使用者指南

AWS Fault Injection Service

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Fault Injection Service: 使用者指南

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS FIS？ 1
概念 1
動作 2
目標 2
停止條件 2
支援的 AWS 服務 2
存取 AWS FIS 3
定價 4
規劃您的實驗 5
基本原則和指導方針 5
實驗規劃準則 6
實驗範本元件 8
範本語法 8
開始使用 9
動作 9
動作語法 9
動作識別符 10
動作參數 11
動作目標 11
動作持續時間 12
動作範例 12
目標 15
目標語法 16
資源類型 17
識別目標資源 18
選擇模式 22
範例目標 22
範例篩選條件 23
停止條件 28
停止條件語法 28
進一步了解 29
實驗角色 29
先決條件 30
選項 1：建立實驗角色並連接 AWS 受管政策 31

選項 2：建立實驗角色並新增內嵌政策文件 ... 32
實驗報告組態 ... 34
實驗報告組態語法 ... 36
實驗報告許可 ... 37
實驗報告最佳實務 ... 39
實驗選項 ... 40
帳戶目標 ... 40
空目標解析度模式 ... 41
動作模式 ... 42
動作參考 ... 43
錯誤注入動作 ... 44
aws:fis:inject-api-internal-error ... 44
aws:fis:inject-api-throttle-error ... 45
aws:fis:inject-api-unavailable-error ... 45
復原動作 ... 46
aws:arc:start-zonal-autoshift ... 46
等待動作 ... 47
aws:fis:wait ... 47
Amazon CloudWatch 動作 ... 48
aws:cloudwatch:assert-alarm-state ... 48
Amazon DynamoDB 動作 ... 48
aws:dynamodb:global-table-pause-replication ... 48
Amazon EBS 動作 ... 50
aws:ebs:pause-volume-io ... 50
Amazon EC2 動作 ... 51
aws:ec2:api-insufficient-instance-capacity-error ... 52
aws:ec2:asg-insufficient-instance-capacity-error ... 52
aws:ec2:reboot-instances ... 53
aws:ec2:send-spot-instance-interruptions ... 54
aws:ec2:stop-instances ... 54
aws:ec2:terminate-instances ... 55
Amazon ECS 動作 ... 56
aws:ecs:drain-container-instances ... 56
aws:ecs:stop-task ... 57
aws:ecs:task-cpu-stress ... 58
aws:ecs:task-io-stress ... 58

aws:ecs:task-kill-process ... 59
aws:ecs:task-network-blackhole-port ... 60
aws:ecs:task-network-latency ... 60
aws:ecs:task-network-packet-loss ... 61
Amazon EKS 動作 ... 62
aws:eks:inject-kubernetes-custom-resource ... 63
aws:eks:pod-cpu-stress ... 64
aws:eks:pod-delete ... 65
aws:eks:pod-io-stress ... 66
aws:eks:pod-memory-stress ... 67
aws:eks:pod-network-blackhole-port ... 68
aws:eks:pod-network-latency ... 69
aws:eks:pod-network-packet-loss ... 70
aws:eks:terminate-nodegroup-instances ... 71
Amazon ElastiCache 動作 ... 71
aws:elasticache:replicationgroup-interrupt-az-power ... 71
AWS Lambda 動作 ... 72
aws:lambda:invocation-add-delay ... 72
aws:lambda:invocation-error ... 73
aws:lambda:invocation-http-integration-response ... 74
網路動作 ... 75
aws:network:disrupt-connectivity ... 75
aws:network:route-table-disrupt-cross-region-connectivity ... 76
aws:network:transit-gateway-disrupt-cross-region-connectivity ... 77
Amazon RDS 動作 ... 78
aws:rds:failover-db-cluster ... 78
aws:rds:reboot-db-instances ... 79
Amazon S3 動作 ... 80
aws:s3:bucket-pause-replication ... 80
Systems Manager 動作 ... 81
aws:ssm:send-command ... 81
aws:ssm:start-automation-execution ... 82
SSM 文件動作 ... 83
使用 aws:ssm:send-command動作 ... 83
預先設定的 AWS FIS SSM 文件 ... 84
範例 ... 92

故障診斷 ... 92
ECS 任務動作 ... 93
動作 ... 93
限制 ... 94
要求 ... 94
指令碼的參考版本 ... 97
範例實驗範本 ... 99
EKS Pod 動作 ... 100
動作 ... 101
限制 ... 102
要求 ... 103
建立實驗角色 ... 103
設定 Kubernetes 服務帳戶 ... 103
授予 IAM 使用者和角色對 Kubernetes APIs存取權 ... 104
Pod 容器映像 ... 105
範例實驗範本 ... 107
AWS Lambda 動作 ... 109
動作 ... 109
限制 ... 109
先決條件 ... 109
設定 Lambda 函數 ... 111
設定 AWS FIS 實驗 ... 112
日誌 ... 112
進階主題 ... 113
AWS FIS Lambda 延伸模組版本 ... 119
管理實驗範本 ... 122
建立實驗範本 ... 122
檢視實驗範本 ... 124
產生目標預覽 ... 125
從範本開始實驗 ... 126
更新實驗範本 ... 126
標記實驗範本 ... 127
刪除實驗範本 ... 127
範例範本 ... 128
根據篩選條件停止 EC2 執行個體 ... 128
停止指定數量的 EC2 執行個體 ... 130

執行預先設定的 AWS FIS SSM 文件 131
執行預先定義的 Automation Runbook 132
具有目標 IAM 角色之 EC2 執行個體上的調節 API 動作 132
Kubernetes 叢集中 Pod 的壓力測試 CPU 134
管理實驗 136
開始實驗 136
檢視您的實驗 137
實驗狀態 137
動作狀態 138
標記實驗 138
停止實驗 139
列出已解析的目標 139
教學課程 140
測試執行個體停止和啟動 140
先決條件 140
步驟 1：建立實驗範本 140
步驟 2：開始實驗 143
步驟 3：追蹤實驗進度 144
步驟 4：驗證實驗結果 144
步驟 5：清除 144
在執行個體上執行 CPU 應力 145
先決條件 145
步驟 1：為停止條件建立 CloudWatch 警示 146
步驟 2：建立實驗範本 147
步驟 3：開始實驗 149
步驟 4：追蹤實驗進度 149
步驟 5：驗證實驗結果 149
步驟 6：清除 144
測試 Spot 執行個體中斷 151
先決條件 152
步驟 1：建立實驗範本 153
步驟 2：開始實驗 155
步驟 3：追蹤實驗進度 155
步驟 4：驗證實驗結果 156
步驟 5：清除 157
模擬連線事件 157

先決條件 158
步驟 1：建立 AWS FIS 實驗範本 159
步驟 2：Ping Amazon S3 端點 160
步驟 3：啟動您的 AWS FIS 實驗 161
步驟 4：追蹤您的 AWS FIS 實驗進度 161
步驟 5：驗證 Amazon S3 網路中斷 161
步驟 5：清除 162
排程重複實驗 162
先決條件 163
步驟 1：建立 IAM 角色和政策 163
步驟 2：建立 Amazon EventBridge 排程器 165
步驟 3：驗證您的實驗 166
步驟 4：清理 166
使用案例程式庫 167
檢視案例 167
使用案例 167
匯出案例 168
案例參考 168
AZ Availability: Power Interruption 170
Cross-Region: Connectivity 183
使用多帳戶實驗 196
概念 196
最佳實務 197
先決條件 197
許可 197
停止條件 （選用） 200
多帳戶實驗的安全控制桿 （選用） 200
建立多帳戶實驗範本 200
更新目標帳戶組態 202
刪除目標帳戶組態 202
排程實驗 204
建立排程器角色 204
建立實驗排程 207
使用主控台更新排程 208
更新實驗排程 209
停用或刪除實驗排程 209

安全控制桿 210
安全控制桿的概念 210
Saftey 控制桿資源 210
使用安全控制桿 211
檢視安全控制桿 211
使用安全控制桿 211
停用安全控制桿 212
監控實驗 213
使用 CloudWatch 監控 214
監控 AWS FIS 實驗 214
AWS FIS 用量指標 214
使用 EventBridge 監控 215
實驗記錄 217
許可 217
日誌結構描述 217
日誌目的地 219
日誌記錄範例 219
啟用實驗記錄 224
停用實驗記錄 224
使用 記錄 API 呼叫 AWS CloudTrail 225
使用 CloudTrail 225
了解 AWS FIS 日誌檔案項目 226
疑難排解 231
錯誤代碼 231
安全 233
資料保護 233
靜態加密 234
傳輸中加密 234
身分與存取管理 234
目標對象 235
使用身分驗證 235
使用政策管理存取權 238
Fault Injection Service AWS 如何與 IAM 搭配使用 240
政策範例 245
使用服務連結角色 254
AWS 受管政策 256

