 选择保存。

使用 CLI 标记实验模板

运行 tag-resource 命令。

删除实验模板

您可以删除不再使用的实验模板。此操作不会影响正在使用此模板运行的任何实验。实验会继续运行，直到完成或停止。但从控制台的实验页面上，无法查看已删除的实验模板。

使用控制台删除实验模板

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和删除实验模板。

4. 当系统提示您确认时，输入 **delete**，然后选择删除实验模板。

使用 CLI 删除实验模板

使用 delete-experiment-template 命令。

AWS FIS 实验模板示例

如果您使用 AWS FIS API 或命令行工具来创建实验模板，则可以用 JavaScript 对象表示法 (JSON) 构造模板。有关实验模板组件的更多信息，请参阅 模板组件。

要使用其中某个示例模板进行实验，请将其保存为 JSON 文件（如 `my-template.json`），并将*##*占位符值替换为您自己的值，然后运行以下 create-experiment-template 命令。

```
aws fis create-experiment-template --cli-input-json file://my-template.json
```

示例模板

- 根据筛选条件停止 EC2 实例
- 停止运行指定数量的 EC2 实例
- 运行预先配置的 AWS FIS SSM 文档
- 运行预定义的自动化运行手册
- 使用目标 IAM 角色限制 EC2 实例上的 API 操作
- 对 Kubernetes 集群中的容器组（pod）CPU 进行压力测试

根据筛选条件停止 EC2 实例

在指定 VPC 的指定区域中，以下示例会停止运行所有带有指定标签的 Amazon EC2 实例。两分钟后重启实例。

```
{
    "tags": {
        "Name": "StopEC2InstancesWithFilters"
    },
    "description": "Stop and restart all instances in us-east-1b with the tag env=prod
 in the specified VPC",
    "targets": {
        "myInstances": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {
                "env": "prod"
            },
            "filters": [
                {
```

```
                    "path": "Placement.AvailabilityZone",
                    "values": ["us-east-1b"]
                },
                {
                    "path": "State.Name",
                    "values": ["running"]
                },
                {
                    "path": "VpcId",
                    "values": [ "vpc-aabbcc11223344556"]
                }
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "StopInstances": {
            "actionId": "aws:ec2:stop-instances",
            "description": "stop the instances",
            "parameters": {
                "startInstancesAfterDuration": "PT2M"
            },
            "targets": {
                "Instances": "myInstances"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/role-name"
}
```

停止运行指定数量的 EC2 实例

以下示例使用指定标签停止三个实例。 AWS FIS 选择要随机停止的特定实例。两分钟后重启实例。

```
{
    "tags": {
        "Name": "StopEC2InstancesByCount"
```

```
    },
    "description": "Stop and restart three instances with the specified tag",
    "targets": {
        "myInstances": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {
                "env": "prod"
            },
            "selectionMode": "COUNT(3)"
        }
    },
    "actions": {
        "StopInstances": {
            "actionId": "aws:ec2:stop-instances",
            "description": "stop the instances",
            "parameters": {
                "startInstancesAfterDuration": "PT2M"
            },
            "targets": {
                "Instances": "myInstances"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/role-name"
}
```

运行预先配置的 AWS FIS SSM 文档

以下示例使用预配置的 AWS FIS SSM 文档-run-CPU-stress 在指定的 EC2 实例上运行 CPU 故障注入 60 秒。AWSFIS AWS FIS 对实验进行了两分钟的监测。

```
{
    "tags": {
        "Name": "CPUStress"
    },
    "description": "Run a CPU fault injection on the specified instance",
```

```
    "targets": {
        "myInstance": {
            "resourceType": "aws:ec2:instance",
            "resourceArns": ["arn:aws:ec2:us-east-1:111122223333:instance/instance-
id"],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "CPUStress": {
            "actionId": "aws:ssm:send-command",
            "description": "run cpu stress using ssm",
            "parameters": {
                "duration": "PT2M",
                "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-CPU-Stress",
                "documentParameters": "{\"DurationSeconds\": \"60\",
 \"InstallDependencies\": \"True\", \"CPU\": \"0\"}"
            },
            "targets": {
                "Instances": "myInstance"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/role-name"
}
```

运行预定义的自动化运行手册

以下示例使用 Systems Manager 提供的运行手册 (AWS-PublishSNSNotification)，向 Amazon SNS 发送通知。角色必须具有向指定 SNS 主题发布通知的权限。

```
{
    "description": "Publish event through SNS",
    "stopConditions": [
        {
            "source": "none"
```

```
        }
    ],
    "targets": {
    },
    "actions": {
        "sendToSns": {
            "actionId": "aws:ssm:start-automation-execution",
            "description": "Publish message to SNS",
            "parameters": {
                "documentArn": "arn:aws:ssm:us-east-1::document/AWS-
PublishSNSNotification",
                "documentParameters": "{\"Message\": \"Hello, world\", \"TopicArn\":
 \"arn:aws:sns:us-east-1:111122223333:topic-name\"}",
                "maxDuration": "PT1M"
            },
            "targets": {
            }
        }
    },
    "roleArn": "arn:aws:iam::111122223333:role/role-name"
}
```

使用目标 IAM 角色限制 EC2 实例上的 API 操作

以下示例限制了操作定义中为目标定义中指定的 IAM 角色发出的 API 调用的 100% 的 API 调用。

> **Note**
>
> 如果您想定位属于 Auto Scaling 组成员的 EC2 实例，请使用 aws: ec2: asg-实例容量不足错误操作，改用 Auto Scaling 组作为目标。有关更多信息，请参阅
>
> 对目标自动扩缩组发出的请求注入 InsufficientInstanceCapacity 错误响应。此操作仅支持使用启动模板的自动扩缩组。要了解有关实例容量不足错误的更多信息，请参阅 Amazon EC2 用户指南。

资源类型

- aws:ec2:autoscaling-group

参数

- duration— 在 AWS FIS API 中，该值是 ISO 8601 格式的字符串。例如，PT1M 代表一分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

- availabilityzoneIdentifiers：以逗号分隔的可用区列表。支持区域 ID（例如 "use1-az1, use1-az2"）和区域名称（例如 "us-east-1a"）。
- percentage：可选。目标自动扩缩组的启动请求中注入故障的百分比（1-100）。默认值为 100。

权限

- ec2:InjectApiError条件键 ec2: FisActionId 值设置为，aws:ec2:asg-insufficient-instance-capacity-errorec2:FisTargetArns条件键设置为目标 Auto Scaling 组。

- autoscaling:DescribeAutoScalingGroups

有关策略示例，请参阅示例：使用 ec2:InjectApiError 条件键。

。

```
{
    "tags": {
        "Name": "ThrottleEC2APIActions"
    },
    "description": "Throttle the specified EC2 API actions on the specified IAM role",
    "targets": {
        "myRole": {
            "resourceType": "aws:iam:role",
            "resourceArns": ["arn:aws:iam::111122223333:role/role-name"],
            "selectionMode": "ALL"
        }
    },
    "actions": {
```

```
        "ThrottleAPI": {
            "actionId": "aws:fis:inject-api-throttle-error",
            "description": "Throttle APIs for 5 minutes",
            "parameters": {
                "service": "ec2",
                "operations": "DescribeInstances,DescribeVolumes",
                "percentage": "100",
                "duration": "PT2M"
            },
            "targets": {
                "Roles": "myRole"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/role-name"
}
```

对 Kubernetes 集群中的容器组（pod）CPU 进行压力测试

以下示例使用 Chaos Mesh 对 Amazon EKS Kubernetes 集群中的容器组（pod）CPU 进行一分钟压力测试。

```
{
    "description": "ChaosMesh StressChaos example",
    "targets": {
        "Cluster-Target-1": {
            "resourceType": "aws:eks:cluster",
            "resourceArns": [
                "arn:aws:eks:arn:aws::111122223333:cluster/cluster-id"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "TestCPUStress": {
            "actionId": "aws:eks:inject-kubernetes-custom-resource",
```

```
            "parameters": {
                "maxDuration": "PT2M",
                "kubernetesApiVersion": "chaos-mesh.org/v1alpha1",
                "kubernetesKind": "StressChaos",
                "kubernetesNamespace": "default",
                "kubernetesSpec": "{\"selector\":{\"namespaces\":[\"default\"],
\"labelSelectors\":{\"run\":\"nginx\"}},\"mode\":\"all\",\"stressors\": {\"cpu\":
{\"workers\":1,\"load\":50}},\"duration\":\"1m\"}"
            },
            "targets": {
                "Cluster": "Cluster-Target-1"
            }
        }
    },
    "stopConditions": [{
        "source": "none"
    }],
    "roleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {}
}
```

以下示例使用 Litmus 对 Amazon EKS Kubernetes 集群中的容器组（pod）CPU 进行一分钟压力测试。

```
{
    "description": "Litmus CPU Hog",
    "targets": {
        "MyCluster": {
            "resourceType": "aws:eks:cluster",
            "resourceArns": [
                "arn:aws:eks:arn:aws::111122223333:cluster/cluster-id"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "MyAction": {
            "actionId": "aws:eks:inject-kubernetes-custom-resource",
            "parameters": {
                "maxDuration": "PT2M",
                "kubernetesApiVersion": "litmuschaos.io/v1alpha1",
                "kubernetesKind": "ChaosEngine",
                "kubernetesNamespace": "litmus",
```

```
                "kubernetesSpec": "{\"engineState\":\"active\",\"appinfo\":
{\"appns\":\"default\",\"applabel\":\"run=nginx\",\"appkind\":\"deployment\"},
\"chaosServiceAccount\":\"litmus-admin\",\"experiments\":[{\"name\":\"pod-cpu-hog
\",\"spec\":{\"components\":{\"env\":[{\"name\":\"TOTAL_CHAOS_DURATION\",\"value\":
\"60\"},{\"name\":\"CPU_CORES\",\"value\":\"1\"},{\"name\":\"PODS_AFFECTED_PERC\",
\"value\":\"100\"},{\"name\":\"CONTAINER_RUNTIME\",\"value\":\"docker\"},{\"name\":
\"SOCKET_PATH\",\"value\":\"/var/run/docker.sock\"}]},\"probe\":[]}}],\"annotationCheck
\":\"false\"}"
            },
            "targets": {
                "Cluster": "MyCluster"
            }
        }
    },
    "stopConditions": [{
        "source": "none"
    }],
    "roleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {}
}
```

的多账户实验 AWS FIS

通过多账户实验，你可以在跨越一个区域内多个 AWS 账户的应用程序上设置和运行真实的故障场景。您可以使用一个 Orchestrator 账户 运行多账户实验，这会影响多个目标账户 中的资源。

当您运行多账户实验时，拥有受影响资源的目标账户将通过其 AWS Health Dashboard 收到通知，让目标账户中的用户了解情况。通过多账户实验，您可以：

- 使用提供的中央控制和护栏，在跨多个账户的应用程序上运行现实世界的 AWS FIS 故障场景。
- 使用具有精细权限和标签的 IAM 角色来定义每个目标的范围，从而控制多账户实验的效果。
- 从日志 AWS Management Console 和 AWS FIS 日志 AWS FIS 中集中查看每个账户中执行的操作。
- 监控和审计 AWS 每个账户中 AWS FIS 进行的 API 调用 CloudTrail。

本节可帮助您开始多账户实验。

主题

- 多账户实验的概念
- 多账户实验的先决条件
- 使用多账户实验

多账户实验的概念

以下是多账户实验的主要概念：

Orchestrator 账户

Orchestrator 账户充当中央账户，用于在 AWS FIS 控制台中配置和管理实验以及集中日志记录。协调员账户拥有 AWS FIS 实验模板和实验。

目标账户

目标账户是指个人 AWS 账户，其资源可能会受到 AWS FIS 多账户实验的影响。

目标账户配置

可通过在实验模板中添加目标账户配置来定义实验中的目标账户。目标账户配置是实验模板中的一个元素，该元素对于多账户实验是必需的。您可以通过设置账户 ID、IAM 角色和可选描述为每个目标 AWS 账户定义一个账户。

多账户实验的先决条件

要在多账户实验中使用停止条件，必须先配置跨账户警报。IAM 角色是在创建多账户实验模板时定义的。您可以在创建该模板之前创建必要的 IAM 角色。

内容

- 多账户实验的权限
- 多账户实验的停止条件（可选）

多账户实验的权限

多账户实验使用 IAM 角色链接 向 AWS FIS 授予权限，以便对目标账户中的资源执行操作。对于多账户实验，您可以在每个目标账户和 Orchestrator 账户中设置 IAM 角色。这些 IAM 角色要求在目标账户与 Orchestrator 账户之间以及 Orchestrator 账户与 AWS FIS之间建立信任关系。

目标账户的 IAM 角色包含对资源执行操作所需的权限，这些角色是通过添加目标账户配置为实验模板创建的。您可以为 Orchestrator 账户创建一个 IAM 角色，该角色可以代入目标账户的角色，并与 AWS FIS建立信任关系。该 IAM 角色用作实验模板的 `roleArn`。

要了解有关角色链接的更多信息，请参阅《IAM 用户指南》中的角色术语和概念。

在以下示例中，您将为 Orchestrator 账户 A 设置权限，以便在目标账户 B 中使用 `aws:ebs:pause-volume-io` 运行实验。

1. 在账户 B 中，使用运行该操作所需的权限创建一个 IAM 角色。有关每项操作所需的权限，请参阅 the section called “操作参考”。以下示例显示了目标账户为运行 EBS 暂停卷 IO 操作 the section called “aws:ebs:pause-volume-io” 授予的权限。

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

```
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:PauseVolumeIO"
            ],
            "Resource": "arn:aws:ec2:region:accountIdB:volume/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
}
```

2. 接下来，在账户 B 中添加一个信任策略，以创建与账户 A 的信任关系。为账户 A 的 IAM 角色选择一个名称，您将在步骤 3 中创建它。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
           "Effect": "Allow",
            "Principal": {
                "AWS": "AccountIdA"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike":{
                    "sts:ExternalId": "arn:aws:fis:region:accountIdA:experiment/*"
                },
                "ArnEquals": {
                    "aws:PrincipalArn": "arn:aws:iam::accountIdA:role/role_name"
                }
            }
```

```
        }
    ]
}
```

3. 在账户 A 中创建 IAM 角色。该角色名称必须与您在步骤 2 的信任策略中指定的角色相匹配。要将多个账户确定为目标，您可以向 Orchestrator 授予权限以代入每个角色。以下示例显示了账户 A 代入账户 B 的权限。如果您还有其他目标账户，则需要在此策略中添加其他角色 ARN。每个目标账户只能有一个角色 ARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::accountIdB:role/role_name"
            ]
        }
    ]
}
```

4. 账户 A 的该 IAM 角色用作实验模板的 `roleArn`。以下示例显示了 IAM 角色中所需的信任策略，该策略授予代入账户 A（协调器账户）的 AWS FIS 权限。

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

您还可以使用 StackSets 同时配置多个 IAM 角色。要使用 CloudFormation StackSets，您需要在 AWS 账户中设置必要的 StackSet 权限。要了解更多信息，请参阅使用 AWS CloudFormation StackSets。

多账户实验的停止条件（可选）

停止条件 是一种在实验达到定义的警报阈值时停止实验的机制。要为多账户实验设置停止条件，可以使用跨账户警报。必须在每个目标账户中启用共享，才能使用只读权限向 Orchestrator 账户提供警报。共享后，您可以使用指标数学合并来自不同目标账户的指标。然后，您就可以将此警报添加为实验的停止条件。

