



用户指南

AWS 故障注入服务



AWS 故障注入服务: 用户指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

什么 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	3
规划实验	4
基本原则和指南	4
实验规划指南	5
实验模板组件	7
模板语法	7
开始使用	8
操作	8
操作语法	8
操作标识符	9
操作参数	10
操作目标	10
操作持续时间	11
操作示例	11
Targets	14
目标语法	15
资源类型	16
标识目标资源	17
选择模式	21
示例目标	22
示例筛选条件 :	23
停止条件	28
停止条件语法	28
了解更多	28
实验角色	29
前提条件	29
选项 1 : 创建实验角色并附加 AWS 托管策略	31

选项 2：创建实验角色并添加内联策略文档	31
实验报告配置	33
实验报告配置语法	35
实验报告权限	36
实验报告最佳实践	38
实验选项	39
账户定位	40
空目标解析模式	41
操作模式	41
操作参考	42
故障注入操作	43
aws:fis:inject-api-internal-error	43
aws:fis:inject-api-throttle-error	44
aws:fis:inject-api-unavailable-error	44
恢复行动	45
aws:arc:start-zonal-autoshift	45
等待动作	46
aws:fis:wait	46
亚马逊的 CloudWatch 行动	47
aws:cloudwatch:assert-alarm-state	47
Amazon DynamoDB 操作	47
aws:dynamodb:global-table-pause-replication	47
亚马逊 Aurora DSQL 操作	51
aws:dsql:cluster-connection-failure	51
Amazon EBS 操作	52
aws:ebs:pause-volume-io	52
aws:ebs:volume-io-latency	53
Amazon EC2 操作	54
aws:ec2:api-insufficient-instance-capacity-error	54
aws:ec2:asg-insufficient-instance-capacity-error	55
aws:ec2:reboot-instances	56
aws:ec2:send-spot-instance-interruptions	56
aws:ec2:stop-instances	57
aws:ec2:terminate-instances	58
Amazon ECS 操作	58
aws:ecs:drain-container-instances	59

aws:ecs:stop-task	60
aws:ecs:task-cpu-stress	60
aws:ecs:task-io-stress	61
aws:ecs:task-kill-process	62
aws:ecs:task-network-blackhole-port	62
aws:ecs:task-network-latency	63
aws:ecs:task-network-packet-loss	64
Amazon EKS 操作	66
aws:eks:inject-kubernetes-custom-resource	66
aws:eks:pod-cpu-stress	67
aws:eks:pod-delete	68
aws:eks:pod-io-stress	69
aws:eks:pod-memory-stress	70
aws:eks:pod-network-blackhole-port	71
aws:eks:pod-network-latency	72
aws:eks:pod-network-packet-loss	73
aws:eks:terminate-nodegroup-instances	74
亚马逊的 ElastiCache 行动	75
aws:elasticache:replicationgroup-interrupt-az-power	75
亚马逊 Kinesis Data Streams 操作	76
aws:kinesis:stream-provisioned-throughput-exception	76
aws:kinesis:stream-expired-iterator-exception	77
AWS Lambda 行动	77
aws:lambda:invocation-add-delay	77
aws:lambda:invocation-error	78
aws:lambda:invocation-http-integration-response	79
亚马逊 MemoryDB 操作	79
aws:memorydb:multi-region-cluster-pause-replication	80
网络操作	80
aws:network:disrupt-connectivity	81
aws:network:route-table-disrupt-cross-region-connectivity	82
aws:network:transit-gateway-disrupt-cross-region-connectivity	83
aws:network:disrupt-vpc-endpoint	84
Amazon RDS 操作	85
aws:rds:failover-db-cluster	85
aws:rds:reboot-db-instances	86

Amazon S3 操作	86
aws:s3:bucket-pause-replication	87
Systems Manager 操作	87
aws:ssm:send-command	88
aws:ssm:start-automation-execution	88
AWS Direct Connect 行动	89
aws:directconnect:virtual-interface-disconnect	90
SSM 文档操作	90
执行 aws:ssm:send-command 操作。	91
预先配置的 AWS FIS SSM 文档	92
示例	100
限制	101
回滚脚本	101
问题排查	102
ECS 任务操作	103
操作	103
限制	104
要求	104
脚本参考版本	107
实验模板示例	109
EKS Pod 操作	110
操作	111
限制	112
要求	113
创建实验角色	113
配置 Kubernetes 服务账户	113
授予 IAM 用户和角色访问 Kubernetes 的权限 APIs	114
容器组 (pod) 容器映像	115
实验模板示例	118
AWS Lambda 行动	119
操作	119
限制	120
先决条件	120
配置 Lambda 函数	122
配置实 AWS FIS 验	122
日志记录	122

高级主题	124
AWS FIS Lambda 扩展版本	129
管理实验模板	133
创建实验模板	133
查看实验模板	135
生成目标预览	136
通过模板开始实验	137
更新实验模板	137
标记实验模板	138
删除实验模板	138
示例 模板	139
根据过滤器停止 EC2 实例	139
停止指定数量的 EC2 实例	141
运行预配置的 AWS FIS SSM 文档	141
运行预定义的自动化运行手册	142
限制对具有目标 IAM 角色的 EC2 实例的 API 操作	143
对 Kubernetes 集群中的容器组 (pod) CPU 进行压力测试	144
指定数量的 Kinesis Data Streams 的预配置吞吐量异常	146
实验角色权限示例	147
管理实验	149
开始实验	149
查看实验	150
实验状态	150
操作状态	151
标记实验	151
停止实验	152
列出已解析的目标	152
教程	153
测试实例停止和启动	153
先决条件	153
步骤 1：创建实验模板	153
步骤 2：开始实验	156
步骤 3：跟踪实验进度	157
步骤 4：验证实验结果	157
步骤 5：清除	157
在实例上运行 CPU 压力测试	158

先决条件	158
步骤 1：为停止条件创建 CloudWatch 警报	159
步骤 2：创建实验模板	160
步骤 3：开始实验	162
步骤 4：跟踪实验进度	162
步骤 5：验证实验结果	162
步骤 6：清除	157
测试竞价型实例中断情况	164
先决条件	165
步骤 1：创建实验模板	166
步骤 2：开始实验	168
步骤 3：跟踪实验进度	168
步骤 4：验证实验结果	169
第 5 步：清理	170
模拟连接事件	170
先决条件	171
步骤 1：创建 AWS FIS 实验模板	172
步骤 2：对 Amazon S3 端点执行 Ping 操作	173
第 3 步：开始你的 AWS FIS 实验	174
第 4 步：追踪您 AWS 的 FIS 实验进度	174
步骤 5：验证 Amazon S3 网络中断	174
步骤 5：清除	175
安排定期实验	176
先决条件	176
步骤 1：创建 IAM 角色和策略	176
步骤 2：创建 Amazon EventBridge 调度器	178
步骤 3：验证实验	179
步骤 4：清除	180
使用场景库	181
查看场景	181
使用场景	182
导出场景	182
场景参考	183
AZ Availability: Power Interruption	185
AZ: Application Slowdown	199
Cross-AZ: Traffic Slowdown	205

Cross-Region: Connectivity	212
使用多账户实验	226
概念	226
最佳实践	227
先决条件	227
Permissions	227
停止条件 (可选)	230
多账户实验的安全杆 (可选)	230
创建多账户实验模板	231
更新目标账户配置	232
删除目标账户配置	232
调度实验	234
创建调度器角色	234
创建实验计划	238
使用控制台更新计划	238
更新实验计划	239
禁用或删除实验计划	239
安全杆	241
安全杆的概念	241
安全杆资源	241
使用安全杆	242
查看安全杆	242
激活安全杆	242
解除安全杆	243
监控实验	244
监视器使用 CloudWatch	245
监测 AWS FIS 实验	245
AWS FIS 使用情况指标	245
监视器使用 EventBridge	246
实验日志记录	248
权限	248
日志架构	248
日志目的地	250
日志记录示例	250
启用实验日志记录	255
禁用实验日志记录	255

使用记录 API 调用 AWS CloudTrail	256
使用 CloudTrail	256
了解 AWS FIS 日志文件条目	257
故障排除	262
错误代码	262
安全性	264
数据保护	264
静态加密	265
传输中加密	265
Identity and access management	265
受众	266
使用身份进行身份验证	266
使用策略管理访问	267
AWS 故障注入服务如何与 IAM 配合使用	269
策略示例	273
使用服务相关角色	283
AWS 托管策略	285
基础结构安全性	289
AWS PrivateLink	289
注意事项	289
创建接口 VPC 端点	290
创建 VPC 端点策略	290
标记您的资源	292
添加标签限制	292
使用标签	292
配额和限制	294
文档历史记录	304
.....	CCCX

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。

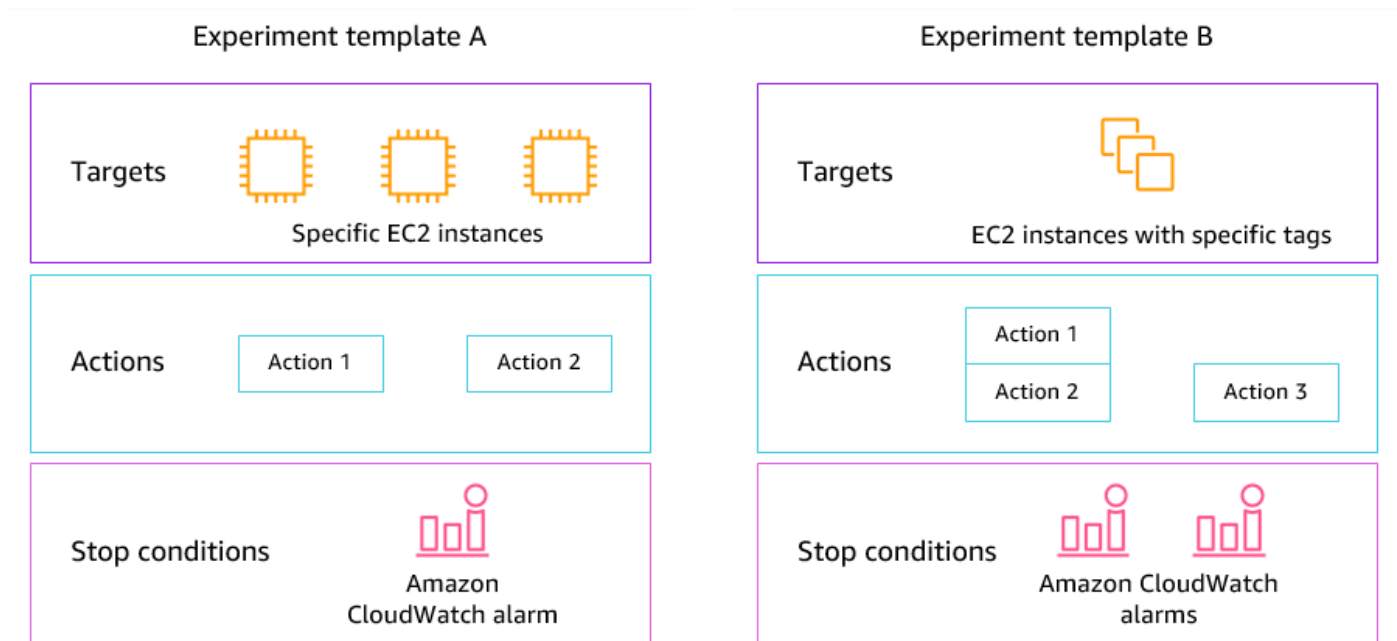
Important

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[规划你的 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。有关支持的服务及其操作的列表，请参阅 [AWS FIS 操作参考](#)。

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 [金融情报 AWS 局的行动](#)。

访问 AWS FIS

您可以通过以下任何 AWS 一种方式与 FIS 合作：

- AWS 管理控制台— 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 管理控制台](#)。
- 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 SDKs— 提供特定语言 APIs 并处理许多连接细节，例如计算签名、处理请求重试次数和处理错误。有关更多信息，请参阅 [AWS SDKs](#)。
- 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](#)。

规划你的 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。

在开始使用 AWS FIS 进行故障注入实验之前，我们建议您熟悉以下原则和指南。

Important

AWS FIS 对系统中的真实 AWS 资源执行实际操作。因此，在开始使用 AWS FIS 进行实验之前，我们强烈建议您先在预生产或测试环境中完成计划阶段和测试。

内容

- [基本原则和指南](#)
- [实验规划指南](#)

基本原则和指南

在开始使用 AWS FIS 进行实验之前，请执行以下步骤：

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 *fault injection action* 已执行，则 *business or technical metric impact* 不应超过 *value*。

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。

我们还建议在使用 AWS FIS 时遵循以下指南：

- 请务必在测试环境中开始试验 AWS FIS。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

使用 AWS FIS，您可以 AWS 利用自己的资源进行实验，以测试应用程序或系统在故障条件下将如何运行的理论。

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。如果您不熟悉 AWS FIS 或者以前从未进行过故障注入实验，我们建议您首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。

我们建议您查看 Well-Archit AWS ected 框架。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，请参阅 [使用 AWS Amazon 监控 FIS 使用量指标 CloudWatch](#)。

- 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 FIS 实验模板组件

可以使用以下组件构造实验模板：

操作

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](#)。

Targets

执行特定操作所依据的 AWS 资源。有关更多信息，请参阅 [Targets](#)。

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](#)。

实验报告配置

启用实验报告的配置。有关更多信息，请参阅 [AWS FIS 的实验报告配置](#)。

实验选项

实验模板的选项。有关更多信息，请参阅 [的实验选项 AWS FIS](#)。

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",
```

```
"experimentReportConfiguration": {},
"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 管理控制台，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](#)。

金融情报 AWS 局的行动

要创建实验模板，必须定义一项或多项操作。有关 AWS FIS 提供的预定义操作列表，请参阅 [操作参考](#)。

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](#)
- [操作标识符](#)
- [操作参数](#)
- [操作目标](#)
- [操作持续时间](#)
- [操作示例](#)

操作语法

以下是操作的语法。

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

```
    "parameters": {
      "name": "value"
    },
    "startAfter": ["action_name", ...],
    "targets": {
      "ResourceType": "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 操作参考](#)。

操作参数

某些 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时 AWS 向 FIS 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": {  
  "ResourceType": "resource_name"  
}
```

AWS FIS 操作支持以下操作目标的资源类型：

- AutoScalingGroups— 亚马逊 EC2 Auto Scaling 群组
- 存储桶：Amazon S3 存储桶
- 集群：Amazon EKS 集群
- 集群 — 亚马逊 ECS、Aurora DSQL 或亚马逊 Aurora 数据库集群
- DBInstances— 亚马逊 RDS 数据库实例
- 函数- AWS Lambda 函数
- 实例：Amazon EC2 实例
- KinesisStreams— Kinesis 数据流

- ManagedResources— 启用 ARC 区域转移的 Amazon EKS 集群、亚马逊 EC2 应用程序和网络负载均衡器以及启用 ARC 区域转移的 Amazon EC2 Auto Scaling 组。
- MultiRegionClusters— Amazon MemoryDB 多区域集群
- 节点组：Amazon EKS 节点组
- 容器组 (pod)：Amazon EKS 上的 Kubernetes 容器组 (pod)
- ReplicationGroups— ElastiCache 复制组
- 角色：IAM 角色
- SpotInstances— 亚马逊 EC2 竞价型实例
- 子网：VPC 子网
- 表 — Amazon DynamoDB 多区域全局表强且最终保持一致
- 任务：Amazon ECS 任务
- TransitGateways— 公交网关
- VirtualInterfaces— Direct Connect 虚拟接口
- 卷：Amazon EBS 卷
- VPCEndpoints— 亚马逊 VPC 终端节点

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

操作示例如下所示。

示例

- [停止 EC2 实例](#)
- [中断竞价型实例](#)
- [中断网络流量](#)
- [终止 EKS Worker](#)
- [启动 ARC 区域自动换档](#)

示例：停止 EC2 实例

以下操作将停止使用名为的目标识别的 EC2 实例 *targetInstances*。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* 为目标标识的 50% 的 EC2 实例。

```

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

```

示例：启动 ARC 区域自动换档

以下操作将启动 ARC 区域自动移动，将托管资源从 *for* 转移出 *az-in-parameters* 去。 *duration-in-parameteres* 在 FI ManagedResources S 实验模板中，资源类型用作目标名称 AWS 的密钥。

```
{
  "description": "aaa",
  "targets": {
    "ManagedResources-Target-1": {
      "resourceType": "aws:arc:zonal-shift-managed-resource",
      "resourceArns": [
        "arn:aws:elasticloadbalancing:us-east-1:0124567890:loadbalancer/app/application/11223312312516",
      ],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "arc": {
      "actionId": "aws:arc:start-zonal-autoshift",
      "parameters": {
        "availabilityZoneIdentifier": "us-east-1a",
        "duration": "PT1M"
      },
      "targets": {
        "ManagedResources": "ManagedResources-Target-1"
      }
    }
  },
  "stopConditions": [
    {
      "source": "none"
    }
  ],
  "roleArn": "arn:aws:iam::718579638765:role/fis",
  "tags": {},
  "experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "fail"
  }
}
```

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[使用多账户实验](#)。

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: arc: zonal-shift-managed-resource` — 在 ARC 区域偏移中注册的 AWS 资源

- aws: directconnect: 虚拟接口 — 虚拟接口 Direct Connect
- aws: dsq: cluster — Amazon Aurora DSQL 集群
- aws: dynamodb: global-table — 亚马逊 DynamoDB 多区域全球表
- aws: ec2: autoscaling-group — Amazon Auto Scaling 群组 EC2
- aws:ec2:ebs-volume : Amazon EBS 卷
- aws: ec2: instance — 亚马逊实例 EC2
- aws: ec2: spot-instance — 亚马逊竞价实例 EC2
- aws:ec2:subnet : Amazon VPC 子网
- aws:ec2:transit-gateway : 中转网关
- aws: ec2: vpc-endpoint — 亚马逊 VPC 终端节点
- 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: replicationgroup — 复制组 ElastiCache
- aws:iam:role : IAM 角色
- aws: kinesis: stream — 亚马逊 Kinesis 数据流
- aws: lambda: function — 一个函数 AWS Lambda
- aws: memorydb: — Amazon MemoryDB 多multi-region-cluster 区域集群
- 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 IDs-特定 IDs 资源的 AWS 资源。所有资源 IDs 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。
- 如果您指定多个标签，则所有指定的标签都必须存在于目标资源上才能被选中 (AND)。

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。每个周期代表一个元素的扩展。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"  
    ]  
  }  
],
```

以下逻辑适用于所有资源筛选条件：

- 如果提供了多个过滤器，包括具有相同路径的过滤器，则必须匹配所有过滤器才能选择资源 — AND
- 如果为单个筛选器提供了多个值，则需要匹配任何一个值才能选择资源 — OR
- 如果在 describe API 调用的路径位置找到多个值，则需要匹配任何一个值才能选择资源 — OR
- 要匹配标签 key/value 对，应改为按标签选择目标资源 (见上文)。

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:arc:zonal-shift-managed-resource	ListManagedResources	list-managed-resources
aws:directconnect:virtual-interface	DescribeVirtualInterfaces	describe-virtual-interfaces
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes
aws:ec2:instance	DescribeInstances	describe-instances
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ec2:vpc-endpoint	DescribeVpcEndpoints	describe-vpc-endpoints
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:replication-group	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	list-roles
aws:kinesis:stream	DescribeStreamSummary	describe-stream-summary
aws:lambda:function	ListFunctions	列表函数

资源类型	API 操作	AWS CLI 命令
aws:memorydb:multi-region-cluster	DescribeMultiRegionClusters	describe-multi-region-clusters
aws:rds:cluster	描述DBClusters	describe-db-clusters
aws:rds:db	描述DBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets
aws:dynamodb:global-table	DescribeTable	描述表
aws:dsq:cluster	GetCluster	获取集群

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:lambda:function

- `functionQualifier` : 可选。要定位的函数的版本或别名。如果未指定限定符，则将考虑将所有调用作为定位。如果指定了具有多个版本的别名，则只要使用包含别名的 ARN 调用别名中包含的所有版本，就会考虑将其定位。如果使用特殊别名，`$LATEST` 则将考虑对基本函数 ARN 的调用以及包含 `$LATEST` 在 ARN 中的调用进行错误注入。有关 Lambda 版本的更多信息，请参阅用户指南中的 [管理 Lambda 函数版本](#)。AWS Lambda

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: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 实例作为目标并将范围添加到 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

此示例可能的目标是指定 VPC 中带有标签的 Amazon EC2 实例env=prod。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 `A EC2 describe-instances mazon` 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。以下是部分输出，其中包括标记为的字段 *italics*。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-0011111111111111",
```

```
"InstanceId": "i-00aaaaaaaaaaaaaaaa",
"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",
...
"NetworkInterfaces": [
{
  ...
  "Groups": [
    {
      "GroupName": "sec-group-1",
      "GroupId": "sg-a0011223344556677"
    },
    {
      "GroupName": "sec-group-1",
      "GroupId": "sg-b9988776655443322"
    }
  ],
  ...
},
...
],
...
{
```

```

        ...
    }
  ],
  "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" ]
  }
],

```

要选择附加了多个安全组中任意一个的实例，请指定一个带有属性路径的过滤 `GroupId` 器和多个安全组 IDs。例如：

```
"filters": [
  {
    "path": "NetworkInterfaces.Groups.GroupId",
    "values": [
      "sg-a0011223344556677",
      "sg-f1100110011001100"
    ]
  }
],
```

要选择连接了所有安全组的实例，请指定多个过滤器，其中包含的属性路径GroupId和每个过滤器的单个安全组 ID。例如：

```
"filters": [
  {
    "path": "NetworkInterfaces.Groups.GroupId",
    "values": [
      "sg-a0011223344556677"
    ]
  },
  {
    "path": "NetworkInterfaces.Groups.GroupId",
    "values": [
      "sg-b9988776655443322"
    ]
  }
],
```

示例：Amazon RDS 集群（数据库集群）

当您为支持 `aws:rds:cluster` 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。以下是部分输出，其中包括标记为的字段 *AllocatedStorage*。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

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) 是一项可帮助管理员安全地控制 AWS 资源访问权限的 AWS 服务。要使用 AWS FIS，您必须创建一个 IAM 角色来授予 AWS FIS 所需的权限，这样 AWS FIS 才能代表您运行实验。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，请参阅 [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。您可以通过在策略中指定特定资源 ARNs 或标签来实现此目的。

为了帮助您快速开始使用 AWS FIS，我们提供了 AWS 托管策略，您可以在创建实验角色时指定这些策略。或者，您也可以创建专属内联策略文档时使用这些策略作为模型。

内容

- [前提条件](#)
- [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](#)

前提条件

在开始之前，请安装 AWS CLI 并创建所需的信任策略。

安装 AWS CLI

在开始之前，请安装并配置 AWS CLI。配置时 AWS CLI，系统会提示您输入 AWS 凭据。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

实验角色必须具有允许 AWS FIS 服务担任该角色的信任关系。创建 `fis-role-trust-policy.json` 文本文件并添加以下信任关系策略。

JSON

```
{
  "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 的其中一个 AWS 托管策略快速入门。

创建实验角色并附加 AWS 托管策略

1. 验证您的实验中是否有 AWS FIS 操作的托管策略。否则，您需要改为创建专有的内联策略。有关更多信息，请参阅 [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。

JSON

```
{
  "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。

JSON

```
{
  "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[AWSFaultInjectionSimulatorEC2访问策略](#)开始。
 - Amazon ECS 操作 — 从[AWSFaultInjectionSimulatorECSAccess策略](#)开始。
 - Amazon EKS 行动 — 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
```

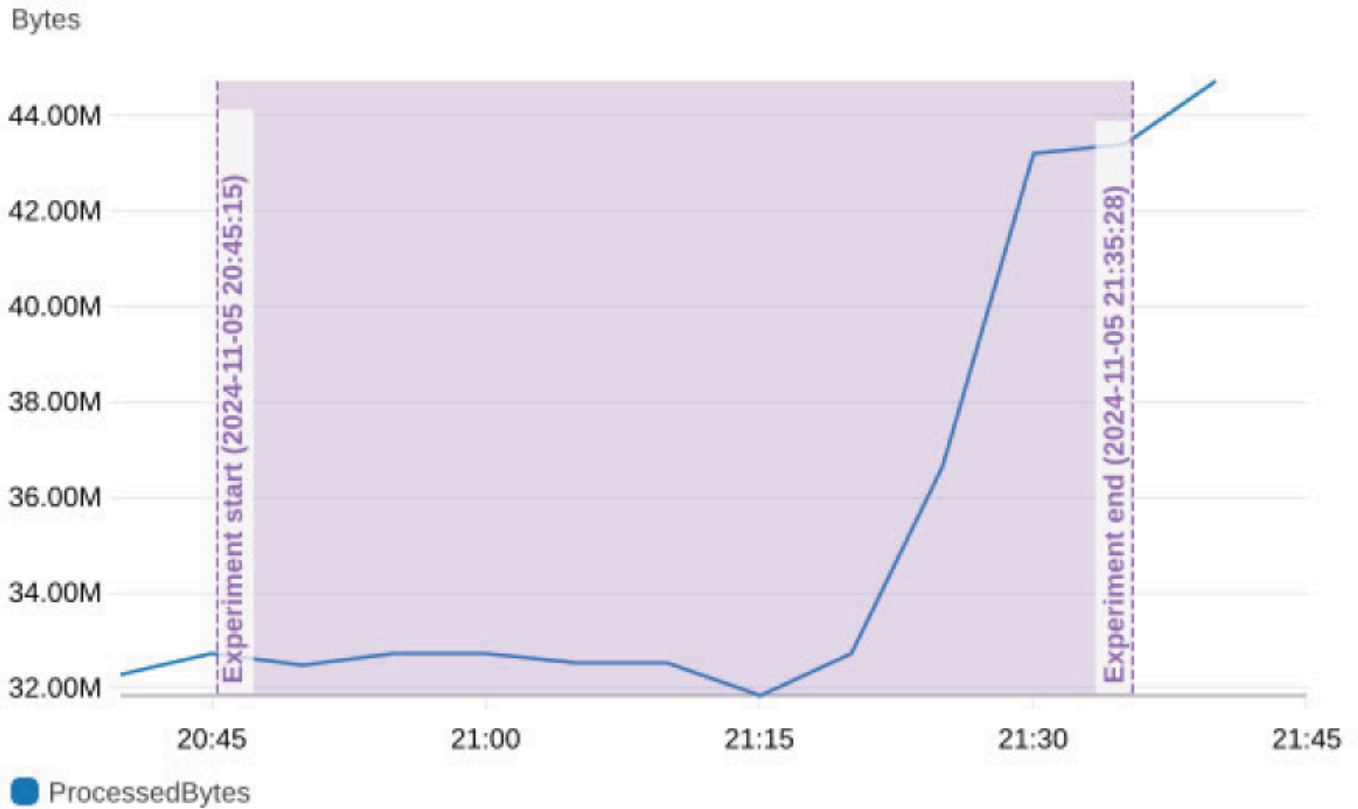
AWS FIS 的实验报告配置

您可以启用 AWS 故障注入服务 (FIS) 来生成实验报告，从而更轻松地生成弹性测试的证据。实验报告是一个 PDF 文档，它汇总了实验操作，并可选择从您指定的 CloudWatch 仪表板捕获应用程序的响应。要查看示例实验报告，请[在此处](#)下载 zip 文件。