基礎架構安全 260
AWS PrivateLink 260
考量事項 261
建立介面 VPC 端點 261
建立 VPC 端點政策 261
標記您的 資源 263
標記限制 263
使用標籤 263
限制和配額 265
文件歷史紀錄 277
..... cclxxxii

什麼是 AWS Fault Injection Service？

AWS Fault Injection Service (AWS FIS) 是一種受管服務，可讓您在 AWS 工作負載上執行故障注入實驗。錯誤注入是以混亂工程的原則為基礎。這些實驗會透過建立破壞性事件來強調應用程式，以便您可以觀察應用程式回應的方式。然後，您可以使用此資訊來改善應用程式的效能和彈性，使其如預期般運作。

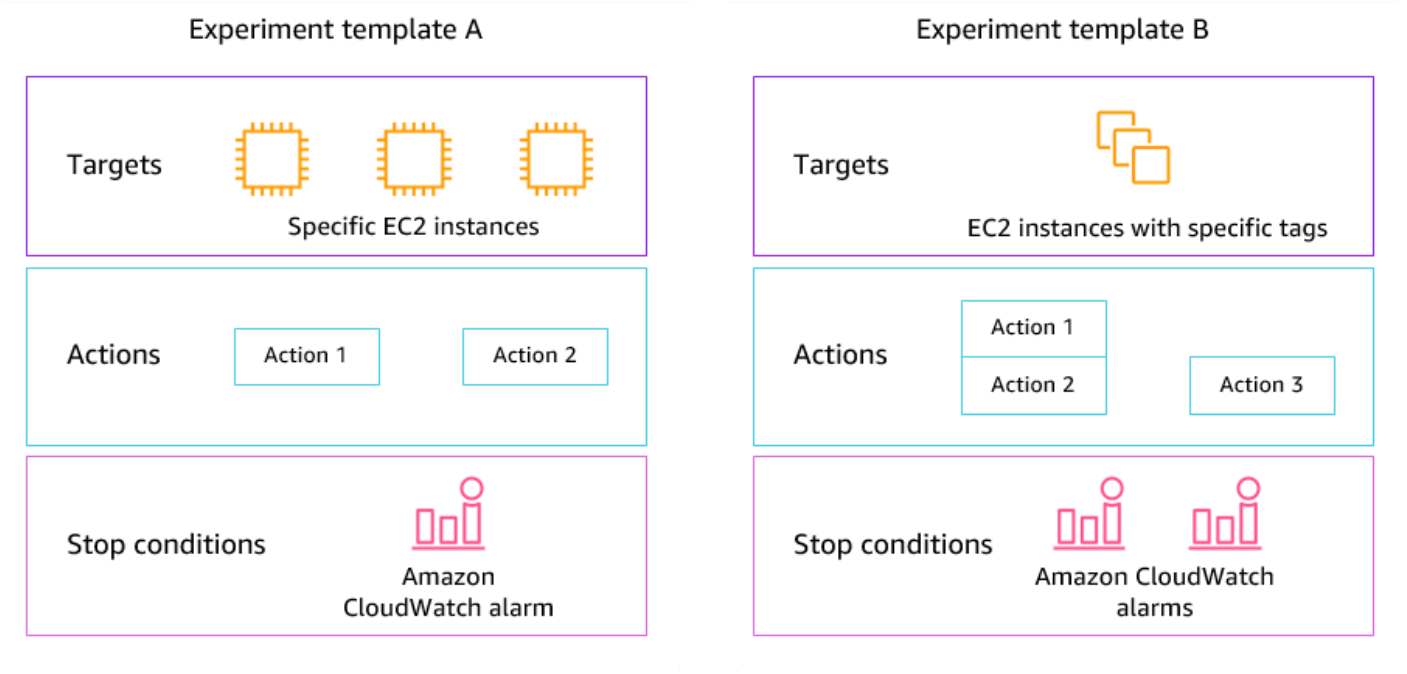
若要使用 AWS FIS，您可以設定並執行實驗，協助您建立所需的真實環境，以發現在其他情況下可能難以發現的應用程式問題。 AWS FIS 提供會產生中斷的範本，以及您在生產環境中執行實驗所需的控制項和護欄，例如在符合特定條件時自動復原或停止實驗。

> **Important**
>
> AWS FIS 會對系統中的真實 AWS 資源執行實際動作。因此，在您使用 AWS FIS 在生產環境中執行實驗之前，強烈建議您完成規劃階段，並在生產前環境中執行實驗。

如需規劃實驗的詳細資訊，請參閱測試可靠性和 規劃您的 AWS FIS 實驗。如需 AWS FIS 的詳細資訊，請參閱 AWS Fault Injection Service。

AWS FIS 概念

若要使用 AWS FIS，請在 AWS 資源上執行實驗，以測試應用程式或系統在故障條件下如何執行的理論。若要執行實驗，請先建立實驗範本。實驗範本是您實驗的藍圖。它包含實驗的動作、目標和停止條件。建立實驗範本之後，您可以使用它來執行實驗。實驗執行時，您可以追蹤其進度並檢視其狀態。當實驗中的所有動作都已執行時，實驗即完成。



動作

動作是 AWS FIS 在實驗期間對 AWS 資源執行的活動。 AWS FIS 根據 AWS 資源類型提供一組預先設定的動作。每個動作會在實驗期間執行指定的持續時間，或直到您停止實驗為止。動作可以循序或同時執行（平行）。

目標

目標是一或多個 AWS 資源，其中 AWS FIS 會在實驗期間對其執行動作。您可以選擇特定資源，也可以根據特定條件選擇資源群組，例如標籤或狀態。

停止條件

AWS FIS 提供在 AWS 工作負載上安全地執行實驗所需的控制項和護欄。停止條件是一種機制，可在實驗達到您定義為 Amazon CloudWatch 警示的閾值時停止實驗。如果在實驗執行時觸發停止條件，AWS FIS 會停止實驗。

支援的 AWS 服務

AWS FIS 為跨 AWS 服務的特定類型目標提供預先設定的動作。 AWS FIS 支援目標資源的動作如下 AWS 服務：

- Amazon CloudWatch

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

對於單一帳戶實驗，目標資源必須與 AWS 帳戶 實驗相同。您可以使用 AWS FIS 多 AWS 帳戶 帳戶實驗，執行以不同帳戶中資源為目標的 AWS FIS 實驗。

如需詳細資訊，請參閱AWS FIS 的動作。

存取 AWS FIS

您可以透過下列任何方式使用 AWS FIS：

- AWS Management Console — 提供可用來存取 AWS FIS 的 Web 界面。如需詳細資訊，請參閱使用 AWS Management Console。
- AWS Command Line Interface (AWS CLI) — 為廣泛的 AWS 服務提供命令，包括 AWS FIS，並在 Windows、macOS 和 Linux 上支援 。如需詳細資訊，請參閱AWS Command Line Interface。如需 AWS FIS 命令的詳細資訊，請參閱 AWS CLI 命令參考中的 fis。
- AWS CloudFormation — 建立描述 AWS 資源的範本。您可以使用範本，佈建並管理這些資源做為單一單位。如需詳細資訊，請參閱 AWS Fault Injection Service 資源類型參考。
- AWS SDKs — 提供語言特定的 APIs並負責許多連線詳細資訊，例如計算簽章、處理請求重試和處理錯誤。如需詳細資訊，請參閱 AWS 開發套件。
- HTTPS API — 提供您可以使用 HTTPS 請求呼叫的低階 API 動作。如需詳細資訊，請參閱 AWS Fault Injection Service API 參考。

AWS FIS 的定價

根據實驗的目標帳戶數量，您每分鐘需支付動作從開始到結束執行的費用。如需詳細資訊，請參閱 AWS FIS 定價。

規劃您的 AWS FIS 實驗

故障注入是藉由建立破壞性事件，例如伺服器中斷或 API 限流，在測試或生產環境中對應用程式施加壓力的程序。透過觀察系統回應的方式，您可以實作改善。當您在系統上執行實驗時，它可協助您以受控方式識別系統性弱點，以免這些弱點影響依賴您系統的客戶。然後，您可以主動解決問題，以協助防止無法預測的結果。

開始使用 AWS FIS 執行故障注入實驗之前，建議您先熟悉下列原則和準則。

> **Important**
>
> AWS FIS 會對系統中的真實 AWS 資源執行實際動作。因此，在您開始使用 AWS FIS 執行實驗之前，強烈建議您先在生產前或測試環境中完成規劃階段和測試。