要了解有关跨账户控制面板的更多信息，请参阅中的启用跨账户功能。 CloudWatch

使用多账户实验

您可以使用 AWS FIS 控制台或命令行创建和管理多账户实验模板。您可以通过将账户定位实验选项指定为 `"multi-account"`，然后添加目标账户配置来创建多账户实验。您可以使用创建的多账户实验模板运行实验。

内容

- 多账户实验最佳实践
- 创建多账户实验模板
- 更新目标账户配置
- 删除目标账户配置

多账户实验最佳实践

以下是使用多账户实验的最佳实践：

- 为多账户实验配置目标时，建议您在所有目标账户中使用一致的资源标签来确定目标资源。 AWS FIS 实验将解析每个目标账户中标签一致的资源。除了将 `emptyTargetResolutionMode` 设置为 `skip` 的实验外，一项操作必须解析任何目标账户中的至少一个目标资源，否则将失败。操作配额按账户应用。如果您想按资源 ARN 来确定目标资源，则每项操作的单账户限制同样适用。
- 使用参数或筛选条件来确定一个或多个可用区中的目标资源时，应指定可用区 ID，而不是可用区名称。可用区 ID 是跨账户的可用区的唯一且一致的标识符。要了解如何在账户中查找可用区的可用区 ID，请参阅您的 AWS 资源的可用区 ID。

创建多账户实验模板

要学习如何创建实验模板，请访问 AWS Management Console

请参阅 创建实验模板。

使用 CLI 创建实验模板

1. 打开 AWS Command Line Interface
2. 要使用保存的 JSON 文件创建实验，并将账户定位实验选项设置为 "multi-account"（如 my-template.json），请将## 占位符值替换为您自己的值，然后运行以下 create-experiment-template 命令。

   ```
   aws fis create-experiment-template --cli-input-json file://my-template.json
   ```

   这将在响应中返回实验模板。从响应中复制 id，即实验模板的 ID。
3. 运行 create-target-account-configuration 命令，将目标账户配置添加到实验模板中。将## 占位符值替换为您自己的值，使用步骤 2 中的 id 作为 --experiment-template-id 参数的值，然后运行以下命令。--description 参数是可选的。对每个目标账户重复此步骤。

   ```
   aws fis create-target-account-configuration --experiment-template-id EXTxxxxxxxxx
    --account-id 111122223333 --role-arn arn:aws:iam::111122223333:role/role-name --
   description "my description"
   ```
4. 运行 get-target-account-configuration 命令，以检索特定目标账户配置的详细信息。

   ```
   aws fis get-target-account-configuration --experiment-template-id EXTxxxxxxxxx --
   account-id 111122223333
   ```
5. 添加完所有目标账户配置后，就可以运行 list-target-account-configurations 命令来查看是否已创建目标账户配置。

   ```
   aws fis list-target-account-configurations --experiment-template-id EXTxxxxxxxxx
   ```

   还可以通过运行 get-experiment-template 命令来验证是否已添加目标账户配置。该模板将返回一个只读字段 targetAccountConfigurationsCount，该字段是实验模板上所有目标账户配置的计数。
6. 准备就绪后，您可以使用 start-experiment 命令运行实验模板。

```
aws fis start-experiment --experiment-template-id EXTxxxxxxxxx
```

更新目标账户配置

如果要更改账户的角色 ARN 或描述，则可以更新现有目标账户配置。更新目标账户配置时，所做的更改不会影响正在使用此模板运行的任何实验。

要更新目标账户配置，请使用 AWS Management Console

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板
3. 选择实验模板，然后依次选择操作和更新实验模板。
4. 修改目标账户配置，然后选择更新实验模板。

使用 CLI 更新目标账户配置

运行 update-target-account-configuration 命令，将## 占位符值替换为您自己的值。`--role-arn` 和 `--description` 是可选参数，如果不包括在内，将不会更新。

```
aws fis update-target-account-configuration --experiment-template-id EXTxxxxxxxxx
 --account-id 111122223333 --role-arn arn:aws:iam::111122223333:role/role-name --
description "my description"
```

删除目标账户配置

如果您不再需要某个目标账户配置，则可以将其删除。删除目标账户配置时，正在使用该模板运行的任何实验都不会受影响。实验会继续运行，直到完成或停止。

要删除目标账户配置，请使用 AWS Management Console

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和更新。
4. 在目标账户配置下，对于要删除的目标账户角色 ARN，请选择删除。

使用 CLI 删除目标账户配置

运行 delete-target-account-configuration 命令，将## 占位符值替换为您自己的值。

```
aws fis update-target-account-configuration --experiment-template-id EXTxxxxxxxxx --
account-id 111122223333
```

AWS FIS 场景库

这些场景定义了客户可用于测试应用程序弹性的事件或条件，如运行应用程序的计算资源中断。AWS 创建并拥有这些场景，可针对常见的应用程序问题为您提供预定义目标和故障操作（如停止运行自动扩展组中 30% 的实例），最大限度地减少无差别的繁重工作。

主题

- 处理 AWS FIS 场景
- 场景库中的 AWS FIS 场景
- AZ Availability: Power Interruption
- Cross-Region: Connectivity

处理 AWS FIS 场景

这些场景由仅限控制台的场景库提供，并基于 AWS FIS 实验模板运行。要使用场景运行实验，您需要从库中选择场景，然后指定与工作负载详细信息相匹配的参数，最后将所选场景作为实验模板保存到账户中。

主题

- 查看场景
- 使用场景
- 导出场景

查看场景

使用控制台查看场景：

1. 打开 AWS FIS 控制台，网址为https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择场景库。
3. 要查看有关特定场景的信息，请选择场景卡片以打开拆分面板。
   - 您可以在页面底部的拆分面板的描述选项卡中查看对场景的简短描述。您还可以找到先决条件的简短摘要，其中包含所需目标资源的摘要以及为准备搭配场景使用的资源所需采取的任何操作。最后，您可以查看有关场景目标和操作的更多信息，并查看使用默认设置成功运行实验时的预期持续时间。

- 您可以在页面底部的拆分面板的内容选项卡中预览要使用场景创建的实验模板的部分填充版本。
- 您可以在页面底部的拆分面板的详细信息选项卡中找到如何实施场景的详细说明。其中可能包含有关如何概算场景各方面的详细信息。您还会了解哪些指标可以作为停止条件，并提供可从实验中掌握的可观测性（如果适用）。最后，您将获得有关如何扩展生成的实验模板的建议。

使用场景

使用控制台使用场景：

1. 打开 AWS FIS 控制台，网址为https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择场景库。
3. 要查看有关特定场景的信息，请选择场景卡片以打开拆分面板
4. 要使用场景，请选择场景卡并选择使用场景创建模板。
5. 在创建实验模板视图中，填写所有缺失的项目。
   a. 您可以在某些场景中对多个操作或目标之间共享的参数进行批量编辑。此功能将在您更改场景后（包括通过批量编辑功能进行更改）禁用。要使用此功能，请选择批量编辑参数按钮。在模态中编辑参数，然后选择保存按钮。
   b. 某些实验模板可能缺少动作或目标参数，这些参数会在每个动作和目标卡片上突出显示。选择每张卡片的编辑按钮，添加缺失信息，然后选择卡片上的保存按钮。
   c. 所有模板都要用到服务访问执行角色。您可以选用现有角色，为实验模板新建角色。
   d. 我们建议通过选择现有 AWS CloudWatch 警报来定义一个或多个可选的停止条件。了解有关 AWS FIS 的停止条件 的更多信息。如果您尚未配置警报，则可以按照使用 Amazon CloudWatch Alarms 中的说明进行操作，稍后更新实验模板。
   e. 我们建议对亚马逊 CloudWatch 日志或 Amazon S3 存储桶启用可选实验日志。了解有关 AWS FIS 的实验日志记录 的更多信息。如果您尚未配置合适的资源，则可以在稍后更新实验模板。
6. 在创建实验模板中，选择创建实验模板。
7. 在 AWS FIS 控制台的实验模板视图中，选择开始实验。了解有关 AWS FIS 的实验 的更多信息。

导出场景

场景体验仅适用于控制台。场景类似于实验模板，但又并不完全相同，它能直接导入 AWS FIS。如果您想将场景添加到自动化流程中，则可以使用以下两种方法之一：

1. 按照中的使用场景步骤创建有效的 AWS FIS 实验模板并导出该模板。

2. 按照 查看场景 和步骤 3 中的步骤，复制内容选项卡中的场景内容并保存，然后手动添加缺失的参数，以创建有效实验模板。

场景库中的 AWS FIS 场景

场景库中的场景旨在尽可能使用标签，其中场景描述的先决条件和工作原理部分介绍了各场景所需的标签。您可以使用这些预定义标签来标记资源，也可以使用批量参数编辑体验来设置专属标签（请参阅使用场景）。

本参考描述了 AWS FIS 场景库中的常见场景。您也可以使用 AWS FIS 控制台列出支持的场景。

有关更多信息，请参阅使用场景。

AWS FIS 支持以下 Amazon EC2 场景。这些场景使用标签来确定目标实例。您可以使用自己的标签，也可以使用场景中包含的默认标签。其中一些场景会使用 SSM 文档。

- EC2 压力：实例故障：停止一个或多个 EC2 实例，探索实例故障产生的影响。

  定位当前区域中带有特定标签的实例。在此场景中，亚马逊将停止运行实例，并在操作持续时间结束时重启实例，默认为 5 分钟。

- EC2 压力：磁盘：探索磁盘利用率增加对 EC2 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中带有特定标签的 EC2 实例。在此场景中，您可以自定义在操作持续时间内向目标 EC2 实例注入越来越高的磁盘利用率，每项磁盘压力操作的时间均默认为 5 分钟。

- EC2 压力：CPU：探索 CPU 增加对 EC2 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中带有特定标签的 EC2 实例。在此场景中，您可以自定义在操作持续时间内向目标 EC2 实例注入越来越高的 CPU 压力，每项 CPU 压力操作的时间均默认为 5 分钟。

- EC2 压力：内存：探索内存利用率增加对 EC2 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中带有特定标签的 EC2 实例。在此场景中，您可以自定义在操作持续时间内向目标 EC2 实例注入越来越高的内存压力，每项内存压力操作的时间均默认为 5 分钟。

- EC2 压力：网络延迟：探索网络延迟增加对 EC2 应用程序的影响。

在此场景中，亚马逊将定位当前区域中带有特定标签的 EC2 实例。在此场景中，您可以自定义在操作持续时间内向目标 EC2 实例注入越来越高的网络延迟，每项网络延迟操作的时间均默认为 5 分钟。

AWS FIS 支持以下 Amazon EKS 场景。这些场景使用 Kubernetes 应用程序标记来确定目标 EKS 容器组（Pod）。您可以使用自己的标记，也可以使用场景中包含的默认标记。有关将 EKS 与 FIS 配合使用的更多信息，请参阅执行 EKS 容器组（pod）操作。

- EKS 压力：删除容器组（pod）：删除一个或多个容器组（pod），探索 EKS 容器故障产生的影响。

  在此场景中，亚马逊将定位当前区域中与应用程序标签关联的目标容器组（pod）。在此场景中，亚马逊将终止所有匹配的容器组（pod）。Kubernetes 配置将控制容器组（pod）的重新创建过程。

- EKS 压力：CPU：探索 CPU 增加对 EKS 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中与应用程序标签关联的目标容器组（pod）。在此场景中，您可以自定义在操作持续时间内向目标 EKS 容器组（pod）注入越来越高的 CPU 压力，每项 CPU 压力操作的时间均默认为 5 分钟。

- EKS 压力：磁盘：探索磁盘利用率增加对 EKS 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中与应用程序标签关联的目标容器组（pod）。在此场景中，您可以自定义在操作持续时间内向目标 EKS 容器组（pod）注入越来越高的磁盘压力，每项 CPU 压力操作的时间均默认为 5 分钟。

- EKS 压力：内存：探索内存利用率增加对 EKS 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中与应用程序标签关联的目标容器组（pod）。在此场景中，您可以自定义在操作持续时间内向目标 EKS 容器组（pod）注入越来越高的内存压力，每项内存压力操作的时间均默认为 5 分钟。

- EKS 压力：网络延迟：探索网络延迟增加对 EKS 应用程序的影响。

  在此场景中，亚马逊将定位当前区域中与应用程序标签关联的目标容器组（pod）。在此场景中，您可以自定义在操作持续时间内向目标 EKS 容器组（pod）注入越来越高的网络延迟，每项网络延迟操作的时间均默认为 5 分钟。

对于多可用区和多区域应用程序，AWS FIS 支持以下场景。这些场景以多种资源类型为目标。

- AZ Availability: Power Interruption：注入可用区（AZ）电力完全中断的预期症状。了解有关 AZ Availability: Power Interruption 的更多信息。
- Cross-Region: Connectivity：阻止从实验区域到目标区域的应用程序网络流量，并暂停跨区域数据复制。了解有关使用 Cross-Region: Connectivity 的更多信息。

AZ Availability: Power Interruption

您可以使用 AZ Availability: Power Interruption 场景来诱发可用区（AZ）电力完全中断的预期症状。

可通过此场景来演示多可用区应用程序在单个可用区电力完全中断期间能够按预期运行。它包括区域计算（Amazon EC2、EKS 和 ECS）丢失、可用区内无法重新扩展计算、子网连接丢失、RDS 故障转移、 ElastiCache 故障转移以及 EBS 卷无响应。默认情况下，未找到目标的操作将被跳过。

操作

以下操作相结合，会产生单个可用区电力完全中断的许多预期症状。“可用区可用性：电力中断”仅会影响在单个可用区电力中断期间预计会受到影响的服务。默认情况下，该场景会注入电力中断症状 30 分钟，然后再注入恢复期间可能出现的症状 30 分钟。

Stop-Instances

在可用区电力中断期间，受影响可用区中的 EC2 实例将关闭。恢复供电后，实例将重启。AZ Availability: Power Interruption 包括 aws:ec2:stop-instances，用于在中断持续时间内停止受影响可用区中的所有实例。该持续时间过后，这些实例将重新启动。停止由 Amazon EKS 管理的 EC2 实例会导致相关 EKS 容器组（Pod）被删除。停止由 Amazon ECS 管理的 EC2 实例会导致相关 ECS 任务停止。

此操作以在受影响可用区中运行的 EC2 实例为目标。默认情况下，它以标签名为 `AzImpairmentPower`、值为 `StopInstances` 的实例为目标。您可以将此标签添加到实例中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的实例，则会跳过此操作。

Stop-ASG-Instances

在可用区电力中断期间，受影响可用区中由自动扩缩组管理的 EC2 实例将关闭。恢复供电后，实例将重启。AZ Availability: Power Interruption 包括 aws:ec2:stop-instances，用于在中断持续时间内停止受影响可用区中的所有实例，包括由自动扩缩管理的实例。该持续时间过后，这些实例将重新启动。

此操作以在受影响可用区中运行的 EC2 实例为目标。默认情况下，它以标签名为 `AzImpairmentPower`、值为 `IceAsg` 的实例为目标。您可以将此标签添加到实例中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的实例，则会跳过此操作。

暂停实例启动

在可用区电力中断期间，对该可用区中预置容量的 EC2 API 调用将失败。具体而言，以下 API 将受影响：`ec2:StartInstances`、`ec2:CreateFleet` 和 `ec2:RunInstances`。AZ Availability: Power Interruption includes 包括 aws:ec2:api-insufficient-instance-capacity-error，以防止在受影响的可用区中预置新实例。