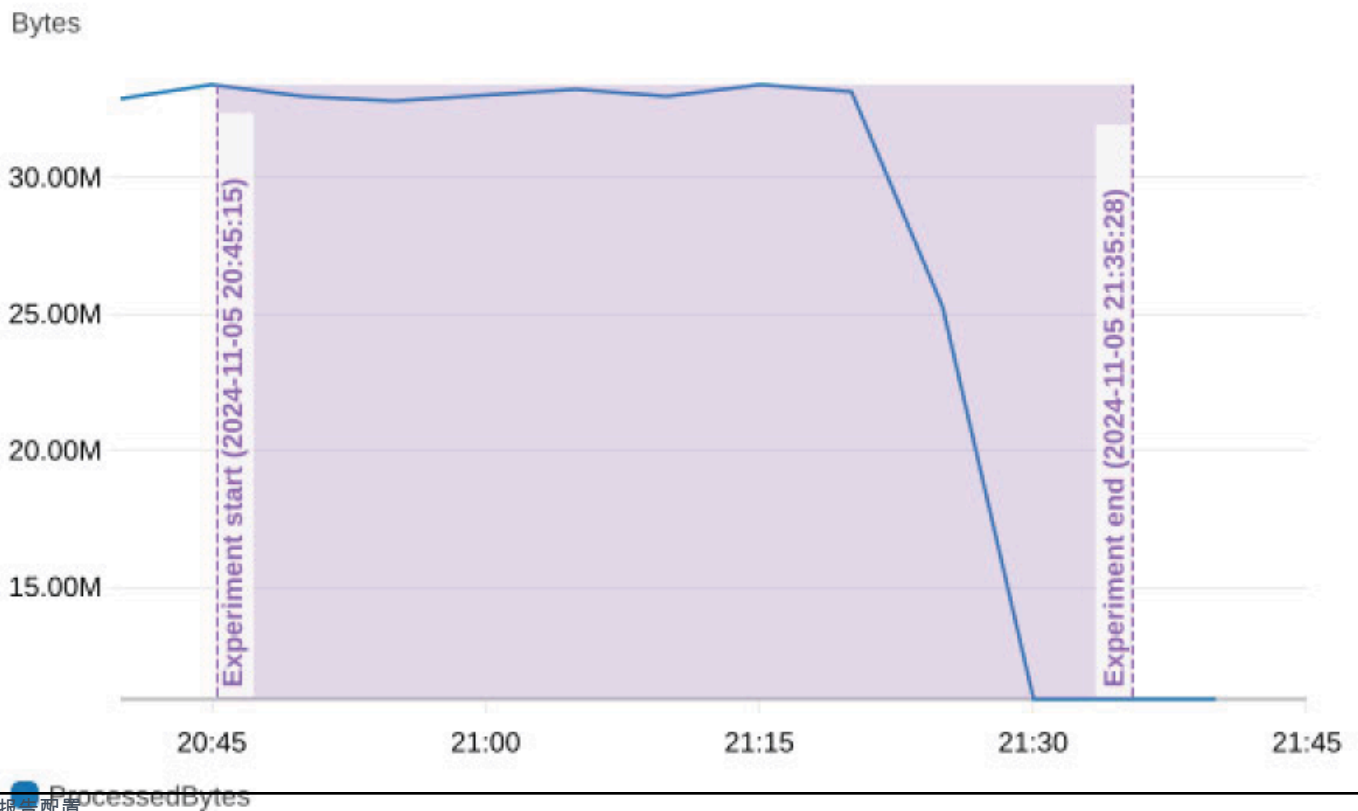
要启用和配置为实验生成的报告的内容，您需要为实验模板定义实验报告配置。当您指定 CloudWatch 仪表板时，AWS FIS 会包含给定仪表板中所有小组件的快照图，并在您指定的持续时间内注释实验的开始和结束时间，如下例所示。

此示例演示了可用区 (AZ) 中丢包实验的影响。当在 AZ use1-az6 中引入数据包丢失时，流量会从 use1-az6 转移到 use1-az4，因此该可用区中负载均衡器处理的字节数会减少。

NLB ProcessedBytes use1-az4



NLB ProcessedBytes use1-az6



实验结束后，可以从 AWS FIS 控制台下载报告，也可以将其存储在 Amazon S3 存储桶中。如果您在报告配置中包含 CloudWatch 仪表板，则还会提供每个小组件的图像。对于作为目标预览一部分 cancelled 或作为目标预览一部分运行的实验（ActionMode 设置为），不会生成报告。skip-all 一旦实验超过实验数据保留期限，则只能从 Amazon S3 存储桶中获取报告。AWS 每份交付的报告均收取 FIS 费用，但因内部错误而失败的报告除外。有关更多信息，请参阅 [AWS 故障注入服务定价](#) 和 [AWS 故障注入服务的配额和限制](#)。可能会收取 Amazon S3 的摄取和存储费用，GetMetricWidgetImage 以及 GetDashboard 请求的 CloudWatch API 费用。有关更多信息，请参阅 [Amazon S3 的定价](#) 和 [CloudWatch 定价](#)。

内容

- [实验报告配置语法](#)
- [实验报告权限](#)
- [实验报告最佳实践](#)

实验报告配置语法

以下是实验报告配置的语法，这是实验模板的可选部分。

```
{
  "experimentReportConfiguration": {
    "outputs": {
      "s3Configuration": {
        "bucketName": "my-bucket-name",
        "prefix": "report-storage-prefix"
      }
    },
    "dataSources": {
      "cloudWatchDashboards": [
        {
          "dashboardIdentifier": "arn:aws:cloudwatch::123456789012:dashboard/MyDashboard"
        }
      ]
    },
    "preExperimentDuration": "PT20M",
    "postExperimentDuration": "PT20M"
  }
}
```

使用 `experimentReportConfiguration`，您可以自定义要包含在实验报告中的数据的输出目的地、输入数据和时间窗口，这可以帮助您更好地了解 AWS FIS 实验的影响和结果。在定义实验报告配置时，您需要提供以下内容：

输出

`experimentReportConfiguration` 的部分指定了实验报告的交付地点。在中 `outputs`，您可以以 `s3Configuration` 通过提供以下内容来指定：

- `bucketName`-存储报告的 Amazon S3 存储桶的名称。存储桶必须与实验位于同一区域。
- `prefix` (可选)-存储报告的 Amazon S3 存储桶中的前缀。强烈建议使用此字段，以便您只能访问前缀。

数据源

的可选部分指定实验报告中将包含的其他数据源。 `experimentReportConfiguration`

- `cloudWatchDashboards`-将包含在报告中的 CloudWatch 仪表板阵列。仅限于一个 CloudWatch 仪表板。
- `dashboardIdentifier`-控制面板的 ARN。CloudWatch 报告中将包含此控制面板 `metric` 中该类型的每个控件的快照图，但跨区域指标除外。

`preExperimentDuration`

的可选部分 `experimentReportConfiguration`，用于定义要包含在报告中的 CloudWatch 仪表板指标的实验前持续时间，最多 30 分钟。这段时间应该代表您的应用程序的稳定状态。例如，实验前持续时间为 5 分钟意味着快照图表将在实验开始前 5 分钟包含指标。持续时间的格式为 ISO 8601，默认格式为 20 分钟。

`postExperimentDuration`

的可选部分 `experimentReportConfiguration`，用于定义要包含在报告中的 CloudWatch 仪表板指标的实验后持续时间，最多 2 小时。该持续时间应代表您的应用程序的稳定状态或恢复期。例如，如果您将实验后持续时间指定为 5 分钟，则快照图表将包含直到实验结束 5 分钟后的指标。持续时间的格式为 ISO 8601，默认格式为 20 分钟。

实验报告权限

要让 AWS FIS 生成和存储实验报告，您需要允许 AWS FIS 实验 IAM 角色执行以下操作：

- `cloudwatch:GetDashboard`

- cloudwatch:GetMetricWidgetImage
- s3:GetObject
- s3:PutObject

我们建议您遵循 AWS 安全最佳实践，将实验角色限制为存储桶和前缀。以下是限制实验角色访问权限的策略声明示例。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::my-experiment-report-bucket/my-prefix/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "cloudwatch:GetDashboard"
      ],
      "Resource": "arn:aws:cloudwatch::012345678912:dashboard/my-experiment-report-dashboard",
      "Effect": "Allow"
    },
    {
      "Action": [
        "cloudwatch:GetMetricWidgetImage"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

向使用客户托管密钥 (CMK) 加密的 Amazon S3 存储桶提交报告的额外权限

如果您在中指定S3Configuration的 Amazon S3 存储桶已使用 CMK 加密，则需要根据您的 KMS 密钥策略向 FIS 实验角色授予以下额外权限：

- kms:GenerateDataKey
- kms:Decrypt

以下是 KMS 密钥策略声明示例，该声明允许 FIS 实验角色向加密存储桶写入报告：

```
{
  "Sid": "Allow FIS experiment report",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::012345678912:role/FISExperimentRole",
    ]
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

实验报告最佳实践

以下是使用 AWS FIS 实验报告配置的最佳实践：

- 在开始实验之前，请生成目标预览，以验证您的实验模板是否按预期进行配置。目标预览将为您提供有关实验预期目标的信息。要了解更多信息，请参阅[使用实验模板生成目标预览](#)。
- 该报告不应用于对失败的实验进行故障排除。取而代之的是使用实验日志来排除实验错误。我们建议您仅在之前运行并成功完成的实验中使用该报告。
- 限制实验 IAM 角色放置并获取对 S3 目标存储桶和前缀的对象访问权限。我们建议您仅将 bucket/前缀指定 AWS 给 FIS 实验报告，不要向其他 AWS 服务授予访问该存储桶和前缀的权限。
- 使用 Amazon S3 对象锁定可在固定时间内或无限期地防止报告被删除或覆盖。要了解更多信息，请参阅[使用对象锁定锁定对象](#)。

- 如果您的 CloudWatch 控制面板位于同一区域的单独账户中，则可以使用 CloudWatch 跨账户可观察性通过 CloudWatch 控制台或和 API 中的 Observability Access Manager 命令将您的 FIS Orchestrator 账户启用为监控账户，将单独的账户设置为源账户。AWS CLI 要了解更多信息，请参阅[CloudWatch 跨账户可观察性](#)。

的实验选项 AWS FIS

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。在开始实验时设置其他实验选项。

以下是可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。

以下是可在开始实验时设置的实验选项的语法。

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

如果在开始实验时未指定任何实验选项，则使用默认的 run-all。

内容

- [账户定位](#)
- [空目标解析模式](#)
- [操作模式](#)

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，这会影响到多个目标账户 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 [使用多账户实验 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 IDs 的目标账户配置账号必须是唯一的。

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[使用多账户实验 AWS FIS](#)。

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定义的目标的操作 ARNs，例如。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`。

操作模式

操作模式是可以在开始实验时指定的可选参数。您可以将操作模式设置为 `skip-all`，以便在向目标资源注入故障之前生成目标预览。通过目标预览，您能够验证以下各项：

- 是否已将实验模板配置为以预期资源为目标。开始此实验时作为目标的实际资源可能会与预览不同，因为资源可能会被随机删除、更新或采样。
- 日志记录配置是否已正确设置。
- 对于多账户实验，您是否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`：（默认值）实验将对目标资源采取操作。
- `skip-all`：实验将跳过对目标资源执行的所有操作。

要了解有关如何在开始实验时设置操作模式参数的更多信息，请参阅[使用实验模板生成目标预览](#)。

AWS FIS 操作参考

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。

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](#)。在得知特定操作的名称后，您可以使用 [get-action 命令](#) 查看有关该操作的详细信息。有关在中使用 AWS FIS 命令的更多信息 AWS CLI，请参阅《AWS CLI 命令参考》中的 [《AWS Command Line Interface 用户指南》](#) 和 [fis](#)。

有关 AWS FIS 操作工作原理的更多信息，请参阅 [金融情报 AWS 局的行动](#) 和 [AWS 故障注入服务如何与 IAM 配合使用](#)。

操作

- [故障注入操作](#)
- [恢复行动](#)
- [等待动作](#)
- [亚马逊的 CloudWatch 行动](#)
- [Amazon DynamoDB 操作](#)
- [亚马逊 Aurora DSQL 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](#)
- [亚马逊 Kinesis Data Streams 操作](#)
- [AWS Lambda 行动](#)
- [亚马逊 MemoryDB 操作](#)
- [网络操作](#)
- [Amazon RDS 操作](#)

- [Amazon S3 操作](#)
- [Systems Manager 操作](#)
- [AWS Direct Connect 行动](#)
- [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 AWS FIS aws: lambda: 函数操作](#)

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。具体响应取决于每项服务和 API。有关更多信息，请查看服务的 SDK 和 API 文档。

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` 和 `kinesis`。
- `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 [亚马逊 EC2 API 参考](#) 和 [Amazon Kinesis Data Streams API 参考](#)。

Permissions

- `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。具体响应取决于每项服务和 API。有关更多信息，请查看服务的 SDK 和 API 文档。

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` 和 `kinesis`。
- `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 [亚马逊 EC2 API 参考](#) 和 [Amazon Kinesis Data Streams API 参考](#)。

Permissions

- `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。具体响应取决于每项服务和 API。有关更多信息，请查看服务的 SDK 和 API 文档。

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` 和 `kinesis`。
- `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 [亚马逊 EC2 API 参考](#) 和 [Amazon Kinesis Data Streams API 参考](#)。

Permissions

- `fis:InjectApiUnavailableError`

恢复行动

执行恢复操作是为了降低风险或保护受损后的应用程序。

AWS FIS 支持以下恢复操作。

`aws:arc:start-zonal-autoshift`

自动将受支持资源的流量从可能受损的可用区 (AZ) 转移出去，并将其重新路由到同一 AWS 区域 AZs 的正常运行状态。这允许通过 FIS 体验区域自动换档。区域自动切换是 Amazon 应用程序恢复控制器 (ARC) 中的一项功能，它允许 AWS 在 AWS 确定存在可能影响可用区客户的损害时，代表您将资源流量从可用区转移出去。

运行 `aws:arc:start-zonal-autoshift` 操作时，使用 `StartZonalShift` 和 `UpdateZonalShift`，并将这些请求 `CancelZonalShift` APIs 的 `expiresIn` 字段设置为 1 分钟作为安全机制。这使得 AWS FIS 在发生任何意外事件（例如网络中断或系统问题）时，可以快速回滚区域移动。在 ARC 控制台中，到期时间字段将显示 `AWS FIS-managed`，实际的预期到期时间由区域移位操作中指定的持续时间决定。

资源类型

- `aws:arc:zonal-shift-managed-resource`

区域转移托管资源是资源类型，包括 Amazon EKS 集群、Amazon EC2 应用程序和网络负载均衡器以及可以启用 ARC 区域自动切换的 Amazon EC2 Auto Scaling 组。有关更多信息，请参阅 ARC 开发人员指南中的[支持的资源](#)和[启用区域自动移位资源](#)。

参数

- `duration`— 流量转移的时间长度。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。
- `availabilityZoneIdentifier`— 流量从该区域移开。这可以是可用区名称 (us-east-1a) 或可用区 ID (use1-az1)。
- `managedResourceTypes`— 将从中转移流量的资源类型，用逗号分隔。可能的选项有 ASG (Auto Scaling Group)、ALB (应用程序负载均衡器)、NLB (网络负载均衡器) 和 EKS (Amazon EKS)。
- `zonalAutoshiftStatus`— 您要定位的资源的 `zonalAutoshiftStatus` 状态。可能的选项有 ENABLED、DISABLED、和 ANY。默认值为 ENABLED。

Permissions

- `arc-zonal-shift:StartZonalShift`
- `arc-zonal-shift:GetManagedResource`
- `arc-zonal-shift:UpdateZonalShift`
- `arc-zonal-shift:CancelZonalShift`
- `arc-zonal-shift:ListManagedResources`
- 自动缩放 : `DescribeTags`
- 标签 : `GetResources`

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。

Permissions

- 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 ARNs 中的一个，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。

Permissions

- `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 多区域全局表复制。操作开始后，表可能会继续复制最多 5 分钟。

多区域强一致 (MRSC) 全局表

以下语句将动态附加到目标 DynamoDB MRSC 全局表的策略中：

```
{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "dynamodb:UpdateTable"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
      "Condition": {
        "DateLessThan": {
          "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
        },
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        }
      }
    },
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxxxxxxForApplicationAutoScaling",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:UpdateTable"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
      "Condition": {
        "DateLessThan": {
          "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
        },
        "ArnEquals": {
```

```

        "aws:PrincipalArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable"
    }
}
]
}

```

如果目标表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，会将错误语句插入现有策略中，不会对现有策略语句进行任何其他修改。然后，在实验结束时，从策略中删除错误声明。

目标 Amazon DynamoDB MRSC 全局表需要额外配额。该配额规定，在7天的滚动窗口内，任何一张桌子的减值时间不得超过5,040分钟。

多区域最终一致 (MREC) 全局表

以下语句将动态附加到目标 DynamoDB MREC 全局表的策略中：

```

{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "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"
        }
      }
    }
  ]
}

```

```

    },
    "ArnEquals": {
      "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
    }
  }
}
]
}

```

以下语句将动态附加到目标 DynamoDB MREC 全局表的流策略中：