目錄

- 基本原則和指導方針
- 實驗規劃準則

基本原則和指導方針

開始使用 AWS FIS 進行實驗之前，請執行下列步驟：

1. 識別實驗的目標部署 — 首先識別目標部署。如果這是您的第一個實驗，我們建議您從生產前或測試環境中開始。
2. 檢閱應用程式架構：您必須確定已識別每個元件的所有應用程式元件、相依性和復原程序。從檢閱應用程式架構開始。視應用程式而定，請參閱 AWS Well-Architected Framework。
3. 定義穩定狀態行為：根據重要的技術和業務指標來定義系統的穩定狀態行為，例如延遲、CPU 負載、每分鐘登入失敗、重試次數或頁面載入速度。
4. 形成假設 — 形成您預期系統行為在實驗期間如何變更的假設。假設定義遵循以下格式：

   如果執行######，則#########不應超過 #。

   例如，身分驗證服務的假設可能會如下讀取：「如果網路延遲增加 10%，則登入失敗增加少於 1%。」 實驗完成後，您會評估應用程式彈性是否符合您的業務和技術預期。

在使用 AWS FIS 時，我們也建議遵循這些準則：

- 一律在測試環境中開始試驗 AWS FIS。永遠不要從生產環境開始。隨著您進行故障注入實驗，您可以在測試環境以外的其他受控環境中進行實驗。
- 從小型、簡單的實驗開始，例如在一個目標上執行 aws：ec2：stop-instances 動作，以建立您團隊對應用程式彈性的信心。
- 錯誤注入可能會導致實際問題。請謹慎進行，並確保您的第一次故障注入是在測試執行個體上，這樣客戶就不會受到影響。
- 測試、測試和測試更多。故障注入旨在透過妥善規劃的實驗在受控環境中實作。這可讓您對應用程式和工具的能力建立信心，以承受動盪的環境。
- 我們強烈建議您在開始之前，先備妥卓越的監控和提醒計畫。如果沒有它，您就無法了解或測量實驗的影響，這對於持續的故障注入實務至關重要。

實驗規劃準則

使用 AWS FIS，您可以在 AWS 資源上執行實驗，以測試應用程式或系統在故障條件下如何執行的理論。

以下是規劃 AWS FIS 實驗的建議準則。

- 檢閱中斷歷史記錄 — 檢閱您系統的先前中斷和事件。這可協助您建立系統整體運作狀態和彈性的概觀。開始在系統上執行實驗之前，您應該解決系統中的已知問題和弱點。
- 識別具有最大影響的服務 — 檢閱您的服務，並識別那些對最終使用者或客戶有最大影響的服務。
- 識別目標系統 — 目標系統是您執行實驗的系統。如果您初次使用 AWS FIS，或從未執行過故障注入實驗，建議您先在生產前或測試系統上執行實驗。
- 諮詢您的團隊：詢問他們擔心什麼。您可以建立假設來證明或拒絕他們的疑慮。您也可以詢問您的團隊他們不擔心什麼。此問題可以顯示兩個常見的缺點： 沉降成本和確認偏差缺失。根據團隊的答案形成假設，有助於提供有關系統狀態現實的詳細資訊。
- 檢閱您的應用程式架構：對您的系統或應用程式進行檢閱，並確保您已識別每個元件的所有應用程式元件、相依性和復原程序。

  建議您檢閱 AWS Well-Architected 架構。該架構可協助您為應用程式和工作負載建置安全、高效能、彈性且高效率的基礎設施。如需詳細資訊，請參閱 AWS Well-Architected。
- 識別適用的指標 — 您可以使用 Amazon CloudWatch 指標來監控實驗對 AWS 資源的影響。您可以使用這些指標來判斷應用程式以最佳方式執行時的基準或「穩定狀態」。然後，您可以在實驗期間或

之後監控這些指標，以判斷影響。如需詳細資訊，請參閱使用 Amazon CloudWatch 監控 AWS FIS 用量指標。

- 為您的系統定義可接受的效能閾值 — 識別代表系統可接受、穩定狀態的指標。您將使用此指標來建立一或多個 CloudWatch 警示，其代表實驗的停止條件。如果觸發警示，實驗會自動停止。如需詳細資訊，請參閱AWS FIS 的停止條件。

AWS FIS 實驗範本元件

您可以使用下列元件來建構實驗範本：

動作

您要執行的 AWS FIS 動作。動作可以依您指定的設定順序執行，也可以同時執行。如需詳細資訊，請參閱動作。

目標

執行特定動作 AWS 的資源。如需詳細資訊，請參閱目標。

停止條件

定義應用程式效能無法接受之閾值的 CloudWatch 警示。如果在實驗執行時觸發停止條件， AWS FIS 會停止實驗。如需詳細資訊，請參閱停止條件。

實驗角色

授予 AWS FIS 所需許可的 IAM 角色，以便其代表您執行實驗。如需詳細資訊，請參閱實驗角色。

實驗報告組態

啟用實驗報告的組態。如需詳細資訊，請參閱AWS FIS 的實驗報告組態。

實驗選項

實驗範本的選項。如需詳細資訊，請參閱的實驗選項 AWS FIS。

您的帳戶具有與 AWS FIS 相關的配額。例如，每個實驗範本的動作數量有配額。如需詳細資訊，請參閱限制和配額。

範本語法

以下是實驗範本的語法。

```
{
            "description": "string",
            "targets": {},
            "actions": {},
            "stopConditions": [],
            "roleArn": "arn:aws:iam::123456789012:role/AllowFISActions",
            "experimentReportConfiguration":{},
```

```
            "experimentOptions":{},
            "tags": {}
        }
```

如需範例，請參閱 範例範本。

開始使用

若要使用 建立實驗範本 AWS Management Console，請參閱 建立實驗範本。

若要使用 建立實驗範本 AWS CLI，請參閱 AWS FIS 實驗範本範例。

AWS FIS 的動作

若要建立實驗範本，您必須定義一或多個動作。如需 AWS FIS 提供的預先定義動作清單，請參閱 動作參考。

您只能在實驗期間執行動作一次。若要在相同的實驗中多次執行相同的 AWS FIS 動作，請使用不同的名稱多次將其新增至範本。

目錄

- 動作語法
- 動作識別符
- 動作參數
- 動作目標
- 動作持續時間
- 動作範例

動作語法

以下是 動作的語法。

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

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

當您定義動作時，請提供下列項目：

action_name

動作的名稱。

actionId

動作識別符。

description

選擇性的描述。

parameters

任何動作參數。

startAfter

必須先完成的任何動作，才能啟動此動作。否則，動作會在實驗開始時執行。

targets

任何動作目標。

如需範例，請參閱 the section called “動作範例”。

動作識別符

每個 AWS FIS 動作都有以下格式的識別符：

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

例如，下列動作會停止目標 Amazon EC2 執行個體：

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

如需動作的完整清單，請參閱 AWS FIS 動作參考。

動作參數

有些 AWS FIS 動作具有動作特有的其他參數。這些參數用於在動作執行時將資訊傳遞給 AWS FIS。

AWS FIS 支援使用 `aws:ssm:send-command`動作的自訂錯誤類型，該動作使用 SSM Agent 和 SSM 命令文件在目標執行個體上建立錯誤條件。`aws:ssm:send-command` 動作包含以 SSM 文件的 Amazon Resource Name (ARN) 做為值的`documentArn`參數。當您將 動作新增至實驗範本時，您可以指定參數的值。

如需指定 `aws:ssm:send-command`動作參數的詳細資訊，請參閱 使用 aws:ssm:send-command動作。

在可能的情況下，您可以在動作參數中輸入轉返組態（也稱為後置動作)。後置動作會將目標回復為動作執行前原有的狀態。後置動作會在動作持續時間指定的時間之後執行。並非所有動作都可以支援貼文動作。例如，如果動作終止 Amazon EC2 執行個體，則您無法在執行個體終止後復原執行個體。

動作目標

動作會在您指定的目標資源上執行。定義目標之後，您可以在定義動作時指定其名稱。

```
"targets": {
    "ResourceType": "resource_name"
}
```

AWS FIS 動作支援動作目標的下列資源類型：

- AutoScalingGroups – Amazon EC2 Auto Scaling 群組
- 儲存貯體 – Amazon S3 儲存貯體
- 叢集 – Amazon EKS 叢集
- 叢集 – Amazon ECS 叢集或 Amazon Aurora 資料庫叢集
- DBInstances – Amazon RDS 資料庫執行個體
- 執行個體 – Amazon EC2 執行個體
- ManagedResources – 啟用 ARC 區域轉移的 Amazon EKS 叢集、Amazon EC2 Application and Network Load Balancer 和 Amazon EC2 Auto Scaling 群組。

- 節點群組 – Amazon EKS 節點群組
- Pod：Amazon EKS 上的 Kubernetes Pod
- ReplicationGroups – ElastiCache 複寫群組
- 角色 – IAM 角色
- SpotInstances – Amazon EC2 Spot 執行個體
- 子網路 – VPC 子網路
- 資料表 – Amazon DynamoDB 全域資料表
- 任務 – Amazon ECS 任務
- TransitGateways – 傳輸閘道
- 磁碟區 – Amazon EBS 磁碟區