此操作确定用于预置实例的目标 IAM 角色。这些角色必须通过 ARN 进行确定。默认情况下，如果找不到有效的 IAM 角色，则会跳过此操作。

暂停 ASG 扩缩

在可用区电力中断期间，自动扩缩控制面板为恢复可用区中丢失的容量而进行的 EC2 API 调用将失败。具体而言，以下 API 将受影响：`ec2:StartInstances`、`ec2:CreateFleet` 和 `ec2:RunInstances`。AZ Availability: Power Interruption 包括 aws:ec2:asg-insufficient-instance-capacity-error，以防止在受影响的可用区中预置新实例。这还会阻止 Amazon EKS 和 Amazon ECS 在受影响的可用区中进行扩缩。

此操作以自动扩缩组为目标。默认情况下，它以标签名为 `AzImpairmentPower`、值为 `IceAsg` 的自动扩缩组为目标。您可以将此标签添加到自动扩缩组中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的自动扩缩组，则会跳过此操作。

暂停网络连接

在可用区电力中断期间，可用区中的联网将不可用。发生这种情况时，某些 AWS 服务可能需要几分钟时间更新 DNS，以反映受影响可用区中的私有端点不可用。在此期间，DNS 查询可能会返回无法访问的 IP 地址。AZ Availability: Power Interruption 包括 aws:network:disrupt-connectivity，用于阻止受影响可用区中所有子网的所有网络连接并持续 2 分钟。这将强制大多数应用程序超时和 DNS 刷新。2 分钟后结束该操作，可以随后在可用区仍然不可用的情况下恢复区域服务 DNS。

此操作以子网为目标。默认情况下，它以标签名为 `AzImpairmentPower`、值为 `DisruptSubnet` 的集群为目标。您可以将此标签添加到子网中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的子网，则会跳过此操作。

失效转移 RDS

在可用区电力中断期间，受影响可用区中的 RDS 节点将关闭。受影响可用区中的单个可用区 RDS 节点将完全不可用。对于多可用区集群，写入器节点将失效转移到未受影响的可用区，受影响可用区中的读取器节点将不可用。对于多可用区集群，AZ Availability: Power Interruption 包括 aws:rds:failover-db-cluster，用于在写入器位于受影响的可用区时进行失效转移。

此操作以 RDS 集群为目标。默认情况下，它以标签名为 `AzImpairmentPower`、值为 `DisruptRds` 的集群为目标。您可以将此标签添加到集群中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的集群，则会跳过此操作。

暂停 ElastiCache Redis

在可用区电源中断期间，可用区中的 ElastiCache 节点不可用。 AZ Availability: Power Interruption包括 aws: elasticache: interrupt-cluster-az-power，用于终止受影响可用区中的节点。 ElastiCache 中断期间，不会在受影响的可用区中预置新实例，因此集群的容量将保持在较低水平。

此操作以 ElastiCache 集群为目标。默认情况下，它以标签名为 `AzImpairmentPower`、值为 `ElasticacheImpact` 的集群为目标。您可以将此标签添加到集群中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的集群，则会跳过此操作。请注意，只有写入器节点位于受影响可用区的集群才会被视为有效目标。

暂停 EBS I/O

可用区电力中断后，一旦恢复供电，极少数实例可能会遇到 EBS 卷无响应的情况。AZ Availability: Power Interruption 包括 aws:ebs:pause-io，以使 1 个 EBS 卷处于无响应状态。

默认情况下，仅以设置为在实例终止后持续存在的卷为目标。此操作以标签名为 `AzImpairmentPower`、值为 `APIPauseVolume` 的卷为目标。您可以将此标签添加到卷中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的卷，则会跳过此操作。

限制

- 此场景不包括停止条件。应将适用于您应用程序的正确停止条件添加到实验模板中。
- 不支持在 AWS Fargate 上运行的 Amazon EKS 容器组（Pod）。
- 不支持在 AWS Fargate 上运行的 Amazon ECS 任务。
- 不支持带有两个可读备用数据库实例的 Amazon RDS Multi-AZ。在这种情况下，实例将被终止，RDS 将进行失效转移，容量将立即预置回受影响的可用区。受影响可用区中的可读备用副本将仍然可用。

要求

- 向 AWS FIS 实验角色添加所需的权限。
- 必须将资源标签应用于实验的目标资源。它们可以使用您自己的标签约定，也可以使用场景中定义的默认标签。

权限

以下策略授予 AWS FIS 在 AZ Availability: Power Interruption 场景中执行实验所需的权限。必须将此策略附加到实验角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFISExperimentLoggingActionsCloudwatch",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-acl/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkAcl",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkAcl",
            "Resource": "arn:aws:ec2:*:*:network-acl/*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkAclEntry",
                "ec2:DeleteNetworkAcl"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-acl/*",
                "arn:aws:ec2:*:*:vpc/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkAcl",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeSubnets",
                "ec2:DescribeNetworkAcls"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:ReplaceNetworkAclAssociation",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-acl/*"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "rds:FailoverDBCluster"
            ],
            "Resource": [
                "arn:aws:rds:*:*:cluster:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "rds:RebootDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:*:db:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:DescribeReplicationGroups",
                "elasticache:InterruptClusterAzPower"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Sid": "TargetResolutionByTags",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": [
                "arn:aws:kms:*:*:key/*"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "ec2.*.amazonaws.com"
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:PauseVolumeIO"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*"
        },
        {
            "Sid": "AllowInjectAPI",
            "Effect": "Allow",
            "Action": [
                "ec2:InjectApiError"
            ],
```

```
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:api-insufficient-instance-capacity-error",
                        "aws:ec2:asg-insufficient-instance-capacity-error"
                    ]
                }
            }
        },
        {
            "Sid": "DescribeAsg",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

场景内容

以下内容定义了场景。可以保存此 JSON，并将其与 AWS 命令行界面（AWS CLI）中的 create-experiment-template 命令结合使用以创建实验模板。有关该场景的最新版本，请访问 FIS 控制台中的场景库。

```
{
    "targets": {
        "IAM-role": {
            "resourceType": "aws:iam:role",
            "resourceArns": [],
            "selectionMode": "ALL"
        },
        "EBS-Volumes": {
            "resourceType": "aws:ec2:ebs-volume",
            "resourceTags": {
                "AzImpairmentPower": "ApiPauseVolume"
```

```
            },
            "selectionMode": "COUNT(1)",
            "parameters": {
                "availabilityZoneIdentifier": "us-east-1a"
            },
            "filters": [
                {
                    "path": "Attachments.DeleteOnTermination",
                    "values": [
                        "false"
                    ]
                }
            ]
        },
        "EC2-Instances": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {
                "AzImpairmentPower": "StopInstances"
            },
            "filters": [
                {
                    "path": "State.Name",
                    "values": [
                        "running"
                    ]
                },
                {
                    "path": "Placement.AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "selectionMode": "ALL"
        },
        "ASG": {
            "resourceType": "aws:ec2:autoscaling-group",
            "resourceTags": {
                "AzImpairmentPower": "IceAsg"
            },
            "selectionMode": "ALL"
        },
        "ASG-EC2-Instances": {
            "resourceType": "aws:ec2:instance",
```

```
            "resourceTags": {
                "AzImpairmentPower": "IceAsg"
            },
            "filters": [
                {
                    "path": "State.Name",
                    "values": [
                        "running"
                    ]
                },
                {
                    "path": "Placement.AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "selectionMode": "ALL"
        },
        "Subnet": {
            "resourceType": "aws:ec2:subnet",
            "resourceTags": {
                "AzImpairmentPower": "DisruptSubnet"
            },
            "filters": [
                {
                    "path": "AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "selectionMode": "ALL",
            "parameters": {}
        },
        "RDS-Cluster": {
            "resourceType": "aws:rds:cluster",
            "resourceTags": {
                "AzImpairmentPower": "DisruptRds"
            },
            "selectionMode": "ALL",
            "parameters": {
                "writerAvailabilityZoneIdentifiers": "us-east-1a"
            }
```

```
        },
        "ElastiCache-Cluster": {
            "resourceType": "aws:elasticache:redis-replicationgroup",
            "resourceTags": {
                "AzImpairmentPower": "DisruptElasticache"
            },
            "selectionMode": "ALL",
            "parameters": {
                "availabilityZoneIdentifier": "us-east-1a"
            }
        }
    },
    "actions": {
        "Pause-Instance-Launches": {
            "actionId": "aws:ec2:api-insufficient-instance-capacity-error",
            "parameters": {
                "availabilityZoneIdentifiers": "us-east-1a",
                "duration": "PT30M",
                "percentage": "100"
            },
            "targets": {
                "Roles": "IAM-role"
            }
        },
        "Pause-EBS-IO": {
            "actionId": "aws:ebs:pause-volume-io",
            "parameters": {
                "duration": "PT30M"
            },
            "targets": {
                "Volumes": "EBS-Volumes"
            },
            "startAfter": [
                "Stop-Instances",
                "Stop-ASG-Instances"
            ]
        },
        "Stop-Instances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "completeIfInstancesTerminated": "true",
                "startInstancesAfterDuration": "PT30M"
            },
            "targets": {
```

```
                "Instances": "EC2-Instances"
            }
        },
        "Pause-ASG-Scaling": {
            "actionId": "aws:ec2:asg-insufficient-instance-capacity-error",
            "parameters": {
                "availabilityZoneIdentifiers": "us-east-1a",
                "duration": "PT30M",
                "percentage": "100"
            },
            "targets": {
                "AutoScalingGroups": "ASG"
            }
        },
        "Stop-ASG-Instances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "completeIfInstancesTerminated": "true",
                "startInstancesAfterDuration": "PT30M"
            },
            "targets": {
                "Instances": "ASG-EC2-Instances"
            }
        },
        "Pause-network-connectivity": {
            "actionId": "aws:network:disrupt-connectivity",
            "parameters": {
                "duration": "PT2M",
                "scope": "all"
            },
            "targets": {
                "Subnets": "Subnet"
            }
        },
        "Failover-RDS": {
            "actionId": "aws:rds:failover-db-cluster",
            "parameters": {},
            "targets": {
                "Clusters": "RDS-Cluster"
            }
        },
        "Pause-ElastiCache": {
            "actionId": "aws:elasticache:interrupt-cluster-az-power",
            "parameters": {
```

```
                "duration": "PT30M"
            },
            "targets": {
                "ReplicationGroups": "ElastiCache-Cluster"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": ""
        }
    ],
    "roleArn": "",
    "tags": {
        "Name": "AZ Impairment: Power Interruption"
    },
    "logConfiguration": {
        "logSchemaVersion": 2
    },
    "experimentOptions": {
        "accountTargeting": "single-account",
        "emptyTargetResolutionMode": "skip"
    },
    "description": "Affect multiple resource types in a single AZ, targeting by tags
 and explicit ARNs, to approximate power interruption in one AZ."
}
```

Cross-Region: Connectivity

您可以使用 Cross-Region: Connectivity 场景来阻止从实验区域到目标区域的应用程序网络流量，并暂停 Amazon S3 和 Amazon DynamoDB 的跨区域复制。“跨区域：连接”会影响您运行实验所在区域（实验区域）的出站应用程序流量。可能不会阻止源自您希望与实验区域隔离的区域（目标区域）的无状态入站流量。可能不会阻止源自 AWS 托管服务的流量。

此场景可用于进行如下演示：当无法从实验区域访问目标区域中的资源时，多区域应用程序能够按预期运行。它包括通过以中转网关和路由表为目标来阻止从实验区域到目标区域的网络流量。它还会暂停 S3 和 DynamoDB 的跨区域复制。默认情况下，未找到目标的操作将被跳过。

操作

以下操作相结合，会阻止所含 AWS 服务的跨区域连接。这些操作并行运行。默认情况下，该场景会阻止流量 3 小时，您最多可以将时间延长到 12 小时。

中断中转网关连接

Cross Region: Connectivity 包括 aws:network:transit-gateway-disrupt-cross-region-connectivity，用于阻止从实验区域中的 VPC 到目标区域中的 VPC 之间（通过中转网关连接）的跨区域网络流量。这不会影响对实验区域中的 VPC 端点的访问，但会阻止从实验区域发往目标区域中的 VPC 端点的流量。

此操作以连接实验区域和目标区域的中转网关为目标。默认情况下，它以标签名为 `DisruptTransitGateway`、值为 `Allowed` 的中转网关为目标。您可以将此标签添加到中转网关中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的中转网关，则会跳过此操作。

中断子网连接

Cross Region: Connectivity 包括 aws:network:route-table-disrupt-cross-region-connectivity，用于阻止从实验区域中的 VPC 到目标区域中的公有 AWS IP 块之间的跨区域网络流量。这些公有 IP 块包括目标区域中的 AWS 服务端点（如 S3 区域端点）和托管服务的 AWS IP 块（如用于负载均衡器和 Amazon API Gateway 的 IP 地址）。此操作还会阻止通过跨区域 VPC 对等连接从实验区域到目标区域的网络连接。它不会影响对实验区域中 VPC 端点的访问，但会阻止从实验区域发往目标区域中 VPC 端点的流量。

此操作以实验区域中的子网为目标。默认情况下，它以标签名为 `DisruptSubnet`、值为 `Allowed` 的子网为目标。您可以将此标签添加到子网中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的子网，则会跳过此操作。

暂停 S3 复制

Cross Region: Connectivity 包括 aws:s3:bucket-pause-replication，用于暂停目标存储桶从实验区域到目标区域的 S3 复制。从目标区域到实验区域的复制将不受影响。该场景结束后，存储桶复制将从暂停之处恢复。请注意，复制直至所有对象保持同步所需的时间将因实验持续时间和对象上传到存储桶的速度而异。

此操作的目标是实验区域中启用了跨区域复制（CRR）到目标区域 S3 存储桶的 S3 存储桶。默认情况下，它以标签名为 `DisruptS3`、值为 `Allowed` 的存储桶为目标。您可以将此标签添加到存储桶中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的存储桶，则会跳过此操作。

暂停 DynamoDB 复制

Cross-Region: Connectivity包括 aws: dynamodb: global-table-pause-replic ation，用于暂停实验区域与包括目标区域在内的所有其他区域之间的复制。这可防止进出实验区域的复制，但不会影响其他区域之间的复制。该场景结束后，表复制将从暂停之处恢复。请注意，复制使所有数据保持同步所需的时间将因实验持续时间和表的变化速度而异。

此操作针对实验区域中的 D ynamoDB 全局表。默认情况下，它以标签名为 `DisruptDynamoDb`、值为 `Allowed` 的表为目标。您可以将此标签添加到表中，也可以在实验模板中用自己的标签替换默认标签。默认情况下，如果找不到有效的全局表，则会跳过此操作。

限制

- 此场景不包括停止条件。应将适用于您应用程序的正确停止条件添加到实验模板中。

要求

- 向 AWS FIS 实验角色添加所需的权限。
- 必须将资源标签应用于实验的目标资源。它们可以使用您自己的标签约定，也可以使用场景中定义的默认标签。

权限

以下策略授予 AWS FIS 在 Cross-Region: Connectivity 场景中执行实验所需的权限。必须将此策略附加到实验角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RouteTableDisruptConnectivity1",
            "Effect": "Allow",
            "Action": "ec2:CreateRouteTable",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
```

```
        },
        {
            "Sid": "RouteTableDisruptConnectivity2",
            "Effect": "Allow",
            "Action": "ec2:CreateRouteTable",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity21",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateRouteTable",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity3",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity4",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateManagedPrefixList",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
```