```

{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "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"
        },
        "ArnEquals": {
          "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        }
      }
    }
  ]
}

```

如果目标表或流未附加任何资源策略，则会在实验期间创建资源策略，并在实验结束时自动将其删除。否则，会将错误语句插入现有策略中，不会对现有策略语句进行任何其他修改。然后，在实验结束时，从策略中删除错误声明。

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。

Permissions

- `dynamodb:PutResourcePolicy`
- `dynamodb>DeleteResourcePolicy`
- `dynamodb:GetResourcePolicy`
- `dynamodb:DescribeTable`
- `tag:GetResources`
- `dynamodb:InjectError` *

* 仅当您以 MRSC 全局表为目标时，才需要该权限

亚马逊 Aurora DSQL 操作

AWS FIS 支持以下 Amazon Aurora DSQL 操作。

`aws:dsql:cluster-connection-failure`

在 Aurora DSQL 集群中创建指定持续时间的受控连接故障，以测试应用程序的弹性。

资源类型

- `aws:dsql:cluster`

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。
- `percentage` : 注入故障信息的调用百分比 (1 - 100)。

Permissions

- `dsql:InjectError`
- `dsql:GetCluster`
- `tag:GetResources`

要使用 Aurora DSQL 启动实验，请参阅 Aurora DSQL 用户指南中的[故障注入测试](#)。

Amazon EBS 操作

AWS FIS 支持以下 Amazon EBS 操作。

操作

- [aws:ebs:pause-volume-io](#)
- [aws:ebs:volume-io-latency](#)

aws:ebs:pause-volume-io

暂停对目标 EBS 卷的 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（例如 PT5 S），I/O 则会在指定的持续时间内暂停实验，但由于初始化实验需要时间，因此可能需要更长的时间才能完成实验。

Permissions

- `ec2:DescribeVolumes`

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

aws:ebs:volume-io-latency

为目标 EBS 卷的 I/O 操作注入延迟。目标卷必须位于同一个可用区，并且必须连接到[基于 Nitro 的实例](#)。这些卷无法连接到前哨基地中的实例。

要使用亚马逊 EC2 控制台启动实验，请参阅[亚马逊 EBS 用户指南中的亚马逊 EBS 故障测试](#)。

资源类型

- `aws:ec2:ebs-volume`

参数

- `readIOPercentage`— 将注入延迟的读取 I/O 操作的百分比，从 0.1% 到 100.% 不等。这是实验期间将受到影响的卷上所有读取 I/O 操作的百分比。默认值为 100。
- `readIOLatencyMilliseconds`— 读取 I/O 操作注入的延迟量，以毫秒为单位，从 1 毫秒 (io2 卷) 或 10 毫秒 (非 io2 卷) 到 60 秒。这是实验 I/O 期间在读取的指定百分比上观察到的延迟值。默认值为 100。
- `writeIOPercentage`— 将注入延迟的写入 I/O 操作的百分比，从 0.1% 到 100.% 不等。这是实验期间卷上所有写入 I/O 操作将受到影响的百分比。默认值为 100。
- `writeIOLatencyMilliseconds`— 写入 I/O 操作注入的延迟量，以毫秒为单位，从 1 毫秒 (io2 卷) 或 10 毫秒 (非 io2 卷) 到 60 秒。这是实验 I/O 期间将根据读取的特定百分比观察到的延迟值。默认值为 100。
- `duration`— 注入延迟的持续时间，从 1 秒到 12 小时不等。

Permissions

- `ec2:DescribeVolumes`
- `ec2:InjectVolumeIOLatency`
- `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 “SSM 文档操作”](#)。

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。

对于 Auto Scaling `LaunchInstances` 操作，将在响应的 `errors` 字段中返回 `InsufficientInstanceCapacity` 错误，但是 Auto Scaling 组的所需容量仍会更新，从而允许异步扩展过程有可能启动实例。要对容量不足的处理进行更广泛的测试 `LaunchInstances`，请考虑将此操作与一起使用 [the section called “aws:ec2:asg-insufficient-instance-capacity-error”](#)。

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 IDs（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)。

Permissions

- `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 IDs (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。

Permissions

- `ec2:InjectApiError` 条件键 `ec2:FisActionId` 值设置为 , `aws:ec2:asg-insufficient-instance-capacity-error` `ec2: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

Permissions

- ec2:RebootInstances
- ec2:DescribeInstances

AWS 托管策略

- [AWSFaultInjectionSimulatorEC2访问](#)

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durationBeforeInterruption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durationBeforeInterruption，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。

Permissions

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

AWS 托管策略

- [AWSFaultInjectionSimulatorEC2访问](#)

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。
- `completelyInstancesTerminated` : 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。

Permissions

- `ec2:StopInstances`
- `ec2:StartInstances`

- `ec2:DescribeInstances` : 可选。需要使用“completeInstances已终止”，才能在操作结束时验证实例状态。
- `kms:CreateGrant` : 可选。需要使用startInstancesAfter持续时间来重启带有加密卷的实例。

AWS 托管策略

- [AWSFaultInjectionSimulatorEC2访问](#)

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

Permissions

- `ec2:TerminateInstances`
- `ec2:DescribeInstances`

AWS 托管策略

- [AWSFaultInjectionSimulatorEC2访问](#)

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。

Permissions

- `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

Permissions

- 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-stress](#) SSM 文档。任务必须由管理 AWS Systems Manager。有关更多信息，请参阅 [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` 。

Permissions

- `ecs:DescribeTasks`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-io-stress

对目标任务运行 str I/O ess。使用 [AWS FIS-Run-io-stress SSM 文档](#)。任务必须由管理 AWS Systems Manager。有关更多信息，请参阅 [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，文件系统中使用的空闲空间百分比。默认为 80%。
- `workers` : 可选。工作程序数量。Workers 混合执行顺序、随机和内存映射 read/write 操作、强制同步和缓存丢弃。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` 。

Permissions

- `ecs:DescribeTasks`
- `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-Process 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 [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`。

Permissions

- `ecs:DescribeTasks`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-blackhole-port

使用 [Amazon ECS 故障注入终端节点](#) 丢弃指定协议和端口的入站或出站流量。使用 `-n` [AWSFIS-Runnetwork-blackHole-port-EC S 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 [ECS 任务操作](#)。

设置 `useEcsFaultInjectionEndpoints` 为 `false` 时，故障将使用该 `iptables` 工具，并使用 [AWSFIS-Run-Network-Blackhole-Port SSM 文档](#)。

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`。依赖关系是 `atdcurl-minimal`、`dig` 和 `jq`。
- `useEcsFaultInjectionEndpoints` : 可选。如果设置为 `true`，则 APIs 将使用 Amazon ECS 故障注入。默认值为 `false`。

Permissions

- `ecs:DescribeTasks`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-latency

使用 [Amazon ECS 故障注入](#) 终端节点，为网络接口增加延迟和抖动，以便向特定来源传出流量。使用 `-n AWSFIS-Runnetwork-Latency-EC` 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 [ECS 任务操作](#)。

设置 `useEcsFaultInjectionEndpoints` 为 `true` 时，故障将使用该 `tc` 工具 `false`，并使用 [AWSFIS-Run-Network-Latency-Sources SSM](#) 文档。

使用 `flowsPercent` 参数来增加一定百分比的连接的延迟。要使用该 `flowsPercent` 参数，ECS 代理版本应为 `1.100.0` 或更高。

要在 `sources` 参数 IDs 中使用可用区名称或可用区，操作的所有目标都必须位于同一 VPC 上。

资源类型

- `aws:ecs:task`

参数

- `duration` : 测试的持续时间，格式为 ISO 8601。
- `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。
- `flowsPercent` : 可选。将受该操作影响的网络流量的百分比。默认为 100%。
- `sources` : 可选。源代码用逗号分隔，不含空格。可能的值是：IPv4 地址、IPv4 CIDR 块、域名、可用区名称 (us-east-1a)、可用区 ID (use1-az1)、ALL 和 DYNAMODB 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。默认值为 ALL，它匹配所有 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。依赖关系是 `atdcurl-minimal`、`dig`、`jq` 和 `Isuf`。
- `useEcsFaultInjectionEndpoints` : 可选。如果设置为 true，则 APIs 将使用 Amazon ECS 故障注入。默认值为 false。

Permissions

- `ecs:DescribeTasks`
- `ecs:DescribeContainerInstances`
- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-packet-loss

使用 [Amazon ECS 故障注入终端节点](#)，将数据包丢失添加到网络接口，用于向特定来源的出口流量。使用 [-network-AWSFIS-RunPacket-loss-ecs](#) 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 [ECS 任务操作](#)。

设置 `useEcsFaultInjectionEndpoints` 为 `true` 时，故障将使用该 `tc` 工具 `false`，并使用 `Network-AWSFIS-RunPacket-Loss-Sources` SSM 文档。

使用 `flowsPercent` 参数按一定百分比的连接注入丢包信息。要使用该 `flowsPercent` 参数，ECS 代理版本应为 `1.100.0` 或更高。

要在 `sources` 参数 IDs 中使用可用区名称或可用区，操作的所有目标都必须位于同一 VPC 上。

资源类型

- `aws:ecs:task`

参数

- `duration`：测试的持续时间，格式为 ISO 8601。
- `lossPercent`：可选。丢包率。默认为 7%。
- `flowsPercent`：可选。将受该操作影响的网络流量的百分比。默认为 100%。
- `sources`：可选。源代码用逗号分隔，不含空格。可能的值是：IPv4 地址、IPv4 CIDR 块、域名、可用区名称 (`us-east-1a`)、可用区 ID (`use1-az1`)、ALL 和 `DYNAMODB 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。默认值为 ALL，它匹配所有 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`。依赖关系是 `atdcurl-minimal`、`dig`、`jq` 和 `Isuf`。
- `useEcsFaultInjectionEndpoints`：可选。如果设置为 `true`，则 APIs 将使用 Amazon ECS 故障注入。默认值为 `false`。

Permissions

- `ecs:DescribeTasks`
- `ecs:DescribeContainerInstances`
- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`
- `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` | `I0Chaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodI0Chaos` | `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。

Permissions

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[EKS 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组 (pod) 的 Kubernetes 标签。
- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。
- `fisPodSecurityPolicy` : 可选。用于 FIS 创建的故障编排容器组 (pod) 和临时容器的 [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。

Permissions

- `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 [EKS 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组 (pod) 的 Kubernetes 标签。
- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。

- `fisPodSecurityPolicy` : 可选。用于 FIS 创建的故障编排容器组 (pod) 和临时容器的 [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。

Permissions

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

AWS 托管策略

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-io-stress`

在目标 pod 上运行 I/O stress。有关更多信息，请参阅 [EKS 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。Workers 混合执行顺序、随机和内存映射 read/write 操作、强制同步和缓存丢弃。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组 (pod) 的 Kubernetes 标签。
- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。

- `fisPodSecurityPolicy` : 可选。用于 FIS 创建的故障编排容器组 (pod) 和临时容器的 [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。

Permissions

- `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 [EKS 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组 (pod) 的 Kubernetes 标签。
- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。

- `fisPodSecurityPolicy` : 可选。用于 FIS 创建的故障编排容器组 (pod) 和临时容器的 [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。

Permissions

- `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](#) `privileged` 策略兼容。有关更多信息，请参阅 [EKS 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` : 协议。可能的值为 `tcp` 和 `udp`。
- `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组 (pod) 的 Kubernetes 标签。

- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。

Permissions

- `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 privileged](#) 策略兼容。有关更多信息，请参阅 [EKS Pod 操作](#)。

使用 `flowsPercent` 参数来增加一定百分比的连接的延迟。

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，用逗号分隔。支持“全部”和“默认值”。默认为 `DEFAULT`，它将以操作系统的主网络接口为目标。
- `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。
- `flowsPercent` : 可选。将受该操作影响的网络流量的百分比。默认为 100%。
- `sources` : 可选。源代码用逗号分隔，不含空格。可能的值是：IPv4 地址、IPv4 CIDR 块、域名、可用区名称 (`us-east-1a`)、可用区 ID (`use1-az1`)、`ALL` 和 `DYNAMODB 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。对于域名，会尝试进行 10 次 DNS 解析以收集 IP 地址。由于 DNS 负载平衡和轮换，此操作可能不会损害域可以解析到的所有可能的 IP 地址。默认值为 `ALL`，它匹配所有 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组 (pod) 的 Kubernetes 标签。
- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。

Permissions

- `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](#) privileged 策略兼容。有关更多信息，请参阅 [EKS Pod 操作](#)。

使用 `flowsPercent` 参数按一定百分比的连接注入丢包信息。

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，用逗号分隔。支持“全部”和“默认值”。默认为 `DEFAULT`，它将以操作系统的主网络接口为目标。
- `lossPercent` : 可选。丢包率。默认为 7%。
- `flowsPercent` : 可选。将受该操作影响的网络流量的百分比。默认为 100%。

- `sources` : 可选。源代码用逗号分隔，不含空格。可能的值是：IPv4 地址、IPv4 CIDR 块、域名、可用区名称 (us-east-1a)、可用区 ID (use1-az1)、ALL 和。DYNAMODB 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。对于域名，会尝试进行 10 次 DNS 解析以收集 IP 地址。由于 DNS 负载均衡和轮换，此操作可能不会损害域可以解析到的所有可能的 IP 地址。默认值为 ALL，它匹配所有 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组 (pod) 的 Kubernetes 标签。
- `fisPodAnnotations` : 可选。附加到 FIS 创建的故障编排容器组 (pod) 的 Kubernetes 注释。

Permissions

- `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。仅与 Amazon EKS 托管节点组兼容。不支持自建节点组。有关更多信息，请参阅 [EKS 管理计算](#)。

资源类型

- `aws:eks:nodegroup`

参数

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

Permissions

- `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:replicationgroup-interrupt-az-power`

在启用了多可用区的目标 ElastiCache 复制组的指定可用区中断节点的电源。每个复制组一次只能有一个可用区受到影响。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。此操作的目标同时支持 Redis 和 Valkey 引擎。该操作不支持“无服务器”部署选项。

资源类型

- `aws:elasticache: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。

Permissions

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

Note

中 ElastiCache 断可用区电源操作现在支持所有复制组类型，包括 Valkey 和 Redis。为了更好地呈现此功能，该操作已被重命名。如果您当前正在使用 `aws:elasticache:interrupt-cluster-az-power`，我们建议您迁移 `aws:elasticache:replicationgroup-interrupt-az-power` 到新操作以利用最新功能。

亚马逊 Kinesis Data Streams 操作

亚马逊 Kinesis Data Streams 支持以下 Kinesis 操作。

操作

- [aws:kinesis:stream-provisioned-throughput-exception](#)
- [aws:kinesis:stream-expired-iterator-exception](#)

aws:kinesis:stream-provisioned-throughput-exception

针对目标 Kinesis Data Streams 的请求注入 `ProvisionedThroughputExceededException` 错误响应。支持的操作包括：`GetRecordsGetShardIterator`、`PutRecord`、和 `PutRecords`。

资源类型

- `aws:kinesis:stream`

参数

- 持续时间-持续时间，范围从 1 分钟到 12 小时。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。
- 百分比-要将故障注入的呼叫的百分比 (1-100)。

Permissions

- `kinesis:InjectApiError`

aws:kinesis:stream-expired-iterator-exception

针对指定 Kinesis Data Streams 的 GetRecords 呼叫注入 ExpiredIteratorException 错误响应。

资源类型

- aws:kinesis:stream

参数

- 持续时间-持续时间，范围从 1 分钟到 12 小时。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。
- 百分比-要将故障注入的呼叫的百分比 (1-100)。

Permissions

- kinesis:InjectApiError

AWS Lambda 行动

AWS Lambda 支持以下 Lambda 操作

操作

- [aws:lambda:invocation-add-delay](#)
- [aws:lambda:invocation-error](#)
- [aws:lambda:invocation-http-integration-response](#)

aws:lambda:invocation-add-delay

在您指定的时间内延迟启动函数的毫秒数。此操作的效果与 Lambda 冷启动类似，但额外的时间是计费时长的一部分，适用于所有执行环境，而不仅仅影响新的执行环境。这意味着您可能会遇到 Lambda 冷启动和延迟。通过将延迟值设置为高于 Lambda 函数上配置的超时值，此操作还将提供对高保真超时事件的访问权限。

资源类型

- aws:lambda:function

参数

- 持续时间-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。
- 调用百分比-可选。要将错误注入的函数调用的百分比 (1-100)。默认值为 100。
- startupDelayMilliseconds：可选。在函数代码调用和执行之间等待的时间，以毫秒 (0-900,000) 为单位。默认值为 1000。

Permissions

- s3:PutObject
- s3>DeleteObject
- lambda:GetFunction
- tag:GetResources

aws:lambda:invocation-error

将 Lambda 函数调用标记为失败。此操作对于测试错误处理机制（例如警报和重试配置）非常有用。使用此操作时，您可以选择是否在返回错误之前运行函数代码。

资源类型

- aws:lambda:function

参数

- 持续时间-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。
- 调用百分比-可选。要将错误注入的函数调用的百分比 (1-100)。默认值为 100。
- p@@@ revertExecution — 如果该值为真，则操作将在不执行函数的情况下返回错误。

Permissions

- s3:PutObject
- s3>DeleteObject

- `lambda:GetFunction`
- `tag:GetResources`

aws:lambda:invocation-http-integration-response

修改函数的行为。您可以选择内容类型和 HTTP 响应代码来支持与 ALB、API-GW 和 VPC Lattice 的集成。要启用有选择地影响上游或下游集成，您可以选择是直接返回修改后的响应，还是运行函数并在函数完成执行后替换响应。

资源类型

- `aws:lambda:function`

参数

- `contentTypeHeader`— 要从 Lambda 函数返回的 HTTP 内容类型标头的字符串值。
- `持续时间`-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。
- `调用百分比`-可选。要将错误注入的函数调用的百分比 (1-100)。默认值为 100。
- `p@@@ revertExecution` — 如果该值为真，则操作将在不执行函数的情况下返回响应。
- `Status Code` — 从 Lambda 函数返回的 HTTP 状态码 (000-999) 的值。

Permissions

- `s3:PutObject`
- `s3>DeleteObject`
- `lambda:GetFunction`
- `tag:GetResources`

亚马逊 MemoryDB 操作

AWS FIS 支持以下 MemoryDB 操作。

aws:memorydb:multi-region-cluster-pause-replication

暂停一个区域集群与多区域集群内所有其他区域集群之间的复制。目标区域集群是金融信息系统实验正在进行区域的集群。在复制暂停期间，无法更新多区域集群。操作完成后，多区域集群可能需要几分钟才能恢复到可用状态。要了解有关 Amazon MemoryDB 多区域的更多信息，请参阅 [Amazon MemoryDB 多区域开发者指南](#)。有关区域可用性，请参阅 [MemoryDB 多区域先决条件](#) 和限制。

资源类型

- aws:memorydb:multi-region-cluster

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。

Permissions

- memorydb:DescribeMultiRegionClusters
- memorydb:PauseMultiRegionClusterReplication
- 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-vpc-endpoint](#)

aws:network:disrupt-connectivity

通过临时克隆与目标子网关联的原始网络访问控制列表 (网络 ACL)，拒绝向目标子网发送指定流量。FIS 向克隆的网络 ACL 添加了拒绝规则，其标签为 `managedByfis=True`，并在操作期间将其与子网关联。操作完成后，FIS 会删除克隆的网络 ACL 并恢复原始的网络 ACL 关联。

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`，则网络中的最大条目数 ACLs 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。可以在 VPC 中作为目标的子网的最大数量为 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。
 - `s3express`— 拒绝进出目标子网可用区中 Amazon S3 Express One 区域的区域终端节点的流量。目标子网必须位于 S3 Express One 区域当前可用 AZs 的地方。有关更多信息，请参阅 [S3 Express One 区域可用区和区域](#)。
 - `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。

Permissions

- `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。创建路由表，使其包含要隔离的区域的所有路由。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 (如果 `region` 参数为 `us-east-1`，则为 350) 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。

Permissions

- `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。您无需管理此标签。AWS FIS 在实验期间添加和移除此标签。

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。

Permissions

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

AWS 托管策略

- [AWSFaultInjectionSimulatorNetworkAccess](#)

aws:network:disrupt-vpc-endpoint

阻止目标接口 VPC 终端节点的入站和出站流量。FIS 使用空规则创建托管安全组，并使用该托管安全组临时替换目标 VPC 终端节点的安全组。如果在操作执行期间对目标资源进行了修改，则操作将失败，资源也不会恢复到实验前的状态。此外，如果在操作执行期间修改了 FIS 管理的安全组，则 FIS 不会将其删除。

资源类型

- `aws:ec2:vpc-endpoint`

参数

- `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。

Permissions

- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyVpcEndpoint`

- ec2:CreateSecurityGroup
- ec2>DeleteSecurityGroup
- ec2:RevokeSecurityGroupEgress
- ec2:CreateTags
- vpce:AllowMultiRegion *

* 仅当您的目标是跨区域 VPC 终端节点时，才需要该权限

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DBCluster上运行 Amazon RDS API 操作[故障转移](#)。支持 RDS 集群和文档数据库集群。

资源类型

- aws:rds:cluster

参数

- 无

Permissions

- 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DBInstance上运行 Amazon RDS API 操作[重启](#)。支持 RDS 集群和文档数据库集群。

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。

Permissions

- 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。

Permissions

- 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。

Permissions

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

AWS 托管策略

- [AWSFaultInjectionSimulatorEC2访问](#)

aws:ssm:start-automation-execution

运行 Systems Manager API 操作 [StartAutomationExecution](#)。

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。

Permissions

- `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](#)

AWS Direct Connect 行动

AWS FIS 支持以下 AWS Direct Connect 操作。

操作

- [aws:directconnect:virtual-interface-disconnect](#)

aws:directconnect:virtual-interface-disconnect

通过暂时中断本地网络与与目标虚拟 AWS Direct Connect 接口 () 关联的对等方之间的边界网关协议 (BGP) 会话来测试连接的弹性。VIFs在启动实验之前，FIS 会验证实验中的所有 VIFs 目标是否都处于“可用”状态，并且每个 VIF 的所有 BGP 对等体都处于“可用”状态和“向上”BGP 状态。在实验期间，目标虚拟接口的 BGP 对等会话将处于关闭状态。有关 Direct Connect 故障转移测试的详细信息，请参阅[AWS Direct Connect 文档](#)。

资源类型

- aws:directconnect:虚拟接口

参数

- duration— 持续时间，从 10 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，PT10M 代表十分钟。在 AWS FIS 控制台中，您可以输入秒数、分钟数或小时数。

Permissions

- directconnect:DescribeVirtualInterfaces
- directconnect:StartBgpFailoverTest
- directconnect:ListVirtualInterfaceTestHistory
- directconnect:StopBgpFailoverTest
- tag:GetResources

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 代理是可以在 Amazon EC2 实例、本地服务器或虚拟机 (VMs) 上安装和配置的亚马逊软件。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 代理默认安装在某些 Amazon 系统映像上 (AMIs)。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。[您可以在 AWS Systems Manager 控制台的“文档：https://console.aws.amazon.com/systems-manager/文档”下查看预配置的 SSM 文档](#)。您也可以从 AWS FIS 控制台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。

了解有关操作状态的更多信息

SSM 操作状态由 [SSM 命令](#)状态决定。

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：
 - 亚马逊 Linux 2023、亚马逊 Linux 2
 - Ubuntu
 - RHEL 8、9
 - CentOS 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](#)

行动持续时间和 AWS FIS SSM DurationSeconds M 文档中的区别

某些 SSM 文档会限制自己的执行时间，例如，一些预先配置的 AWS FIS SSM 文档会使用该 DurationSeconds 参数。因此，您需要在 FIS 操作定义中指定两个独立 AWS 的持续时间：

- Action duration：对于具有单个操作的实验，操作持续时间等于实验持续时间。对于多个动作，实验持续时间取决于各个动作的持续时间及其运行顺序。AWS FIS 会监视每个动作，直到其动作持续时间过去。
- 文档参数 DurationSeconds：SSM 文档将执行的持续时间，以秒为单位指定。

您可以为两种持续时间选择不同的值：

- Action duration exceeds DurationSeconds：SSM 文档的执行在操作完成之前完成。AWS FIS 会等到动作持续时间过后才开始后续行动。
- Action duration is shorter than DurationSeconds：操作完成后，SSM 文档继续执行。如果 SSM 文档仍在执行中，并且操作持续时间已过，则操作状态将设置为“已完成”。AWS FIS 只监视执行情况，直到操作持续时间过去。

请注意，某些 SSM 文档设置有可变持续时间。例如，AWS FIS SSM 文档可以选择安装先决条件，这可以将总体执行持续时间延长到指定参数 DurationSeconds 之外。因此，如果将操作持续时间和值设置 DurationSeconds 为相同的值，则 SSM 脚本的运行时间可能会超过操作持续时间。

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-stress](#) SSM 文档。

操作类型（仅限控制台）

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

进行筛选

```
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-D [AWSFIS-Runisk-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
```

进行筛选

```
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`。依赖关系是 `atd`、`kmod` 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 SSM 文档](#)。

操作类型 (仅限控制台)

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

进行筛选

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。混合执行顺序、随机和内存映射 read/write 操作、强制同步和缓存删除的工作程序数量。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-Process SSM 文档](#)。

操作类型 (仅限控制台)

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

进行筛选

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-M [AWSFIS-Runmemory-Stress SSM 文档](#)。

操作类型 (仅限控制台)

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

进行筛选

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

进行筛选

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、Isof 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-Run-网络延迟](#) SSM 文档。

操作类型 (仅限控制台)

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

进行筛选

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-Sources](#) SSM 文档。

使用 `FlowsPercent` 参数来增加一定百分比的连接的延迟。

操作类型 (仅限控制台)

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

进行筛选

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

文档参数

- `Interface` : 可选。网络接口，用逗号分隔。支持“全部”和“默认值”。默认为 `DEFAULT`，它将以操作系统的主网络接口为目标。
- `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。
- `FlowsPercent` : 可选。将受该操作影响的网络流量的百分比。默认为 100%。
- `Sources` – 必需。源代码用逗号分隔，不含空格。可能的值是：IPv4 地址、IPv4 CIDR 块、域名、可用区名称 (`us-east-1a`)、可用区 ID (`use1-az1`)、`ALL` 和 `DYNAMODB 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`。依赖关系是 `atddig`、`jq`、`lsof`、和 `tc`。

使用本文档时，实验角色需要以下权限：

- `ec2:DescribeInstances`

- `ec2:DescribeSubnets`

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-Loss](#) SSM 文档。

操作类型（仅限控制台）

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

进行筛选

`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`、`lsuf`、`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 [Network-AWSFIS-RunPacket-Loss-Sources](#) SSM 文档。

使用 `FlowsPercent` 参数按一定百分比的连接注入丢包信息。

操作类型 (仅限控制台)

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

进行筛选

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

文档参数

- **Interface** : 可选。网络接口，用逗号分隔。支持“全部”和“默认值”。默认为DEFAULT，它将以操作系统的主网络接口为目标。
- **LossPercent** : 可选。丢包率。默认为 7%。
- **FlowsPercent** : 可选。将受该操作影响的网络流量的百分比。默认为 100%。
- **Sources** – 必需。源代码用逗号分隔，不含空格。可能的值是：IPv4 地址、IPv4 CIDR 块、域名、可用区名称 (us-east-1a)、可用区 ID (use1-az1)、ALL 和。DYNAMODB 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。依赖关系是atddig、jq、Isof、和tc。

使用本文档时，实验角色需要以下权限：

- ec2:DescribeInstances
- ec2:DescribeSubnets

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](#)。

限制

- 以下文档不能并行运行：
 - 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

回滚脚本

AWS FIS SSM 文档会自动创建回滚脚本，作为一种安全机制，用于在故障注入实验后恢复系统状态。这些脚本可确保删除注入的故障，即使操作失败或意外终止。

创建回滚脚本

当故障注入实验开始时，会自动创建回滚脚本。

创作详情

- 位置-脚本在 `/var/lib/amazon/ssm/` 目录中创建。
- 命名模式 — `FAULT_NAME-FAULT_IDENTIFIER-Rollback.sh` 其中 `FAULT_IDENTIFIER` 是随机生成的 32 个字符的字符串
- 计时 — 在每个故障注入实验开始时，在故障注入开始之前创建。
- 内容-包含所有必需的环境变量和命令以纠正特定故障。

例如，网络延迟实验可能会在上创建回滚脚本。`/var/lib/amazon/ssm/NetworkLatency-abc123-Rollback.sh`

回滚日志

回滚脚本实现双重日志记录以捕获所有回滚活动，以进行故障排除和审计。

日志文件位置

执行回滚脚本时，它会在两个位置创建日志：

- 临时文件 — /tmp/aws-fis-rollback-*TIMESTAMP*-*PID*.log
- 系统日志-发送到带有设备的 syslog local0.info

日志文件命名

临时日志文件使用以下命名约定：

```
/tmp/aws-fis-rollback-YYYY-MM-DDTHH:MM:SSZ-PID.log
```

其中 *YYYY-MM-DDTHH:MM:SSZ* 是 UTC 时间戳，*PID* 是回滚脚本的进程 ID。

系统日志配置

使用以下配置将回滚日志发送到 syslog：

- 标签 — aws-fis-rollback
- 优先级：local0.info
- 格式-[*YYYY-MM-DDTHH:MM:SSZ*] *log_message*

查看回滚日志

使用以下命令查看 systemd 日志中的所有回滚日志：

```
sudo journalctl -t aws-fis-rollback
```

问题排查

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。

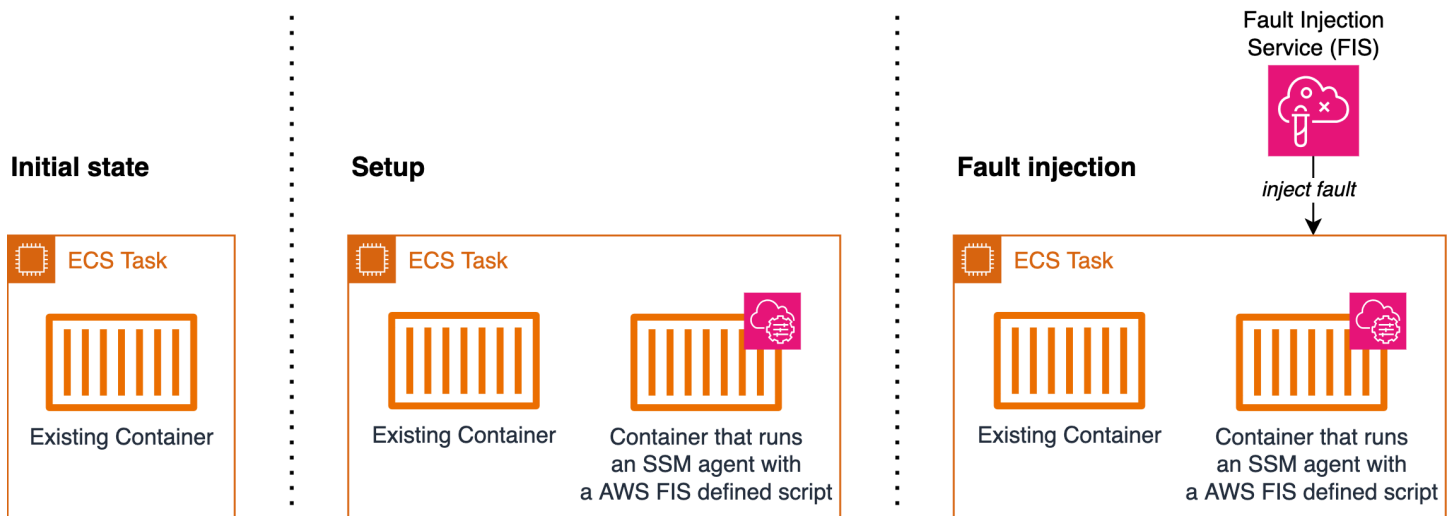
使用 AWS FIS aws: ecs: task 操作

您可以执行 `aws:ecs:task` 操作，为 Amazon ECS 任务注入故障。支持 Amazon EC2 和 Fargate 容量类型。

这些操作使用 [AWS Systems Manager \(SSM\) 文档](#) 来注入故障。要使用 `aws:ecs:task` 操作，您需要将带有 SSM 代理的容器添加到您的亚马逊弹性容器服务 (Amazon ECS) 任务定义中。该容器运行一个 [AWS FIS 定义的脚本](#)，以在 SSM 服务中将 Amazon ECS 任务注册为托管实例。此外，该脚本还会检索任务元数据以向托管实例中添加标签。该设置将允许 AWS FIS 解决目标任务。本段落指的是下图中的设置。

当您运行 FIS 实验定位 AWS 时 `aws:ecs:task`，AWS FIS 会使用资源标签将您在 AWS FIS 实验模板中指定的目标 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。本段落指的是下图中的故障注入。

下图通过包含一个现有容器的任务举例说明了设置和故障注入。



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

- 以下操作无法并行运行：
 - aws:ecs:task-network-blackhole-port
 - aws:ecs:task-network-latency
 - aws:ecs:task-network-packet-loss
- 如果您启用了 Amazon ECS Exec，则必须先将其禁用，然后才能使用这些操作。
- 即使实验的状态为“已完成”，SSM 文档的执行也可能会处于“已取消”状态。在执行 Amazon ECS 操作时，客户提供的持续时间用于实验中的操作持续时间和 Amazon EC2 Systems Manager (SSM) 文档持续时间。操作启动后，SSM 文档需要一段时间才能开始运行。因此，当达到指定的操作持续时间时，SSM 文档可能还需要几秒钟时间才能完成执行。当达到实验操作持续时间时，操作将停止，并且 SSM 文档执行将被取消。故障注入成功。

要求

- 向 AWS FIS [实验角色](#) 添加以下权限：
 - ecs:DescribeTasks
 - 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 [亚马逊 EC2 Task Execution Role Policy](#) 托管策略。
- 在任务定义中，将环境变量 MANAGED_INSTANCE_ROLE_NAME 设置为 [托管实例角色](#) 的名称。该角色将附加到在 SSM 中注册为托管实例的任务。
- 为托管实例角色添加以下权限：

- `ssm:DeleteActivation`
- `ssm:DeregisterManagedInstance`
- 将 [Amazon SSManaged InstanceCore](#) 托管策略添加到托管实例角色。
- 将 SSM 代理容器添加到 Amazon ECS 任务定义中。命令脚本将 Amazon 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; dnf upgrade -y; dnf install jq procs 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_SIDEAR,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”](#)。

- 通过在 Amazon ECS 任务定义中设置以下 `enableFaultInjection` 字段 APIs，启用 Amazon ECS 故障注入：

```
"enableFaultInjection": true,
```

- 在 Fargate 任务上使用 `aws:ecs:task-network-blackhole-port`、`aws:ecs:task-network-latency`、或 `aws:ecs:task-network-packet-loss` 操作时，操作必须将 `useEcsFaultInjectionEndpoints` 参数设置为 `true`

- 使用aws:ecs:task-kill-process、aws:ecs:task-network-blackhole-port、或aws:ecs:task-network-packet-loss操作时aws:ecs:task-network-latency，Amazon ECS 任务定义必须pidMode设置为task。
- 对具有 EC2 启动类型的任务使用aws:ecs:task-network-blackhole-portaws:ecs:task-network-latency、或aws:ecs:task-network-packet-loss操作时，[任务定义中的联网选项](#)必须设置为awsvpc或host。

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_SIDE CAR: 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

dnf upgrade -y
dnf install jq procs 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
fi
fi
```

```
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_SIDE CAR,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",
```