如需範例，請參閱 the section called “動作範例”。

動作持續時間

如果動作包含參數，您可以用來指定動作的持續時間，根據預設，只有在指定的持續時間過後，才會將動作視為完成。如果您已將`emptyTargetResolutionMode`實驗選項設定為 `skip`，則在未解析任何目標時，動作會立即完成，狀態為「略過」。例如，如果您指定 5 分鐘的持續時間， AWS FIS 會將動作視為在 5 分鐘後完成。然後，它會開始下一個動作，直到所有動作完成為止。

持續時間可以是維持動作條件的時間長度，或監控指標的時間長度。例如，延遲會在指定的時間內注入。對於近乎即時的動作類型，例如終止執行個體，會在指定的時間內監控停止條件。

如果動作在動作參數中包含後置動作，則後置動作會在動作完成後執行。完成後置動作所需的時間可能會導致指定動作持續時間與下一個動作開始（或實驗結束，如果所有其他動作都完成）之間的延遲。

動作範例

以下是範例動作。

範例

- 停止 EC2 執行個體
- 中斷 Spot 執行個體
- 中斷網路流量
- 終止 EKS 工作者

- 啟動 ARC 區域自動轉移

範例：停止 EC2 執行個體

下列動作會停止使用名為 *targetInstances* 的目標所識別的 EC2 執行個體。兩分鐘後，它會重新啟動目標執行個體。

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

範例：中斷 Spot 執行個體

下列動作會停止使用名為 *targetSpotInstances* 的目標所識別的 Spot 執行個體。它會等待兩分鐘再中斷 Spot 執行個體。

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

範例：中斷網路流量

下列動作會拒絕目標子網路與其他可用區域中子網路之間的流量。

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

範例：終止 EKS 工作者

下列動作會在使用名為 *targetNodeGroups* 的目標所識別的 EKS 叢集中終止 50% 的 EC2 執行個體。

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

範例：啟動 ARC 區域自動轉移

下列動作會啟動 ARC 區域自動轉移，將受管資源從參數*duration-in-parameteres*的 *az-in-parameters* 轉移。資源類型ManagedResources會用作 AWS FIS 實驗範本中目標名稱的索引鍵。

```
{
    "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 FIS 的目標

目標為一或多個 AWS 資源，其中 Fault Injection Service AWS (AWS FIS) 會在實驗期間對其執行動作。目標可以位於與實驗相同的 AWS 帳戶中，也可以位於使用多帳戶實驗的不同帳戶中。若要進一步了解如何鎖定不同帳戶中的資源，請參閱 使用多帳戶實驗。

您可以在建立實驗範本時定義目標。您可以在實驗範本中針對多個動作使用相同的目標。

AWS FIS 會在實驗開始時識別所有目標，再啟動動作集中的任何動作。 AWS FIS 會使用其為整個實驗選取的目標資源。如果找不到目標，實驗會失敗。

內容

- 目標語法
- 資源類型
- 識別目標資源
  - 資源篩選條件
  - 資源參數
- 選擇模式
- 範例目標
- 範例篩選條件

目標語法

以下是目標的語法。

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

當您定義目標時，請提供下列項目：

target_name

目標的名稱。

resourceType

資源類型。

resourceArns

特定資源的 Amazon Resource Name (ARN)。

resourceTags

套用至特定資源的標籤。

parameters

使用特定屬性識別目標的參數。

filters

資源篩選條件會使用特定屬性來限定已識別的目標資源範圍。

selectionMode

已識別資源的選擇模式。

如需範例，請參閱 the section called “範例目標”。

資源類型

每個 AWS FIS 動作都會對特定 AWS 資源類型執行。當您定義目標時，必須指定一個資源類型。當您指定動作的目標時，目標必須是 動作支援的資源類型。

AWS FIS 支援下列資源類型：

- aws：arc：zonal-shift-managed-resource – 向 ARC 區域轉移註冊 AWS 的資源
- aws：dynamodb：global-table – Amazon DynamoDB 全域資料表
- aws：ec2：autoscaling-group – Amazon EC2 Auto Scaling 群組
- aws：ec2：ebs-volume – Amazon EBS 磁碟區
- aws：ec2：instance – Amazon EC2 執行個體

- aws：ec2：spot-instance – Amazon EC2 Spot 執行個體
- aws：ec2：subnet – Amazon VPC 子網路
- aws：ec2：transit-gateway – 傳輸閘道
- aws：ecs：cluster – Amazon ECS 叢集
- aws：ecs：task – Amazon ECS 任務
- aws：eks：cluster – Amazon EKS 叢集
- aws：eks：nodegroup – Amazon EKS 節點群組
- aws：eks：pod – Kubernetes Pod
- aws：elasticache：replicationgroup – ElastiCache 複寫群組
- aws：iam：role – IAM 角色
- aws：lambda：function – AWS Lambda 函數
- aws：rds：cluster – Amazon Aurora 資料庫叢集
- aws：rds：db – Amazon RDS 資料庫執行個體
- aws：s3：bucket – Amazon S3 儲存貯體

識別目標資源

當您在 AWS FIS 主控台中定義目標時，您可以選擇要鎖定的特定 AWS 資源（特定資源類型）。或者，您可以讓 AWS FIS 根據您提供的條件來識別資源群組。

若要識別目標資源，您可以指定下列項目：

- 資源 IDs – 特定資源的資源 IDs AWS 。所有資源 IDs都必須代表相同類型的資源。
- 資源標籤 – 套用至特定 AWS 資源的標籤。
- 資源篩選條件 – 代表具有特定屬性之資源的路徑和值。如需詳細資訊，請參閱資源篩選條件。
- 資源參數 – 代表符合特定條件之資源的參數。如需詳細資訊，請參閱資源參數。

考量事項

- 您無法同時指定相同目標的資源 ID 和資源標籤。
- 您無法同時指定相同目標的資源 ID 和資源篩選條件。
- 如果您指定具有空白標籤值的資源標籤，則它不等同於萬用字元。它會比對具有具有指定標籤索引鍵和空白標籤值之標籤的資源。

- 如果您指定多個標籤，則目標資源上必須存在所有指定的標籤，才能選取該標籤 (AND)。

資源篩選條件

資源篩選條件是根據特定屬性識別目標資源的查詢。 AWS FIS 會根據您指定的資源類型，將查詢套用至包含 AWS 資源正式描述的 API 動作輸出。具有符合查詢之屬性的資源會包含在目標定義中。

每個篩選條件都以屬性路徑和可能的值表示。路徑是以句點分隔的元素序列，描述在資源的描述動作輸出中到達屬性的路徑。每個句點代表 元素的擴展。每個元素必須以 Pascal 大小寫表示，即使資源的描述動作的輸出為駱駝大小寫。例如，您應該使用 `AvailabilityZone`，而不是 `availablityZone` 做為屬性元素。

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

下列邏輯適用於所有資源篩選條件：

- 如果提供多個篩選條件，包括具有相同路徑的篩選條件，則所有篩選條件都必須符合才能選取資源： AND
- 如果為單一篩選條件提供多個值，則任何一個值都必須符合才能選取資源： OR
- 如果在描述 API 呼叫的路徑位置找到多個值，則任何一個值都需要符合才能選取資源： OR
- 若要比對標籤鍵/值對，您應該改為依標籤選取目標資源 （請參閱上述）。

下表包含 API 動作和 AWS CLI 命令，您可以用來取得每個資源類型的正式描述。 AWS FIS 會代表您執行這些動作，以套用您指定的篩選條件。根據預設，對應的文件會描述結果中包含的資源。例如，最近終止執行個體DescribeInstances的狀態文件可能會出現在結果中。

資源類型	API 動作	AWS CLI 命令
aws:arc:zonal-shift-managed-resource	ListManagedResources	list-managed-resources

資源類型	API 動作	AWS CLI 命令
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes
aws:ec2:instance	DescribeInstances	describe-instances
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ecs:cluster	DescribeClusters	describe-clusters
aws:ecs:task	DescribeTasks	describe-tasks
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	describe-nodegroup
aws:elasticache:replication group	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	list-roles
aws:lambda:function	ListFunctions	list-functions
aws:rds:cluster	DescribeDBClusters	describe-db-clusters
aws:rds:db	DescribeDBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets
aws:dynamodb:global-table	DescribeTable	describe-table

如需範例，請參閱 the section called “範例篩選條件”。

資源參數

資源參數會根據特定條件識別目標資源。

下列資源類型支援參數。

aws:ec2:ebs-volume

- `availabilityZoneIdentifier` – 包含目標磁碟區的可用區域的程式碼（例如 us-east-1a)。

aws:ec2:subnet

- `availabilityZoneIdentifier` – 包含目標子網路之可用區域的程式碼（例如 us-east-1a) 或 AZ ID（例如 use1-az1)。
- `vpc` – 包含目標子網路的 VPC。每個帳戶不支援多個 VPC。

aws:ecs:task

- `cluster` – 包含目標任務的叢集。
- `service` – 包含目標任務的服務。

aws:eks:pod

- `availabilityZoneIdentifier` - 選用。包含目標 Pod 的可用區域。例如 `us-east-1d`。我們透過比較其 hostIP 和叢集子網路的 CIDR 來判斷 Pod 的可用區域。
- `clusterIdentifier` - 必要。目標 EKS 叢集的名稱或 ARN。
- `namespace` - 必要。目標 Pod 的 Kubernetes 命名空間。
- `selectorType` - 必要。選擇器類型。可能的值為 `labelSelector`、`deploymentName` 和 `podName`。
- `selectorValue` - 必要。選擇器值。此值取決於 的值`selectorType`。
- `targetContainerName` - 選用。Pod 規格中定義的目標容器名稱。預設值是每個目標 Pod 規格中定義的第一個容器。

aws:lambda:function

- `functionQualifier` - 選用。要鎖定的函數版本或別名。如果未指定任何限定詞，則會考慮將所有調用設為目標。如果指定具有多個版本的別名，只要使用包含別名的 ARN 叫用別名，別名中包含的所有版本都會被視為目標。如果`$LATEST`使用特殊別名，則會考慮對基本函數 ARN 的叫用，以及包含在 ARN `$LATEST`中的叫用，以進行錯誤注入。如需 Lambda 版本的詳細資訊，請參閱《 AWS Lambda 使用者指南》中的管理 Lambda 函數版本。

aws:rds:cluster

- `writerAvailabilityZoneIdentifiers` - 選用。資料庫叢集寫入器的可用區域。可能的值為：可用區域識別符的逗號分隔清單，`all`。

aws:rds:db

- `availabilityZoneIdentifiers` - 選用。要受影響的資料庫執行個體可用區域。可能的值為：可用區域識別符的逗號分隔清單，`all`。

aws:elasticache:replicationgroup

- `availabilityZoneIdentifier` - 必要。包含目標節點之可用區域的程式碼（例如 us-east-1a) 或 AZ ID（例如 use1-az1)。

選擇模式

您可以透過指定選取模式來限定已識別資源的範圍。 AWS FIS 支援下列選取模式：

- `ALL` – 在所有目標上執行 動作。
- `COUNT(n)` – 在指定數量的目標上執行動作，隨機從已識別的目標中選擇。例如，COUNT(1) 會選取其中一個已識別的目標。
- `PERCENT(n)` – 在指定百分比的目標上執行動作，以隨機方式從已識別的目標中選擇。例如，PERCENT(25) 會選取 25% 的已識別目標。

如果您有奇數的資源並指定 50%， AWS FIS 會捨棄。例如，如果您新增五個 Amazon EC2 執行個體做為目標，範圍為 50%， AWS FIS 會捨入為兩個執行個體。您無法指定少於一個資源的百分比。例如，如果您將四個 Amazon EC2 執行個體和範圍新增至 5%， AWS FIS 就無法選取執行個體。

如果您使用相同的目標資源類型定義多個目標， AWS FIS 可以多次選取相同的資源。

無論您使用哪種選擇模式，如果您指定的範圍未識別任何資源，則實驗會失敗。

範例目標

以下是範例目標。

範例

- 具有指定標籤之指定 VPC 中的執行個體
- 具有指定參數的任務

範例：指定 VPC 中具有指定標籤的執行個體

此範例的可能目標是指定 VPC 中的 Amazon EC2 執行個體，其標籤為 env=prod。選擇模式會指定 AWS FIS 隨機選擇其中一個目標。