```
            "Sid": "RouteTableDisruptConnectivity5",
            "Effect": "Allow",
            "Action": "ec2:DeleteRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*",
                "arn:aws:ec2:*:*:vpc/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity6",
            "Effect": "Allow",
            "Action": "ec2:CreateRoute",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity7",
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity8",
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
```

```
        {
            "Sid": "RouteTableDisruptConnectivity9",
            "Effect": "Allow",
            "Action": "ec2:DeleteNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity10",
            "Effect": "Allow",
            "Action": "ec2:CreateManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity11",
            "Effect": "Allow",
            "Action": "ec2:DeleteManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity12",
            "Effect": "Allow",
            "Action": "ec2:ModifyManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
```

```
        {
            "Sid": "RouteTableDisruptConnectivity13",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeSubnets",
                "ec2:DescribeRouteTables",
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity14",
            "Effect": "Allow",
            "Action": "ec2:ReplaceRouteTableAssociation",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:route-table/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity15",
            "Effect": "Allow",
            "Action": "ec2:GetManagedPrefixListEntries",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity16",
            "Effect": "Allow",
            "Action": "ec2:AssociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:route-table/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity17",
            "Effect": "Allow",
            "Action": "ec2:DisassociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity18",
            "Effect": "Allow",
            "Action": "ec2:DisassociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity19",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity20",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity1",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateTransitGatewayRouteTable",
                "ec2:AssociateTransitGatewayRouteTable"
            ],
            "Resource": [
```

```
                "arn:aws:ec2:*:*:transit-gateway-route-table/*",
                "arn:aws:ec2:*:*:transit-gateway-attachment/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayAttachments",
                "ec2:DescribeTransitGateways"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion1",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion2",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion3",
            "Effect": "Allow",
            "Action": [
                "s3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "StringLike": {
                    "s3:DestinationRegion": "*"
                }
            }
        },
        {
```

```
            "Sid": "S3CrossRegion4",
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:PutReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "BoolIfExists": {
                    "s3:isReplicationPauseRequest": "true"
                }
            }
        },
        {
            "Sid": "DdbCrossRegion1",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DdbCrossRegion2",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:*:*:table/*",
                "arn:aws:dynamodb:*:*:global-table/*"
            ]
        },
        {
            "Sid": "DdbCrossRegion3",
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey",
                "kms:GetKeyPolicy",
                "kms:PutKeyPolicy"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        }
    ]
```

```
}
```

场景内容

以下内容定义了场景。可以保存此 JSON，并将其与 AWS 命令行界面（AWS CLI）中的 create-experiment-template 命令结合使用以创建实验模板。有关该场景的最新版本，请访问 FIS 控制台中的场景库。

```
{
        "targets": {
                "Transit-Gateway": {
                        "resourceType": "aws:ec2:transit-gateway",
                        "resourceTags": {
                                "TgwTag": "TgwValue"
                        },
                        "selectionMode": "ALL"
                },
                "Subnet": {
                        "resourceType": "aws:ec2:subnet",
                        "resourceTags": {
                                "SubnetKey": "SubnetValue"
                        },
                        "selectionMode": "ALL",
                        "parameters": {}
                },
                "S3-Bucket": {
                        "resourceType": "aws:s3:bucket",
                        "resourceTags": {
                                "S3Impact": "Allowed"
                        },
                        "selectionMode": "ALL"
                },
                "DynamoDB-Global-Table": {
                        "resourceType": "aws:dynamodb:encrypted-global-table",
                        "resourceTags": {
                                "DisruptDynamoDb": "Allowed"
                        },
                        "selectionMode": "ALL"
                }
        },
        "actions": {
                "Disrupt-Transit-Gateway-Connectivity": {
```

```
                        "actionId": "aws:network:transit-gateway-disrupt-cross-region-
connectivity",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "TransitGateways": "Transit-Gateway"
                        }
                },
                "Disrupt-Subnet-Connectivity": {
                        "actionId": "aws:network:route-table-disrupt-cross-region-
connectivity",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "Subnets": "Subnet"
                        }
                },
                "Pause-S3-Replication": {
                        "actionId": "aws:s3:bucket-pause-replication",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "Buckets": "S3-Bucket"
                        }
                },
                "Pause-DynamoDB-Replication": {
                        "actionId": "aws:dynamodb:encrypted-global-table-pause-
replication",
                        "parameters": {
                                "duration": "PT3H"
                        },
                        "targets": {
                                "Tables": "DynamoDB-Global-Table"
                        }
                }
        },
        "stopConditions": [
                {
```

```
                        "source": "none"
                }
        ],
        "roleArn": "",
        "logConfiguration": {
                "logSchemaVersion": 2
        },
        "tags": {
                "Name": "Cross-Region: Connectivity"
        },
        "experimentOptions": {
                "accountTargeting": "single-account",
                "emptyTargetResolutionMode": "skip"
        },
        "description": "Block application network traffic from experiment Region to
 target Region and pause cross-Region replication"
}
```

AWS FIS 的实验

AWS FIS 使您能够对 AWS 工作负载执行故障注入实验。首先，创建实验模板。然后，使用此模板开始实验。

发生下列情况之一时，实验结束：

- 模板中的所有操作均已顺利完成。
- 触发停止条件。
- 发生错误，无法完成操作。例如，未发现目标。
- 实验已手动停止。

您无法恢复已停止或失败的实验。您也无法重新运行已完成的实验。但是，您可以使用同一个实验模板开始新实验。您可以先更新实验模板，再将其指定为新实验的模板。

任务

- 开始实验
- 查看实验
- 标记实验
- 停止实验
- 列出已解析的目标

开始实验

您可以使用实验模板开始实验。有关更多信息，请参阅 通过模板开始实验。

您可以使用 Amazon EventBridge，将实验安排为一次性任务或循环任务。有关更多信息，请参阅 教程：安排定期实验。

您可以使用以下任意功能来监控实验：

- 在 AWS FIS 控制台中查看您的实验。有关更多信息，请参阅 查看实验。
- 在实验中查看目标资源的亚马逊 CloudWatch 指标或查看 AWS FIS 使用指标。有关更多信息，请参阅 使用 CloudWatch 进行监控。

- 启用实验日志记录，以便在实验运行时捕获有关详细信息。有关更多信息，请参阅 实验日志记录。

查看实验

您可以查看正在运行的实验的进度，也可以查看已完成、已停止或已失败的实验。

您的账户将在 120 天后自动删除已完成、已停止或已失败的实验。

使用控制台查看指标

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验。
3. 选择实验 ID，打开详细信息页面。
4. 执行以下一个或多个操作：
   - 查看详细信息和状态，了解实验状态。
   - 选择操作选项卡，了解有关实验操作的信息。
   - 选择目标选项卡，了解有关实验目标的信息。
   - 选择时间轴选项卡，根据开始和结束时间直观呈现这些操作。

使用 CLI 查看实验

运行 list-experiments 命令获取实验列表，并运行 get-experiment 命令获取有关特定实验的信息。

实验状态

实验可能处于以下某种状态：

- 待处理：实验正待处理。
- 正在启动：实验正准备开始。
- 正在运行：实验正在运行。
- 已完成：所有实验操作均已顺利完成。
- 正在停止：停止条件已触发或实验已手动停止。
- 已停止：所有正在运行或待处理的实验操作均已停止。
- 已失败：由于权限不足或语法不正确等错误，实验已经失败。

操作状态

操作可能处于以下某种状态：

- 待处理：由于实验尚未开始或操作要在实验后期开始，此操作正处于待处理状态。
- 正在启动：操作正准备开始。
- 正在运行：操作正在运行。
- 已完成：操作已顺利完成。
- 已取消：在操作开始前，实验就已停止。
- 已跳过：已跳过该操作。
- 正在停止：操作正在停止。
- 已停止：所有正在运行或待处理的实验操作均已停止。
- 已失败：由于权限不足或语法不正确等客户端错误，操作已经失败。

标记实验

您可以对实验应用标签，以便梳理。您还可以实施基于标签的 IAM 策略，控制对实验的访问权限。

使用控制台标记实验

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验。
3. 选择实验，然后依次选择操作和管理标签。
4. 要添加新标签，请选择添加新标签，然后指定键和值。

   要删除标签，请选择删除。
5. 选择保存。

使用 CLI 标记实验

运行 tag-resource 命令。

停止实验

您可以随时停止正在运行的实验。停止实验时，所有未完成的后置操作都会在实验停止前完成。您无法恢复已停止的实验。

使用控制台停止实验

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验。
3. 选择实验，然后选择停止实验。
4. 在确认对话框中，选择停止实验。

使用 CLI 停止实验

运行 stop-experiment 命令。

列出已解析的目标

您可以在目标解析结束后查看实验中已解析目标的信息。

使用控制台查看已解析目标

1. 打开 AWS FIS 控制台，网址为 https://console.aws.amazon.com/fis/。
2. 在导航窗格中，选择实验。
3. 选择实验，然后选择报告。
4. 在资源下查看已解析目标的信息。

使用 CLI 查看已解析目标

使用 list-experiment-resolved-targets 命令。

实验调度器

您可以通过 AWS Fault Injection Service (FIS) 对 AWS 工作负载进行故障注入实验。这些实验运行在模板上，其中包含要在指定目标上运行的一项或多项操作。现在，您可以使用 FIS 控制台将实验安排为一次性任务或本地循环任务。除计划规则外，FIS 目前还提供全新的调度功能。FIS 现在可与 EventBridge 调度程序集成，并代表您创建规则。 EventBridge Scheduler 是一种无服务器调度程序，允许您通过一个中央托管服务创建、运行和管理任务。

> **Important**
>
> 带的 AWS Fault Injection Service 实验计划程序在 AWS GovCloud （美国东部）和 AWS（美国西部）中不可用。 GovCloud

主题

- 开始使用
- 安排 FIS 实验
- 使用控制台更新计划
- 更新实验计划
- 使用控制台禁用或删除实验执行

开始使用

执行角色是 AWS Fault Injection Service 为了与 EventBridge 调度器交互以及让事件桥调度器启动 FIS 实验而担任的 IAM 角色。您可以将权限策略附加到此角色以授予 EventBridge调度程序调用 FIS 实验的权限。以下步骤描述了如何创建新的执行角色和 EventBridge 允许启动实验的策略。

使用 AWS CLI 创建调度器角色

EventBridge 需要使用此 IAM 角色代表客户安排实验。

1. 复制以下 JSON 格式的角色代入策略，并作为 `fis-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。

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

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

2. 在 AWS 命令行界面 (AWS CLI) 中，输入以下新建角色的命令。将 FisSchedulerExecutionRole 替换为您想授予此角色的名称。

```
aws iam create-role --role-name FisSchedulerExecutionRole --assume-role-policy-
document file://fis-execution-role.json
```

成功替换后，您将看到以下输出内容：

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FisSchedulerExecutionRole",
        "RoleId": "AROAZL22PDN5A6WKRBQNU",
        "Arn": "arn:aws:iam::123456789012:role/FisSchedulerExecutionRole",
        "CreateDate": "2023-08-24T17:23:05+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "scheduler.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

3. 要创建允许 EventBridge 调度器调用实验的新策略，请复制以下 JSON 并将其保存为 `fis-start-experiment-permissions.json`本地。以下政策允许 S EventBridge cheduler 对您账户中的所有实验模板进行`fis:StartExperiment`操作。如要将角色限制为单个实验模板，则使用您实验模板的 ID，在 `"arn:aws:fis:*:*:experiment-template/*"` 的底部替换 `*`。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": [
                "arn:aws:fis:*:*:experiment-template/*",
                "arn:aws:fis:*:*:experiment/*"
            ]
        }
    ]
}
```

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

```
aws iam create-policy --policy-name FisSchedulerPolicy --policy-document file://fis-start-experiment-permissions.json
```

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

```
{
    "Policy": {
        "PolicyName": "FisSchedulerPolicy",
        "PolicyId": "ANPAZL22PDN5ESVUWXLBD",
        "Arn": "arn:aws:iam::123456789012:policy/FisSchedulerPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2023-08-24T17:34:45+00:00",
        "UpdateDate": "2023-08-24T17:34:45+00:00"
    }
}
```

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 `FisSchedulerExecutionRole` 替换为您执行角色的名称。

```
aws iam attach-role-policy --policy-arn your-policy-arn --role-name
 FisSchedulerExecutionRole
```

   `attach-role-policy` 操作不会在命令行上返回响应。

6. 您可以限制调度器，使其只运行具有特定标签值的 AWS FIS 实验。例如，以下策略为所有 AWS FIS 实验模板授予 `fis:StartExperiment` 权限，但限制调度器只运行带有 `Purpose=Schedule` 标签的实验。