```

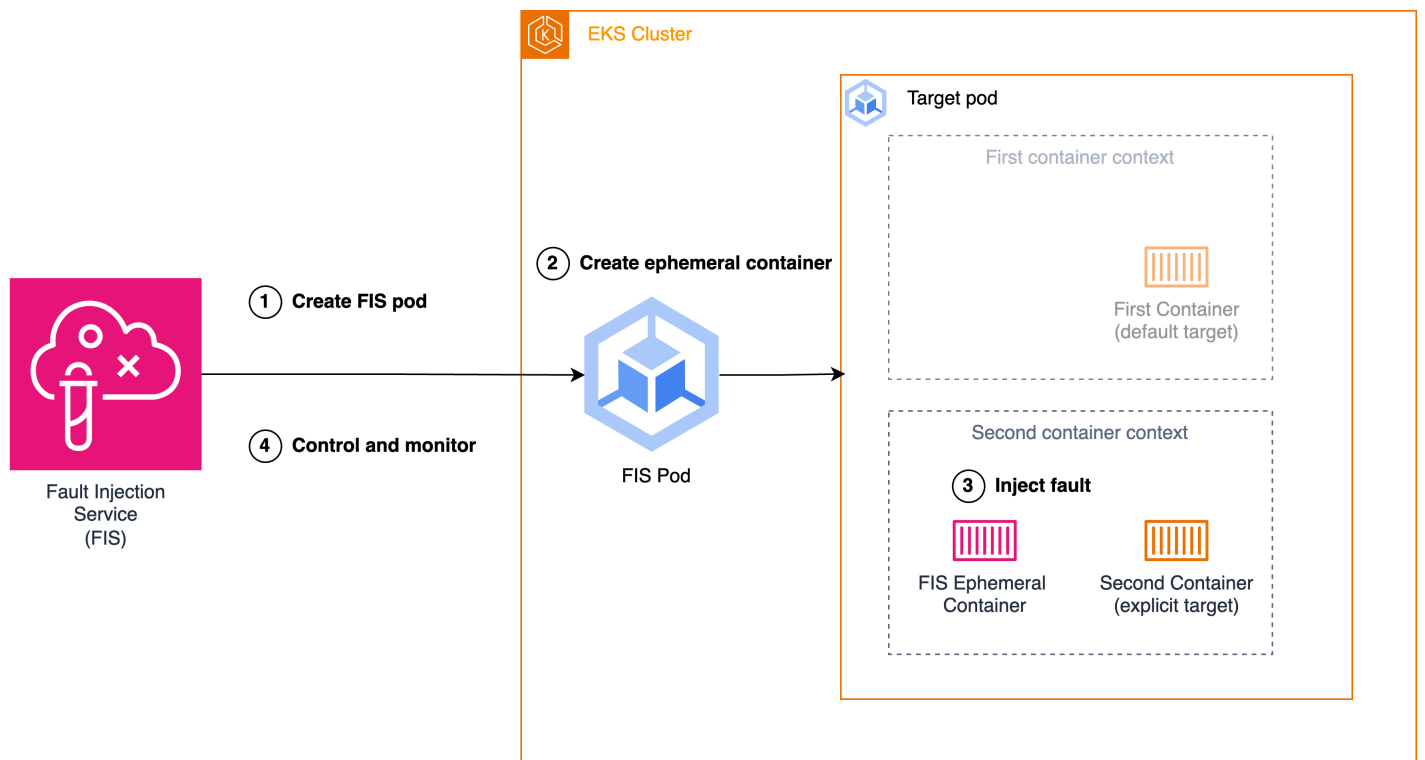
"targets": {
  "myTasks": {
    "resourceType": "aws:ecs:task",
    "resourceArns": [
      "arn:aws:ecs:us-east-1:111122223333:task/my-
cluster/09821742c0e24250b187dfed8EXAMPLE"
    ],
    "selectionMode": "ALL"
  }
},
"actions": {
  "EcsTask-cpu-stress": {
    "actionId": "aws:ecs:task-cpu-stress",
    "parameters": {
      "duration": "PT1M"
    },
    "targets": {
      "Tasks": "myTasks"
    }
  }
},
"stopConditions": [
  {
    "source": "none",
  }
],
"roleArn": "arn:aws:iam::111122223333:role/fis-experiment-role",
"tags": {}
}

```

使用 FIS `aws:eks:pod` 操作

你可以使用 `aws:eks:pod` 操作将错误注入在 EKS 集群中运行的 Kubernetes Pod 中。

操作启动后，FIS 将检索 [FIS 容器组容器映像](#)。然后使用此镜像在目标 EKS 集群中创建 Pod。新创建的 Pod 负责注入、控制和监控故障。对于所有 FIS EKS 操作，除 [aws:eks:pod-delete](#) 之外，错误注入都是通过使用临时容器来实现的，临时容器是一项 [Kubernetes 功能，允许在现有 Pod 中创建临时容器](#)。临时容器在与目标容器相同的命名空间中启动，并执行所需的错误注入任务。如果未指定目标容器，则选择 Pod 规范中的第一个容器作为目标。



1. FIS 在实验模板中指定的目标集群中创建 FIS Pod。
2. FIS Pod 在目标 Pod 中创建一个临时容器，其命名空间与目标容器相同。
3. 临时容器在目标容器的命名空间中注入错误。
4. FIS 容器组 (pod) 控制和监视临时容器的故障注入，而 FIS 控制和监视 FIS 容器组 (pod) 。

实验完成后或发生错误时，将删除临时容器和 FIS 容器组 (pod) 。

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
- 以下操作需要临时容器内的 root 权限。
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency
 - aws:eks:pod-network-packet-loss

临时容器将从目标 Pod 的安全上下文中继承其权限。如果您需要以非 root 用户身份运行 Pod 中的容器，则可以为目标 Pod 中的容器设置单独的安全上下文。

- 您无法使用资源或资源 ARNs 标签在实验模板中识别 aws: eks: pod 类型的目标。必须使用必要资源参数来标识目标。
- 动作 aws: eks: pod-network-latency 和 aws: eks: pod-network-packet-loss 不应并行运行并针对同一 Pod。根据您的指定的 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。
 - 你可以从目标 Pod 中注入的临时容器的日志中识别出这些故障。它将失败，并显示 Exit Code: 16。
- 动作 aws: eks: pod-network-blackhole-port 不应与其他针对同一 Pod 并使用相同的 Pod 的操作并行运行。trafficType 支持使用不同流量类型的并行操作。
- 只有当目标 Pod 的设置 `readOnlyRootFilesystem: false` 时，FIS 才能监控故障注入 securityContext 的状态。如果没有此配置，所有 EKS 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。欲了解更多信息，请参阅 <https://kubernetes.io/docs/tasks/tools/>。
- 当前支持的 EKS 最低版本为 1.23。

创建实验角色

要运行实验，您需要为实验配置 IAM 角色。有关更多信息，请参阅 [the section called “实验角色”](#)。此角色所需的权限取决于正在使用的操作。请参阅[以 aws:eks:pod 为目标的 AWS FIS 操作](#)，查找执行您的操作所需的权限。

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。在以下示例中，服务帐号为 *myserviceaccount*，命名空间为 *default*。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. 运行如下命令。

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

授予 IAM 用户和角色访问 Kubernetes 的权限 APIs

按照 EKS 文档中[将 IAM 身份与 Kubernetes 权限相关联](#)中介绍的步骤进行操作。

选项 1：创建访问条目

我们建议使用 Access Entries。您可以使用以下命令来创建将 IAM 角色与 Kubernetes 用户关联的访问条目。[fis-experiment](#) 有关更多信息，请参阅使用 EK [S 访问条目向 IAM 用户授予访问 Kubernetes 的访问权限](#)。

```
aws eks create-access-entry \  
    --principal-arn arn:aws:iam::123456789012:role/fis-experiment-role \  
    --username fis-experiment \  
    --cluster-name my-cluster
```

Important

为了利用访问条目，必须将 EKS 集群的身份验证模式配置为 API_AND_CONFIG_MAP 或 API 模式。

选项 2：向 aws-auth 添加条目 ConfigMap

还可以使用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
```

Important

利用 eksctl 工具包配置身份映射将导致在中创建条目。aws-auth ConfigMap 需要注意的是，生成的这些条目不支持包含路径组件。因此，作为输入提供的 ARN 不得包含路径分段（例如 arn:aws:iam::*123456789012*:role/service-role/fis-experiment-role）。

容器组 (pod) 容器映像

AWS FIS 提供的 Pod 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。

Pod 容器镜像也可在 [AWS ECR 公共图库](#) 中找到。

AWS 区域	映像 URI
美国东部 (俄亥俄州)	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 (北加利福尼亚)	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
亚太地区 (孟买)	524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1
亚太地区 (大阪)	148336246925.dkr.ecr.ap-northeast-3.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

AWS 区域	映像 URI
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/aw s-fis-pod:0.1
欧洲地区 (伦敦)	327424803546.dkr.ecr.eu-west-2.amazonaws.com/aw s-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/aw s-fis-pod:0.1
欧洲 (西班牙)	395402409451.dkr.ecr.eu-south-2.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
欧洲 (苏黎世)	604225987275.dkr.ecr.eu-central-2.amazonaws.com/ aws-fis-pod:0.1
中东 (巴林)	065825543785.dkr.ecr.me-south-1.amazonaws.com/ aws-fis-pod:0.1
中东 (阿联酋) :	438374459301.dkr.ecr.me-central-1.amazonaws.com/ aws-fis-pod:0.1
南美洲 (圣保罗)	767113787785.dkr.ecr.sa-east-1.amazonaws.com/aw s-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 区域	映像 URI
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"
  }
}

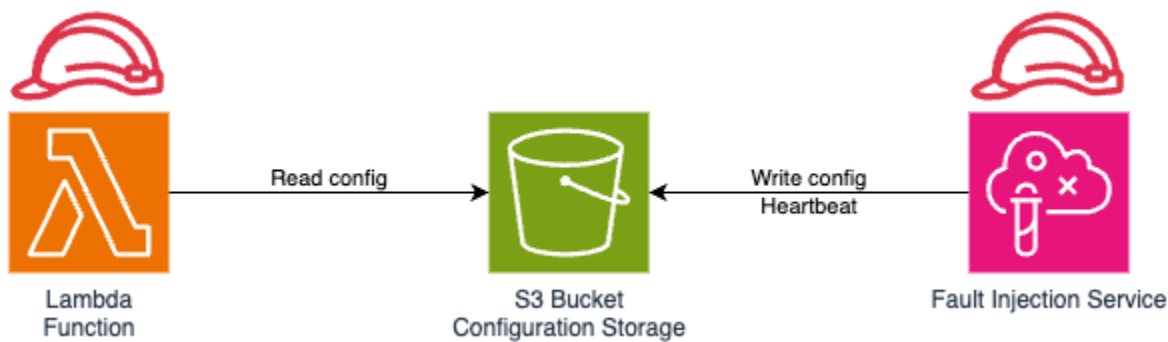
```

使用 AWS FIS aws: lambda: 函数操作

您可以使用 `aws: lambda: function` 操作在函数的调用中注入错误。AWS Lambda

这些操作使用 AWS FIS 托管扩展来注入错误。要使用 `aws: lambda: function` 操作，您需要将扩展作为一个层附加到您的 Lambda 函数，并配置 Amazon S3 存储桶以在和扩展之间进行通信。AWS FIS

当您运行以 `aws: lambda: function` 为目标的 AWS FIS 实验时，从您的 Lambda 函数中读取 AWS FIS Amazon S3 配置，并将错误注入信息写入指定的 Amazon S3 位置，如下图所示。



操作

- [the section called “aws:lambda:invocation-add-delay”](#)
- [the section called “aws:lambda:invocation-error”](#)
- [the section called “aws:lambda:invocation-http-integration-response”](#)

限制

- L AWS FIS lambda 扩展不能与使用响应流的函数一起使用。即使没有出现任何故障，L AWS FIS lambda 扩展也会抑制流配置。有关更多信息，请参阅AWS Lambda 用户指南中的 [Lambda 函数的响应流](#)。

先决条件

在使用 AWS FIS Lambda 操作之前，请确保您已完成以下一次性任务：

- 在您计划开始实验的区域创建 Amazon S3 存储桶 — 您可以使用单个 Amazon S3 存储桶进行多个实验，并在多个 AWS 账户之间共享该存储桶。但是，每个存储桶都必须有单独的存储桶 AWS 区域。
- 创建 IAM 策略以授予 Lambda 扩展对 Amazon S3 存储桶的读取权限 — 在以下模板中，my-config-distribution-bucket 替换为您在上面创建的 Amazon S3 存储桶的 FisConfigs 名称以及您要使用的 Amazon S3 存储桶中文件夹的名称。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListingConfigLocation",
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["arn:aws:s3::my-config-distribution-bucket"],
      "Condition": {
        "StringLike": {
          "s3:prefix": ["FisConfigs/*"]
        }
      }
    },
    {
      "Sid": "AllowReadingObjectFromConfigLocation",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": ["arn:aws:s3::my-config-distribution-bucket/FisConfigs/*"]
    }
  ]
}
```

```
}

```

- 创建 IAM 策略以授予 AWS FIS 实验对 Amazon S3 存储桶的写入权限 — 在以下模板中，my-config-distribution-bucket 使用您在上面创建的 Amazon S3 存储桶的 FisConfigs 名称以及您要使用的 Amazon S3 存储桶中的文件夹名称替换。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFisToWriteAndDeleteFaultConfigurations",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::my-config-distribution-bucket/FisConfigs/*"
    },
    {
      "Sid": "AllowFisToInspectLambdaFunctions",
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowFisToDoTagLookups",
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": "*"
    }
  ]
}
```

配置 Lambda 函数

对于要影响的每个 Lambda 函数，请按照以下步骤操作：

1. 将上面创建的 Amazon S3 读取访问策略附加到 Lambda 函数。
2. 将 AWS FIS 扩展作为图层附加到函数上。有关该图层的更多信息 ARNs，请参阅[适用于 Lambda 的 AWS FIS 扩展程序的可用版本](#)。
3. 例如，将 `AWS_FIS_CONFIGURATION_LOCATION` 变量设置为 Amazon S3 配置文件夹的 ARN。 `arn:aws:s3:::my-config-distribution-bucket/FisConfigs/`
4. 将 `AWS_LAMBDA_EXEC_WRAPPER` 变量设置为 `/opt/aws-fis/bootstrap`。

配置实 AWS FIS 验

在运行实验之前，请确保已将您在先决条件中创建的 Amazon S3 写入权限策略附加到将使用 AWS FIS Lambda 操作的实验角色中。有关如何设置 AWS FIS 实验的更多信息，请参阅[管理 AWS FIS 实验模板](#)。

日志记录

AWS FIS Lambda 扩展程序将日志写入控制台并 CloudWatch 写入日志。可以使用 `AWS_FIS_LOG_LEVEL` 变量配置日志记录。支持的值有 `INFO`、`WARN` 和 `ERROR`。日志将以为您的 Lambda 函数配置的日志格式写入。

以下是文本格式的日志示例：

```
2024-08-09T18:51:38.599984Z INFO AWS FIS EXTENSION - extension enabled 1.0.1
```

以下是 JSON 格式的日志示例：

```
{
  "timestamp": "2024-10-08T17:15:36.953905Z",
  "level": "INFO",
  "fields": {
    "message": "AWS FIS EXTENSION - adding 5000 milliseconds of latency to function invocation",
    "requestId": "0608bf70-908f-4a17-bbfe-3782cd783d8b"
  }
}
```

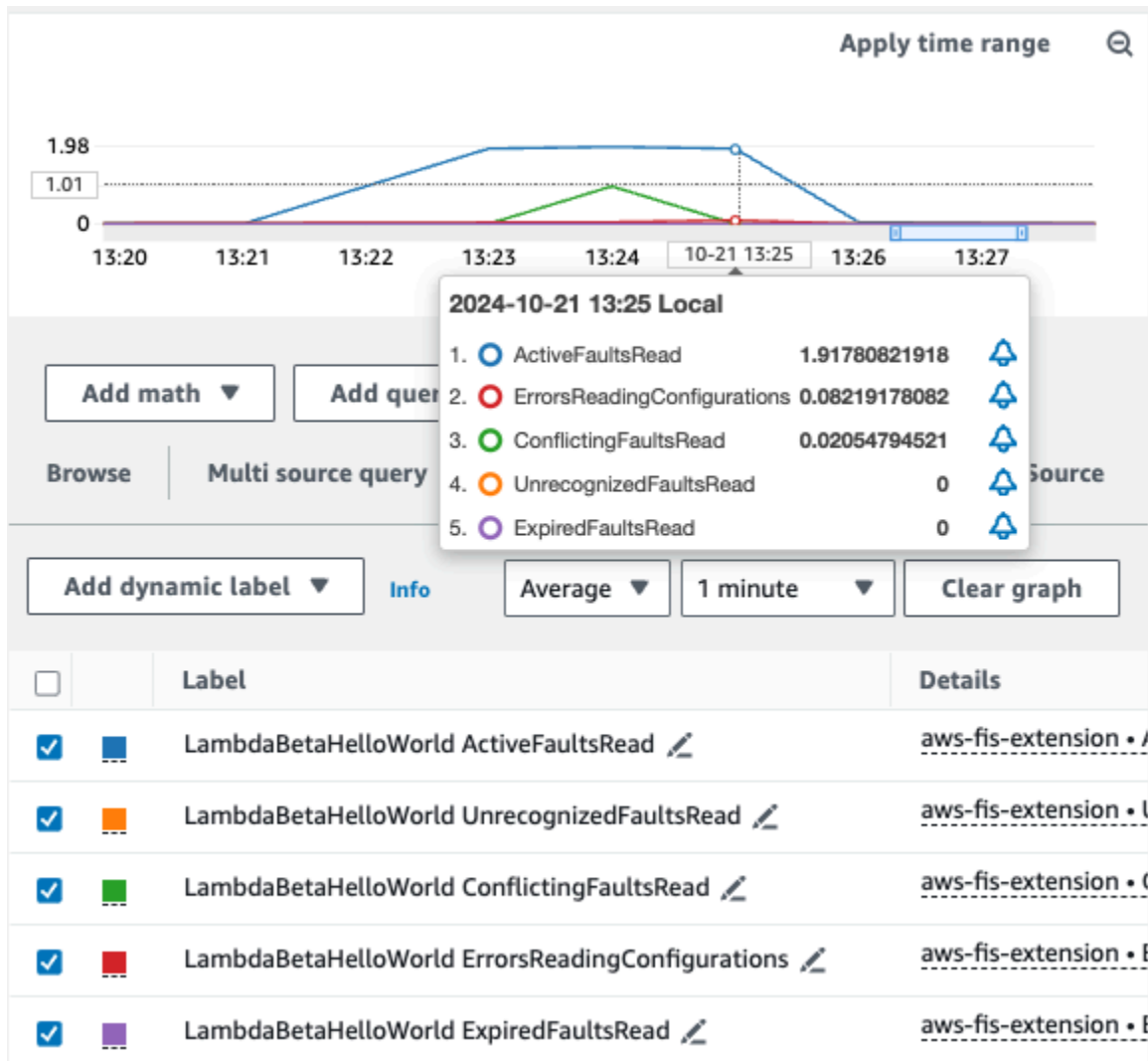
```
}
}
```

发出的日志可以与 Amazon CloudWatch 指标筛选器一起使用，以生成自定义指标。有关指标筛选条件的更多信息，请参阅 Amazon Logs 用户指南中的使用筛选条件从 CloudWatch 日志[事件创建指标](#)。

使用 CloudWatch 嵌入式指标格式 (EMF)

您可以通过将变量设置为，将 AWS FIS Lambda 扩展配置为发出 EMF 日志。AWS_FIS_EXTENSION_METRICS all默认情况下，扩展程序不发出 EMF 日志，AWS_FIS_EXTENSION_METRICS默认为。noneEMF 日志发布在 CloudWatch 控制台aws-fis-extension namespace上。

在aws-fis-extension命名空间中，您可以选择要在图表中显示的某些指标。以下示例显示了aws-fis-extension命名空间中的一些可用指标。



高级主题

本节提供了有关如何 AWS FIS 使用 Lambda 扩展和特殊用例的更多信息。

主题

- [了解投票](#)
- [了解并行性](#)
- [了解调用百分比](#)
- [的特殊注意事项 SnapStart](#)
- [快速不频繁函数的特殊注意事项](#)
- [使用 Lambda 运行时 API 代理配置多个扩展](#)
- [AWS FIS 与容器运行时一起使用](#)
- [AWS FIS Lambda 环境变量](#)

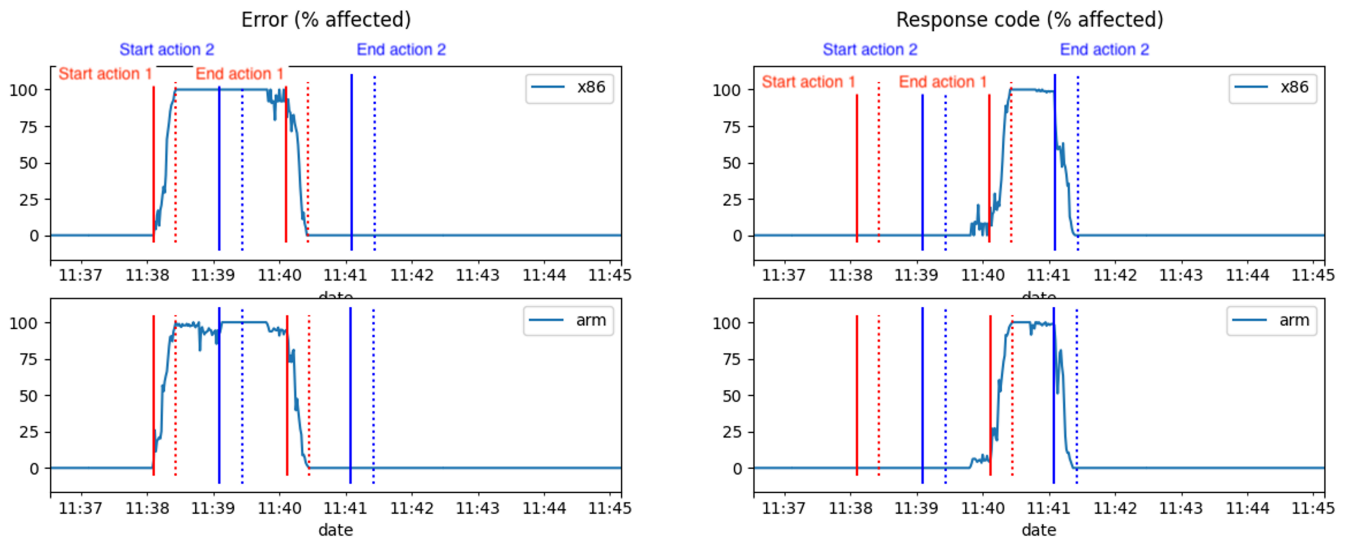
了解投票

你可能会注意到，在故障开始影响所有调用之前，会有长达 60 秒的上升期。这是因为 Lambda 扩展在等待实验开始时很少轮询配置信息。您可以通过设置 `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` 环境变量（默认为 60 秒）来调整轮询间隔。值越低，轮询频率越高，但会对性能产生更大的影响和成本。您可能还会注意到，在注入故障后，会有长达 20 秒的缓冲期。这是因为在实验运行期间，扩展程序的轮询频率更高。

了解并行性

您可以同时使用多个操作来定位相同的 Lambda 函数。如果操作彼此不同，则将应用所有操作。例如，您可以在返回错误之前添加初始延迟。如果将两个相同或相互冲突的操作应用于同一个函数，则只会应用开始日期最早的操作。

下图显示了两个相互冲突的操作，`aws: lambda: invocation-error` 和 `aws: lambda: invocation-http-integration-response` 最初，`aws: lambda: invocation-error` 在 11:38 上升并运行 2 分钟。然后，`aws: lambda: invocation-http-integration-response` 尝试从 11:39 开始，但直到第一个操作结束后 11:40 才生效。为了保持实验时间，`aws: lambda: invocation-http-integration-response` 仍会在原定时间 11:41 结束。



了解调用百分比

AWS Fault Injection Service Lambda 操作使用 `aws:lambda:function` 目标，该目标允许您选择一个或多个函数。AWS Lambda ARNs 使用这些 ARNs，AWS Fault Injection Service Lambda 操作可以在每次调用所选 Lambda 函数时注入错误。为了只允许将错误注入到一小部分调用中，每个操作都允许您指定一个值介于 0 到 100 之间的 `invocationPercentage` 参数。使用该 `invocationPercentage` 参数，即使调用百分比低于 100%，也可以确保操作是并发的。

的特殊注意事项 SnapStart

AWS Lambda 即使实验已经在运行，SnapStart 启用的函数也更有可能会等待整个持续时间后 `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` 才开始第一个故障配置。这是因为 Lambda SnapStart 使用单个快照作为多个执行环境的初始状态并保留临时存储。对于 AWS Fault Injection Service Lambda 扩展，它将保持轮询频率，并在执行环境初始化时跳过初始配置检查。有关 Lambda 的更多信息 SnapStart，请参阅用户指南 SnapStart 中的使用 [Lambda 提高启动性能](#)。AWS Lambda

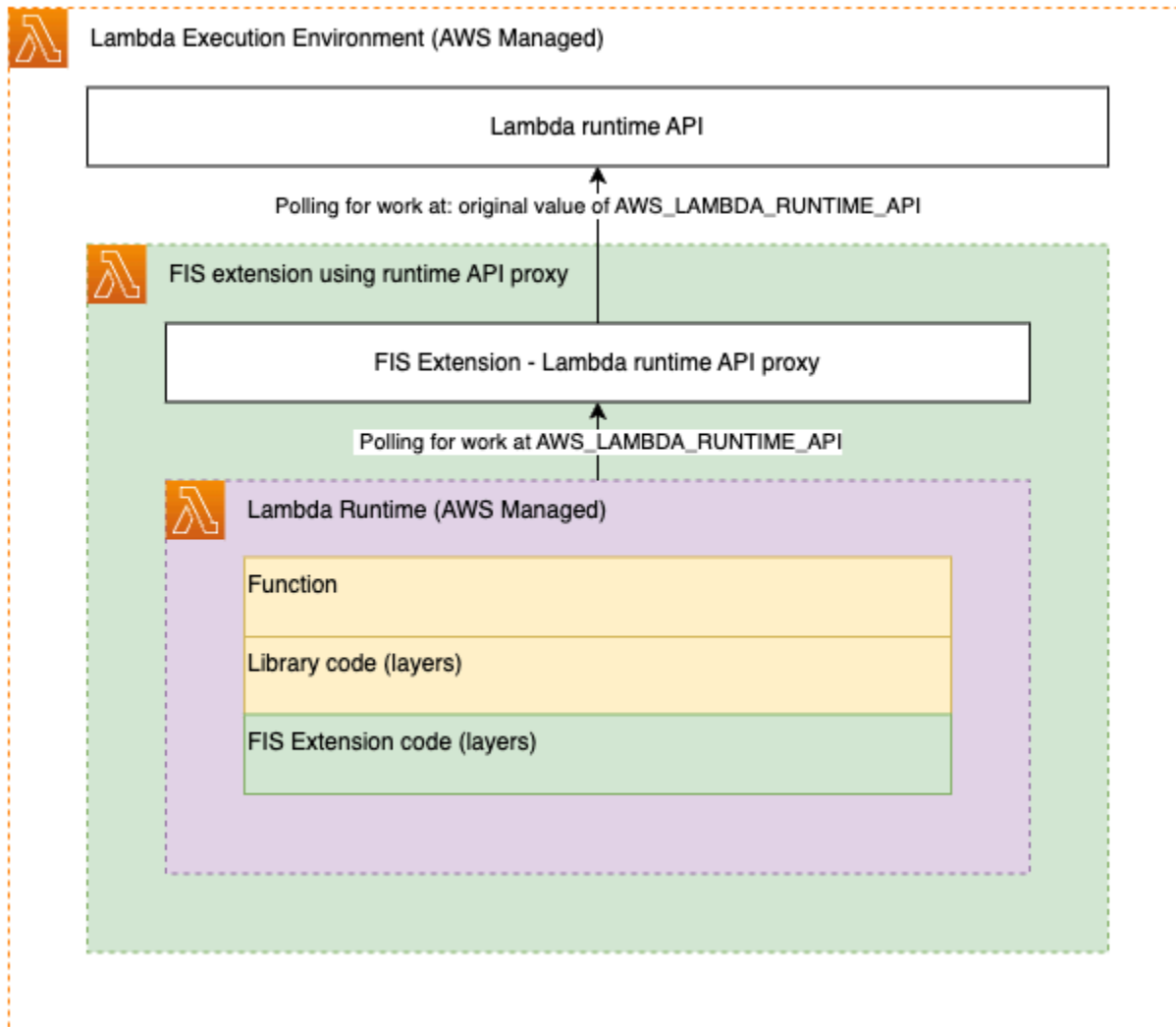
快速不频繁函数的特殊注意事项

如果您的 Lambda 函数运行时间少于 70 毫秒的平均轮询持续时间，则轮询线程可能需要多次调用才能获得错误配置。如果该函数不经常运行，例如每 15 分钟运行一次，则投票将永远无法完成。为确保轮询线程可以完成，请设置 `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` 参数。在启动该功能之前，扩展程序将等到您设置的机上民意调查完成的时长。请注意，这将增加计费函数的持续时间，并导致某些调用的额外延迟。

使用 Lambda 运行时 API 代理配置多个扩展

Lambda 扩展使用 AWS Lambda 运行时 API 代理在函数调用到达运行时之前对其进行拦截。它通过向 AWS Lambda 运行时公开 Runtime API 的代理并在 `AWS_LAMBDA_RUNTIME_API` 变量中公布其位置来实现此目的。

下图显示了使用 Lambda 运行时 API 代理的单个扩展的配置：

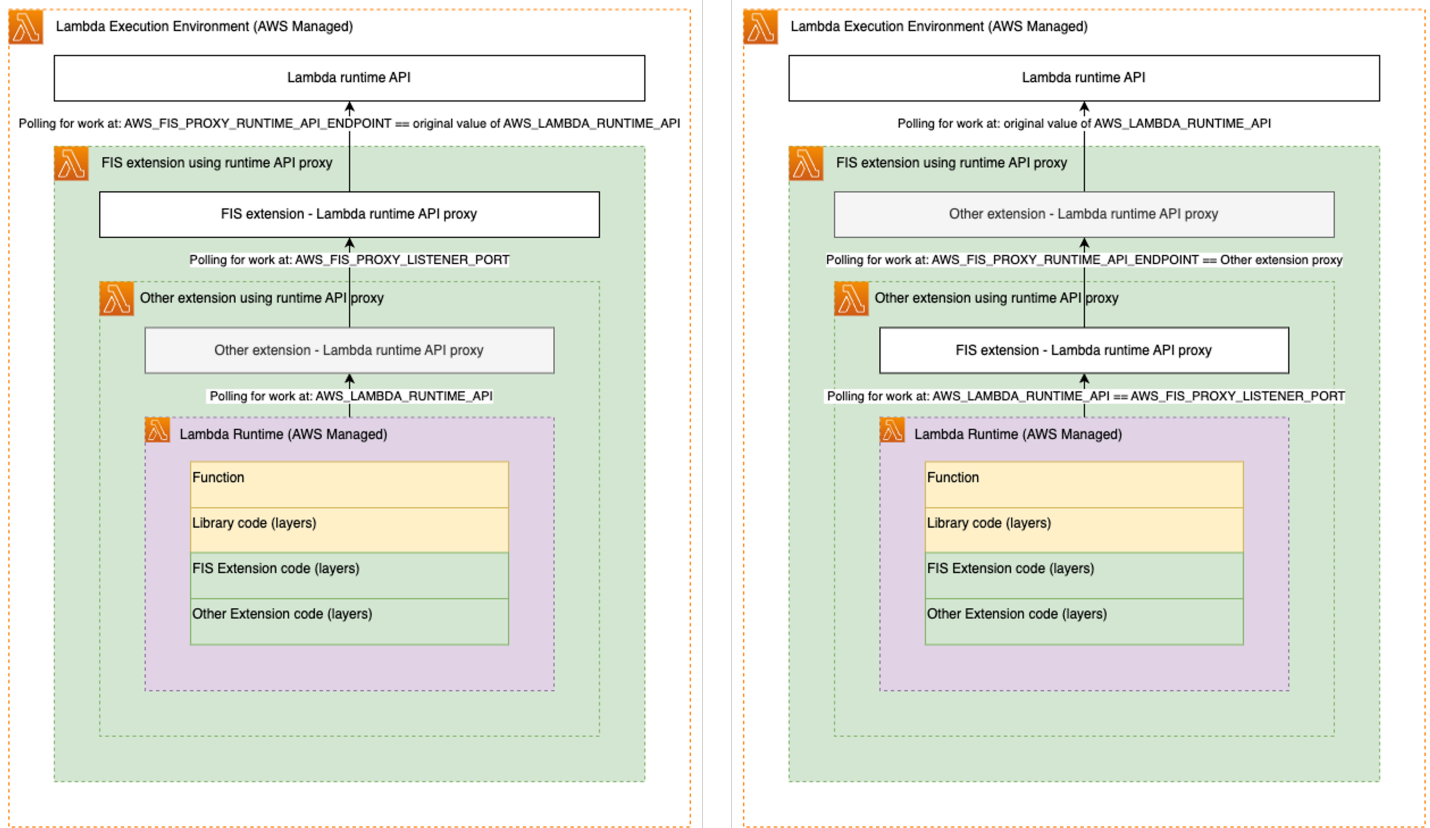


要使用 AWS Lambda Runtime API 代理模式将 AWS FIS Lambda 扩展与其他扩展一起使用，您需要使用自定义引导脚本将代理链接起来。AWS FIS lambda 扩展接受以下环境变量：

- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT`-以 `127.0.0.1:9876` 表示 AWS Lambda 运行时 API 的本地 IP 和侦听器端口的形式取一个字符串。这可能是另一个代理的原始值 `AWS_LAMBDA_RUNTIME_API` 或位置。

- `AWS_FIS_PROXY_LISTENER_PORT`-取一个端口号，默认情况下，AWS FIS 扩展程序应在该端口号上启动自己的代理9100。

通过这些设置，您可以使用 Lambda Runtime API 代理按两个不同的顺序将 AWS FIS 扩展与其他扩展程序链接起来。



有关 AWS Lambda 运行时 API 代理的更多信息，请参阅用户指南中的使用[运行时 API 代理扩展增强 AWS Lambda 运行时安全和治理](#)和使用[Lambda 运行时 API 进行自定义运行时](#)。AWS Lambda

AWS FIS 与容器运行时一起使用

对于使用接受 `AWS_LAMBDA_RUNTIME_API` 环境变量的容器镜像的 AWS Lambda 函数，您可以按照以下步骤将 AWS FIS Lambda 扩展打包到您的容器镜像中：

1. 确定要从中提取扩展的图层的 ARN。有关如何查找 ARN 的更多信息，请参阅。[配置 Lambda 函数](#)
2. 使用 AWS Command Line Interface (CLI) 请求有关扩展的详细信息 `aws lambda get-layer-version-by-arn --arn fis-extension-arn`。响应中将包含一个包含预签名 URL 的 `Location` 字段，您可以从中将 FIS 扩展名下载为 ZIP 文件。

3. 将扩展的内容解压缩到 Docker 文件系统中/opt。以下是基于 NodeJS Lambda 运行时的 Dockerfile 示例：