```
{
```

```
    "targets": {
        "randomInstance": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {
                "env": "prod"
            },
            "filters": [
                {
                    "path": "VpcId",
                    "values": [
                        "vpc-aabbcc11223344556"
                    ]
                }
            ],
            "selectionMode": "COUNT(1)"
        }
    }
}
```

範例：具有指定參數的任務

此範例的可能目標為具有指定叢集和服務的 Amazon ECS 任務。選擇模式會指定 AWS FIS 隨機選擇其中一個目標。

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

範例篩選條件

以下是範例篩選條件。

範例

- EC2 執行個體
- 資料庫叢集

範例：EC2 執行個體

當您為支援 aws：ec2：instance 資源類型的動作指定篩選條件時， AWS FIS 會使用 Amazon EC2 describe-instances命令並套用篩選條件來識別目標。

describe-instances 命令會傳回 JSON 輸出，其中每個執行個體都是 下的結構`Instances`。以下是部分輸出，其中包含以`##`標示的欄位。我們將提供範例，這些範例使用這些欄位從 JSON 輸出的結構指定屬性路徑。

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

```
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-aabbcc11223344556",
                    "VpcId": "vpc-00bbbbbbbbbbbbbbbb",
                    ...
                    "NetworkInterfaces": [
                    {
                        ...
                        "Groups": [
                            {
                                "GroupName": "sec-group-1",
                                "GroupId": "sg-a0011223344556677"
                            },
                            {
                                "GroupName": "sec-group-1",
                                "GroupId": "sg-b9988776655443322"
                            }
                        ],
                        ...
                    },
                    ...
                },
                ...
                {
                    ...
                }
            ],
            "OwnerId": "123456789012",
            "ReservationId": "r-aaaaaabbbbb111111"
        },
        ...
    ]
}
```

若要使用資源篩選條件選取特定可用區域中的執行個體，請指定 的屬性路徑`AvailabilityZone`和可用區域的程式碼做為 值。例如：

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

若要使用資源篩選條件選取特定子網路中的執行個體，請指定 的屬性路徑SubnetId和子網路 ID 做為值。例如：

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

若要選取處於特定執行個體狀態的執行個體，請指定 的屬性路徑Name，並將下列其中一個狀態名稱指定為值： pending | running | shutting-down | terminated | stopping || stopped。例如：

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

若要選取已連接多個安全群組的任何執行個體，請指定具有 GroupId和多個安全群組 IDs單一篩選條件。例如：

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

若要選取已連接所有數量安全群組的執行個體，請指定多個篩選條件，其中包含屬性路徑 GroupId和每個篩選條件的單一安全群組 ID。例如：

```
"filters": [
    {
        "path": "NetworkInterfaces.Groups.GroupId",
        "values": [
```

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

範例：Amazon RDS 叢集（資料庫叢集）

當您為支援 aws：rds：cluster 資源類型的動作指定篩選條件時， AWS FIS 會執行 Amazon RDS describe-db-clusters命令並套用篩選條件以識別目標。

describe-db-clusters 命令會針對每個資料庫叢集傳回類似下列的 JSON 輸出。以下是部分輸出，其中包含以##標示的欄位。我們將提供範例，這些範例使用這些欄位從 JSON 輸出的結構指定屬性路徑。

```
[
    {
        "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",
        ...
    }
]
```

若要套用資源篩選條件，只傳回使用特定資料庫引擎的資料庫叢集，請將屬性路徑指定為 `Engine`，並將值指定為 `aurora-postgresql`，如下列範例所示。

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

若要套用僅傳回特定可用區域中資料庫叢集的資源篩選條件，請指定屬性路徑和值，如下列範例所示。

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

AWS FIS 的停止條件

AWS Fault Injection Service (AWS FIS) 提供控制項和護欄，讓您安全地在 AWS 工作負載上執行實驗。停止條件是一種機制，可在實驗達到您定義為 Amazon CloudWatch 警示的閾值時停止實驗。如果在實驗期間觸發停止條件， AWS FIS 會停止實驗。您無法繼續停止的實驗。

若要建立停止條件，請先為您的應用程式或服務定義穩定狀態。穩定狀態是應用程式以最佳方式執行時，以業務或技術指標定義。例如，延遲、CPU 負載或重試次數。您可以使用穩定狀態來建立 CloudWatch 警示，如果您的應用程式或服務達到無法接受其效能的狀態，您可以使用該警示來停止實驗。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的使用 Amazon CloudWatch 警示。

您的帳戶對於您可以在實驗範本中指定的停止條件數量有配額。如需詳細資訊，請參閱Fault Injection Service AWS 的配額和限制。

停止條件語法

建立實驗範本時，您可以透過指定您建立的 CloudWatch 警示來指定一或多個停止條件。