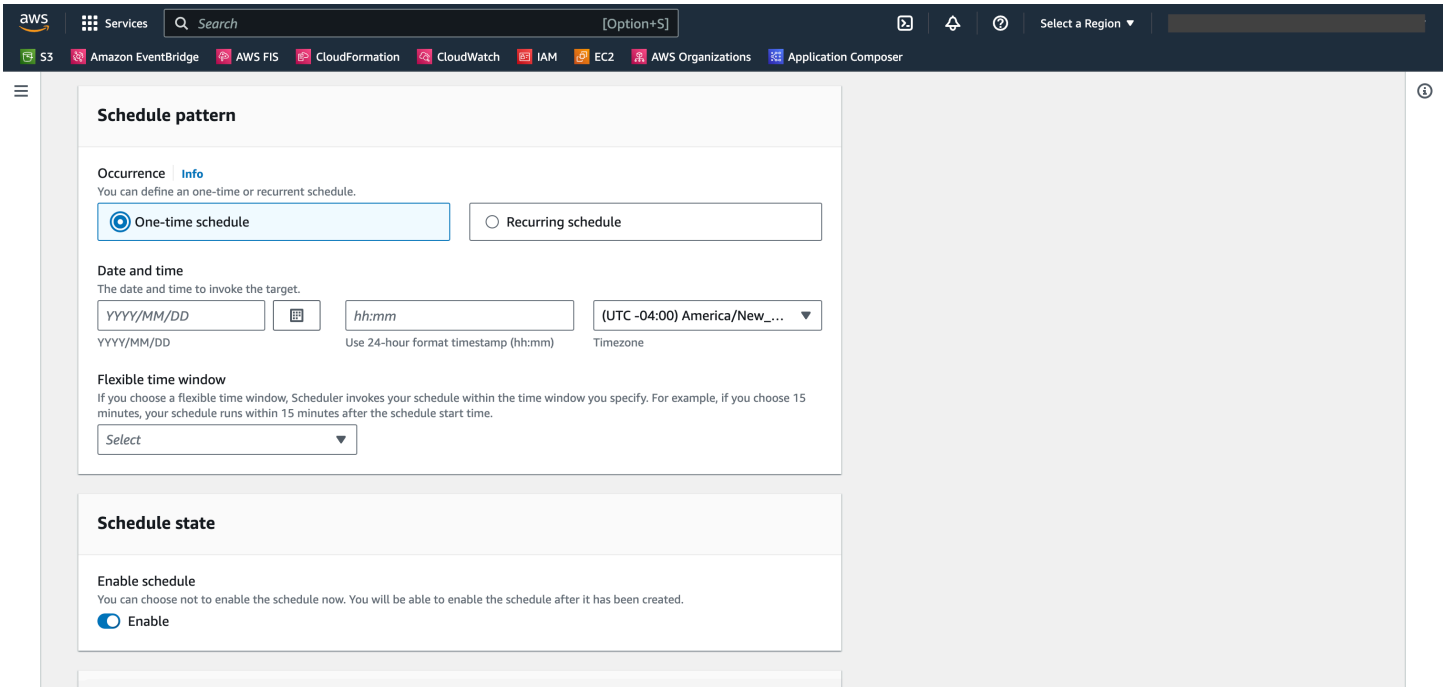
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment/*"
        },
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment-template/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Schedule"
                }
            }
        }
    ]
}
```

安排 FIS 实验

您需要先为此安排调用一个或多个 实验模板，才能安排实验。您可以使用现有的 AWS 资源或新建资源。

创建实验模板后，单击操作，然后选择安排实验。您将重定向至安排实验页面。系统将为您填写计划名称。

请按照计划模式部分的要求，选择一次性任务或循环任务。填写必填项并导航到权限。



系统将默认启用计划状态。注意：如果禁用计划状态，那么即使您创建了计划，系统也不会安排实验。

AWS FIS 实验调度器建立在EventBridge 调度器之上。您可以参考文档，了解支持的各种计划类型。

使用控制台更新计划

1. 打开AWS FIS 控制台。
2. 在左侧导航窗格中，选择实验模板。
3. 选择要创建计划的实验模板。
4. 单击操作，然后从下拉列表中选择安排实验。
   a. 在计划名称下，自动填充名称。
   b. 在计划模式下，选择定期计划。
   c. 在计划类型下，您可以选择基于速率的计划，请参阅计划类型。
   d. 在 Rate 表达式下，选择比实验执行时间更慢的速率，如 5 分钟。
   e. 在时间范围下，选择您的时区。
   f. 在开始日期和时间下，指定开始日期和时间。

g. 在结束日期和时间下，指定结束日期和时间

h. 在计划状态下，切换启用计划选项。

i. 在权限下，选择使用现有角色，然后搜索 `FisSchedulerExecutionRole`。

j. 选择下一步。

5. 选择查看并创建计划，查看调度器的详细信息，然后选择创建计划。

更新实验计划

您可以更新实验计划，将其安排为适合您需求的日期和时间。

使用控制台更新实验执行

1. 打开 Amazon FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择资源类型：实验模板，已为此模板创建计划。
4. 点击模板的实验 ID。然后导航到“计划”选项卡。
5. 检查是否存在与实验关联的现有计划。选择关联计划，然后单击更新计划按钮。

使用控制台禁用或删除实验执行

要阻止实验按计划执行或运行，可以删除或禁用此规则。以下步骤会引导您删除或禁用实验执行。

删除或禁用规则

1. 打开 Amazon FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择资源类型：实验模板，已为此模板创建计划。
4. 点击模板的实验 ID。然后导航到“计划”选项卡。
5. 检查是否存在与实验关联的现有计划。选择关联计划，然后单击更新计划按钮。
6. 请执行以下操作之一：

a. 要删除计划，请选择删除计划规则旁的按钮。键入 `delete`，然后单击删除计划按钮。

b. 要禁用计划，请选择禁用计划规则旁的按钮。键入 `disable`，然后单击禁用计划按钮。

监控 AWS FIS

您可以使用以下工具，监控 AWS Fault Injection Service (AWS FIS) 实验的进度和影响。

AWS FIS 控制台和 AWS CLI

使用 AWS FIS 控制台或 AWS CLI，监控正在运行的实验的进度。您可以查看实验中各项操作的状态及结果。有关更多信息，请参见 the section called “查看实验”。

CloudWatch 使用情况指标和警报

使用 CloudWatch 使用量指标来了解您账户的资源使用情况。 AWSFIS 使用情况指标与 AWS 服务限额对应。您可以配置警报，以在用量接近服务限额时向您发出警报。有关更多信息，请参见 使用 CloudWatch 进行监控。

您还可以通过创建 CloudWatch 警报来定义实验何时越界，从而AWS为 FIS 实验创建停止条件。一旦触发警报，实验就会停止。有关更多信息，请参见 停止条件。有关创建 CloudWatch 警报的更多信息，请参阅 Amazon CloudWatch 用户指南中的基于静态阈值创建 CloudWatch 警报和基于异常检测创建警报。 CloudWatch

AWSFIS 实验日志记录

启用实验日志记录，以便在实验运行时捕获有关详细信息。有关更多信息，请参阅 实验日志记录。

实验状态更改事件

Amazon EventBridge 使您能够自动响应系统事件或资源更改。 AWSFIS 会在实验状态更改时发出通知。您可以为关注事件创建规则，为符合规则的事件指定要执行的自动化操作。例如，向 Amazon SNS 主题发送通知或调用 Lambda 函数。有关更多信息，请参见 监视器使用 EventBridge。

CloudTrail 日志

用于捕获AWS CloudTrail有关 AWS FIS API 调用的详细信息，并将其作为日志文件存储在 Amazon S3 中。 CloudTrail 还会记录对你正在运行实验的资源的服务 API 的调用。您可以使用这些 CloudTrail 日志来确定拨打了哪些呼叫、呼叫来自哪个源 IP 地址、谁拨打了电话、何时拨打了呼叫等。

AWS Health Dashboard 通知

AWS Health 使您可以随时掌握您的资源性能以及 AWS 服务和账户的可用性。当您开始实验时，AWS FIS 会向您的 AWS Health Dashboard 发出通知。该通知在实验持续期间会出现

在包含实验（包括多账户实验）中的目标资源的每个账户中。仅执行不包含目标的操作（如 `aws:ssm:start-automation-execution` 和 `aws:fis:wait`）的多账户实验不会发出通知。有关用于允许实验的角色的信息将列在受影响的资源下。要了解有关 AWS Health Dashboard 的更多信息，请参阅《AWS Health 用户指南》中的 AWS Health Dashboard。

> **Note**
>
> AWS Health 将尽最大效能 传送事件。

使用 Amazon CloudWatch 监控 AWS FIS 的使用情况指标

您可以使用 Amazon CloudWatch，监控 AWS FIS 实验对目标的影响。还可以监控 AWS FIS 的使用情况。

有关查看实验状态的更多信息，请参阅 查看实验。

监测 AWS FIS 实验

在规划 AWS FIS 实验时，请识别出可用于标识实验目标资源类型的基准或“稳定状态”的 CloudWatch 指标。开始实验后，您可以监控实验模板所选目标的 CloudWatch 指标。

有关 AWS FIS 支持的目标资源类型的可用 CloudWatch 指标的更多信息，请参阅以下内容：

- 使用 CloudWatch 监控实例
- Amazon ECS CloudWatch 指标
- 使用 CloudWatch 监控 Amazon RDS 指标
- 使用 CloudWatch 监控 Run Command 指标

AWS FIS 使用情况指标

您可以使用 CloudWatch 使用情况指标来提供账户资源使用情况的可见性。这些指标可在 CloudWatch 图表和控制面板上直观呈现当前的服务使用情况。

AWS FIS 使用情况指标与 AWS 服务限额对应。您可以配置警报，以在用量接近服务配额时向您发出警报。有关 CloudWatch 警报的更多信息，请参阅 Amazon CloudWatch 用户指南。

AWS FIS 在 AWS/Usage 命名空间中发布了以下指标。

指标	描述
ResourceCount	您账户中运行的指定资源的总数量。资源由与指标关联的维度定义。

以下维度用于优化由 AWS FIS 发布的使用情况指标。

维度	描述
Service	包含该资源的 AWS 服务的名称。对于 AWS FIS 使用情况指标，维度值为 FIS。
Type	正在报告的实体的类型。目前，AWS FIS 使用情况指标的唯一有效值为 Resource。
Resource	正在运行的资源的类型。实验模板和进行中实验的值分别可能为 ExperimentTemplates 和 ActiveExperiments 。
Class	保留此维度，以备将来使用。

使用 AWS Amazon 监控 FIS 实验 EventBridge

当实验状态发生变化时， AWS FIS 会发出通知。这些通知通过 Amazon 以事件形式提供 EventBridge（以前称为 CloudWatch 事件）。 AWS FIS 会尽最大努力发布这些事件。近乎实时 EventBridge 地向其发送事件。

使用 EventBridge，您可以创建触发程序化操作以响应事件的规则。例如，您可以配置规则，以调用 SNS 主题发送电子邮件通知，或者调用 Lambda 函数执行某些操作。

有关更多信息 EventBridge，请参阅《亚马逊 EventBridge 用户指南》 EventBridge中的 “亚马逊入门”。

以下是实验状态更改事件的语法：

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
```

```
    "detail-type": "FIS Experiment State Change",
    "source": "aws.fis",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "region",
    "resources": [
        "arn:aws:fis:region:account_id:experiment/experiment-id"
    ],
    "detail": {
        "experiment-id": "EXPaBCD1efg2HIJkL3",
        "experiment-template-id": "EXTa1b2c3de5f6g7h",
        "new-state": {
            "status": "new_value",
            "reason": "reason_string"
        },
        "old-state": {
           "status": "old_value",
           "reason": "reason_string"
        }
    }
}
```

`experiment-id`

状态更改的实验的 ID。

`experiment-template-id`

实验所用实验模板的 ID。

`new_value`

实验的新状态。可能的值包括：

- `completed`
- `failed`
- `initiating`
- `running`
- `stopped`
- `stopping`

`old_value`

实验的上一个状态。可能的值包括：

- initiating
- pending
- running
- stopping

AWS FIS 的实验日志记录

您可以使用实验日志记录，以便在实验运行时捕获有关详细信息。

您需要根据与每种日志目的地类型关联的成本付费使用实验日志记录。有关更多信息，请参阅 Amazon CloudWatch 定价（在 “付费套餐、日志、销售日志” 下）和 Amazon S3 定价。

权限

您必须授予 AWS FIS 权限，才能向配置的各个日志目的地发送日志。有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南中的以下内容：

- 发送到日志的 CloudWatch 日志
- 发送到 Amazon S3 的日志

日志架构

实验日志记录使用以下架构。当前架构采用版本 2。用于 `details` 的字段取决于 `log_type` 值。用于 `resolved_targets` 的字段取决于 `target_type` 值。有关更多信息，请参见 the section called “日志记录示例”。

```
{
    "id": "EXP123abc456def789",
    "log_type": "experiment-start | target-resolution-start | target-resolution-detail
 | target-resolution-end | action-start | action-error | action-end | experiment-end",
    "event_timestamp": "yyyy-mm-ddThh:mm:ssZ",
    "version": "2",
    "details": {
        "account_id":"123456789012",
        "action_end_time": "yyyy-mm-ddThh:mm:ssZ",
        "action_id": "String",
        "action_name": "String",
```

```
        "action_start_time": "yyyy-mm-ddThh:mm:ssZ",
        "action_state": {
            "status": "pending | initiating | running | completed | cancelled |
 stopping | stopped | failed",
            "reason": "String"
        },
        "action_targets": "String to string map",
        "error_information": "String",
        "experiment_end_time": "yyyy-mm-ddThh:mm:ssZ",
        "experiment_state": {
            "status": "pending | initiating | running | completed | stopping | stopped
 | failed",
            "reason": "String"
        },
        "experiment_start_time": "yyyy-mm-ddThh:mm:ssZ",
        "experiment_template_id": "String",
        "page": Number,
        "parameters": "String to string map",
        "resolved_targets": [
            {
               "field": "value"
            }
        ],
        "resolved_targets_count": Number,
        "status": "failed | completed",
        "target_name": "String",
        "target_resolution_end_time": "yyyy-mm-ddThh:mm:ssZ",
        "target_resolution_start_time": "yyyy-mm-ddThh:mm:ssZ",
        "target_type": "String",
        "total_pages": Number,
        "total_resolved_targets_count": Number

    }
}
```

发布说明

- 版本 2 引入了：
  - target_type 字段，并将 resolved_targets 字段从 ARN 列表更改为对象列表。resolved_targets 对象的有效字段取决于 target_type 值，即目标的资源类型。
  - action-error 和 target-resolution-detail 事件类型（添加了 account_id 字段）。
- 初始版本为版本 1。

日志目的地

AWS FIS 支持将日志传输到以下目的地：

- 一个 Amazon S3 存储桶
- Amazon CloudWatch 日志组

S3 日志传输

这些日志会传输到以下位置。

```
bucket-and-optional-prefix/AWSLogs/account-id/fis/region/experiment-
id/YYYY/MM/DD/account-id_awsfislogs_region_experiment-id_YYYYMMDDHHMMZ_hash.log
```

日志会在几分钟后传输到存储桶。

CloudWatch 日志日志传输

这些日志会传输到 /aws/fis/*experiment-id* 日志流中。

日志会在一分钟以内传输到日志组。

日志记录示例

以下随机选取了在 EC2 实例上运行 aws:ec2:reboot-instances 操作的实验日志记录示例。

记录

- experiment-start
- target-resolution-start
- target-resolution-detail
- target-resolution-end
- action-start
- action-end
- action-error
- experiment-end

experiment-start

以下是 experiment-start 事件的示例记录。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "experiment-start",
    "event_timestamp": "2023-05-31T18:50:45Z",
    "version": "2",
    "details": {
        "experiment_template_id": "EXTCDh1M8HHkhxoaQ",
        "experiment_start_time": "2023-05-31T18:50:43Z"
    }
}
```

target-resolution-start

以下是 target-resolution-start 事件的示例记录。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "target-resolution-start",
    "event_timestamp": "2023-05-31T18:50:45Z",
    "version": "2",
    "details": {
        "target_resolution_start_time": "2023-05-31T18:50:45Z",
        "target_name": "EC2InstancesToReboot"
    }
}
```

target-resolution-detail

以下是 target-resolution-detail 事件的示例记录。如果目标分辨率失败，则记录中还有 error_information 字段。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "target-resolution-detail",
    "event_timestamp": "2023-05-31T18:50:45Z",
    "version": "2",
    "details": {
        "target_resolution_end_time": "2023-05-31T18:50:45Z",
        "target_name": "EC2InstancesToReboot",
```

```
        "target_type": "aws:ec2:instance",
        "account_id": "123456789012",
        "resolved_targets_count": 2,
        "status": "completed"

    }
}
```

target-resolution-end

如果目标分辨率失败，则记录中还有 error_information 字段。如果 total_pages 大于 1，则表示已解析目标的数量超出一条记录的大小限制。还有其他包含剩余已解决目标的 target-resolution-end 记录。

以下是 EC2 操作 target-resolution-end 事件的示例记录。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "target-resolution-end",
    "event_timestamp": "2023-05-31T18:50:45Z",
    "version": "2",
    "details": {
        "target_resolution_end_time": "2023-05-31T18:50:46Z",
        "target_name": "EC2InstanceToReboot",
        "target_type": "aws:ec2:instance",
        "resolved_targets": [
            {
                "arn": "arn:aws:ec2:us-east-1:123456789012:instance/
i-0f7ee2abffc330de5"
            }
        ],
        "page": 1,
        "total_pages": 1
    }
}
```