```
# extension installation #
FROM amazon/aws-lambda-nodejs:12 AS builder
COPY extension.zip extension.zip
RUN yum install -y unzip
RUN mkdir -p /opt
RUN unzip extension.zip -d /opt
RUN rm -f extension.zip
FROM amazon/aws-lambda-nodejs:12
WORKDIR /opt
COPY --from=builder /opt .
# extension installation finished #
# JS example. Modify as required by your runtime
WORKDIR ${LAMBDA_TASK_ROOT}
COPY index.js package.json .
RUN npm install
CMD [ "index.handler" ]
```

有关容器镜像的更多信息，请参阅AWS Lambda 用户指南中的[使用容器镜像创建 Lambda 函数](#)。

AWS FIS Lambda 环境变量

以下是 AWS FIS Lambda 扩展的环境变量列表

- **AWS_FIS_CONFIGURATION_LOCATION**-必填项。写入活动故障配置的位置 AWS FIS ，扩展程序将读取故障配置的位置。这些位置应采用 Amazon S3 ARN 格式，包括存储桶和路径。例如 `arn:aws:s3:::my-fis-config-bucket/FisConfigs/`。
- **AWS_LAMBDA_EXEC_WRAPPER**-必填项。用于配置 Lambda AWS FIS a 扩展的 AWS Lambda [包装脚本](#)的位置。应将其设置为扩展中包含的 `/opt/aws-fis/bootstrap` 脚本。
- **AWS_FIS_LOG_LEVEL**-可选。Lambda AWS FIS a 扩展程序发出的消息的日志级别。支持的值有 INFO、WARN 和 ERROR。如果未设置，则 AWS FIS 扩展名将默认为 INFO。
- **AWS_FIS_EXTENSION_METRICS**-可选。可能的值为 `all` 和 `none`。如果设置为扩展名 `all`，则将在下方发出 EMF 指标。 `aws-fis-extension namespace`
- **AWS_FIS_SLOW_POLL_INTERVAL_SECONDS**-可选。如果已设置，则将在扩展程序不注入故障并等待将故障配置添加到配置位置时覆盖轮询间隔（以秒为单位）。默认值为 60。

- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT`-可选。如果设置了则将覆盖的值 `AWS_LAMBDA_RUNTIME_API` 以定义 AWS FIS 扩展与 AWS Lambda 运行时 API 交互的位置，以控制函数调用。例如，期望 IP: 端口。127.0.0.1:9000 有关更多信息 `AWS_LAMBDA_RUNTIME_API`，请参阅用户指南 [中的将 Lambda 运行时 API 用于自定义运行时](#)。AWS Lambda
- `AWS_FIS_PROXY_LISTENER_PORT`-可选。定义 AWS FIS Lambda 扩展公开 AWS Lambda 运行时 API 代理的端口，该代理可供其他扩展或运行时使用。默认值为 9100。
- `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS`-可选。如果设置为非零值，则此变量定义了扩展程序在评估故障配置和开始调用运行时之前等待运行中异步轮询完成的毫秒数。默认值为 0。

适用于 Lambda 的 AWS FIS 扩展程序的可用版本

本节包含有关 AWS FIS Lambda 扩展版本的信息。该扩展支持为 x86-64 和 ARM64 (Graviton2) 平台开发的 Lambda 函数。您的 Lambda 函数必须配置为使用其当前托管 AWS 区域位置的特定亚马逊资源名称 (ARN)。您可以在下面查看 AWS 区域 和 ARN 详情。

主题

- [AWS FIS Lambda 扩展版本说明](#)
- [Lambda 扩展程序访问指南 ARNs](#)
- [查找您的 Lambda 扩展程序版本号](#)

AWS FIS Lambda 扩展版本说明

下表描述了对 AWS FIS Lambda 扩展的最新版本所做的更改

版本	推出日期	注意
1.0.0	2024-10-29	初始版本

Lambda 扩展程序访问指南 ARNs

在使用控制台搜索公共参数 `AWS 区域` 之前，您的 AWS 账户 和中必须至少有一个参数。要发现公共参数，请参阅 [在参数存储中发现公共参数](#)。

控制台访问权限：

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 Parameter Store。
3. 选择 Public parameters (公有参数) 选项卡。
4. 选择 Select a service (选择服务) 下拉菜单。从下拉选项中选择 fis。
5. (可选) 通过在搜索栏中输入更多信息，筛选您选择的参数。对于 arm64 架构，通过输入 “arm 64” 来筛选参数。对于 x86_64 架构，通过输入 “x86_64” 来筛选参数。
6. 选择要使用的公有参数。
7. 从参数详细信息中找到 ARN 值。复制 ARN 以用于在目标 Lambda 函数上配置层扩展。

AWS CLI 访问权限：

SSM 参数名称

以下 SSM 参数名称适用于不同的架构：

1. arm64 : /aws/service/fis/lambda-extension/AWS-FIS-extension-arm64/1.x.x
2. x86_64 : /aws/service/fis/lambda-extension/AWS-FIS-extension-x86_64/1.x.x

AWS CLI 命令格式

要检索扩展 ARNs，请使用以下 AWS CLI 命令格式，其中 ParameterName 是架构的名称，区域是目标：AWS 区域

```
aws ssm get-parameter --name parameterName --region region
```

示例用法

```
aws ssm get-parameter --name /aws/service/fis/lambda-extension/AWS-FIS-extension-x86_64/1.x.x --region ap-southeast-2
```

响应格式

该命令返回一个包含参数详细信息的 JSON 对象，如下所示。Lambda 层的 ARN 包含在参数对象的值字段中。复制 ARN 以用于在目标 Lambda 函数上配置层扩展。

```
{
  "Parameter":
  {
    "Name": "/aws/service/fis/lambda-extension/AWS-FIS-extension-
x86_64/1.x.x",
    "Type": "String",
    "Value": "arn:aws:lambda:ap-southeast-2:211125361907:layer:aws-fis-
extension-x86_64:9",
    "Version": 1,
    "LastModifiedDate": "2025-01-02T15:13:54.465000-05:00",
    "ARN": "arn:aws:ssm:ap-southeast-2::parameter/aws/service/fis/lambda-
extension/AWS-FIS-extension-x86_64/1.x.x",
    "DataType": "text"
  }
}
```

编程访问：

使用基础设施即代码 (IaC) 构建或配置 Lambda 函数时，以编程方式检索这些公共参数。这种方法有助于使用最新层版本 ARN 维护您的 Lambda 函数，而无需手动更新扩展 AWS FIS 层 ARN 时所需的代码。以下资源展示了如何使用常用 IaC 平台检索公共参数：

- [使用 AWS SDK 获取公共参数](#)
- [使用 AWS CDK 从 AWS Systems Manager 参数存储库获取公共参数](#)
- [使用 Terraform 获取公共参数](#)

查找您的 Lambda 扩展程序版本号

使用以下步骤查找您当前配置的 AWS FIS Lambda 扩展的版本号。

1. 打开 AWS Lambda 控制台，网址为 <https://console.aws.amazon.com/lambda/>。
2. 选择要添加 AWS-FIS-Extension 层的 Lambda 函数。
3. 在层部分中，选择编辑。
4. 在编辑图层部分中，选择添加图层。
5. 在选择图层部分，选择指定 ARN。
6. 输入与您的 AWS 区域 和架构对应的 AWS FIS 扩展层的 ARN。您可以使用前几节中描述的控制台或编程访问方法查找 ARN。AWS CLI

7. 选择验证以确认图层 ARN 有效，然后选择添加。
8. 使用测试选项卡测试此函数。
9. 测试完成后，查看日志输出。在“执行详情”部分中找到 AWS FIS Lambda 扩展版本。

管理 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。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。有关实验模板组件的更多信息，请参阅[实验模板组件](#)。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](#)
- [使用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](#)
- [AWS FIS 实验模板示例](#)

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<https://console.aws.amazon.com/fis/>为。
2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 对于步骤 1，指定模板详细信息，请执行以下操作：
 - a. 在描述和名称中，输入模板的描述，例如 Amazon S3 Network Disrupt Connectivity。

- b. (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。
 - c. 选择下一步, 然后进入步骤 2, 指定操作和目标。
5. 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。
6. 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, IDs 然后选择或添加资源 IDs。
 - 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。
7. 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。

8. (可选) 对于实验选项, 请选择空目标分辨率模式的行为。
9. 选择下一步进入步骤 3, 配置服务访问权限。
10. 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”](#)。
11. (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 ARNs 带有 CSV 文件的目标账户角色, 请选择 ARNs 为所有目标账户上传角色, 然后选择选择 CSV 文件
12. 选择下一步进入步骤 4, 配置可选设置。
13. (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](#)。
14. (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。
15. (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。
16. 选择“下一步”进入步骤 5, “查看并创建”。
17. 查看模板并选择创建实验模板。当提示您确认时, 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](#)。有关模板示例, 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<https://console.aws.amazon.com/fis/>为。
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，然后使用该[get-experiment-template](#)命令获取有关特定实验模板的信息。

使用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，以验证是否已将实验模板配置为以预期资源为目标。开始实际实验时作为目标资源可能会与预览中的资源不同，因为资源可能会被随机删除、更新或采样。生成目标预览时，开始的实验会跳过所有操作。

Note

生成目标预览时，不允许验证您是否具有对资源采取操作所需的权限。

使用控制台启动目标预览

1. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
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，此目标预览将随着最新实验中的目标自动更新。

使用 CLI 启动目标预览

- 运行以下 [start-experiment](#) 命令。将斜体值替换为您自己的值。

```
aws fis start-experiment \  
  --experiment-options actionsMode=skip-all \  
  --experiment-template-id EXTxxxxxxxx
```

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<https://console.aws.amazon.com/fis/>为。
2. 在导航窗格中，选择实验模板。
3. 选择实验模板，然后选择开始实验。
4. （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。
5. 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<https://console.aws.amazon.com/fis/>为。

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<https://console.aws.amazon.com/fis/>为。
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<https://console.aws.amazon.com/fis/>为。
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 [AWS FIS 实验模板组件](#)。

要使用其中一个示例模板创建实验，请将其保存到 JSON 文件（例如 `my-template.json`），将中的 *italics* 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

- [根据过滤器停止 EC2 实例](#)
- [停止指定数量的 EC2 实例](#)
- [运行预配置的 AWS FIS SSM 文档](#)
- [运行预定义的自动化运行手册](#)
- [限制对具有目标 IAM 角色的 EC2 实例的 API 操作](#)
- [对 Kubernetes 集群中的容器组 \(pod \) CPU 进行压力测试](#)
- [指定数量的 Kinesis Data Streams 的预配置吞吐量异常](#)
- [实验角色权限示例](#)

根据过滤器停止 EC2 实例

以下示例使用指定 VPC 中的指定标签停止指定区域中所有正在运行的 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"
}
```

运行预配置的 AWS FIS SSM 文档

以下示例使用预配置的 [AWS FIS SSM 文档-CPU-stress](#) 在指定 EC2 实例上运行 CPU 故障注入 60 秒。[AWSFIS-Run](#) 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 AWS-Publish](#) 提供的运行手册向亚马逊 SNS 发布通知。[SNSNotification](#) 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"
}
```

限制对具有目标 IAM 角色的 EC2 实例的 API 操作

以下示例限制 100% API 调用，该调用是在目标定义中指定的 IAM 角色发出的 API 调用的操作定义中指定的。

Note

如果您想定位属于 Auto Scaling 组成员的 EC2 实例，请使用 `aws:ec2:asg-insufficient-instance-capacity-error` 操作，改用 Auto Scaling 组进行定位。有关更多信息，请参阅 [aws:ec2:asg-insufficient-instance-capacity-error](#)。

```
{
  "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\"],\n\"labelSelectors\":{\"run\":{\"nginx\"}},\n\"mode\":{\"all\"},\n\"stressors\":{\"cpu\":{\"workers\":1,\n\"load\":50}},\n\"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": "{\n  \"engineState\": \"active\",\n  \"appinfo\":\n  {\n    \"appns\": \"default\",\n    \"applabel\": \"run=nginx\",\n    \"appkind\": \"deployment\"\n  },\n  \"chaosServiceAccount\": \"litmus-admin\",\n  \"experiments\": [\n    {\n      \"name\": \"pod-cpu-hog\",\n      \"spec\": {\n        \"components\": {\n          \"env\": [\n            {\n              \"name\": \"TOTAL_CHAOS_DURATION\",\n              \"value\": \"60\"\n            },\n            {\n              \"name\": \"CPU_CORES\",\n              \"value\": \"1\"\n            },\n            {\n              \"name\": \"PODS_AFFECTED_PERC\",\n              \"value\": \"100\"\n            },\n            {\n              \"name\": \"CONTAINER_RUNTIME\",\n              \"value\": \"docker\"\n            },\n            {\n              \"name\": \"SOCKET_PATH\",\n              \"value\": \"/var/run/docker.sock\"\n            }\n          ]\n        },\n        \"probe\": []\n      }\n    }\n  ],\n  \"annotationCheck\": \"false\"\n}"
      },
      "targets": {
        "Cluster": "MyCluster"
      }
    }
  },
  "stopConditions": [{
    "source": "none"
  }],
  "roleArn": "arn:aws:iam::111122223333:role/role-name",
  "tags": {}
}

```

指定数量的 Kinesis Data Streams 的预配置吞吐量异常

以下示例针对最多五个带有指定标签的 Kinesis Data Streams 请求的 100% 抛出预配置吞吐量异常。AWS FIS 随机选择要影响的直播。5 分钟后，故障被清除。

```

{
  "description": "Kinesis stream experiment",
  "targets": {
    "KinesisStreams-Target-1": {
      "resourceType": "aws:kinesis:stream",
      "resourceTags": {
        "tag-key": "tag-value"
      }
    }
  }
}

```

```
    },
    "selectionMode": "COUNT(5)"
  }
},
"actions": {
  "kinesis": {
    "actionId": "aws:kinesis:stream-provisioned-throughput-exception",
    "description": "my-stream",
    "parameters": {
      "duration": "PT5M",
      "percentage": "100",
      "service": "kinesis"
    },
    "targets": {
      "KinesisStreams": "KinesisStreams-Target-1"
    }
  }
},
"stopConditions": [
  {
    "source": "none"
  }
],
"roleArn": "arn:aws:iam::111122223333:role/role-name",
"tags": {},
"experimentOptions": {
  "accountTargeting": "single-account",
  "emptyTargetResolutionMode": "fail"
}
}
```

实验角色权限示例

以下权限允许您在影响 50% 请求的特定流上运行 `aws:kinesis:stream-provisioned-throughput-exception` 和 `aws:kinesis:stream-expired-iterator-exception` 操作。

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

```
    "Condition": {
      "ForAllValues:StringEquals": {
        "kinesis:FisActionId": [
          "aws:kinesis:stream-provisioned-throughput-exception",
          "aws:kinesis:stream-expired-iterator-exception"
        ],
        "kinesis:FisTargetArns": [
          "arn:aws:kinesis:us-east-1:111122223333:stream/stream-name"
        ],
      },
      "NumericEquals": {
        "kinesis:FisInjectPercentage": "50"
      }
    },
    {
      "Action": [
        "kinesis:DescribeStreamSummary",
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

管理你的 AWS FIS 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 使用率指标。有关更多信息，请参阅[监视器使用 CloudWatch](#)。

- 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<https://console.aws.amazon.com/fis/>为。
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。

- 已取消：由于激活了安全杆，实验已停止或无法启动。

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<https://console.aws.amazon.com/fis/>为。
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<https://console.aws.amazon.com/fis/>为。
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<https://console.aws.amazon.com/fis/>为。
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。

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 停止和启动测试实例](#)
- [教程：使用 AWS FIS 在实例上运行 CPU stress](#)
- [教程：使用 AWS FIS 测试点实例中断](#)
- [教程：模拟连接事件](#)
- [教程：安排定期实验](#)

教程：使用 AWS FIS 停止和启动测试实例

您可以使用 AWS 故障注入服务 (AWS FIS) 来测试您的应用程序如何处理实例停止和启动。使用本教程创建实验模板，该模板使用 AWS FIS `aws:ec2:stop-instances` 操作停止一个实例，然后停止第二个实例。

先决条件

要完成本教程，请确保您已做好以下准备：

- 在您的账户中启动两个测试 EC2 实例。启动实例后，请记下 IDs 两个实例的情况。
- 创建一个 IAM 角色以使 AWS FIS 服务能够代表您执行 `aws:ec2:stop-instances` 操作。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](#)。

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。第一个操作会停止 AWS FIS 随机选择的其中一个测试实例。第二项操作会停止所有测试实例。

创建实验模板

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

2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 对于步骤 1，指定模板详细信息，请执行以下操作：
 - a. 在描述和名称中，输入模板的描述，例如 Amazon S3 Network Disrupt Connectivity。
 - b. 选择“下一步”，然后移至步骤 2，指定操作和目标。
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 Target，请保留 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. AWS 为 FIS 在上一步中自动为您创建的目标选择“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 Target 方法 IDs，选择资源，然后选择两个测试实例中的一个。IDs
 - 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 Target 方法 IDs，选择资源，然后选择两个测试实例中的一个。IDs
 - 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. 选择下一步进入步骤 3，配置服务访问权限。
 10. 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”](#)。
 11. 选择下一步进入步骤 4，配置可选设置。
 12. (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。
 13. 选择“下一步”进入步骤 5，“查看并创建”。
 14. 查看模板并选择创建实验模板。当提示您确认时，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. 在新的浏览器选项卡或窗口<https://console.aws.amazon.com/ec2/>中打开 Amazon EC2 控制台。这使您可以继续在 AWS FIS 控制台中跟踪实验进度，同时在 Amazon 控制 EC2 台中查看实验结果。
2. 在导航窗格中，选择实例。
3. 当第一个操作的状态从“待处理”变AWS为“正在运行”（FIS 控制台）时，其中一个目标实例的状态将从“正在运行”变为“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

如果您不再需要为此实验创建的测试 EC2 实例，则可以将其终止。

终止实例

1. 打开亚马逊 EC2 控制台，网址为<https://console.aws.amazon.com/ec2/>。

2. 在导航窗格中，选择实例。
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. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
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 stress

您可以使用 AWS 故障注入服务 (AWS FIS) 来测试您的应用程序如何处理 CPU 压力。使用本教程创建实验模板，该模板使用 AWS FIS 来运行预先配置的 SSM 文档，该文档在实例上运行 CPU stress。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，请参阅 [the section called “预先配置的 AWS FIS SSM 文档”](#)。

先决条件

在使用 AWS FIS 运行 CPU stress 之前，请先完成以下先决条件。

创建一个 IAM 角色

创建一个角色并附加一个策略，使 AWS FIS 能够代表您使用该 `aws:ssm:send-command` 操作。有关更多信息，请参阅 [适用于 AWS FIS 实验的 IAM 角色](#)。

验证对 AWS FIS 的访问权限

确保您可以访问 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，并且该实例具有与 Amazon SSMManaged InstanceCore zon 策略关联的 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，请参阅 [停止条件](#)。

创建警报，指明 CPU 利用率何时超过阈值

1. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。
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-Run end-command/-CPU-stress](#)。