```
{
    "stopConditions": [
```

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

下列範例指出實驗範本未指定停止條件。

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

進一步了解

如需示範如何建立 CloudWatch 警示並將停止條件新增至實驗範本的教學課程，請參閱 在執行個體上執行 CPU 應力。

如需 FIS 支援之資源類型可用的 CloudWatch AWS 指標的詳細資訊，請參閱以下內容：

- 使用 CloudWatch 監控您的執行個體
- Amazon ECS CloudWatch 指標
- 使用 CloudWatch 監控 Amazon RDS 指標
- 使用 CloudWatch 監控執行命令指標

AWS FIS 實驗的 IAM 角色

AWS Identity and Access Management (IAM) 是一種 AWS 服務，可協助管理員安全地控制對 AWS 資源的存取。若要使用 AWS FIS，您必須建立 IAM 角色來授予 AWS FIS 所需的許可，讓 AWS FIS 可以代表您執行實驗。您可以在建立實驗範本時指定此實驗角色。對於單一帳戶實驗，實驗角色的 IAM 政策必須授予許可，以修改您在實驗範本中指定為目標的資源。對於多帳戶實驗，實驗角色必須授予協調者角色許可，以擔任每個目標帳戶的 IAM 角色。如需詳細資訊，請參閱多帳戶實驗的許可。

我們建議您遵循授予最低權限的標準安全實務。您可以在政策中指定特定資源 ARNs或標籤來執行此操作。

為了協助您快速開始使用 AWS FIS，我們提供 AWS 受管政策，您可以在建立實驗角色時指定這些政策。或者，您也可以在建立自己的內嵌政策文件時，使用這些政策做為模型。

目錄

- 先決條件
- 選項 1：建立實驗角色並連接 AWS 受管政策
- 選項 2：建立實驗角色並新增內嵌政策文件

先決條件

開始之前，請安裝 AWS CLI 並建立所需的信任政策。

安裝 AWS CLI

開始之前，請先安裝並設定 AWS CLI。當您設定 時 AWS CLI，系統會提示您輸入 AWS 登入資料。本程序中的範例假設您也設定了預設「區域」。否則，請將 `--region` 選項新增至每個命令。如需詳細資訊，請參閱安裝或更新 AWS CLI 和設定 AWS CLI。

建立信任關係政策

實驗角色必須具有信任關係，允許 AWS FIS 服務擔任該角色。建立名為 的文字檔案，`fis-role-trust-policy.json`並新增下列信任關係政策。

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

建議您使用 `aws:SourceAccount` 和 `aws:SourceArn` 條件索引鍵，保護自己免受混淆代理人問題的困擾。來源帳戶是實驗的擁有者，而來源 ARN 是實驗的 ARN。例如，您應該將下列條件區塊新增至信任政策。

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

新增許可以擔任目標帳戶角色（僅限多帳戶實驗）

對於多帳戶實驗，您需要允許協調器帳戶擔任目標帳戶角色的許可。您可以修改下列範例，並將 新增為內嵌政策文件，以擔任目標帳戶角色：

```
{
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource":[
        "arn:aws:iam::target_account_id:role/role_name"
    ]
}
```

選項 1：建立實驗角色並連接 AWS 受管政策

使用 AWS FIS 的其中一個 AWS 受管政策快速入門。

建立實驗角色並連接 AWS 受管政策

1. 確認您的實驗中的 AWS FIS 動作有受管政策。否則，您將需要改為建立自己的內嵌政策文件。如需詳細資訊，請參閱the section called “AWS 受管政策”。
2. 使用下列 create-role 命令來建立角色，並新增您在先決條件中建立的信任政策。

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

3. 使用下列 attach-role-policy 命令來連接 AWS 受管政策。

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

其中 *fis-policy-arn* 為下列其中一項：

- 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

選項 2：建立實驗角色並新增內嵌政策文件

針對沒有受管政策的動作使用此選項，或僅包含特定實驗所需的許可。

建立實驗並新增內嵌政策文件

1. 使用下列 create-role 命令來建立角色，並新增您在先決條件中建立的信任政策。

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

2. 建立名為 的文字檔案，`fis-role-permissions-policy.json`並新增許可政策。如需可用作起點的範例，請參閱以下內容。

   - 錯誤注入動作 – 從下列政策開始。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowFISExperimentRoleFaultInjectionActions",
            "Effect": "Allow",
            "Action": [
                "fis:InjectApiInternalError",
                "fis:InjectApiThrottleError",
                "fis:InjectApiUnavailableError"
            ],
```

```
            "Resource": "arn:*:fis:*:*:experiment/*"
        }
    ]
}
```

- Amazon EBS 動作 – 從下列政策開始。

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

- Amazon EC2 動作 – 從 AWSFaultInjectionSimulatorEC2Access 政策開始。
- Amazon ECS 動作 – 從 AWSFaultInjectionSimulatorECSAccess 政策開始。
- Amazon EKS 動作 – 從 AWSFaultInjectionSimulatorEKSAccess 政策開始。
- 網路動作 – 從 AWSFaultInjectionSimulatorNetworkAccess 政策開始。
- Amazon RDS 動作 – 從 AWSFaultInjectionSimulatorRDSAccess 政策開始。
- Systems Manager 動作 – 從 AWSFaultInjectionSimulatorSSMAccess 政策開始。

3. 使用下列 put-role-policy 命令來新增您在上一個步驟中建立的許可政策。

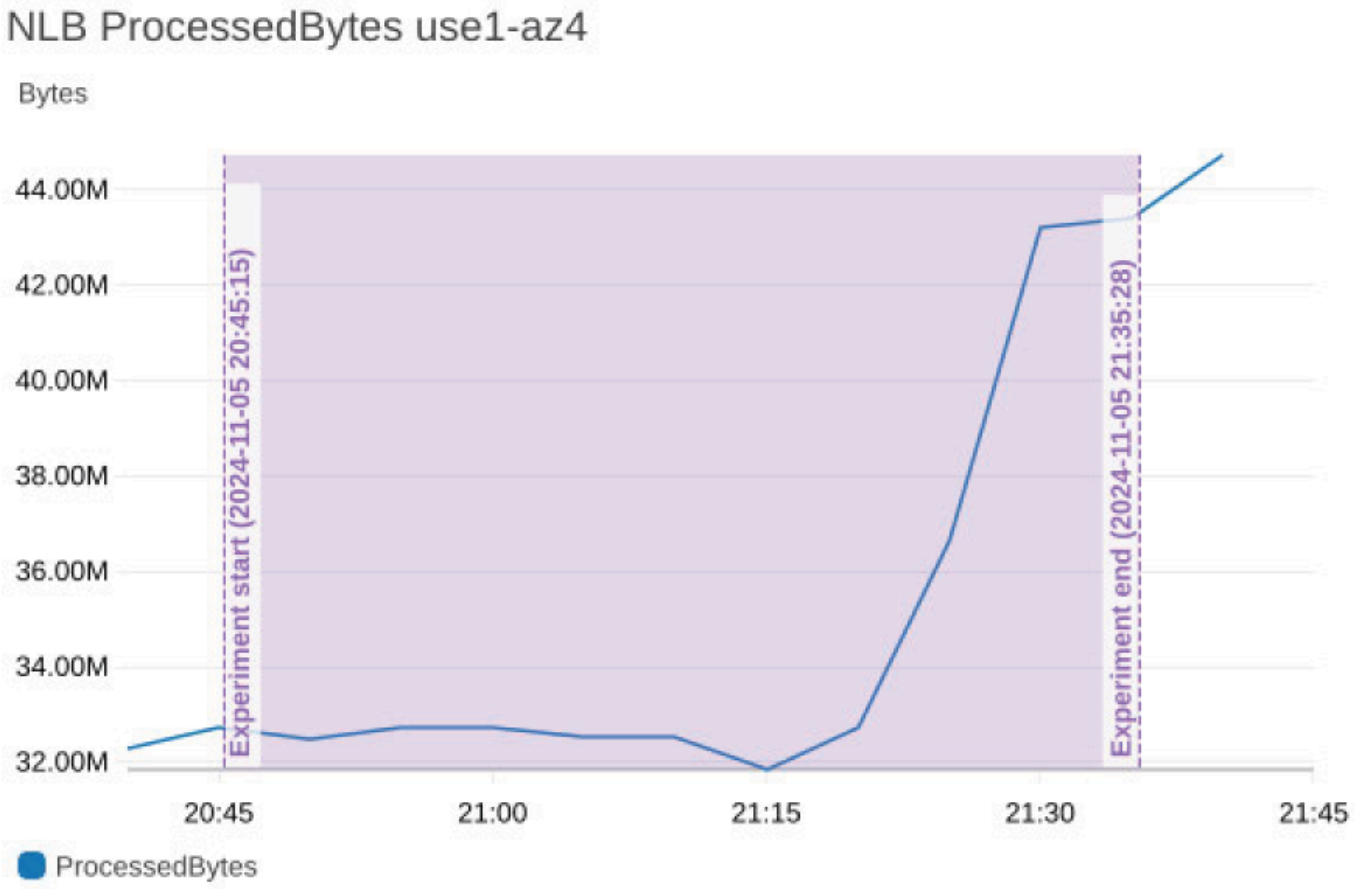
```
aws iam put-role-policy --role-name my-fis-role --policy-name my-fis-policy --
policy-document file://fis-role-permissions-policy.json
```