以下是 EKS 操作 target-resolution-end 事件的示例记录。

```
{
    "id": "EXP24YfiucfyVPJpEJn",
    "log_type": "target-resolution-end",
    "event_timestamp": "2023-05-31T18:50:45Z",
```

```
    "version": "2",
    "details": {
        "target_resolution_end_time": "2023-05-31T18:50:46Z",
        "target_name": "myPods",
        "target_type": "aws:eks:pod",
        "resolved_targets": [
            {
                "pod_name": "example-696fb6498b-sxhw5",
                "namespace": "default",
                "cluster_arn": "arn:aws:eks:us-east-1:123456789012:cluster/fis-demo-
cluster",
                "target_container_name": "example"
            }
        ],
        "page": 1,
        "total_pages": 1
    }
}
```

action-start

以下是 `action-start` 事件的示例记录。如果实验模板指定了操作参数，则记录中还有 `parameters` 字段。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "action-start",
    "event_timestamp": "2023-05-31T18:50:56Z",
    "version": "2",
    "details": {
        "action_name": "Reboot",
        "action_id": "aws:ec2:reboot-instances",
        "action_start_time": "2023-05-31T18:50:56Z",
        "action_targets": {"Instances":"EC2InstancesToReboot"}
    }
}
```

action-error

以下是 `action-error` 事件的示例记录。只有操作失败时才会返回该事件。操作失败的每个账户都会返回该事件。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "action-error",
    "event_timestamp": "2023-05-31T18:50:56Z",
    "version": "2",
    "details": {
        "action_name": "pause-io",
        "action_id": "aws:ebs:pause-volume-io",
        "account_id": "123456789012",
        "action_state": {
            "status": "failed",
            "reason":"Unable to start Pause Volume IO. Target volumes must be attached
 to an instance type based on the Nitro system. VolumeId(s): [vol-1234567890abcdef0]:"
        }
    }
}
```

action-end

以下是 `action-end` 事件的示例记录。

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "action-end",
    "event_timestamp": "2023-05-31T18:50:56Z",
    "version": "2",
    "details": {
        "action_name": "Reboot",
        "action_id": "aws:ec2:reboot-instances",
        "action_end_time": "2023-05-31T18:50:56Z",
        "action_state": {
            "status": "completed",
            "reason": "Action was completed."
        }
    }
}
```

experiment-end

以下是 `experiment-end` 事件的示例记录。

```
{
```

```
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "experiment-end",
    "event_timestamp": "2023-05-31T18:50:57Z",
    "version": "2",
    "details": {
        "experiment_end_time": "2023-05-31T18:50:57Z",
        "experiment_state": {
            "status": "completed",
            "reason": "Experiment completed"
        }
    }
}
```

启用实验日志记录

默认禁用实验日志记录。要接收实验日志记录，您必须通过启用了日志记录的实验模板来创建实验。如果某个实验配置为使用一个以前未用于日志记录的目的地，则当您首次运行该实验时，我们会将此实验延迟约 15 秒，以便将日志配置为传输到此目的地。

使用控制台启用实验日志记录

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和更新实验模板。
4. 对于日志，配置目的地选项。要向 S3 存储桶发送日志，请选择发送到 Amazon S3 存储桶，然后输入存储桶名称和前缀。要将日志发送到 CloudWatch 日志，请选择发送到 CloudWatch 日志并输入日志组。
5. 选择更新实验模板。

使用 AWS CLI 启用实验日志记录

使用update-experiment-template命令并指定日志配置。

禁用实验日志记录

如果不想再接收实验日志，则可以禁用实验日志记录。

使用控制台禁用实验日志记录

1. 您可以访问 https://console.aws.amazon.com/fis/，打开 AWS FIS 控制台。

2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后依次选择操作和更新实验模板。
4. 对于日志，清除 “发送到 Amazon S3 存储桶” 和 “发送到 CloudWatch 日志”。
5. 选择更新实验模板。

使用 AWS CLI 禁用实验日志记录

使用update-experiment-template命令并指定空日志配置。

使用 AWS CloudTrail 记录 API 调用

AWS故障注入服务 (AWSFIS) 与AWS CloudTrail一项服务集成，该服务提供用户、角色或AWS服务在 AWS FIS 中采取的操作的记录。 CloudTrail 将 AWS FIS 的所有 API 调用捕获为事件。这些调用包括来自 AWS FIS 控制台的调用，以及对 AWS FIS API 操作的代码调用。如果您创建跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 AWS FIS 的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向 AWS FIS 发出的请求、发出请求的 IP 地址、谁提出了请求、何时提出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《AWS CloudTrail用户指南》。

使用 CloudTrail

CloudTrail 在您创建账户AWS 账户时已在您的账户上启用。当 AWS FIS 中发生活动时，该活动会与其他AWS服务 CloudTrail 事件一起记录在事件历史记录中。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，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录 AWS 账户 中的事件（包括 AWS FIS 事件），请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪记录在 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 Simple Storage Service（Amazon S3）存储桶。此外，您可以配置其他AWS服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- 为您的 AWS 账户创建跟踪。
- CloudTrail 支持的服务和集成
- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有 AWS FIS 操作均由《故障注入服务 API 参考》记录 CloudTrail 并记录在《AWS故障注入服务 API 参考》中。有关在目标资源上执行的实验操作，请查看拥有该资源的服务的 API 参考文档。例如，有关在 Amazon EC2 实例上执行的操作，请参阅 Amazon EC2 API 参考。

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根凭证还是用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 AWS FIS 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下是调用 AWS FIS `StopExperiment` 操作的 CloudTrail 日志条目示例。


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
    "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/example",
        "accountId": "111122223333",
        "userName": "example"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2020-12-03T09:40:42Z",
        "mfaAuthenticated": "false"
```

```
      }
    }
  },
  "eventTime": "2020-12-03T09:44:20Z",
  "eventSource": "fis.amazonaws.com",
  "eventName": "StopExperiment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.51.100.25",
  "userAgent": "Boto3/1.22.9 Python/3.8.13 Linux/5.4.186-113.361.amzn2int.x86_64
Botocore/1.25.9",
  "requestParameters": {
    "clientToken": "1234abc5-6def-789g-012h-ijklm34no56p",
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",
    "tags": {}
  },
  "responseElements": {
    "experiment": {
      "actions": {
        "exampleAction1": {
          "actionId": "aws:ec2:stop-instances",
          "duration": "PT10M",
          "state": {
            "reason": "Initial state",
            "status": "pending"
          },
          "targets": {
            "Instances": "exampleTag1"
          }
        },
        "exampleAction2": {
          "actionId": "aws:ec2:stop-instances",
          "duration": "PT10M",
          "state": {
            "reason": "Initial state",
            "status": "pending"
          },
          "targets": {
            "Instances": "exampleTag2"
          }
        }
      },
      "creationTime": 1605788649.95,
      "endTime": 1606988660.846,
      "experimentTemplateId": "ABCDE1fgHIJkLmNop",
```

```
      "id": "ABCDE1fgHIJkLmNop",
      "roleArn": "arn:aws:iam::111122223333:role/AllowFISActions",
      "startTime": 1605788650.109,
      "state": {
        "reason": "Experiment stopped",
        "status": "stopping"
      },
      "stopConditions": [
        {
          "source": "aws:cloudwatch:alarm",
          "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:example"
        }
      ],
      "tags": {},
      "targets": {
        "ExampleTag1": {
          "resourceTags": {
            "Example": "tag1"
          },
          "resourceType": "aws:ec2:instance",
          "selectionMode": "RANDOM(1)"
        },
        "ExampleTag2": {
          "resourceTags": {
            "Example": "tag2"
          },
          "resourceType": "aws:ec2:instance",
          "selectionMode": "RANDOM(1)"
        }
      }
    }
  },
  "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
  "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"

}
```

以下是 AWS FIS 在包含 FIS 操作的实验中调用的 API 操作的示例 CloudTrail 日志条目。aws:ssm:send-command AWSuserIdentity 元素反映了使用临时凭证发出的请求，此凭证可通过代入角色获得。userName 会显示代入角色的名称。实验 ID (EXP21nT17WMzA6dnUgz) 会作为带入角色 ARN 的一部分，出现在 principalId 中。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROATZZZ4JPIXUEXAMPLE:EXP21nT17WMzA6dnUgz",
        "arn": "arn:aws:sts::111122223333:assumed-role/AllowActions/
EXP21nT17WMzA6dnUgz",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROATZZZ4JPIXUEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/AllowActions",
                "accountId": "111122223333",
                "userName": "AllowActions"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-05-30T13:23:19Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "fis.amazonaws.com"
    },
    "eventTime": "2022-05-30T13:23:19Z",
    "eventSource": "ssm.amazonaws.com",
    "eventName": "ListCommands",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "fis.amazonaws.com",
    "userAgent": "fis.amazonaws.com",
    "requestParameters": {
        "commandId": "51dab97f-489b-41a8-a8a9-c9854955dc65"
    },
    "responseElements": null,
    "requestID": "23709ced-c19e-471a-9d95-cf1a06b50ee6",
    "eventID": "145fe5a6-e9d5-45cc-be25-b7923b950c83",
    "readOnly": true,
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

AWS 故障注入服务中的安全性

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 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 AWS 故障注入AWS 服务的合规计划，请参阅合规性计划范围内按合规计划AWS。
- 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 AWS FIS 时如何应用分担责任模型。以下主题向您介绍如何配置 AWS FIS 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 AWS 的 FIS 资源。

内容

- AWS 故障注入服务中的数据保护
- AWS 故障注入服务的身份和访问管理
- AWS 故障注入服务中的基础设施安全
- 使用接口 VPC 终端节点访问 AWS FIS ()AWS PrivateLink

AWS 故障注入服务中的数据保护

分 AWS 担责任模型适用于 AWS 故障注入服务中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础架构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

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

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

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

静态加密

AWS FIS 始终对您的静态数据进行加密。 AWS FIS 中的数据使用透明的服务器端加密进行静态加密。这样可以帮助减少在保护敏感数据时涉及的操作负担和复杂性。通过静态加密，您可以构建符合加密合规性和法规要求的安全敏感型应用程序。

传输中加密

AWS FIS 对服务与其他集成 AWS 服务之间传输的数据进行加密。在 AWS FIS 和集成服务之间传递的所有数据均使用传输层安全 (TLS) 进行加密。有关其他集成 AWS 服务的更多信息，请参阅支持的 AWS 服务。

AWS 故障注入服务的身份和访问管理

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 AWS FIS 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

内容

- 受众
- 使用身份进行身份验证

- 使用策略管理访问
- AWS 故障注入服务如何与 IAM 配合使用
- AWS 故障注入服务策略示例
- 为 AWS 故障注入服务使用服务相关角色
- AWSAWS 故障注入服务的托管策略

受众

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

服务用户-如果您使用 AWS FIS 服务完成工作，则您的管理员会为您提供所需的凭证和权限。当你使用更多 AWS 的 FIS 功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。

服务管理员-如果您负责公司 AWS 的 FIS 资源，则可能拥有对 AWS FIS 的完全访问权限。您的工作 AWS 是确定您的服务用户应访问哪些 FIS 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。

IAM 管理员 — 如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理 AWS FIS 的访问权限。

使用身份进行身份验证

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

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

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

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

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

AWS 账户 root 用户

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

联合身份

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

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

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

IAM 用户和群组

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

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

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

IAM 角色

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

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的为第三方身份提供商创建角色。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的 权限集。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解角色和基于资源的跨账户访问策略之间的区别，请参阅 IA M 用户指南中的跨账户资源访问。
- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service (Amazon S3) 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向

EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

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

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

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

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

基于身份的策略

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

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式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式策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资

源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。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。

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

访问控制列表 (ACL)

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

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

其他策略类型

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

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

多个策略类型

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

AWS 故障注入服务如何与 IAM 配合使用

在使用 IAM 管理对 AWS FIS 的访问权限之前，请先了解有哪些 IAM 功能可用于 AWS FIS。

您可以与 AWS 故障注入服务一起使用的 IAM 功能

IAM 功能	AWS 金融情报局支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是
ACL	否
ABAC（策略中的标签）	是
临时凭证	是
主体权限	是
服务角色	是
服务相关角色	是

要全面了解 AWS FIS 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。

FIS 基于身份的政策 AWS

支持基于身份的策略	是

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

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

FIS 基于身份的政策示例 AWS

要查看 AWS FIS 基于身份的策略的示例，请参阅。AWS 故障注入服务策略示例

金融服务机构内部 AWS 基于资源的政策

支持基于资源的策略	否

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

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

AWS 金融情报机构的政策行动

支持策略操作	是

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

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

在策略中包含操作以授予执行关联操作的权限。

要查看 AWS FIS 操作列表，请参阅《服务授权参考》中的 “ AWS 故障注入服务” 定义的操作。

AWS FIS 中的策略操作在操作前使用以下前缀：

```
fis
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

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

```
"Action": "fis:List*"
```

AWS 金融情报机构的政策资源

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。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称 (ARN) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

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

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

某些 AWS FIS API 操作支持多种资源。要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [
    "resource1",
```

```
    "resource2"
]
```

要查看 AWS FIS 资源类型及其 ARN 的列表，请参阅《服务授权参考》中的“ AWS 故障注入服务” 定义的资源类型。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅AWS 故障注入服务定义的操作。

FIS 的政策条件 AWS 密钥

支持特定于服务的策略条件键	是

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

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。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。

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

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

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

要查看 AWS FIS 条件键列表，请参阅《服务授权参考》中的“ AWS 故障注入服务” 的条件密钥。要了解可以使用条件键的操作和资源，请参阅AWS 故障注入服务定义的操作。

要查看 AWS FIS 基于身份的策略的示例，请参阅。AWS 故障注入服务策略示例

FIS 中的 AWS ACL

支持 ACL	否

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

带有 FIS 的 ABA AWS C

支持 ABAC（策略中的标签）	是

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

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

要基于标签控制访问，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》中的什么是 ABAC？。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

要查看基于身份的策略示例，以根据资源标签限制资源访问权限，请参阅 示例：使用标签控制资源使用率。

在 AWS FIS 中使用临时证书

支持临时凭证	是

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

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的 切换到角色（控制台）。

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

FIS 的跨服务主体 AWS 权限

支持转发访问会话 (FAS)	是

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

适用于 AWS FIS 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。

FIS 的服务相关角色 AWS

支持服务相关角色	是

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

有关创建或管理 AWS FIS 服务相关角色的详细信息，请参阅。为 AWS 故障注入服务使用服务相关角色

AWS 故障注入服务策略示例

默认情况下，用户和角色无权创建或修改 AWS FIS 资源。他们也无法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源

执行操作的权限，IAM 管理员可以创建 IAM policy。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅 IAM 用户指南中的 创建 IAM 策略。

有关 AWS FIS 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 “AWS 故障注入服务” 的操作、资源和条件键。

内容

- 策略最佳实践
- 示例：使用 AWS FIS 控制台
- 示例：列出可用 AWS 的 FIS 操作
- 示例：为特定操作创建实验模板
- 示例：开始实验
- 示例：使用标签控制资源使用率
- 示例：删除带有特定标签的实验模板
- 示例：允许用户查看自己的权限
- 示例：使用 ec2:InjectApiError 条件键
- 示例：使用 aws:s3:bucket-pause-replication 条件键

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS FIS 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 policy 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 policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，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 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件 。

- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM policy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 IAM Acess Analyzer 策略验证。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实操。

示例：使用 AWS FIS 控制台

要访问 AWS 故障注入服务控制台，您必须拥有一组最低权限。这些权限必须允许您在中列出和查看有关 AWS FIS 资源的详细信息。 AWS 账户如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