创建试验模板

1. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 对于步骤 1，指定模板详细信息，请执行以下操作：
 - a. 在描述和名称中，输入模板的描述。
 - b. 选择“下一步”，然后移至步骤 2，指定操作和目标。
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-Run end-command/-CPU-stress`。此操作会自动向 Document ARN 添加 SSM 文档的 ARN。
 - d. 对于 Target，请保留 AWS FIS 为你创建的目标。
 - e. 对于操作参数，文档参数，输入以下内容：

```
 {"DurationSeconds": "120"} 
```
 - f. 在“操作”参数“持续时间”中，指定 5 分钟 (PT5M)。
 - g. 选择保存。
6. 对于目标，请执行以下操作：
 - a. AWS 为 FIS 在上一步中自动为您创建的目标选择“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”IDs，选择“资源”，然后选择测试实例的 ID。
 - e. 对于选择模式，选择全部。
 - f. 选择保存。

7. 选择下一步进入步骤 3，配置服务访问权限。
8. 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”](#)。
9. 选择“下一步”进入步骤 4 “配置可选设置”。
10. 对于停止条件，请选择您在步骤 1 中创建的 CloudWatch 警报。
11. （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。
12. 选择“下一步”进入步骤 5，“查看并创建”。
13. 查看模板并选择创建实验模板。当提示您确认时，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 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/AllowFISSMActions",
"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

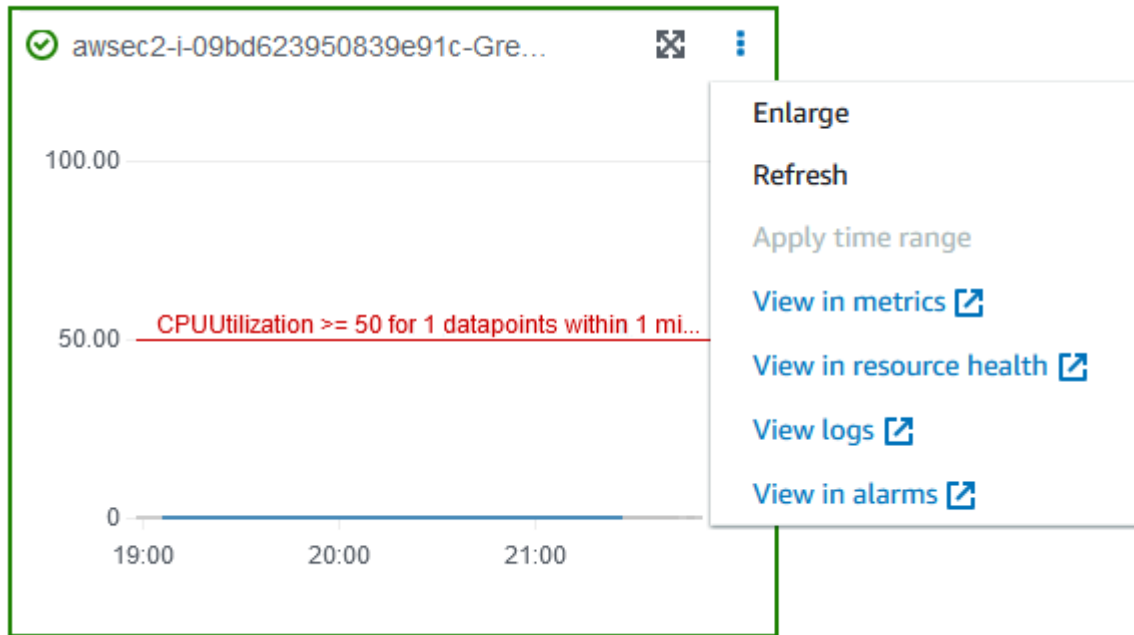
1. 您应该位于刚开始的实验的详细信息页面。否则，请选择实验，然后选择对应 ID 打开详细信息页面。
2. 要查看实验状态，请在详细信息窗格中选择状态。有关更多信息，请参阅[实验状态](#)。
3. 当实验状态为正在运行时，转到下一步。

步骤 5：验证实验结果

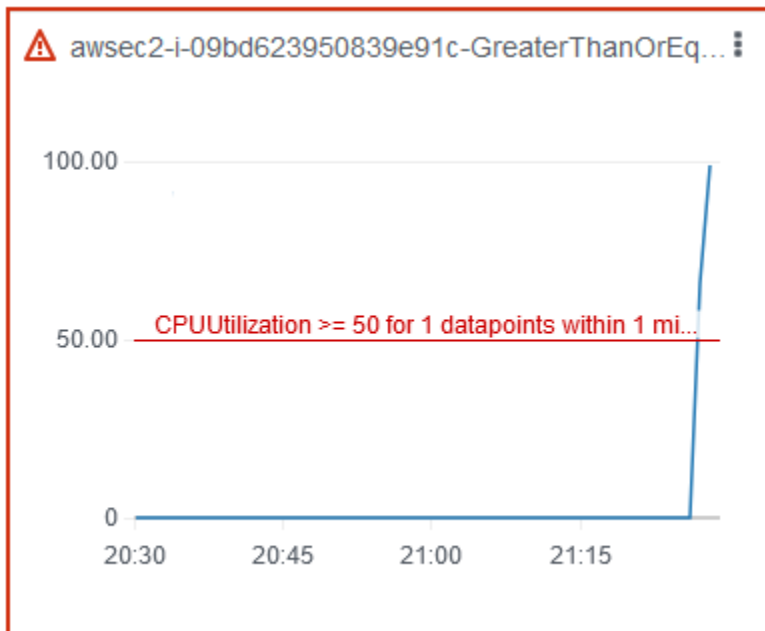
您可以在实验运行期间监控实例的 CPU 利用率。CPU 利用率达到阈值时会触发警报，实验也会因停止条件而停止。

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：清除

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. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。
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. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
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 测试点实例中断

竞价型实例使用可用的备用 EC2 容量，与按需定价相比，可享受高达 90% 的折扣。但是，Amazon EC2 可以在需要恢复容量时中断您的竞价型实例。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](#)。

您可以使用 AWS 故障注入服务 (AWS FIS) 来测试您的应用程序如何处理竞价型实例中断。使用本教程创建实验模板，该模板使用 AWS FIS `aws:ec2:send-spot-instance-interruptions` 操作来中断您的一个竞价型实例。

或者，要使用亚马逊 EC2 控制台启动实验，请参阅亚马逊 EC2 用户指南中的[启动竞价型实例中断](#)。

先决条件

在使用 AWS FIS 中断竞价型实例之前，请先完成以下先决条件。

1. 创建 IAM 角色

创建一个角色并附加一个策略，使 AWS FIS 能够代表您执行 `aws:ec2:send-spot-instance-interruptions` 操作。有关更多信息，请参阅 [适用于 AWS FIS 实验的 IAM 角色](#)。

2. 验证对 AWS FIS 的访问权限

确保您可以访问 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 可以识别目标竞价型实例

使用 [create-tags 命令添加标签](#) Name=interruptMe 到您的目标竞价型实例。

```
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。

创建试验模板

1. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
2. 在导航窗格中，选择实验模板。
3. 选择创建实验模板。
4. 对于步骤 1，指定模板详细信息，请执行以下操作：
 - a. 对于描述和名称，输入模板的描述和名称。
 - b. 选择下一步，然后进入步骤 2，指定操作和目标。
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 Target，请保留 AWS FIS 为您创建的目标。
 - e. 在“操作”参数“中断前持续时间”中，指定 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. 选择下一步进入步骤 3，配置服务访问权限。
 8. 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”](#)。
 9. 选择下一步进入步骤 4，配置可选设置。
 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“下一步”进入步骤 5，“查看并创建”。
 12. 查看模板并选择创建实验模板。当提示您确认时，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](#)。
- [竞价型实例中断通知](#)在 Amazon EC2 终止或停止您的实例前两分钟发出。
- 两分钟后终止或停止竞价型实例。
- 被 AWS FIS 停止的竞价型实例将保持停止状态，直到您重新启动该实例。

验证实验是否已经中断实例

1. 打开亚马逊 EC2 控制台，网址为<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。

要取消请求并终止实例，请使用 AWS CLI

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. 打开 AWS FIS 控制台，网址 <https://console.aws.amazon.com/fis/> 为。
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。

- `s3express`— 拒绝进出目标子网可用区中 Amazon S3 Express One 区域的区域终端节点的流量。目标子网必须位于 S3 Express One 区域当前可用 AZs 的地方。有关更多信息，请参阅 [S3 Express One 区域可用区和区域](#)。
- `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 [IAM 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。按照以下教程，启动并连接 Amazon EC2 实例：《Amazon EC2 用户指南》中的[教程：开始使用 Amazon EC2 Linux 实例](#)。

步骤 1：创建 AWS FIS 实验模板

使用 AWS FIS 管理控制台 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。

创建试验模板

1. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
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. 对于步骤 1，指定模板详细信息，请执行以下操作：
 - a. 在描述和名称中，输入模板的描述，例如 Amazon S3 Network Disrupt Connectivity。
 - b. 选择下一步，然后进入步骤 2，指定操作和目标。
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 IDs，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. 选择下一步进入步骤 3，配置服务访问权限。
8. 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”](#)。
9. 选择下一步进入步骤 4，配置可选设置。
10. (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](#)。
11. (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。
12. 选择“下一步”进入步骤 5，“查看并创建”。
13. 查看模板并选择创建实验模板。当提示您确认时，输入 `create`，然后选择创建实验模板。

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。

要排查相关问题，请参阅《Amazon EC2 用户指南》中的[排查实例的连接问题](#)。

2. 查看您的实例 AWS 区域 所在的位置。您可以在 Amazon EC2 控制台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<https://console.aws.amazon.com/fis/>为。
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. 打开位于 <https://console.aws.amazon.com/ec2/> 的 Amazon EC2 控制台。
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<https://console.aws.amazon.com/fis/>为。
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

借助 AWS 故障注入服务 (AWS FIS)，您可以对 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。当你同时使用时 Amazon EventBridge，您可以将实验安排为一次性任务或重复任务。

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

在开始本教程之前，必须有一个要按计划运行 AWS 的 FIS 实验模板。如果已有可用的实验模板，请记住模板 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. 使用 <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”，然后插入以下片段以允许 S Amazon EventBridge scheduler 代表您担任该角色。

JSON

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

```

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

```

选择下一步。

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。

JSON

```

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

```

您可以将调度程序限制为仅运行具有特定标签值 AWS 的 FIS 实验模板。例如，以下策略授予所有 AWS FIS 实验的 StartExperiment 权限，但将调度程序限制为只能运行带有标签的实验模板。Purpose=Schedule

JSON

```
{
```

```
"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 日程安排器

1. 打开 Amazon 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. 确认您与您的 AWS FIS 实验模板 AWS 区域 相同。
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。
 - 选择下一步。
5. 在“选择目标”下，选择“全部” APIs，然后搜索 AWS FIS。
 6. 选择 AWS FIS，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。在本教程中，我们使用的是 Amazon EventBridge 调度器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"
}
```

选择下一步。

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. 选择下一步。
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

验证您的 AWS FIS 实验是否按计划进行

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

禁用您的 Amazon EventBridge 日程安排

1. 打开 Amazon 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。

使用场 AWS FIS 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。

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](#)
- [场景参考](#)

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. 某些方案允许您编辑在多个操作或目标之间共享的参数。一旦您对场景进行任何更改（包括通过共享参数编辑进行更改），此功能将被禁用。要使用此功能，请选择编辑共享参数按钮。在模式中编辑参数，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](#) 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 [管理 AWS FIS 实验模板](#) 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。

场景参考

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的标签来标记资源，也可以使用共享的参数编辑体验设置自己的标签（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 AWS FIS 景库](#)。

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[EKS 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。

AWS FIS 支持以下单可用区、多可用区和多区域应用程序场景。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。
- AZ: Application Slowdown-在单个可用区 (AZ) 内的资源之间增加延迟，以减慢应用程序的速度。了解有关 [AZ: Application Slowdown](#) 的更多信息。

- **Cross-AZ: Traffic Slowdown**-注入数据包丢失以中断和减慢可用区之间的流量 (AZs)。了解有关 [Cross-AZ: Traffic Slowdown](#) 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。

AWS FIS 支持 Amazon EBS 卷的以下场景。这些场景使用标签来定位卷。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。目标卷必须位于同一个可用区内。有关更多信息，请参阅 [Amazon EBS 上的故障测试](#)。

- **EBS: Sustained Latency**— 探索持续延 I/O 迟对应用程序的影响。

在这种情况下，我们将瞄准当前可用区中附加特定标签的卷。此场景在 15 分钟内使用单个延迟操作，在卷的 50% 读取和 100% 的写入操作中注入 500 毫秒的持续延迟。在这种情况下，您可以自定义注入的延迟时间、注 I/O 入的百分比和操作的持续时间。

- **EBS: Increasing Latency**— 探索 I/O 延迟增加对应用程序的影响。

在这种情况下，我们将瞄准当前可用区中附加特定标签的卷。这种情况会在 15 分钟内使用五个延迟操作对卷进行 10% 的读取和 25% 的写入操作注入 50 毫秒、200 毫秒、700 毫秒、1 秒和 15 秒的延迟。在这种情况下，您可以为每个延迟操作自定义注入的延迟量、I/O 注入的百分比和操作持续时间。

- **EBS: Intermittent Latency**— 探索间歇性 I/O 延迟峰值对应用程序的影响。

在这种情况下，我们将瞄准当前可用区中附加特定标签的卷。此场景使用三个延迟操作，在卷的 0.1% 的读取和写入 I/O 操作上注入三个间歇性尖峰，分别为 30 秒、10 秒和 20 秒，每个峰值之间的恢复间隔为 15 分钟。在这种情况下，您可以为每个延迟操作自定义注入的延迟量、I/O 注入的百分比和操作持续时间。

- **EBS: Decreasing Latency**— 探索减少延 I/O 迟对应用程序的影响。

在这种情况下，我们将瞄准当前可用区中附加特定标签的卷。此场景在 15 分钟内使用五个延迟操作，为卷的 10% 读取和写入操作注入 20 秒、5 秒、900 毫秒、300 毫秒和 40 毫秒的延迟。在这种情况下，您可以为每个延迟操作自定义注入的延迟量、I/O 注入的百分比和操作持续时间。

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、可用区无法重新扩展计算、子网连接丢失、RDS 故障转移、故障转移、ElastiCache 对 S3 Express One Zone 目录存储桶的访问受损以及 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務。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。特别是，以下内容 APIs 将受到影响：`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配置新实例。

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。特别是，以下内容 APIs 将受到影响：`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配置新实例。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。

故障转移 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，写入器节点将故障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，如果写入器位于受影响的可用区中，则 AZ Availability: Power Interruption 包括 [aws: rds: failover-db-cluster](#) 以进行故障转移。

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 复制组

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:replicationgroup-interrupt-az-power](#)，用于终止受影响可用区中的 ElastiCache 节点。在中断期间，不会在受影响的可用区中配置新实例，因此复制组的容量将保持在减少的状态。

此操作以 ElastiCache 复制组为目标。默认情况下，它以复制组为目标，其标签名为 `AzImpairmentPower`，值为 `ElasticacheImpact`。您可以将此标签添加到复制组中，也可以将默认标签替换为实验模板中您自己的标签。默认情况下，如果未找到有效的复制组，则将跳过此操作。请注意，只有节点位于受影响可用区的复制组才会被视为有效目标。

启动 ARC 区域自动换档

在可用区电源中断开始五分钟后，在剩余的 25 分钟电源中断期间，恢复操作 `aws:arc:start-zonal-autoshift` 会自动将资源流量从指定的可用区转移出去。在这段时间之后，流量会转移回原来的可用区。请注意，在现实世界的可用区中，如果启用了自动换档，则电源中断 AWS 将检测到损耗并转移资源流量。虽然这种转移的时间各不相同，但估计是在减值开始后五分钟内发生的。

此操作针对支持自动移位的 Amazon 应用程序恢复控制器 (ARC) 资源。默认情况下，它以标签键 `AzImpairmentPower` 和值为目标的资源 `RecoverAutoshiftResources`。您可以将此标签添加到您的资源中，也可以将默认标签替换为实验模板中您自己的标签。例如，您可能想要使用特定于应用程序的标签。默认情况下，如果未找到有效资源，则会跳过此操作。

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:efs: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。

中断与 S3 Express One 区域目录存储桶的连接

在可用区电源中断期间，无法访问存储在可用区中 S3 Express One Zone 目录存储桶中的数据。可用区可用性：电源中断包括 [aws:network:disrupt-connectivity](#)，用于在实验期间中断子网与受影响可用区中单区域目录存储桶之间的连接，从而导致区域端点数据平面 API 操作超时。使用此操作可以测试计算与存储位于同一可用区内的中断情况。

此操作以子网为目标。默认情况下，它以 `AzImpairmentPower` 带有名为、值为的标签的子网为目标。 `DisruptSubnet` 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。

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。
- 在目标可用区中，在 EC2 上运行的 Amazon EKS 容器组将通过 EC2 Worker 节点终止，并且将阻止新 EC2 节点启动。但是，不支持在 AWS Fargate 上运行的 Amazon EKS 容器组。
- 在目标可用区中，在 EC2 上运行的 Amazon ECS 任务将通过 EC2 Worker 节点终止，并且将阻止新 EC2 节点启动。但是，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 将进行故障转移，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

- 为 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。

Permissions

ARC 区域自动切换使用 IAM 服务相关角色代

表 [AWSServiceRoleForZonalAutoshiftPracticeRun](#) 您执行区域切换。此角色使用 IAM 托管策略 [AWSZonalAutoshiftPracticeRunSLRPolicy](#)。您无需手动创建角色。当您在 AWS 管理控制台、或 AWS SDK 中根据可用区电源中断场景创建实验模板时，ARC 会为您创建服务相关角色。AWS CLI 有关更多信息，请参阅在 [ARC 中使用服务相关角色进行区域自动切换](#)。

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](#)。

JSON

```
{
  "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，并使用 AWS [create-experiment-template](#) 命令行界面 (AWS CLI) Line CLI 中的命令创建 [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: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-I0": {
    "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:replicationgroup-interrupt-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."
}
```

AZ: Application Slowdown

您可以使用“可用区：应用程序减速”场景在单个可用区 (AZ) 内的资源之间引入额外的延迟。这种延迟会产生许多应用程序速度减慢的症状，即部分中断，有时被称为灰色故障。它会增加目标资源之间的网络流延迟。网络流量代表计算资源之间的流量，即在服务器、容器和服务之间传输请求、响应和其他通信的数据包。该场景可以帮助验证可观测性设置，调整警报阈值，发现应用程序对减速的敏感性，并练习关键的运营决策，例如AZ疏散。

默认情况下，该场景会将选定可用区内目标资源之间的 100% 网络流量增加 200 毫秒的延迟，持续时间为 30 分钟。您可以使用 AWS FIS 控制台中的编辑共享参数对话框在场景级别调整以下参数，然后将其应用于基础操作：

- 可用区-您可以选择场景中要损坏的可用区。
- 毫秒 (ms) 延迟-根据应用程序的灵敏度和需求进行调整。例如，您可以为更敏感的应用程序设置较低的延迟，也可以设置更高的延迟以测试超时处理。考虑使用当前应用程序延迟的倍数作为基准。
- 流量百分比-减少以损害部分流量。例如，您可以添加 200 毫秒的延迟，影响 25% 的网络流，以进行更微妙的测试。
- 持续时间-设置实验的运行时长。您可以缩短时间以加快测试速度，也可以运行更长的持续测试。例如，将持续时间设置为 2 小时，以测试受损条件下的恢复机制。
- 资源定向-您可以使用标签（用于 EC2 或 Fargate 上的 EC2 实例或 ECS 任务）或标签（适用于 EC2 上的 EKS 容器）来定义整个场景的目标资源。您可以指定自己的标签和标签，也可以使用场景中提供的默认值。如果您不想使用标签或标签，则可以通过指定其他参数来编辑目标资源的操作。
- 自定义-如果您不想以 EC2 或 ECS 资源为目标，则可以将操作保留为默认标签。实验找不到任何可以瞄准的资源，因此该操作将被跳过。但是，如果您不想以 EKS 资源为目标，则应将 EKS 操作和目标完全从场景中删除，因为它需要提供 EKS 集群标识符。要进行更精细的自定义，您可以直接修改实验模板中的单个操作。

操作

通过在网络流上引入额外的延迟，然后通过应用程序传播，以下操作共同帮助在单个可用区中创建应用程序速度减慢的许多症状。这些操作并行运行，默认情况下，每个操作都会增加 200 毫秒的延迟，持续 30 分钟。在这段时间之后，延迟将恢复到正常水平。该场景至少需要以下一种资源类型才能运行：EC2 实例、ECS 任务或 EKS 容器。

ECS 网络延迟

AZ：应用程序减速包括 [aws: ecs: 用于引入 task-network-latency 入 ECS 任务的延迟](#)。该操作的目标是所选可用区中的任务。默认情况下，它的目标任务的 [标签](#) 名为 AZApplicationSlowdown，值为 LatencyForECS。您可以用自己的标签替换默认标签，也可以将场景标签添加到任务中。如果未找到有效任务，则将跳过此操作。在 ECS 上运行实验之前，应按照 [ECS 任务操作的设置步骤进行操作](#)。

EKS 网络延迟

AZ：应用程序减速包括 [aws: eks: 用于引入 EKS pod 的延迟](#)。pod-network-latency 该操作的目标是选定可用区中的 Pod。默认情况下，它的目标是集群中带有格式为 key=value 的标签的 pod。提供的默认标签是 AZApplicationSlowdown=LatencyForEKS。你可以用自己的标签替换默认标签，也可以将此标签添加到你的 pod 中。如果未找到有效的 pod，则将跳过此操作。在 EKS 上运行实验之前，应按照 [EKS pod 操作的设置步骤进行操作](#)。

EC2 网络延迟

AZ：应用程序减速使用 [aws: ssm: send-command 操作来运行-AWS FIS-Run Network-Latency-Sources 文档](#)，引入 EC2 实例的延迟。该操作的目标是所选可用区中的实例。默认情况下，它的目标实例的 [标签](#) 名为 AZApplicationSlowdown，值为 LatencyForEC2。您可以用自己的标签替换默认标签，也可以将此标签添加到您的实例中。如果未找到有效实例，则将跳过此操作。在使用 SSM 在 EC2 上运行实验之前，您应该 [配置 AWS Systems Manager 代理](#)。

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。
- 您需要将选定可用区内以下 3 种类型中的任何一种中的一种或多种资源作为目标：EC2 实例、ECS 任务或 EKS 容器。

- 场景中的所有目标都必须位于同一 VPC 中。

Permissions

要运行此场景，您需要一个具有允许 FIS 代入角色的信任策略的 IAM 角色以及实验中目标资源类型的托管策略：EC2、ECS 和 EKS。当您从 AZ：应用程序减速场景中创建实验模板时，FIS 会使用信任策略和以下 AWS 托管策略为您创建角色：

- [AWSFaultInjectionSimulatorEC2访问](#)
- [AWSFaultInjectionSimulatorECSAccess](#)
- [AWSFaultInjectionSimulatorEKSAccess](#)

如果您使用现有 [IAM 角色](#) 运行“AZ：应用程序减速”场景，则可以附加以下策略来向 AWS FIS 授予必要的权限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribeTasks",
      "Effect": "Allow",
      "Action": "ecs:DescribeTasks",
      "Resource": "*"
    },
    {
      "Sid": "DescribeContainerInstances",
      "Effect": "Allow",
      "Action": "ecs:DescribeContainerInstances",
      "Resource": "arn:aws:ecs:*:*:container-instance/*/*"
    },
    {
      "Sid": "DescribeInstances",
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Sid": "DescribeSubnets",
      "Effect": "Allow",
      "Action": "ec2:DescribeSubnets",

```

```
    "Resource": "*"
  },
  {
    "Sid": "DescribeCluster",
    "Effect": "Allow",
    "Action": "eks:DescribeCluster",
    "Resource": "arn:aws:eks:*:*:cluster/*"
  },
  {
    "Sid": "TargetResolutionByTags",
    "Effect": "Allow",
    "Action": "tag:GetResources",
    "Resource": "*"
  },
  {
    "Sid": "SendCommand",
    "Effect": "Allow",
    "Action": [
      "ssm:SendCommand"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:instance/*",
      "arn:aws:ssm:*:*:managed-instance/*",
      "arn:aws:ssm:*:*:document/*"
    ]
  },
  {
    "Sid": "ListCommands",
    "Effect": "Allow",
    "Action": [
      "ssm:ListCommands"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CancelCommand",
    "Effect": "Allow",
    "Action": [
      "ssm:CancelCommand"
    ],
    "Resource": "*"
  }
]
```

```
}
```

场景内容

以下内容定义了场景。可以保存此 JSON，并使用 AWS [create-experiment-template](#) 命令行界面 (AWS CLI) Line CLI 中的命令创建 [实验模板](#)。要获取最新版本的场景，请访问 FIS 控制台中的场景库并转到“内容”选项卡。

```
{
  "tags": {
    "Name": "AZ: Application Slowdown"
  },
  "description": "Add latency between resources within a single AZ.",
  "actions": {
    "LatencyForEKS": {
      "actionId": "aws:eks:pod-network-latency",
      "parameters": {
        "delayMilliseconds": "200",
        "duration": "PT30M",
        "flowsPercent": "100",
        "interface": "DEFAULT",
        "kubernetesServiceAccount": "fis-service-account",
        "sources": "us-east-1a"
      },
      "targets": {
        "Pods": "TargetsForEKS"
      }
    },
    "LatencyForEC2": {
      "actionId": "aws:ssm:send-command",
      "parameters": {
        "duration": "PT30M",
        "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-Latency-Sources",
        "documentParameters": "{\"DelayMilliseconds\": \"200\", \"Sources\": \"us-east-1a\", \"Interface\": \"DEFAULT\", \"TrafficType\": \"egress\", \"DurationSeconds\": \"1800\", \"FlowsPercent\": \"100\", \"InstallDependencies\": \"True\"}"
      },
      "targets": {
        "Instances": "TargetsForEC2"
      }
    },
    "LatencyForECS": {
```

```
    "actionId": "aws:ecs:task-network-latency",
    "parameters": {
      "delayMilliseconds": "200",
      "duration": "PT30M",
      "flowsPercent": "100",
      "installDependencies": "true",
      "sources": "us-east-1a",
      "useEcsFaultInjectionEndpoints": "true"
    },
    "targets": {
      "Tasks": "TargetsForECS"
    },
    "startAfter": []
  }
},
"targets": {
  "TargetsForEKS": {
    "parameters": {
      "availabilityZoneIdentifier": "us-east-1a",
      "clusterIdentifier": "",
      "namespace": "default",
      "selectorType": "labelSelector",
      "selectorValue": "AZApplicationSlowdown=LatencyForEKS"
    },
    "resourceType": "aws:eks:pod",
    "selectionMode": "ALL"
  },
  "TargetsForEC2": {
    "filters": [
      {
        "path": "Placement.AvailabilityZone",
        "values": [
          "us-east-1a"
        ]
      }
    ],
    "resourceTags": {
      "AZApplicationSlowdown": "LatencyForEC2"
    },
    "resourceType": "aws:ec2:instance",
    "selectionMode": "ALL"
  },
  "TargetsForECS": {
    "filters": [
```

```
        {
          "path": "AvailabilityZone",
          "values": [
            "us-east-1a"
          ]
        }
      ],
      "resourceTags": {
        "AZApplicationSlowdown": "LatencyForECS"
      },
      "resourceType": "aws:ecs:task",
      "selectionMode": "ALL"
    }
  },
  "experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "skip"
  },
  "stopConditions": [
    {
      "source": "none"
    }
  ]
}
```