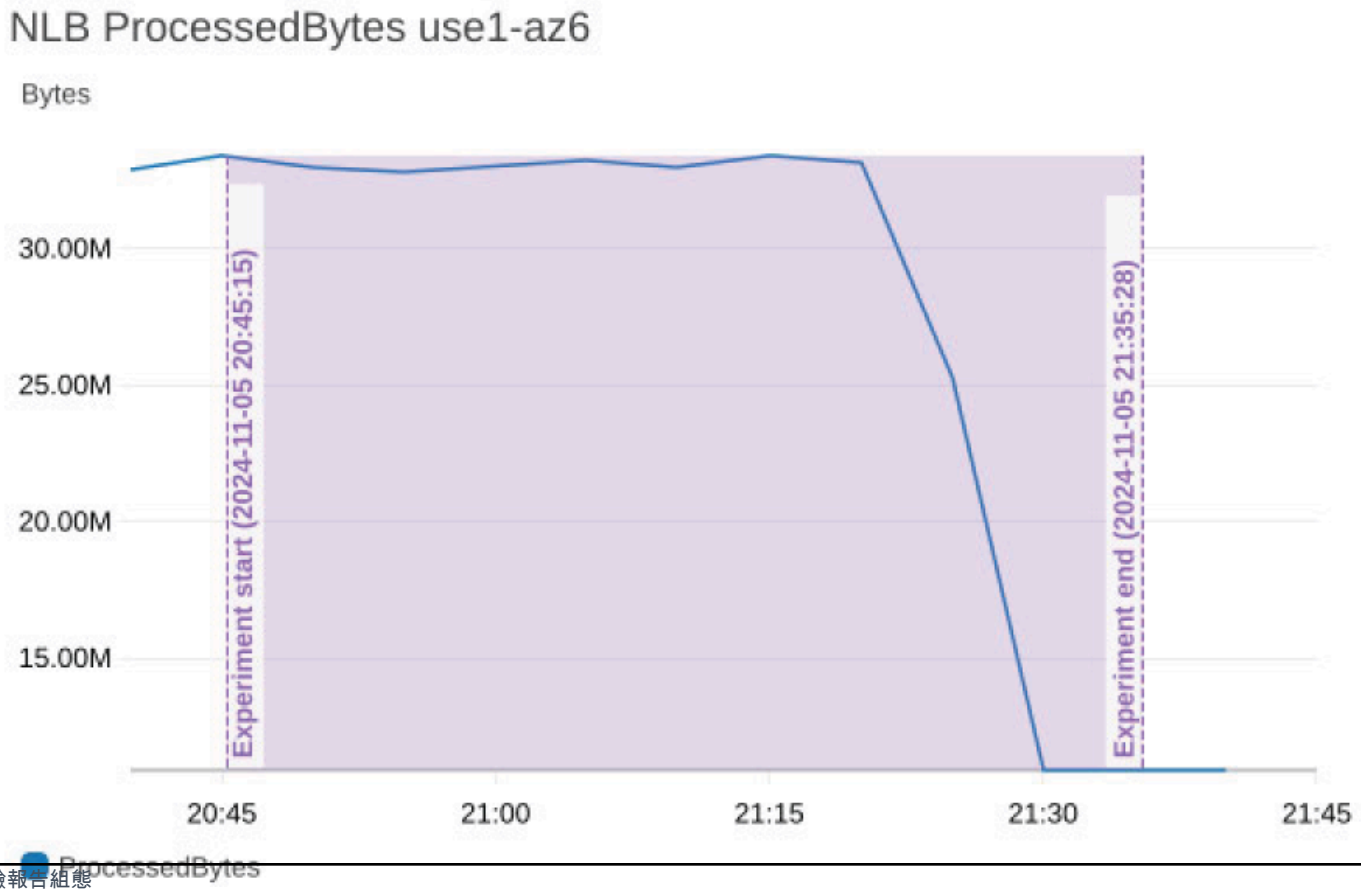
AWS FIS 的實驗報告組態

您可以啟用 AWS Fault Injection Service (FIS) 來產生實驗的報告，以便更輕鬆地產生彈性測試的證據。實驗報告是 PDF 文件，摘要實驗動作，並選擇性地從您指定的 CloudWatch 儀表板擷取應用程式回應。若要查看範例實驗報告，請在此處下載 zip 檔案。

若要啟用和設定為實驗產生的報告內容，您可以定義實驗範本的實驗報告組態。當您指定 CloudWatch 儀表板時， AWS FIS 會包含指定儀表板中所有小工具的快照圖形，並在您指定的持續時間內以實驗開始和結束時間標註，如以下範例所示。

此範例示範封包遺失實驗在可用區域 (AZ) 中的影響。在 AZ use1-az6 中引入封包遺失時，流量會從 use1-az6 轉移到 use1-az4，使得該 AZ 中負載平衡器處理的位元組數會減少。


NLB ProcessedBytes use1-az4
Bytes
44.00M
42.00M
40.00M
38.00M
36.00M
34.00M
32.00M
Experiment start (2024-11-05 20:45:15)
Experiment end (2024-11-05 21:35:28)
20:45
21:00
21:15
21:30
21:45
ProcessedBytes


NLB ProcessedBytes use1-az6
Bytes
30.00M
25.00M
20.00M
15.00M
Experiment start (2024-11-05 20:45:15)
Experiment end (2024-11-05 21:35:28)
20:45
21:00
21:15
21:30
21:45
ProcessedBytes

當實驗結束時，報告可以從 AWS FIS 主控台下載，並且也存放在 Amazon S3 儲存貯體中。如果您在報告組態中包含 CloudWatch 儀表板，也會傳送每個小工具的影像。對於做為目標預覽一部分`cancelled`或執行的實驗（將 actionsMode 設定為），不會產生報告`skip-all`。一旦實驗超過實驗資料保留限制，報告將只能從 Amazon S3 儲存貯體取得。除發生內部錯誤而失敗的報告外，每個交付的報告都會收取 AWS FIS 費用。如需詳細資訊，請參閱 AWS Fault Injection Service 定價和 Fault Injection Service AWS 的配額和限制。Amazon S3 和 CloudWatch API 費用的擷取和儲存費用適用於 GetMetricWidgetImage 和 GetDashboard 請求。如需詳細資訊，請參閱 Amazon S3 定價和 CloudWatch 定價。