以下示例策略授予使用 FIS 控制台列出和查看所有 AWS AWS FIS 资源的权限，但不允许创建、更新或删除这些资源。它还授予查看可在实验模板中指定的所有 AWS FIS 操作所使用的可用资源的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "FISReadOnlyActions",
            "Effect": "Allow",
            "Action": [
                "fis:List*",
                "fis:Get*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AdditionalReadOnlyActions",
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:Describe*",
                "ssm:Get*",
                "ssm:List*",
                "ec2:DescribeInstances",
                "rds:DescribeDBClusters",
                "ecs:DescribeClusters",
                "ecs:ListContainerInstances",
                "eks:DescribeNodegroup",
                "cloudwatch:DescribeAlarms",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PermissionsToCreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "fis.amazonaws.com"
                }
            }
        }
    ]
}
```

示例：列出可用 AWS 的 FIS 操作

以下策略授予列出可用 AWS FIS 操作的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fis:ListActions"
            ],
            "Resource": "arn:aws:fis:*:*:action/*"
        }
    ]
```

```
}
```

示例：为特定操作创建实验模板

以下策略授予为 aws:ec2:stop-instances 操作创建实验模板的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "fis:CreateExperimentTemplate"
      ],
      "Resource": [
        "arn:aws:fis:*:*:action/aws:ec2:stop-instances",
        "arn:aws:fis:*:*:experiment-template/*"
      ]
    },
    {
      "Sid": "PolicyPassRoleExample",
      "Effect": "Allow",
      "Action": [
          "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::account-id:role/role-name"
      ]
    }
  ]
}
```

示例：开始实验

以下策略授予使用指定的 IAM 角色和实验模板开始实验的权限。它还允许 AWS FIS 代表用户创建服务相关角色。有关更多信息，请参阅 为 AWS 故障注入服务使用服务相关角色。

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

```
      "Effect": "Allow",
      "Action": [
          "fis:StartExperiment"
      ],
      "Resource": [
        "arn:aws:fis:*:*:experiment-template/experiment-template-id",
        "arn:aws:fis:*:*:experiment/*"
      ]
    },
    {
        "Sid": "PolicyExampleforServiceLinkedRole",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": "fis.amazonaws.com"
            }
        }
    }
  ]
}
```

示例：使用标签控制资源使用率

以下策略授予使用带有 Purpose=Test 标签的实验模板运行实验的权限。但不授予创建或修改实验模板的权限，也不授予使用无指定标签模板运行实验的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment-template/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

示例：删除带有特定标签的实验模板

以下策略授予删除带有 Purpose=Test 标签的实验模板的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fis:DeleteExperimentTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

示例：允许用户查看自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
```

```
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

示例：使用 **ec2:InjectApiError** 条件键

以下示例策略使用 ec2:FisTargetArns 条件键来限定目标资源的范围。该政策允许 AWS FIS 采取行动aws:ec2:api-insufficient-instance-capacity-error和aws:ec2:asg-insufficient-instance-capacity-error.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:api-insufficient-instance-capacity-error",
                    ],
                    "ec2:FisTargetArns": [
                        "arn:aws:iam:*:*:role:role-name"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:asg-insufficient-instance-capacity-error"
                    ],
                    "ec2:FisTargetArns": [

 "arn:aws:autoscaling:*:*:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:DescribeAutoScalingGroups",
            "Resource": "*"
        }
    ]
}
```

示例：使用 aws:s3:bucket-pause-replication 条件键

以下示例策略使用S3:IsReplicationPauseRequest条件密钥来允许PutReplicationConfiguration且GetReplicationConfiguration仅当 FIS 在 AWS FIS 操作的上下文中使用时才允许。 AWS aws:s3:bucket-pause-replication

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "S3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "StringEquals": {
                    "s3:DestinationRegion": "region"
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "S3:PutReplicationConfiguration",
                "S3:GetReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "BoolIfExists": {
                    "s3:IsReplicationPauseRequest": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "S3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
    }
```

为 AWS 故障注入服务使用服务相关角色

AWS 故障注入服务使用 AWS Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种与 AWS FIS 直接关联的独特的 IAM 角色。服务相关角色由 AWS FIS 预定义，其中包含此服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可以更轻松地设置 AWS FIS，因为您不必手动添加必要的权限来管理实验的监控和资源选择。 AWS FIS 定义其服务相关角色的权限，除非另有定义，否则只有 AWS FIS 可以担任其角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其它 IAM 实体。

除服务相关角色外，您还必须指定一个 IAM 角色，授予对实验模板中指定为目标的资源的修改权限。有关更多信息，请参阅 适用于 AWS FIS 实验的 IAM 角色。

只有在先删除相关 资源后，才能删除服务相关角色。这可以保护您 AWS 的 FIS 资源，因为您不能无意中删除访问这些资源的权限。

FIS 的服务相关角色权限 AWS

AWS FIS 使用名为的服务关联角色AWSServiceRoleForFIS来管理实验的监控和资源选择。

AWSServiceRoleForFIS服务相关角色信任以下服务来代入该角色：

- `fis.amazonaws.com`

AWSServiceRoleForFIS服务相关角色使用托管策略 AmazonFi ServiceRole s 政策。该政策使 AWS FIS 能够管理实验的监测和资源选择。有关更多信息，请参阅《AWS 托管ServiceRole政策参考》中的AmazonFis 政策。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。要成功创建AWSServiceRoleForFIS服务相关角色，您使用 AWS FIS 的 IAM 身份必须具有所需的权限。要授予所需的权限，请将以下策略附加到 IAM 身份。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "fis.amazonaws.com"
                }
            }
        }
    ]
}
```

有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 FIS 创建服务相关角色 AWS

您无需手动创建服务相关角色。当您在 AWS Management Console、或 AWS API 中启动 AWS FIS 实验时 AWS CLI，AWS FIS 会为您创建服务相关角色。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您开始 AWS FIS 实验时， AWS FIS 会再次为您创建服务相关角色。

编辑 FIS 的服务相关角色 AWS

AWS FIS 不允许您编辑AWSServiceRoleForFIS服务相关角色。创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 FIS 的服务相关角色 AWS

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样就没有未被主动监控或维护的未使用实体。但是，必须先清除服务相关角色的资源，然后才能手动删除它。

> **Note**
>
> 如果您尝试清理资源时， AWS FIS 服务正在使用该角色，则清理可能会失败。如果发生这种情况，请等待几分钟后重试。

清理使用 AWS 的 FIS 资源 AWSServiceRoleForFIS

请确保您目前没有运行任何实验。如有必要，请停止实验。有关更多信息，请参阅 停止实验。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除AWSServiceRoleForFIS服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

AWS FIS 服务相关角色支持的区域

AWS FIS 支持在提供服务的所有地区使用服务相关角色。有关更多信息，请参阅 AWS Fault Injection Service 端点和限额。

AWSAWS 故障注入服务的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AWS 托管政策：亚马逊 FiS ServiceRole 政策

此策略附加AWSServiceRoleForFIS到名为的服务相关角色上，允许 AWS FIS 管理实验的监控和资源选择。有关更多信息，请参阅 为 AWS 故障注入服务使用服务相关角色。

AWS 托管策略： AWSFaultInjectionSimulatorEC2Access

在实验角色中使用此策略授予 AWS FIS 运行使用适用于 Amazon EC2 AWS 的 FIS 操作的实验的权限。有关更多信息，请参阅 the section called “实验角色”。

要查看此策略的权限，请参阅AWS 托管策略参考AWSFaultInjectionSimulatorEC2Access中的。

AWS 托管策略： AWSFaultInjectionSimulatorECSAccess

在实验角色中使用此策略授予 AWS FIS 运行使用 Amazon EC AWS S 的 FIS 操作的实验的权限。有关更多信息，请参阅 the section called “实验角色”。

要查看此策略的权限，请参阅AWS 托管策略参考AWSFaultInjectionSimulatorECSAccess中的。

AWS 托管策略： AWSFaultInjectionSimulatorEKSAccess

在实验角色中使用此策略授予 AWS FIS 运行使用 Amazon EK AWS S 的 FIS 操作的实验的权限。有关更多信息，请参阅 the section called “实验角色”。

要查看此策略的权限，请参阅AWS 托管策略参考AWSFaultInjectionSimulatorEKSAccess中的。

AWS 托管策略： AWSFaultInjectionSimulatorNetworkAccess

在实验角色中使用此策略授予 AWS FIS 运行使用 FIS 联网操作的实验AWS 的权限。有关更多信息，请参阅 the section called “实验角色”。

要查看此策略的权限，请参阅AWS 托管策略参考AWSFaultInjectionSimulatorNetworkAccess中的。

AWS 托管策略： AWSFaultInjectionSimulatorRDSAccess

在实验角色中使用此策略授予 AWS FIS 运行使用 Amazon R AWS DS 的 FIS 操作的实验的权限。有关更多信息，请参阅 the section called “实验角色”。

要查看此策略的权限，请参阅AWS 托管策略参考AWSFaultInjectionSimulatorRDSAccess中的。

AWS 托管策略： AWSFaultInjectionSimulatorSSMAccess

在实验角色中使用此策略授予 AWS FIS 运行使用 S ystems Manager AWS 的 FIS 操作的实验的权限。有关更多信息，请参阅 the section called “实验角色”。

要查看此策略的权限，请参阅AWS 托管策略参考AWSFaultInjectionSimulatorSSMAccess中的。

AWS FIS 对 AWS 托管策略的更新

查看自该服务开始跟踪这些更改以来对 AWS FIS AWS 托管策略的更新的详细信息。

更改	描述	日期
AWSFaultInjectionSimulatorECSAccess – 对现有策略的更新	添加了允许 AWS FIS 解析 ECS 目标的权限。	2024 年 1 月 25 日
AWSFaultInjectionSimulatorNetworkAccess – 更新了现有策略	添加了允许 AWS FIS 使用aws:network:route-table-disrupt-cross-region-connectivity和aws:network:transit-gateway-disrupt-cross-region-connectivity操作运行实验的权限。	2024 年 1 月 25 日
AWSFaultInjectionSimulatorEC2Access – 更新了现有策略	增加了允许 AWS FIS 解析 EC2 实例的权限。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorEKSAccess – 更新了现有策略	增加了允许 AWS FIS 解析 EKS 目标的权限。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorRDSAccess – 更新了现有策略	增加了允许 AWS FIS 解析 RDS 目标的权限。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorEC2Access – 更新了现有策略	增加了允许 AWS FIS 在 EC2 实例上运行 SSM 文档和终止 EC2 实例的权限。	2023 年 6 月 2 日
AWSFaultInjectionSimulatorSSMAccess – 更新了现有策略	增加了允许 AWS FIS 在 EC2 实例上运行 SSM 文档的权限。	2023 年 6 月 2 日

更改	描述	日期
AWSFaultInjectionSimulatorECSAccess – 更新了现有策略	增加了允许 AWS FIS 使用新aws:ecs:task操作运行实验的权限。	2023 年 6 月 1 日
AWSFaultInjectionSimulatorEKSAccess – 更新了现有策略	增加了允许 AWS FIS 使用新aws:eks:pod操作运行实验的权限。	2023 年 6 月 1 日
AWSFaultInjectionSimulatorEC2Access：新策略	添加了一项政策，允许 AWS FIS 运行使用针对 Amazon EC2 AWS 的 FIS 操作的实验。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorECSAccess：新策略	添加了一项政策，允许 AWS FIS 在 Amazon ECS 上运行使用 AWS FIS 操作的实验。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorEKSAccess：新策略	添加了一项政策，允许 AWS FIS 在 Amazon EKS 上运行使用 AWS FIS 操作的实验。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorNetworkAccess：新策略	添加了一项政策，允许 AWS FIS 运行使用 AWS FIS 联网操作的实验。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorRDSAccess：新策略	添加了一项政策，允许 AWS FIS 运行使用适用于 Amazon R AWS DS 的 FIS 操作的实验。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorSSMAccess：新策略	添加了一项政策，允许 AWS FIS 运行使用 Systems Manager AWS 的 FIS 操作的实验。	2022 年 10 月 26 日
亚马逊 FiS ServiceRole 政策-对现有政策的更新	增加了允许 AWS FIS 描述子网的权限。	2022 年 10 月 26 日
亚马逊 FiS ServiceRole 政策-对现有政策的更新	增加了允许 AWS FIS 描述 EKS 集群的权限。	2022 年 7 月 7 日

更改	描述	日期
亚马逊 FiS ServiceRole 政策-对现有政策的更新	添加了允许 AWS FIS 列出和描述集群中任务的权限。	2022 年 2 月 7 日
亚马逊 FiS ServiceRole 政策-对现有政策的更新	为 `events:DescribeRule` 操作删除 `events:ManagedBy` 条件。	2022 年 1 月 6 日
亚马逊 FiS ServiceRole 政策-对现有政策的更新	增加了允许 AWS FIS 检索停止条件下使用的 CloudWatch 警报历史记录的权限。	2021 年 6 月 30 日
AWS FIS 开始追踪变更	AWS FIS 开始跟踪其 AWS 托管政策的变更	2021 年 3 月 1 日

AWS 故障注入服务中的基础设施安全

作为一项托管服务，AWS 故障注入服务受 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 ecurity Pillar Well-Architected Fram ework 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 AWS FIS。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（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

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

使用接口 VPC 终端节点访问 AWS FIS ()AWS PrivateLink

您可以通过创建接口 VPC 终端节点在您的 VPC 和 AWS 故障注入服务之间建立私有连接。VPC 终端节点由一项技术提供支持 AWS PrivateLink，该技术使您无需互联网网关、NAT 设备、VPN 连接或 Di AWS rect Connect 连接即可私密访问 AWS FIS API。您的 VPC 中的实例不需要公有 IP 地址即可与 AWS FIS API 通信。

每个接口端点均由子网中的一个或多个弹性网络接口表示。

有关更多信息，请参阅AWS PrivateLink 指南 AWS PrivateLink中的AWS 服务 通过访问。

AWS FIS VPC 终端节点的注意事项

在为 AWS FIS 设置接口 VPC 终端节点之前，请查看AWS PrivateLink 指南中的使用接口 VPC 终端节点访问和 AWS 服务 使用接口 VPC 终端节点。

AWS FIS 支持从您的 VPC 调用其所有 API 操作。

为 AWS FIS 创建接口 VPC 终端节点

您可以使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI) AWS 为 FIS 服务创建 VPC 终端节点。有关更多信息，请参阅 AWS PrivateLink 指南中的创建 VPC 端点。

使用以下服务名称 AWS 为 FIS 创建 VPC 终端节点：`com.amazonaws.region.fis`。

例如，如果您为终端节点启用私有 DNS， AWS 则可以使用该区域的默认 DNS 名称向 FIS 发出 API 请求。`fis.us-east-1.amazonaws.com`

为 AWS FIS 创建 VPC 终端节点策略

您可以将终端节点策略附加到控制对 AWS FIS 的访问的 VPC 终端节点。该策略指定以下信息：

- 可执行操作的主体。
- 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 VPC 端点的访问权限。

示例：针对特定 AWS FIS 操作的 VPC 终端节点策略

以下 VPC 终端节点策略向所有委托人授予对所有资源执行列出 AWS 的 FIS 操作的访问权限。

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "fis:ListExperimentTemplates",
            "fis:StartExperiment",
```

```
            "fis:StopExperiment",
            "fis:GetExperiment"
         ],
         "Resource":"*",
         "Principal":"*"
      }
   ]
}
```

示例：拒绝特定用户访问的 VPC 终端节点策略 AWS 账户

以下 VPC 终端节点策略拒绝对所有操作和资源的指定 AWS 账户 访问权限，但允许所有其他 AWS 账户 用户访问所有操作和资源。