Cross-AZ: Traffic Slowdown

您可以使用跨可用区：流量减速场景来注入数据包丢失，以中断和减慢可用区之间的流量 (AZs)。数据包丢失会损害跨可用区的通信，即部分中断，有时被称为灰色故障。它会在目标资源之间的网络流量中注入数据包丢失。网络流量代表计算资源之间的流量，即在服务器、容器和服务之间传输请求、响应和其他通信的数据包。该场景可以帮助验证可观测性设置，调整警报阈值，发现跨可用区通信中的应用程序敏感性和依赖性，并练习关键运营决策，例如可用区撤离。

默认情况下，该场景会向选定可用区的目标资源的 100% 出站网络流量注入 15% 的数据包丢失，持续时间为 30 分钟。您可以使用 AWS FIS 控制台中的编辑共享参数对话框在场景级别调整以下参数，然后将其应用于基础操作：

- 可用区-您可以选择要损坏的可用区，数据包丢失将从该可用区注入该区域 AZs 内的另一个可用区。
- 丢包-将数据包丢失调整为较低以进行细微的中断测试，例如5%，或更高以测试严重的通信降级和恢复机制，例如50%甚至100%以测试总体连接影响。

- 流量百分比-减少以损害部分流量。例如，您可以注入 15% 的数据包丢失，影响 25% 的网络流，以进行更微妙的测试。
- 持续时间-设置实验的运行时长。您可以缩短时间以加快测试速度，也可以运行更长的持续测试。例如，将持续时间设置为 2 小时，以帮助测试受损条件下的恢复机制。
- 资源定向-您可以使用标签（用于 EC2 或 Fargate 上的 EC2 实例或 ECS 任务）或标签（适用于 EC2 上的 EKS 容器）来定义整个场景的目标资源。您可以指定自己的标签和标签，也可以使用场景中提供的默认值。如果您不想使用标签或标签，则可以通过指定其他参数来编辑目标资源的操作。
- 自定义-如果您不想以 EC2 或 ECS 资源为目标，则可以将操作保留为默认标签。实验找不到任何可以瞄准的资源，因此该操作将被跳过。但是，如果您不想以 EKS 资源为目标，则应将 EKS 操作和目标完全从场景中删除，因为它需要提供 EKS 集群标识符。要进行更精细的自定义，您可以直接修改实验模板中的单个操作。

操作

以下操作共同导致从目标可用区到该区域其他可用区的出站通信在网络层出现数据包丢失，从而帮助缓解可用区之间流量减慢的症状。这些操作并行运行，默认情况下，每次注入 15% 的数据包丢失，持续 30 分钟。在这段时间之后，沟通恢复正常。该场景需要在所选可用区中至少有一种资源类型才能运行：EC2 实例、ECS 任务或 EKS 容器。

ECS 网络数据包丢失

跨可用区：流量减速包括 [aws: ecs: , 用于task-network-packet-loss为 ECS](#) 任务注入丢包。该操作以所选可用区中的任务为目标，并会影响其与该区域 AZs 中所有其他任务的出站通信。您可以通过编辑操作和在 Sources 字段中添加或移 AZs 除来进一步自定义影响范围。默认情况下，它的目标任务的 [标签](#) 名 CrossAZTrafficSlowdown 为，值为 PacketLossForECS。您可以用自己的标签替换默认标签，也可以将场景标签添加到任务中。如果未找到有效任务，则将跳过此操作。在 ECS 上运行实验之前，应按照 [ECS 任务操作的设置步骤进行操作](#)。

EKS 网络数据包丢失

跨可用区：流量减速包括 [aws: eks: , 用于为 pod-network-packet-loss](#) EKS pod 注入丢包。该操作以所选可用区中的 Pod 为目标，并会影响它们与该区域 AZs 内所有其他容器的出站通信。您可以通过编辑操作和在 Sources 字段中添加或移 AZs 除来进一步自定义影响范围。默认情况下，它的目标是集群中带有格式为 key=value 的标签的 pod。提供的默认标签是 CrossAZTraffic=PacketLossForEKS。你可以用自己的标签替换默认标签，也可以将此标签添加到你的 pod 中。如果未找到有效的 pod，则将跳过此操作。在 EKS 上运行实验之前，应按照 [EKS pod 操作的设置步骤进行操作](#)。

EC2 网络数据包丢失

跨可用区：流量减速使用 a [ws: ssm: send-command 操作运行-Network-AWSFIS-Run Packet-Loss-Sources](#) 文档，为 EC2 实例注入数据包丢失，并损害它们与该地区所有其他实例的出站通信。AZs 您可以通过编辑操作和在 Sources 字段中添加或移 AZs 除来进一步自定义影响范围。该操作的目标是所选可用区中的实例。默认情况下，它的目标实例的标签名 CrossAZTrafficSlowdown 为，值为 PacketLossForEC2。您可以用自己的标签替换默认标签，也可以将此标签添加到您的实例中。如果未找到有效实例，则将跳过此操作。在使用 SSM 在 EC2 上运行实验之前，您应该 [配置 AWS Systems Manager 代理](#)。

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。
- 您需要将选定可用区内以下 3 种类型中的任何一种中的一种或多种资源作为目标：EC2 实例、ECS 任务或 EKS 容器。
- 场景中的所有目标都必须位于同一 VPC 中。

Permissions

要运行此场景，您需要一个具有允许 FIS 代入角色的信任策略的 IAM 角色以及实验中目标资源类型的托管策略：EC2、ECS 和 EKS。当您根据跨可用区：流量减速场景创建实验模板时，FIS 会使用信任策略和以下 AWS 托管策略为您创建角色：

- [AWSFaultInjectionSimulatorEC2访问](#)
- [AWSFaultInjectionSimulatorECSAccess](#)
- [AWSFaultInjectionSimulatorEKSAccess](#)

如果您使用现有 [IAM 角色](#) 运行跨可用区：流量减速场景，则可以附加以下策略向 AWS FIS 授予必要的权限：

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

```
{
  "Sid": "DescribeTasks",
  "Effect": "Allow",
  "Action": "ecs:DescribeTasks",
  "Resource": "*"
},
{
  "Sid": "DescribeContainerInstances",
  "Effect": "Allow",
  "Action": "ecs:DescribeContainerInstances",
  "Resource": "arn:aws:ecs:*:*:container-instance/*/*"
},
{
  "Sid": "DescribeInstances",
  "Effect": "Allow",
  "Action": "ec2:DescribeInstances",
  "Resource": "*"
},
{
  "Sid": "DescribeSubnets",
  "Effect": "Allow",
  "Action": "ec2:DescribeSubnets",
  "Resource": "*"
},
{
  "Sid": "DescribeCluster",
  "Effect": "Allow",
  "Action": "eks:DescribeCluster",
  "Resource": "arn:aws:eks:*:*:cluster/*"
},
{
  "Sid": "TargetResolutionByTags",
  "Effect": "Allow",
  "Action": "tag:GetResources",
  "Resource": "*"
},
{
  "Sid": "SendCommand",
  "Effect": "Allow",
  "Action": [
    "ssm:SendCommand"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:instance/*",
```

```

        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/*"
    ]
},
{
    "Sid": "ListCommands",
    "Effect": "Allow",
    "Action": [
        "ssm:ListCommands"
    ],
    "Resource": "*"
},
{
    "Sid": "CancelCommand",
    "Effect": "Allow",
    "Action": [
        "ssm:CancelCommand"
    ],
    "Resource": "*"
}
]
}

```

场景内容

以下内容定义了场景。可以保存此 JSON，并使用 AWS [create-experiment-template](#) 命令行界面 (AWS CLI) Line CLI 中的命令创建 [实验模板](#)。要获取最新版本的场景，请访问 FIS 控制台中的场景库并转到“内容”选项卡。

```

{
  "tags": {
    "Name": "Cross-AZ: Traffic Slowdown"
  },
  "description": "Inject packet loss to disrupt and slow down traffic between AZs.",
  "actions": {
    "PacketLossForEC2": {
      "actionId": "aws:ssm:send-command",
      "parameters": {
        "duration": "PT30M",
        "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Packet-Loss-Sources",
        "documentParameters": "{\"Sources\": \"us-east-1b,us-east-1c,us-
east-1d,us-east-1e,us-east-1f\", \"LossPercent\": \"15\", \"Interface\": \"DEFAULT\"",

```

```

\"TrafficType\": \"egress\", \"DurationSeconds\": \"1800\", \"FlowsPercent\": \"100\",
\"InstallDependencies\": \"True\"}
    },
    \"targets\": {
      \"Instances\": \"TargetsForEC2\"
    }
  },
  \"PacketLossForECS\": {
    \"actionId\": \"aws:ecs:task-network-packet-loss\",
    \"parameters\": {
      \"sources\": \"us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f\",
      \"lossPercent\": \"15\",
      \"duration\": \"PT30M\",
      \"flowsPercent\": \"100\",
      \"installDependencies\": \"true\",
      \"useEcsFaultInjectionEndpoints\": \"true\"
    },
    \"targets\": {
      \"Tasks\": \"TargetsForECS\"
    }
  },
  \"PacketLossForEKS\": {
    \"actionId\": \"aws:eks:pod-network-packet-loss\",
    \"parameters\": {
      \"sources\": \"us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f\",
      \"lossPercent\": \"15\",
      \"duration\": \"PT30M\",
      \"flowsPercent\": \"100\",
      \"interface\": \"DEFAULT\",
      \"kubernetesServiceAccount\": \"fis-service-account\"
    },
    \"targets\": {
      \"Pods\": \"TargetsForEKS\"
    }
  }
},
\"targets\": {
  \"TargetsForEC2\": {
    \"filters\": [
      {
        \"path\": \"Placement.AvailabilityZone\",
        \"values\": [
          \"us-east-1a\"
        ]
      }
    ]
  }
}

```

```
    }
  ],
  "resourceTags": {
    "CrossAZTrafficSlowdown": "PacketLossForEC2"
  },
  "resourceType": "aws:ec2:instance",
  "selectionMode": "ALL"
},
"TargetsForECS": {
  "filters": [
    {
      "path": "AvailabilityZone",
      "values": [
        "us-east-1a"
      ]
    }
  ],
  "resourceTags": {
    "CrossAZTrafficSlowdown": "PacketLossForECS"
  },
  "resourceType": "aws:ecs:task",
  "selectionMode": "ALL"
},
"TargetsForEKS": {
  "parameters": {
    "availabilityZoneIdentifier": "us-east-1a",
    "clusterIdentifier": "",
    "namespace": "default",
    "selectorType": "labelSelector",
    "selectorValue": "CrossAZTrafficSlowdown=PacketLossForEKS"
  },
  "resourceType": "aws:eks:pod",
  "selectionMode": "ALL"
}
},
"experimentOptions": {
  "accountTargeting": "single-account",
  "emptyTargetResolutionMode": "skip"
},
"stopConditions": [
  {
    "source": "none"
  }
]
]
```

```
}
```

Cross-Region: Connectivity

您可以使用该 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 多区域全局表的跨区域复制。“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 全局表的跨区域复制。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 VPCs 实验区域到 VPCs 通过公交网关连接的目标区域的跨区域网络流量。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 VPCs 实验区域到目标区域公有 AWS IP 区块的跨区域网络流量。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。

中断 VPC 终端节点连接

Cross Region: Connectivity包括 [aws: network : disrupt-vpc-endpoint](#) 断与目标 VPC 终端节点关联的服务的连接。例如，如果 VPC 终端节点创建了指向 `com.amazonaws.us-east-1.ec2` 的私有链接，则与该服务的连接将中断。

此操作的目标是实验区域中的 VPC 终端节点。默认情况下，它以接口 VPC 终端节点为目标，DisruptVpcEndpoint 其[标签](#)名为值Allowed。您可以将此标签添加到您的 VPC 终端节点，也可以将默认标签替换为实验模板中您自己的标签。默认情况下，如果未找到有效的 VPC 终端节点，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 S3 从实验区域向目标存储桶的目标区域的复制。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-replication](#) 用于暂停实验区域与所有其他区域 (包括目标区域) 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，复制直至所有数据保持同步所需的时间将因实验持续时间和表的更改速率而异。

此操作同时针对实验区域中的 DynamoDB 多区域以及最终一致的全局表。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。

暂停 MemoryDB 多区域复制

Cross-Region: Connectivity包括 [aws: memorydb: multi-region-cluster-pause-replication](#) ，用于暂停从实验区域中的区域成员集群向目标多区域集群中其余集群的复制。其他区域成员集群之间的复制将不受影响。场景结束后，复制将从暂停点开始恢复。请注意，在成员集群之间同步数据的复制时间将因实验持续时间和写入集群的数据速率而异。

此操作的目标是 MemoryDB 多区域集群，其区域成员位于实验区域。默认情况下，它以多区域集群为目标，其[标签](#)名 DisruptMemoryDB 为，值为 Allowed。您可以将此标签添加到您的多区域集群中，也可以将默认标签替换为实验模板中您自己的标签。默认情况下，如果找不到有效的集群，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。

Permissions

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",
      "ec2:ModifyManagedPrefixList"
    ],
    "Resource": "arn:aws:ec2:*:*:prefix-list/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "EC2DescribeResources",
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeVpcs",
      "ec2:DescribeVpcPeeringConnections",
      "ec2:DescribeManagedPrefixLists",
      "ec2:DescribeSubnets",
      "ec2:DescribeRouteTables",
      "ec2:DescribeVpcEndpoints",
      "ec2:DescribeTransitGatewayPeeringAttachments",
      "ec2:DescribeTransitGatewayAttachments",
      "ec2:DescribeTransitGateways",
      "ec2:DescribeSecurityGroups"
    ]
  }
}
```

```
    ],
    "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": "TransitGatewayDisruptConnectivity1",
  "Effect": "Allow",
  "Action": [
    "ec2:DisassociateTransitGatewayRouteTable",
    "ec2:AssociateTransitGatewayRouteTable"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:transit-gateway-route-table/*",
    "arn:aws:ec2:*:*:transit-gateway-attachment/*"
  ]
},
{
  "Sid": "S3CrossRegion1",
  "Effect": "Allow",
  "Action": "s3:ListAllMyBuckets",
  "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": "DynamoDbPauseReplication",
    "Effect": "Allow",
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:PutResourcePolicy",
        "dynamodb:GetResourcePolicy",
        "dynamodb>DeleteResourcePolicy"
    ],
    "Resource": [
        "arn:aws:dynamodb:*:*:table/*"
    ]
},
{
    "Sid": "DynamoDbMrscPauseReplication",
    "Effect": "Allow",
    "Action": [
        "dynamodb:InjectError"
    ],
    "Resource": ["*"]
},
{
    "Sid": "ResolveResourcesViaTags",
    "Effect": "Allow",
    "Action": "tag:GetResources",
    "Resource": "*"
},
{
    "Sid": "MemDbCrossRegion",
    "Effect": "Allow",
    "Action": [
        "memorydb:DescribeMultiRegionClusters",
        "memorydb:PauseMultiRegionClusterReplication"
    ],

```

```

    "Resource": [
      "arn:aws:memorydb::*:multiregioncluster/*"
    ]
  },
  {
    "Sid": "DisruptVPCE1",
    "Effect": "Allow",
    "Action": "ec2:CreateSecurityGroup",
    "Resource": [
      "arn:aws:ec2::*:vpc/*",
      "arn:aws:ec2::*:security-group/*"
    ]
  },
  {
    "Sid": "DisruptVPCE2",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2::*:security-group/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateSecurityGroup",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "DisruptVPCE3",
    "Effect": "Allow",
    "Action": [
      "ec2>DeleteSecurityGroup",
      "ec2:RevokeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2::*:security-group/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "DisruptVPCE4",
    "Effect": "Allow",
    "Action": "vpce:AllowMultiRegion",
    "Resource": "arn:aws:ec2::*:vpc-endpoint/*"
  }
}

```

```

    },
    {
      "Sid": "ModifyVPCE",
      "Effect": "Allow",
      "Action": "ec2:ModifyVpcEndpoint",
      "Resource": [
        "arn:aws:ec2:*:*:vpc-endpoint/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    }
  ]
}

```

场景内容

以下内容定义了场景。可以保存此 JSON，并使用 AWS [create-experiment-template](#) 命令行界面 (AWS CLI) Line CLI 中的命令创建 [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": {}
    },
    "VPC-Endpoint": {
      "resourceType": "aws:ec2:vpc-endpoint",
      "resourceTags": {
        "DisruptPrivateLink": "Allowed"
      },
      "selectionMode": "ALL"
    },
    "S3-Bucket": {

```

```

        "resourceType": "aws:s3:bucket",
        "resourceTags": {
            "S3Impact": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "DynamoDB-Global-Table": {
        "resourceType": "aws:dynamodb:global-table",
        "resourceTags": {
            "DisruptDynamoDb": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "MemoryDB-Multi-Region-Cluster": {
        "resourceType": "aws:memorydb:multi-region-cluster",
        "resourceTags": {
            "DisruptMemoryDb": "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"
        }
    }
},

```

```
    "Disrupt-Vpc-Endpoint": {
      "actionId": "aws:network:disrupt-vpc-endpoint",
      "parameters": {
        "duration": "PT3H"
      },
      "targets": {
        "VPCEndpoints": "VPC-Endpoint"
      }
    },
    "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:global-table-pause-replication",
      "parameters": {
        "duration": "PT3H"
      },
      "targets": {
        "Tables": "DynamoDB-Global-Table"
      }
    },
    "Pause-MemoryDB-Multi-Region-Cluster-Replication": {
      "actionId": "aws:memorydb:multi-region-cluster-pause-replication",
      "parameters": {
        "duration": "PT3H",
        "region": "eu-west-1"
      },
      "targets": {
        "MultiRegionClusters": "MemoryDB-Multi-Region-Cluster"
      }
    }
  },
  "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"
}
```

使用多账户实验 AWS FIS

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。

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 管理控制台和 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](#)
- [多账户实验最佳实践](#)
- [多账户实验的先决条件](#)
- [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](#)

多账户实验的概念

以下是多账户实验的主要概念：

- Orchestrator 帐户-协调器帐户充当中央帐户，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-目标账户是指其资源可能受到 AWS FIS 多账户实验影响的个人 AWS 账户。

- **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。如果您想按资源定位资源 ARNs，则每个操作的单账户限制相同。
- 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。要了解如何在您的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 IDs](#)。

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 \(可选 \)](#)
- [多账户实验的安全杆 \(可选 \)](#)

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[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。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:PauseVolumeIO"
      ],
      "Resource": "arn:aws:ec2:us-east-1:123456789012: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。

JSON

```

{
  "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::111122223333:role/role_name"
        }
      }
    }
  ]
}

```

- 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 ARNs 向该策略添加其他角色。每个目标账户只能有一个角色 ARN。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::111122223333: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。

JSON

```
{
  "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

多账户实验的安全杆 (可选)

安全杆用于停止所有正在进行的实验，并防止开始新实验。您可能需要在特定时间段内，或者为了响应应用程序运行状况警报而使用安全杆阻止进行 FIS 实验。每个 AWS 账户都有一个安全杆 AWS 区

域。当激活安全杆时，它会影响与安全杆在同一账户和区域内运行的所有实验。为了停止和阻止多账户实验，必须在运行实验的同一账户和区域内激活安全杆。

创建多账户实验模板

要学习如何创建实验模板，请访问 AWS 管理控制台

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. 要使用已保存的 JSON 文件创建将账户定位实验选项设置为 "multi-account" (例如 my-template.json) 的实验，请将中的 *italics* 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。将中的 *italics* 占位符值替换为您自己的值，使用步骤 2 中的值作为 --experiment-template-id 参数值，然后运行以下命令。id--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，您可以运行 [list-target-account-configurations](#) 命令来查看目标账户配置是否已创建。

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

您还可以通过运行 `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 EXTxxxxxxxxxx
```

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 管理控制台

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. 在侧面板中，选择步骤 3，配置服务访问权限。
5. 修改目标账户配置，然后选择更新实验模板。
6. 选择步骤 5，查看并创建。

使用 CLI 更新目标账户配置

运行 `update-target-account-configuration` 命令到命令，将中的 *italics* 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 EXTxxxxxxxxxx  
--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 管理控制台

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. 在侧面板中，选择步骤 3，配置服务访问权限。
5. 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。
6. 选择步骤 5，查看并创建。
7. 查看模板并选择更新实验模板。当系统提示您确认时，输入update并选择更新实验模板。

使用 CLI 删除目标账户配置

运行[delete-target-account-configuration](#)命令，将中的*italics*占位符值替换为您自己的值。

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

调度实验

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

- [创建调度器角色](#)
- [创建实验计划](#)
- [更新实验计划](#)
- [禁用或删除实验计划](#)

创建调度器角色

执行角色是一个 IAM 角色，该角色 AWS FIS 是为了与 EventBridge 调度器交互以及让事件桥调度器启动 FIS 实验。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 日程安排器代表您担任该角色。

JSON

```
{  
  "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": "AROAZL22PDN5A6WKRQNU",  
    "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。以下策略允许 S EventBridge scheduler 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 `*`。

JSON

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

```
{
  "Policy": {
    "PolicyName": "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. 您可以将调度程序限制为仅运行具有特定标签值的 AWS FIS 实验模板。例如，以下策略授予所有 AWS FIS 实验的 `fis:StartExperiment` 权限，但将调度程序限制为只能运行带有标签的实验模板。 `Purpose=Schedule`

JSON

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

创建实验计划

您需要先为此安排调用一个或多个 [实验模板组件](#)，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。

The screenshot shows the AWS FIS 'Schedule pattern' configuration page. It includes the following sections:

- Schedule pattern**
 - Occurrence**: You can define an one-time or recurrent schedule. One-time schedule, Recurring schedule.
 - Date and time**: The date and time to invoke the target. Fields for YYYY/MM/DD, hh:mm, and a Timezone dropdown (UTC -04:00 America/New...).
 - YYYY/MM/DD: Use 24-hour format timestamp (hh:mm)
 - Timezone
 - Flexible time window**: If you choose a flexible time window, Scheduler invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time. A dropdown menu labeled 'Select' is present.
- Schedule state**
 - Enable schedule**: You can choose not to enable the schedule now. You will be able to enable the schedule after it has been created. Enable.

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。

禁用或删除实验计划

要阻止实验按计划执行或运行，可以删除或禁用此规则。以下步骤将引导您完成如何使用 AWS 控制台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。

安全杆适用于 AWS FIS

安全杆用于停止所有正在进行的实验，并防止开始新实验。您可能需要在特定时间段内，或者为了响应应用程序运行状况警报而使用安全杆阻止进行 FIS 实验。每个 AWS 账户都有一个安全杆 AWS 区域。

对于安全杆停止的正在进行的实验，您只需为实验停止之前运行的操作持续时间付费。已阻止开始的实验不会产生任何费用。以下各节提供了有关如何开始使用安全杆的信息。

主题

- [安全杆的概念](#)
- [使用安全杆](#)

安全杆的概念

可以激活或解除安全杆。

- 如果解除，则允许 FIS 实验。默认情况下，安全杆处于解除状态。
- 如果激活，则正在进行的实验将停止，并且不允许开始新实验。

受安全杆影响的实验将以下面的其中一种状态结束：

- 已停止，如果安全杆激活时实验正在运行。
- 已取消，如果实验是在安全杆已激活时开始的。

您无法恢复或重新运行已停止或取消的实验。但是，安全杆解除后，可以使用同一个实验模板开始新实验。

安全杆资源

安全杆是 Amazon 资源名称 (ARN) 定义的一项资源。安全杆包括以下参数：

- 状态，要么是激活，要么是解除。
- 原因，这是用户输入的字符串，用于记录安全杆状态的更改原因。

使用安全杆

本节将详细介绍如何使用 AWS FIS 控制台或命令行查看、接合和断开安全杆。

查看安全杆

可以按照以下步骤查看账户和区域的安全杆状态。

使用控制台查看安全杆

1. [打开控制 AWS FIS 台](#)
2. 在导航窗格中，选择实验。
3. 如果安全杆已激活，您将在页面顶部看到一个警报横幅。如果没有警报横幅，表明安全杆处于解除状态。

使用 CLI 查看安全杆

- 使用以下命令：

```
aws fis get-safety-lever --id "default"
```

安全杆可处于以下几种状态之一：

- 已解除：安全杆不会影响任何实验。实验可以自由运行。默认情况下，安全杆处于解除状态。
- 正在激活：安全杆正在从解除变为激活状态。可能还有一些实验尚未停止。处于这种状态时，无法更改安全杆。
- 已激活：安全杆处于活动状态，未运行任何实验。在安全杆处于激活状态时，任何试图开始的新实验都将被取消。

激活安全杆

使用控制台激活安全杆

1. [打开控制 AWS FIS 台](#)
2. 在导航窗格中，选择实验。
3. 选择停止所有实验按钮。

4. 输入激活安全杆的原因。
5. 选择确认。

使用 CLI 激活安全杆

- 使用以下命令。在原因字段中填写您自己的回复。

```
aws fis update-safety-lever-state --id "default" --state  
"status=engaged,reason=xxxxx"
```

解除安全杆

使用控制台解除安全杆

1. [打开控制 AWS FIS 台](#)
2. 在导航窗格中，选择实验。
3. 选择解除安全杆按钮。
4. 输入解除安全杆的原因。
5. 选择确认。

使用 CLI 解除安全杆

- 使用以下命令：

```
aws fis update-safety-lever-state --id "default" --state  
"status=disengaged,reason=recovered"
```

监测 AWS FIS 实验

您可以使用以下工具来监控 AWS 故障注入服务 (AWS FIS) 实验的进度和影响。

AWS FIS 控制台和 AWS CLI

使用 AWS FIS 控制台或监控正在运行的实验的进度。AWS CLI 您可以查看实验中各项操作的状态及结果。有关更多信息，请参阅 [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。AWS FIS 使用率指标对应于 AWS 服务配额。您可以配置警报，以在用量接近服务配额时向您发出警报。有关更多信息，请参阅 [监视器使用 CloudWatch](#)。

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，请参阅 [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

AWS FIS 实验记录

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。AWS 当实验状态发生变化时，FIS 会发出通知。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，请参阅 [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 APIs 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 控制面板通知

AWS Health 可让您持续了解您的资源性能以及您的 AWS 服务和账户的可用性。当你开始实验时，AWS FIS 会向你的 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 AWS Amazon 监控 FIS 使用量指标 CloudWatch