```
{
   "Statement":[
      {
         "Effect": "Allow",
         "Action": "*",
         "Resource": "*",
         "Principal": "*"
      },
      {
         "Effect":"Deny",
         "Action": "*",
         "Resource": "*",
         "Principal": {
           "AWS": [ "123456789012" ]
         }
      }
   ]
}
```

标记 AWS FIS 资源

标签是您或 AWS 分配给 AWS 资源的元数据标签。每个标签均包含一个键 和一个值。对于您分配的标签，需要定义键和值。例如，您可以将键定义为 `purpose`，并将资源的值定义为 `test`。

标签可帮助您：

- 标识和整理您的 AWS 资源。许多 AWS 服务支持标记，因此，您可以将同一标签分配给来不同服务的资源，以指示这些资源是相关的。
- 控制对 AWS 资源的访问。有关更多信息，请参阅 IAM 用户指南中的使用标签控制访问。

添加标签限制

以下基本限制适用于 AWS FIS 资源上的标签：

- 您可以分配给资源的最大标签数量：50
- 最大密钥长度：128 个 Unicode 字符
- 最大值长度：256 个 Unicode 字符
- 键和值的有效字符：a-z、A-Z、0-9、空格和以下字符： _ . : / = + - 和 @
- 键和值区分大小写
- `aws:` 只能保留用于 AWS，不能作为键的前缀。

使用标签

以下 AWS Fault Injection Service (AWS FIS) 资源支持标记：

- 操作
- 实验
- 实验模板

您可以使用控制台处理实验和实验模板的标签。有关更多信息，请参阅下列内容：

- 标记实验
- 标记实验模板

您可以运行以下 AWS CLI 命令，处理操作、实验和实验模板的标签：

- tag-resource：添加资源标签。
- untag-resource：删除资源标签。
- list-tags-for-resource— 列出特定资源的标签。

AWS 故障注入服务的配额和限制

您的每项 AWS 服务 AWS 账户 都有默认配额，以前称为限制。除非另有说明，否则，每个限额是区域特定的。您可以请求增加某些配额，但并非所有配额都能增加。

要查看 AWS FIS 的配额，请打开 Ser vice Quotas 控制台。在导航窗格中，依次选择 AWS 服务和 AWS Fault Injection Service。

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。

您 AWS 账户 有以下与 AWS FIS 相关的配额。

名称	默认值	可调整	描述
操作持续时间（以小时计）	每个受支持的区域：12 个	否	当前区域中此账户下允许运行一项操作的最大小时数量。
每个实验模板的操作数	每个受支持的区域：20 个	否	您可以在当前区域中的此账户下以实验模板创建的最大操作数量。
活跃的实验数	每个受支持的区域：5 个	否	您可以在当前区域中的此账户下同时运行的最大活跃实验数量。
已完成实验数据留存（以天计）	每个受支持的区域：120 个	否	在当前区域， AWS FIS 允许在此账户中保留有关已完成实验的数据的最大天数。
实验持续时间（以小时计）	每个受支持的区域：12 个	否	当前区域中的此账户下允许运行一个实验的最大小时数量。

名称	默认值	可调整	描述
实验模板	每个受支持的区域：500 个	否	您可以在当前区域中的此账户下创建的最大实验模板数量。
aws:network:route-table-disrupt-cross-region-connectivity 中的最大托管前缀列表数量	每个受支持的区域：15 个	否	aws: network: route-table-在每个操作中允许的最大托管前缀列表数量。disrupt-cross-region-connectivity
aws:network:route-table-disrupt-cross-region-connectivity 中的最大路由表数量	每个受支持的区域：10 个	否	aws: network: route-table-disrupt-cross-region-connectivity 在每个操作中允许的最大路由表数量。
aws:network:route-table-disrupt-cross-region-connectivity 中的最大路由数量	每个受支持的区域：200 个	否	aws: network: route-table-disrupt-cross-region-connectivity 每次操作允许的最大路由数量。
每个实验的并行操作数	每个受支持的区域：10 个	否	您可以在当前区域中的此账户下实验中平行运行的最大操作数量。
每个实验模板的停止条件数	每个受支持的区域：5 个	否	您可以在当前区域中的此账户下添加到实验模板的最大停止条件数量。
aws:ec2:asg-insufficient-instance-capacity-error 的目标自动扩缩组	每个受支持的区域：5 个	是	每个实验使用标签识别目标时 aws: ec2: asg-insufficient-instance-capacity-error 可以瞄准的最大 Auto Scaling 组数。

名称	默认值	可调整	描述
aws:s3:bucket-pause-replication 的目标存储桶	每个受支持的区域：20 个	是	每个实验使用标签识别目标时 aws: s3: bucket-pause-replication 可以瞄准的最大 S3 存储桶数量。
aws:ecs:drain-container-instances 的目标集群	每个受支持的区域：5 个	是	每个实验使用标签识别目标时 aws: ecs: drain-container-instances 可以瞄准的最大集群数量。
aws:rds:failover-db-cluster 的目标集群	每个受支持的区域：5 个	是	每个实验使用标签识别目标时 aws: rds: failover-db-cluster 可以瞄准的最大集群数量。
aws:rds:reboot-db-instances 的目标数据库实例	每个受支持的区域：5 个	是	每个实验使用标签识别目标时 aws: rds: reboot-db-instances 可以瞄准的最大数据库实例数量。
aws:ec2:reboot-instances 的目标实例	每个受支持的区域：5 个	是	每个实验中，当您使用标签标识目标时 aws:ec2:reboot-instances 可以确定为目标的最大实例数量。
aws:ec2:stop-instances 的目标实例	每个受支持的区域：5 个	是	每个实验中，当您使用标签标识目标时 aws:ec2:stop-instances 可以确定为目标的最大实例数量。

名称	默认值	可调整	描述
aws:ec2:terminate-instances 的目标实例	每个受支持的区域：5 个	是	每个实验中，当您使用标签标识目标时 aws:ec2:terminate-instances 可以确定为目标的最大实例数量。
aws:ssm:send-command 的目标实例	每个受支持的区域：5 个	是	每个实验中，当您使用标签标识目标时 aws:ssm:send-command 可以确定为目标的最大实例数量。
aws:eks:terminate-nodegroup-instances 的目标节点组	每个受支持的区域：5 个	是	每个实验使用标签识别目标时 aws: eks: terminate-nodegroup-instances 可以瞄准的最大节点组数量。
aws:eks:pod-cpu-stress 的目标 Pod	每个受支持的区域：50 个	是	在每个实验中，当你使用参数识别目标时，aws: eks: pod-cpu-stress 可以瞄准的最大 Pod 数量。
aws:eks:pod-delete 的目标 Pod	每个受支持的区域：50 个	是	当您使用参数确定目标时，aws:eks:pod-delete 在每次实验中可以确定为目标的最大 Pod 数。
aws:eks:pod-io-stress 的目标 Pod	每个受支持的区域：50 个	是	在每个实验中，当你使用参数识别目标时，aws: eks: pod-io-stress 可以瞄准的最大 Pod 数量。

名称	默认值	可调整	描述
aws:eks:pod-memory-stress 的目标 Pod	每个受支持的区域：50 个	是	在每个实验中，当你使用参数识别目标时，aws:eks: pod-memory-stress 可以瞄准的最大 Pod 数量。
aws:eks:pod-network-blackhole-port 的目标 Pod	每个受支持的区域：50 个	是	在每个实验中，当你使用参数识别目标时，aws:eks: pod-network-blackhole-port 可以瞄准的最大 Pod 数量。
aws:eks:pod-network-latency 的目标 Pod	每个受支持的区域：50 个	是	在每个实验中，当你使用参数识别目标时，aws:eks: pod-network-latency 可以瞄准的最大 Pod 数量。
aws:eks:pod-network-packet-loss 的目标 Pod	每个受支持的区域：50 个	是	在每个实验中，当你使用参数识别目标时，aws:eks: pod-network-packet-loss 可以瞄准的最大 Pod 数量。
aws ReplicationGroups 的目标:elasticache: interrut-cluster-az-power	每个支持的区域：5 个	是	每个实验使用标签/参数识别目标时 ReplicationGroups ，aws:elasticache: interrupt-cluster-az-power 可以瞄准的最大数量。

名称	默认值	可调整	描述
aws: ec2: send-s SpotInstances pot实例中断的目标	每个支持的区域：5 个	是	每个实验使用标签识别目标时 SpotInstances ，aws: ec2: send-spot-instance-interruptions 可以瞄准的最大数量。
aws:network:disrupt-connectivity 的目标子网	每个受支持的区域：5 个	是	每个实验中，当您使用标签标识目标时 aws:network:disrupt-connectivity 可以确定为目标的最大子网数量。大于 5 的配额仅适用于参数范围:all。如果您需要为其他瞄准镜类型提供更高的配额，请通过 https://console.aws.amazon.com/support/home#/ 联系客户支持。
aws:network:route-table-disrupt-cross-region-connectivity 的目标子网	每个受支持的区域：6 个	是	每个实验使用标签识别目标时，aws: network: route-table-disrupt-cross-region-connectivity 可以瞄准的最大子网数量。
适用于 aws:ecs:stop-task 的目标任务	每个受支持的区域：5 个	是	每个实验中，当您使用标签标识目标时，aws:ecs:stop-task 可以标识为目标的最大任务数量。

名称	默认值	可调整	描述
aws:ecs:stop-task 的目标任务	每个受支持的区域：5 个	是	每个实验使用标签/参数识别目标时 aws: ecs: task-cpu-stress 可以定位的最大任务数。
aws:ecs:task-io-stress 的目标任务	每个受支持的区域：5 个	是	每个实验使用标签/参数识别目标时 aws: ecs: task-io-stress 可以定位的最大任务数。
aws:ecs:task-io-stress 的目标任务	每个受支持的区域：5 个	是	每个实验使用标签/参数识别目标时 aws: ecs: task-kill-process 可以定位的最大任务数。
aws:ecs:task-network-blackhole-port 的目标任务	每个受支持的区域：5 个	是	每个实验使用标签/参数识别目标时 aws: ecs: task-network-blackhole-port 可以定位的最大任务数。
aws:ecs:task-network-latency 的目标任务	每个受支持的区域：5 个	是	每个实验使用标签/参数识别目标时 aws: ecs: task-network-latency 可以定位的最大任务数。
aws:ecs:task-network-packet-loss 的目标任务	每个受支持的区域：5 个	是	每个实验使用标签/参数识别目标时 aws: ecs: task-network-packet-loss 可以定位的最大任务数。

名称	默认值	可调整	描述
aws: network: transit-g TransitGateways ateway-druspt-drustion-	每个支持的区域：5 个	是	每个实验使用标签识别目标时，aws: network: transit-gateway-disrupt-cross-region-connectivity 可以瞄准的最大传输网关数量。
每个实验模板的目标账户配置	每个受支持的区域：10 个	是	您可以在当前区域中的此账户下为实验模板创建的最大目标账户配置数量。
aws: dynamodb: 操作的目标表 global-table-pause-replication	每个支持的区域：5 个	是	每个实验中 aws: dynamodb: global-table-pause-replication 可以定位的最大全局表数量。

您对 AWS FIS 的使用受以下额外限制的约束：

名称	限制
`aws:elasticache:interrupt-cluster-az-power` 行动目标	每个区域的每个账户每天只能受损 10 个`aws:elasticache:redis-replicationgroup` 集群。您可以通过在 AWS Support Center 控制台中创建支持案例来申请加薪。

文档历史记录

下表描述了对 AWS Fault Injection Service 用户指南做出的重要文档更新。

变更	说明	日期
新操作	现在，您可以使用该 `aws:dynamodb:global-table-pause-replication` 操作来暂停目标全局表与其副本表之间的数据复制。该`aws:dynamodb:encrypted-global-table-pause-replication` 操作将不再受支持。	2024 年 4 月 24 日
新的动作模式实验选项	您可以将操作模式设置为`skip-all`，以便在运行实验之前生成目标预览。	2024 年 3 月 13 日
AWS 托管策略更新	AWS FIS 更新了现有的托管策略。	2024 年 1 月 25 日
新场景和操作	现在，您可以使用 AWS FIS 场景跨区域:连接和可用区可用性：电源中断。	2023 年 11 月 30 日
新操作	您现在可以使用 aws:ec2:asg-insufficient-instance-capacity-error 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 aws:ec2:api-insufficient-instance-capacity-error 操作。	2023 年 11 月 30 日

新操作	您现在可以使用 aws:network:route-table-disrupt-cross-region-connectivity 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 aws:network:transit-gateway-disrupt-cross-region-connectivity 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 aws:dynamodb:encrypted-global-table-pause-replication 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 aws:s3:bucket-pause-replication 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 aws:elasticache:interrupt-cluster-az-power 操作。	2023 年 11 月 30 日
新实验选项	现在，您可以使用 AWS FIS 实验选项进行账户定位和空白目标分辨率。	2023 年 11 月 27 日
AWS FIS的名称变更	已将服务名称更新为 AWS 故障注入服务。	2023 年 11 月 15 日
AWS 托管策略更新	AWS FIS 更新了现有的托管策略。	2023 年 11 月 13 日
新场景库	现在，您可以使用 AWS FIS 场景库功能。	2023 年 11 月 7 日
新实验调度器	您现在可以使用 AWS FIS 实验调度器功能。	2023 年 11 月 7 日
AWS 托管策略更新	AWS FIS 更新了现有的托管策略。	2023 年 6 月 2 日

新操作	您可以执行 aws:ecs:task 和 aws:eks:pod 新操作。	2023 年 6 月 1 日
AWS 托管策略更新	AWS FIS 更新了现有的托管策略。	2023 年 6 月 1 日
新的预配置 SSM 文档	您可以使用以下预先配置的 SSM 文档： AWSFIS-Run-Disk-Fill。	2023 年 4 月 28 日
新操作	您可以执行 aws:ebs:pause-volume-io 操作，暂停目标卷和附加实例之间的 I/O。	2023 年 1 月 27 日
新操作	您可以执行 aws:network:disrupt-connectivity 操作，拒绝流向目标子网的特定类型流量。	2022 年 10 月 26 日
新操作	您可以使用该aws:eks:inject-kubernetes-custom-resource操作在单个目标集群上运行 ChaosMesh 或 Litmus 实验。	2022 年 7 月 7 日
实验日志记录	您可以将实验模板配置为将实验活动日志发送到 L CloudWatch ogs 或 S3 存储桶。	2022 年 2 月 28 日
新通知	当实验状态发生变化时， AWS FIS 会发出通知。这些通知通过 Amazon 作为事件提供 EventBridge。	2022 年 2 月 24 日
新操作	您可以执行 aws:ecs:stop-task 操作，停止指定任务。	2022 年 2 月 9 日

新操作	您可以执行 aws:cloudwatch:assert-alarm-state 操作，验证指定警报是否处于指定警报状态之一。	2021 年 11 月 5 日
新的预配置 SSM 文档	你可以使用以下预先配置的 SSM 文档： AWSFIS-run-io-Stress、-Run-Network-Blackhold-Port、-Run-Network-Latency-Sources、-Run-Network-Packet-Latency-Sources、-Run-Network-Packet-Latency-S AWSFIS ources、-Run- AWSFIS AWSFIS AWSFIS	2021 年 11 月 4 日
新操作	您可以执行 aws:ec2:send-spot-instance-interruptions 操作，向目标竞价型实例发送竞价型实例中断通知，然后中断目标竞价型实例。	2021 年 10 月 20 日
新操作	您可以执行 aws:ssm:start-automation-execution 操作，启动 Automation 运行手册执行程序。	2021 年 9 月 17 日
初始版本	《AWS 故障注入服务用户指南》的初始版本。	2021 年 3 月 15 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。