您可以使用 Amazon CloudWatch 来监控 AWS FIS 实验对目标的影响。您还可以监控 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，请确定可用于确定实验目标资源类型的基线或“稳定状态”的 CloudWatch 指标。开始实验后，您可以监控通过实验模板选择的目标的这些 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](#)

AWS FIS 使用情况指标

您可以使用 CloudWatch 使用量指标来了解您的账户的资源使用情况。使用这些指标在 CloudWatch 图表和仪表板上可视化您当前的服务使用情况。

AWS FIS 使用率指标对应于 AWS 服务配额。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。

以下维度用于完善 AWS FIS 发布的使用量指标。

维度	描述
Service	包含资源的 AWS 服务的名称。对 AWS 于 FIS 使用情况指标，此维度的值为 FIS。
Type	正在报告的实体的类型。当前，AWS FIS 使用率指标的唯一有效值是 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

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 权限才能将日志发送到您配置的每个日志目标。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，请参阅 [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
  }
}

```

发行说明

- 版本 2 引入了：
 - target_type 字段和将 resolved_targets 字段从对象列表更改 ARNs 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 支持将日志传送到以下目的地：

- 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

日志将传送到日志流 named `/aws/fis/experiment-id`。

日志会在一分钟以内传输到日志组。

日志记录示例

以下是在随机选择的 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. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。
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。

要启用实验记录，请使用 AWS CLI

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. 打开 AWS FIS 控制台，网址<https://console.aws.amazon.com/fis/>为。

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。

要禁用实验记录，请使用 AWS CLI

使用[update-experiment-template](#)命令并指定空日志配置。

使用记录 API 调用 AWS CloudTrail

AWS 故障注入服务 (AWS FIS) 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。捕获的调用包括来自 FIS 控制台的调用和对 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。您可以在中查看、搜索和下载最近发生的事件 AWS 账户。有关更多信息，请参阅[使用事件历史记录查看 CloudTrail 事件](#)。

要持续记录您的 AWS 账户事件（包括 AWS FIS 的事件），请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，当您在控制台中创建跟踪时，该跟踪将应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，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，有关在亚马逊 EC2 实例上执行的操作，请参阅[亚马逊 EC2 API 参考](#)。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息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。
- 请求是否由其他 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:jd0e",
    "arn": "arn:aws:sts::111122223333:assumed-role/example/jd0e",
    "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 AWS userIdentity 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。实验的 ID EXP21n T17 WMz a6dNugz 出现在假设角色的 ARN 中并principalId作为其一部分出现。

```
{
  "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"  
}
```

故障排除 AWS FIS

要对错误进行故障排除，请 AWS FIS 返回 GetExperiment API 和 FIS 实验日志中的详细错误。当实验状态为失败时，将作为实验状态的一部分返回错误。当多个操作失败时，将返回第一个失败操作作为实验错误。可以查看 FIS 实验日志以了解任何其他错误。要了解如何记录和监控 AWS FIS 实验，请参阅[监测 AWS FIS 实验](#)。

根据故障类型，您可能会收到以下错误之一：

- **原因**：具体故障的详细描述。不应使用原因值来自动排查错误，因为它们可能会发生变化。
- **代码**：故障的类型。除非下表中另有说明，否则不应使用代码值来自动排查错误，因为它们可能会发生变化。
- **位置**：实验模板中故障部分的上下文，例如操作或目标。
- **账户 ID**：发生故障的 AWS 账户。

错误代码

错误代码	代码描述
ConfigurationFailure	操作、目标、实验或日志配置不正确。请检查错误 location 并确保参数和配置正确。
DependentServiceFailure	另一个 AWS 服务出现故障。请尝试再次运行实验。
InternalFailure	运行实验时发生内部错误。可以根据此错误代码自动排查错误。
InvalidTarget	在目标解析期间或操作开始时，无法解析目标。这可能是由以下原因之一导致的： <ul style="list-style-type: none">• 目标不存在，例如目标已被删除或 ARN 不正确。• 您的目标有一个不解析任何资源的标签。• 有一个操作未链接到目标。

错误代码	代码描述
AuthorizationFailure	<p>要排查此错误，请查看日志以确定哪些目标未得到解析。请检查是否所有操作均已链接到目标，以及您的资源 ID 或标签已存在且拼写无误。</p> <p>由于权限错误导致实验失败的主要原因有两个：</p> <ul style="list-style-type: none">• 您作为目标的 IAM 角色没有适当的权限来解析目标或对您的资源采取操作。要修复此错误，请在 FIS 操作参考 中查看操作所需的权限，并将其添加到您的实验 IAM 角色中。• 您的组织的 服务控制策略 (SCP) 拒绝为 FIS 创建 AWS 服务相关角色 (SLR)。FIS 使用 SLR 管理实验的监控和资源选择。有关更多信息，请参阅 FIS 的服务相关角色权限 AWS。
QuotaExceededFailure	<p>已超过该资源类型的配额。要确定是否可以增加配额，请参阅 AWS 故障注入服务的配额和限制。</p>

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](#)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[按合规计划划分的范围内的合规性计划范围](#)划分)。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 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结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。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。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的[使用跟 CloudTrail 踪](#)。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务 (例如 Amazon Macie)，它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-3 版》<https://aws.amazon.com/compliance/fips/>。

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

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](#)
- [AWS AWS 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。当您使用更多 AWS 的 FIS 功能来完成工作时，您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 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

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

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

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

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

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

IAM 用户和群组

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

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

IAM 角色

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

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

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概述](#)。

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

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

基于身份的策略

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

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管策略与内联策略之间进行选择](#)。

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。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中[指定主体](#)。

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。

其他策略类型

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

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCPs)-在中指定组织或组织的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[资源控制策略 \(RCPs\)](#)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

多个策略类型

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

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
ACLs	否
ABAC (策略中的标签)	是
临时凭证	是
主体权限	是
服务角色	是
服务关联角色	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[使用客户管理型策略定义自定义 IAM 权限](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

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。

要启用跨账户访问，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。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

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。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。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

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

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 ARNs 用逗号分隔。

```
"Resource": [
  "resource1",
  "resource2"
]
```

要查看 AWS FIS 资源类型及其列表 ARNs，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。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用[条件运算符](#)（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

要查看 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](#)

ACLs 在 AWS FIS

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。ACLs 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

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

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

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

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

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

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的临时安全凭证](#)和[使用 IAM 的 AWS 服务](#)

FIS 的跨服务主体 AWS 权限

支持转发访问会话 (FAS) : 是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。

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

支持服务关联角色 : 是

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

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。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

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

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

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

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列出和查看有关您 AWS 账户的 FIS 资源的详细信息。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 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。

JSON

```
{
  "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。

JSON

```
{
  "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。

JSON

```
{
  "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::111122223333: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](#)。

JSON

```
{
  "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。

JSON

```

{
    "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。

JSON

```

{
  "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:ListGroupsWithUser",
      "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。该策略允许 AWS FIS 采取行动 `aws:ec2:api-insufficient-instance-capacity-error` 和 `aws:ec2:asg-insufficient-instance-capacity-error`。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",

```

```

    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "ec2:FisActionId": [
          "aws:ec2:api-insufficient-instance-capacity-error",
          "aws:ec2:asg-insufficient-instance-capacity-error"
        ]
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "ForAllValues:ArnLike": {
          "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` 条件密钥允许 `PutReplicationConfiguration` 且 `GetReplicationConfiguration` 仅当 FIS 在 AWS FIS 操作的上下文中使用时。AWS `aws:s3:bucket-pause-replication`

JSON

```

{
  "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他 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 `AWSServiceRoleForFIS` 为 FIS 的服务关联角色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 `AmazonFISServiceRolePolicy`。该策略使 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 托管政策参考》`AmazonFISServiceRolePolicy` 中的 [Amazon](#)。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。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。

JSON

```
{
  "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[服务关联角色权限](#)。

为 FIS 创建服务相关角色 AWS

您无需手动创建服务关联角色。当您在 AWS 管理控制台、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。

如果您删除该服务关联角色，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。创建服务关联角色后，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[编辑服务关联角色](#)。

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关联角色的功能或服务，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 AWS FIS 使用的 FIS 资源 AWSService RoleFor

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 手动删除服务关联角色

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[删除服务关联角色](#)。

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 端点和配额](#)。

AWS AWS 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使用案例的[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 托管策略：Amazon FISService RolePolicy

此策略附加到名为 `AWSServiceRoleForFIS` 的服务相关角色，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：AWSFaultInjectionSimulatorEC2访问权限

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 托管策略参考中的 [AWSFaultInjectionSimulatorEC2访问](#) 权限。

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 EC2 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 托管式策略参考》中的 [AWSFaultInjectionSimulatorECSAccess](#)。

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 托管式策略参考》中的 [AWSFaultInjectionSimulatorEKSAccess](#)。

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 托管式策略参考》中的 [AWSFaultInjectionSimulatorNetworkAccess](#)。

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 托管式策略参考》中的 [AWSFaultInjectionSimulatorRDSAccess](#)。

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 托管式策略参考》中的 [AWSFaultInjectionSimulatorSSMAccess](#)。

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
AWSFaultInjectionSimulatorEC2访问权限-更新现有策略	添加了“AZ：应用程序减速”和“跨可用区：流量减速”场景所需的权限。权限是：ec2：DescribeSubnets	2025 年 11 月 12 日

更改	描述	日期
AWSFaultInjectionSimulatorECSAccess : 对现有策略的更新	添加了“可用区：应用程序减速”和“跨可用区：流量减速”场景所需的权限。权限为：ecs: DescribeContainerInstances、ec2: DescribeSubnets 和 ec2: DescribeInstances	2025 年 11 月 12 日
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 : 对现有策略的更新	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
AWSFaultInjectionSimulatorEC2访问权限-更新现有策略	增加了允许 AWS FIS 解析 EC2 实例的权限。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorEKSAccess : 对现有策略的更新	增加了允许 AWS FIS 解析 EKS 目标的权限。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorRDSAccess : 对现有策略的更新	增加了允许 AWS FIS 解析 RDS 目标的权限。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorEC2访问权限-更新现有策略	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 : 对现有策略的更新	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
AWSFaultInjectionSimulatorECSAccess : 对现有策略的更新	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 : 对现有策略的更新	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

更改	描述	日期
AWSFaultInjectionSimulatorEC2访问权限-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
AWSFaultInjectionSimulatorSMSAccess ：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 FISService RolePolicy -更新现有政策	增加了允许 AWS FIS 描述子网的权限。	2022 年 10 月 26 日
亚马逊 FISService RolePolicy -更新现有政策	增加了允许 AWS FIS 描述 EKS 集群的权限。	2022 年 7 月 7 日
亚马逊 FISService RolePolicy -更新现有政策	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 FISService RolePolicy -更新现有政策	为 events:DescribeRule 操作删除 events:ManagedBy 条件。	2022 年 1 月 6 日

更改	描述	日期
亚马逊 FISService RolePolicy -更新现有政策	增加了允许 AWS FIS 检索停止条件下使用的 CloudWatch 警报历史记录的权利。	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 security Pillar Well-Architected Framework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用通过网络访问 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。

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 APIs 无需互联网网关、NAT 设备、VPN 连接或 Direct Connect 连接即可私密访问 AWS FIS。您的 VPC 中的实例不需要公有 IP 地址即可与 AWS FIS APIs 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 直通访问](#)。

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) 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" ]
      }
    }
  ]
}
```

标记您的 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：

- 识别和整理您的 AWS 资源。许多 AWS 服务都支持标记，因此您可以为来自不同服务的资源分配相同的标签，以表明这些资源是相关的。
- 控制对 AWS 资源的访问权限。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 AWS 于 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 故障注入服务 (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](#)

您可以使用以下 AWS CLI 命令来处理动作、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，每个配额是区域特定的。您可以申请增加下表中标记为可调整的配额。

要查看您账户中 AWS FIS 的配额，请打开 [Service Quotas 控制台](#)。在导航窗格中，依次选择 AWS 服务和 AWS Fault Injection Service。系统会立即应用不超过自动批准的配额的值。下表的描述栏中概述了自动批准的配额。如果您需要的配额超过自动批准的限额，请提交申请。超过自动批准限额的值将由客户支持部门审核，并在可能的情况下予以批准。

要请求提高配额，请参阅《服务配额用户指南》中的[请求提高配额](#)。

您 AWS 账户 有以下与 AWS FIS 相关的配额。

Name	默认值	可调整	说明
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。
aws: dynamodb: action 的每个多区域强一致 (MRSC) 全局表的操作分钟数 global-table-pause-replication	每个支持的区域：5,040	<u>是</u>	aws: dynamodb: 在 7 天滚动窗口内 global-table-pause-replication 可以设定的每个 MRSC 全局表的最大累积操作分钟数。
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

Name	默认值	可调整	说明
			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。
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:-region-connectivity 中托管前缀列表的最 route-table-disrupt-cross	每个受支持的区域 : 15 个	否	aws: route-table-disrupt-cross network:-region-connectivity 每次操作允许的最大托管前缀列表数量。
aws: network:-region-connectivity 中的最大路由表数量 route-table-disrupt-cross	每个受支持的区域 : 10 个	否	aws: network: route-table-disrupt-cross-region-connectivity 每次操作允许的最大路由表数量。
aws: network:-region-conectivity 中的最大路由数 route-table-disrupt-cross	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。

Name	默认值	可调整	说明
将 Aurora DSQL 集群定位为 aws: dsq: action。cluster-connection-failure	每个受支持的区域：100 个	<u>是</u>	每个实验使用标签识别目标时，aws: dsq: cluster-connection-failure 可以定位的最大 Aurora DSQL 集群数量。
将 Auto Scaling 群组设为 aws: ec2:-error asg-insufficient-instance-capacity	每个受支持的区域：500 个	<u>是</u>	在每个实验中，当你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 组数量。
aws 的目标存储桶:s3: bucket-pause-replication	每个受支持的区域：25 个	<u>是</u>	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 的目标集群：drain-container-instances	每个受支持的区域：100 个	<u>是</u>	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 的目标集群：failover-db-cluster	每个受支持的区域：160 个	<u>是</u>	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: DBInstances rds 的目标：reboot-db-instances	每个支持的区域：130	<u>是</u>	每个实验使用标签识别目标时 DBInstances ，aws: rds: reboot-db-instances 可以瞄准的最大数量。

Name	默认值	可调整	说明
aws:ec2:reboot-instances 的目标实例	每个受支持的区域：600 个	是	每个实验中，当您使用标签标识目标时 aws:ec2:reboot-instances 可以确定为目标的实例最大数量。
aws:ec2:stop-instances 的目标实例	每个受支持的区域：400 个	是	每个实验中，当您使用标签标识目标时 aws:ec2:stop-instances 可以确定为目标的实例最大数量。
aws:ec2:terminate-instances 的目标实例	每个受支持的区域：300 个	是	每个实验中，当您使用标签标识目标时 aws:ec2:terminate-instances 可以确定为目标的实例最大数量。
aws:ssm:send-command 的目标实例	每个受支持的区域：200 个	是	每个实验中，当您使用标签标识目标时 aws:ssm:send-command 可以确定为目标的实例最大数量。
aws:kinesis:action 的目标 Kinesis 数据流。stream-expired-iterator-exception	每个支持的区域：280	是	在每个实验中，当您使用标签识别目标时，aws:kinesis:stream-expired-iterator-exception 可以瞄准的最大 Kinesis 数据流数量。
aws:kinesis:action 的目标 Kinesis 数据流。stream-provisioned-throughput-exception	每个支持的区域：280	是	在每个实验中，当您使用标签识别目标时，aws:kinesis:stream-provisioned-throughput-exception 可以瞄准的最大 Kinesis 数据流数量。

Name	默认值	可调整	说明
aw ManagedResources s: arc: start-zonal-autoshift 动作的目标。	每个受支持的区域：200 个	<u>是</u>	每次实验使用标签识别目标时，aws: arc: start-zonal-autoshift 可以定位的最大托管资源数量。
aws: eks 的目标节点组：terminate-nodegroup-instances	每个受支持的区域：100 个	<u>是</u>	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 的目标 Pod：pod-cpu-stress	每个受支持的区域：1,000 个	<u>是</u>	每个实验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：1,000 个	<u>是</u>	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 Pod 数。
aws: eks 的目标 Pod：pod-io-stress	每个受支持的区域：1,000 个	<u>是</u>	每个实验使用参数识别目标时 aws: eks: pod-io-stress 可以瞄准的最大 Pod 数量。
aws: eks 的目标 Pod：pod-memory-stress	每个受支持的区域：1,000 个	<u>是</u>	每个实验使用参数识别目标时 aws: eks: pod-memory-stress 可以瞄准的最大 Pod 数量。

Name	默认值	可调整	说明
aws: eks 的目标 Pod : pod-network-blackhole-port	每个受支持的区域 : 1,000 个	<u>是</u>	每个实验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 的目标 Pod : pod-network-latency	每个受支持的区域 : 1,000 个	<u>是</u>	每个实验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 的目标 Pod : pod-network-packet-loss	每个受支持的区域 : 1,000 个	<u>是</u>	每个实验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 elasticache 的目标 : -计划弃用 interrupt-cluster-az-power	每个受支持的区域 : 5 个	<u>是</u>	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。
aws: elasticache ReplicationGroups 的目标 : replicationgroup-interrupt-az-power	每个受支持的区域 : 20 个	<u>是</u>	每个实验 replicationgroup-interrupt-az-power 可以瞄准 ReplicationGroups 的 aws: elasticache: 的最大数量。每日限额适用于定位 ReplicationGroups。欲了解更多信息，请访问： https://docs.aws.amazon.com/fis/latest/userguide/fis-quotas.html 。

Name	默认值	可调整	说明
aw SpotInstances s: ec2 的目标 : send-spot-instance-interruptions	每个受支持的区域 : 100 个	<u>是</u>	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 : 100 个	<u>是</u>	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。超过 5 的配额仅适用于参数 scope:all。如果其他范围类型需要更高的配额, 请联系客户支持。
aws: network:-区域连接的目标子网 route-table-disrupt-cross	每个受支持的区域 : 50 个	<u>是</u>	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 : 500 个	<u>是</u>	每个实验中, 当您使用标签标识目标时, aws:ecs:s top-task 可以标识为目标的最大任务数量。
aws: ecs 的目标任务 : task-cpu-stress	每个受支持的区域 : 200 个	<u>是</u>	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。

Name	默认值	可调整	说明
aws: ecs 的目标任务 : task-io-stress	每个受支持的区域 : 200 个	<u>是</u>	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 的目标任务 : task-kill-process	每个受支持的区域 : 200 个	<u>是</u>	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 的目标任务 : task-network-blackhole-port	每个受支持的区域 : 200 个	<u>是</u>	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 的目标任务 : task-network-latency	每个受支持的区域 : 200 个	<u>是</u>	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 的目标任务 : task-network-packet-loss	每个受支持的区域 : 200 个	<u>是</u>	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。
aws: TransitGateways network:-区域连接的目标 transit-gateway-disrupt-cross	每个受支持的区域 : 50 个	<u>是</u>	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。

Name	默认值	可调整	说明
aws: directconnect: 虚拟接口的目标虚拟接口	每个受支持的区域：5 个	<u>是</u>	每个实验使用标签识别目标时，aws: directconnect: virtual-interface 可以瞄准的最大虚拟接口数量。
aws: ebs 的目标卷：pause-volume-io	每个受支持的区域：160 个	<u>是</u>	每个实验使用标签识别目标时 aws: ebs: pause-volume-io 可以瞄准的最大卷数。
每个实验模板的目标账户配置	每个受支持的区域：40 个	<u>是</u>	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: lambda: action 的目标函数。invocation-add-delay	每个受支持的区域：140 个	<u>是</u>	每个实验使用标签识别目标时，aws: lambda: invocation-add-delay 可以定位的最大 Lambda 函数数量。
aws:lambda:invocation-error 操作的目标函数。	每个受支持的区域：140 个	<u>是</u>	每个实验中，当您使用标签标识目标时，aws:lambda:invocation-error 可以确定为目标的 Lambda 函数的最大数量。
aws: lambda: action 的目标函数。invocation-http-integration-response	每个受支持的区域：140 个	<u>是</u>	每个实验使用标签识别目标时，aws: lambda: invocation-http-integration-response 可以定位的最大 Lambda 函数数量。

Name	默认值	可调整	说明
瞄准多区域集群进行 aws: memorydb:-replication 操作。multi-region-cluster-pause	每个受支持的区域 : 50 个	<u>是</u>	每个实验使用标签识别目标时，aws: memorydb: multi-region-cluster-pause-replication 可以瞄准的最大 MemoryDB 多区域集群数量。如果需要更高的配额，请联系客户支持。
aws: dynamodb: 操作的目标表 global-table-pause-replication	每个受支持的区域 : 60 个	<u>是</u>	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：

Name	限制
aws: elasticache ReplicationGroups 的每日目标上限 : replicationgroup-interrupt-az-power	限额为每天每个区域每个账户 20 个 ReplicationGroups 目标。可以通过在 AWS Support 中心控制台 中创建支持案例来请求增加此数字。

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
的新动作 AWS Direct Connect	您可以使用 <code>aws: directconnect: virtual-interface</code> 操作通过暂时中断本地网络与与目标虚拟接口关联的对等方之间的边界网关协议会话来测试AWS Direct Connect连接的弹性。	2025 年 12 月 18 日
新场景	现在，您可以使用新的“AZ：应用程序减速”和“跨可用区：流量减速”场景。	2025 年 11 月 12 日
SSM 文档的新参数	AWSFIS-Run-Network-Latency-Sources 和-Network-Packet-Loss-Sources SSM 文档现在支持 Flow AWSFIS-RunPercent 参数。	2025 年 11 月 12 日
ECS 和 EKS 操作的新参数	<code>aws: ecs:</code> 、 <code>aws: ecs: task-network-latency</code> 、 <code>aws: eks: task-network-packet-loss pod-network-latency</code> 和 <code>aws: eks:</code> 操作现在支持 <code>flowPercent</code> 参数。 <code>pod-network-packet-loss</code>	2025 年 11 月 12 日
AWS托管策略更新	更新了托管策略 AWSFaultInjectionSimulatorEC2访问权限 ：添加了 <code>ec2: DescribeSubnets</code> 权限。	2025 年 11 月 12 日
AWS托管策略更新	更新了托管策略 AWSFaultInjectionSimulatorECSAccess ：添加了	2025 年 11 月 12 日

	ecs: DescribeContainerInstances、ec2: DescribeSubnets 和 ec2: DescribeInstances 权限。	
亚马逊 Kinesis Data Streams 的新操作	您可以使用 aws: kinesis: stream 操作。	2025 年 10 月 15 日
亚马逊 EBS 的新实验	您可以使用 aws: ebs: volume-io-latency 操作来模拟 Amazon EBS 卷的 I/O 延迟时间延长。	2025 年 9 月 16 日
新的操作范围 AWS FIS	您可以使用 aws: network: disrupt-connection 操作来中断与 S3 Express One Zone 目录存储桶的连接。此范围现在也包含在“可用区可用性：电源中断”场景中。	2025 年 8 月 18 日
FIS 中支持 MemoryDB AWS	您可以使用 AWS FIS 来测试带有 Amazon MemoryDB 多区域集群的应用程序如何响应跨区域网络中断期间暂停的数据复制。	2025 年 7 月 15 日
AWSFIS 中的 ARC 支持	您可以使用 AWS FIS 来测试 ARC 区域自动换档在 AZ 电源中断期间如何自动恢复您的应用程序。	2025 年 3 月 26 日
新的实验报告配置	现在，您可以启用 AWS 用 FIS 生成实验报告，这些报告汇总了 CloudWatch 仪表板上的实验操作和响应。	2024 年 11 月 12 日

新的 Lambda 操作	现在，您可以使用 <code>aws:lambda:function</code> 操作在 Lambda 函数的调用中注入错误。	2024 年 10 月 31 日
新增安全杆功能	AWS FIS 现在支持安全杠杆，允许您快速停止所有正在运行的实验并防止新的实验开始。	2024 年 9 月 3 日
新增故障排除章节	AWS FIS 添加了故障排除指南，其中包括失败实验的错误代码和上下文。	2024 年 8 月 13 日
新操作	现在，您可以使用 <code>aws:dynamodb:global-table-pause-replication</code> 操作暂停在目标全局表与其副本表之间进行数据复制。 <code>aws:dynamodb:encrypted-global-table-pause-replication</code> 操作将不再受支持。	2024 年 4 月 24 日
新增操作模式实验选项	您可以将操作模式设置为 <code>skip-all</code> ，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	您现在可以使用 AWS FIS 场景 <code>跨区域:连接和可用区可用性:电源中断</code> 。	2023 年 11 月 30 日
新操作	您现在可以使用 <code>aws:ec2:asg-insufficient-instance-capacity-error</code> 操作。	2023 年 11 月 30 日

新操作	您现在可以使用 <code>aws:ec2:api-insufficient-instance-capacity-error</code> 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 <code>aws:network:route-table-disrupt-cross-region-connectivity</code> 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 <code>aws:network:transit-gateway-disrupt-cross-region-connectivity</code> 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 <code>aws:dynamodb:encrypted-global-table-pause-replication</code> 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 <code>aws:s3:bucket-pause-replication</code> 操作。	2023 年 11 月 30 日
新操作	您现在可以使用 <code>aws:elasticache:interrupt-cluster-az-power</code> 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
AWSFIS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	AWSFIS 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	AWSFIS 更新了现有的托管策略。	2023 年 6 月 2 日
新操作	您可以执行 <code>aws:ecs:task</code> 和 <code>aws:eks:pod</code> 新操作。	2023 年 6 月 1 日
AWS托管策略更新	AWSFIS 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	您可以使用该 <code>aws:ebs:pause-volume-io</code> 操作在目标卷和它们所连接的实例 I/O 之间暂停。	2023 年 1 月 27 日
新操作	您可以执行 <code>aws:network:disrupt-connectivity</code> 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 <code>aws:eks:inject-kubernetes-custom-resource</code> 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 <code>CloudWatch logs</code> 或 <code>S3</code> 存储桶。	2022 年 2 月 28 日
新通知	当实验状态发生变化时，AWSFIS 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 <code>aws:ecs:stop-task</code> 操作，停止指定任务。	2022 年 2 月 9 日

新操作	您可以执行 <code>aws:cloudwatch:assert-alarm-state</code> 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	您可以使用以下预先配置的 SSM 文档： <code>AWSFIS-Run-io-Stress</code> 、 <code>-Network-Blackhold-Port</code> 、 <code>-Network-Latency-Sources</code> 、 <code>-Network-Packet-Lass-Lass-Sources</code> 和 <code>-Network AWSFIS-Run-Packet-Lass-Sources</code> <code>AWSFIS-Run AWSFIS-Run AWSFIS-Run</code>	2021 年 11 月 4 日
新操作	您可以执行 <code>aws:ec2:send-spot-instance-interruptions</code> 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 <code>aws:ssm:start-automation-execution</code> 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。