目錄

- 實驗報告組態語法
- 實驗報告許可
- 實驗報告最佳實務

實驗報告組態語法

以下是實驗報告組態的語法，這是實驗範本的選用區段。

```
{
    "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`，您可以自訂輸出目的地、輸入資料和時間範圍，讓資料包含在實驗報告中，這可協助您更了解 AWS FIS 實驗的影響和結果。當您定義實驗報告組態時，請提供下列項目：

outputs

的 區段`experimentReportConfiguration`，指定實驗報告交付的位置。在 中`outputs`，您可以提供下列項目`s3Configuration`來指定 ：

- `bucketName` - 存放報告的 Amazon S3 儲存貯體名稱。儲存貯體必須與實驗位於相同的區域。
- `prefix`（選用）- Amazon S3 儲存貯體中的字首，用於存放報告。強烈建議使用此欄位，以便您僅限制對字首的存取。

dataSources

的選用區段`experimentReportConfiguration`，指定要包含在實驗報告中的其他資料來源。

- `cloudWatchDashboards` - 包含在報告中的 CloudWatch 儀表板陣列。僅限一個 CloudWatch 儀表板。
- `dashboardIdentifier`- CloudWatch 儀表板的 ARN。除了跨區域指標之外，報告中將包含具有此儀表板`metric`類型之每個小工具的快照圖形。

preExperimentDuration

的選用區段`experimentReportConfiguration`，定義 CloudWatch 儀表板指標要包含在報告中的實驗前持續時間，最長 30 分鐘。這應該是代表應用程式穩定狀態的期間。例如，實驗前持續時間為 5 分鐘，表示快照圖形會在實驗開始前 5 分鐘包含指標。持續時間的格式為 ISO 8601，預設值為 20 分鐘。

postExperimentDuration

的選用區段`experimentReportConfiguration`，定義 CloudWatch 儀表板指標要包含在報告中的試驗後持續時間，最長 2 小時。這應該是代表應用程式穩定狀態或復原期間的持續時間。例如，如果您指定 5 分鐘的試驗後持續時間，快照圖形將包含指標，直到實驗結束後 5 分鐘。持續時間的格式為 ISO 8601，預設值為 20 分鐘。

實驗報告許可

若要讓 AWS FIS 產生和存放實驗報告，您需要從 AWS FIS 實驗 IAM 角色允許下列操作：

- `cloudwatch:GetDashboard`

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

我們建議您遵循 AWS 安全最佳實務，並將實驗角色限制為儲存貯體和字首。以下是限制實驗角色存取的政策陳述式範例。

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

使用客戶受管金鑰 (CMK) 加密的 Amazon S3 儲存貯體交付報告的其他許可

如果您在 中指定的 Amazon S3 儲存貯體`S3Configuration`使用 CMK 加密，您需要將下列額外許可授予 KMS 金鑰政策上的 FIS 實驗角色：

- `kms:GenerateDataKey`
- `kms:Decrypt`

以下是允許 FIS 實驗角色將報告寫入加密儲存貯體的範例 KMS 金鑰政策陳述式：

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

實驗報告最佳實務

以下是使用 AWS FIS 實驗報告組態的最佳實務：

- 開始實驗之前，請產生目標預覽，以確認您的實驗範本已如預期般設定。目標預覽會為您提供有關預期實驗目標的資訊。如需進一步了解，請參閱 從實驗範本產生目標預覽。
- 報告不應用於故障診斷失敗的實驗。反之，請使用實驗日誌來疑難排解實驗錯誤。我們建議您僅對先前已執行且成功完成的實驗依賴報告。
- 限制實驗 IAM 角色放置，並取得對 S3 目的地儲存貯體和字首的物件存取權。我們建議您將儲存貯體/字首專用於 AWS FIS 實驗報告，並且不要授予其他服務 AWS 存取此儲存貯體和字首的權限。
- 使用 Amazon S3 物件鎖定，以防止報告在固定時間長度內或無限期遭到刪除或覆寫。若要進一步了解，請參閱使用物件鎖定鎖定物件。
- 如果您的 CloudWatch 儀表板位於相同區域內的個別帳戶，您可以使用 CloudWatch 跨帳戶可觀測性，讓 AWS FIS 協調器帳戶做為監控帳戶，並將個別帳戶做為來源帳戶，從 AWS CLI 和 API 中的 CloudWatch 主控台或可觀測性存取管理員命令。若要進一步了解，請參閱 CloudWatch 跨帳戶可觀測性。

的實驗選項 AWS FIS

實驗選項是實驗的選用設定。您可以在實驗範本上定義特定實驗選項。當您開始實驗時，會設定其他實驗選項。

以下是您在實驗範本上定義的實驗選項語法。

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

如果您在建立實驗範本時未指定任何實驗選項，則會使用每個選項的預設值。

以下是您在開始實驗時設定的實驗選項語法。

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

如果您在開始實驗時未指定任何實驗選項，則會`run-all`使用預設值。

目錄

- 帳戶目標
- 空目標解析度模式
- 動作模式

帳戶目標

如果您有多個 AWS 帳戶，其中包含您想要在實驗中鎖定的資源，您可以使用以實驗為目標的帳戶選項來定義多帳戶實驗。您可以從影響多個目標帳戶中資源的協調器帳戶執行多帳戶實驗。協調器帳戶擁有 AWS FIS 實驗範本和實驗。目標帳戶是具有資源的個別 AWS 帳戶，這些資源可能會受到 AWS FIS 實驗的影響。如需詳細資訊，請參閱使用 的多帳戶實驗 AWS FIS。

您可以使用帳戶目標來指出目標資源的位置。您可以為帳戶目標提供兩個值：

- 單一帳戶 – 預設。實驗只會以 AWS FIS 實驗執行之 AWS 帳戶中的資源為目標。
- 多帳戶 – 實驗可以鎖定多個 AWS 帳戶中的資源。

目標帳戶組態

若要執行多帳戶實驗，您必須定義一或多個目標帳戶組態。目標帳戶組態會指定每個帳戶的 accountId、 roleArn 和描述，其中包含實驗中目標為的資源。實驗範本的目標帳戶組態的帳戶 IDs 必須是唯一的。

當您建立多帳戶實驗範本時，實驗範本會傳回唯讀欄位 `targetAccountConfigurationsCount`，這是實驗範本所有目標帳戶組態的計數。

以下是目標帳戶組態的語法。

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

當您建立目標帳戶組態時，請提供下列項目：

accountId

目標帳戶的 12 位數 AWS 帳戶 ID。

roleArn

授予 AWS FIS 許可以在目標帳戶中採取動作的 IAM 角色。

description

選擇性的描述。

若要進一步了解如何使用目標帳戶組態，請參閱使用 的多帳戶實驗 AWS FIS。

空目標解析度模式

此模式可讓您選擇允許實驗完成，即使目標資源未解析。

- 失敗 – 預設。如果目標未解析任何資源，則實驗會立即終止，狀態為 `failed`。

- 略過 – 如果目標未解析任何資源，則實驗將繼續，並略過任何沒有解析目標的動作。具有使用唯一識別符定義之目標的動作，例如 ARNs，無法略過。如果找不到使用唯一識別符定義的目標，則實驗會立即終止，狀態為 `failed`

動作模式

動作模式是選用參數，您可以在開始實驗時指定。您可以將動作模式設定為 `skip-all`，以在將故障注入目標資源之前產生目標預覽。目標預覽可讓您驗證下列項目：

- 您已設定實驗範本以鎖定您預期的資源。當您開始此實驗時，目標的實際資源可能與預覽不同，因為資源可能會隨機移除、更新或取樣。
- 您的記錄組態已正確設定。
- 對於多帳戶實驗，您已為每個目標帳戶組態正確設定 IAM 角色。

> **Note**
>
> `skip-all` 模式不允許您驗證您是否具有執行 AWS FIS 實驗的必要許可，並對資源採取動作。

動作模式參數接受下列值：

- `run-all` - （預設） 實驗會對目標資源採取動作。
- `skip-all` - 實驗會略過目標資源上的所有動作。

若要進一步了解如何在開始實驗時設定動作模式參數，請參閱從實驗範本產生目標預覽。

AWS FIS 動作參考

動作是您使用 AWS Fault Injection Service (AWS FIS) 在目標上執行的錯誤注入活動。 為跨 AWS 服務的特定類型目標 AWS FIS 提供預先設定的動作。您可以將動作新增至實驗範本，然後用來執行實驗。

此參考說明 中的常見動作 AWS FIS，包括動作參數和所需 IAM 許可的相關資訊。您也可以使用 AWS FIS 主控台或來自 AWS Command Line Interface () 的 list-actions 命令列出支援 AWS FIS 的動作 AWS CLI。取得特定動作的名稱後，您可以使用 get-action 命令來檢視動作的詳細資訊。如需搭配使用 AWS FIS 命令的詳細資訊 AWS CLI，請參閱《 AWS CLI 命令參考》中的AWS Command Line Interface 使用者指南和 fis。

如需 AWS FIS 動作如何運作的詳細資訊，請參閱 AWS FIS 的動作和 Fault Injection Service AWS 如何與 IAM 搭配使用。

動作

- 錯誤注入動作
- 復原動作
- 等待動作
- Amazon CloudWatch 動作
- Amazon DynamoDB 動作
- Amazon EBS 動作
- Amazon EC2 動作
- Amazon ECS 動作
- Amazon EKS 動作
- Amazon ElastiCache 動作
- AWS Lambda 動作
- 網路動作
- Amazon RDS 動作
- Amazon S3 動作
- Systems Manager 動作
- 搭配 AWS FIS 使用 Systems Manager SSM 文件
- 使用 AWS FIS aws：ecs：task 動作

- 使用 AWS FIS aws：eks：pod 動作
- 使用 AWS FIS aws：lambda：function 動作

錯誤注入動作

AWS FIS 支援下列錯誤注入動作。

動作

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

aws:fis:inject-api-internal-error

將內部錯誤注入目標 IAM 角色提出的請求中。特定回應取決於每個服務和 API。如需詳細資訊，請檢閱 服務的 SDK 和 API 文件。

資源類型

- aws:iam:role

參數

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- service – 目標 AWS API 命名空間。支援的值為 `ec2`。
- percentage – 將錯誤注入其中的呼叫百分比 (1-100)。
- operations – 插入錯誤的操作，以逗號分隔。如需`ec2`命名空間的 API 動作清單，請參閱《Amazon EC2 API 參考》中的動作。

許可

- `fis:InjectApiInternalError`

aws:fis:inject-api-throttle-error

將限流錯誤注入目標 IAM 角色提出的請求。特定回應取決於每個服務和 API。如需詳細資訊，請檢閱服務的 SDK 和 API 文件。

資源類型

- aws:iam:role

參數

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- service – 目標 AWS API 命名空間。支援的值為 `ec2`。
- percentage – 將錯誤注入其中的呼叫百分比 (1-100)。
- operations – 插入錯誤的操作，以逗號分隔。如需`ec2`命名空間的 API 動作清單，請參閱《Amazon EC2 API 參考》中的動作。

許可

- `fis:InjectApiThrottleError`

aws:fis:inject-api-unavailable-error

將無法使用的錯誤注入目標 IAM 角色提出的請求中。特定回應取決於每個服務和 API。如需詳細資訊，請檢閱 服務的 SDK 和 API 文件。

資源類型

- aws:iam:role

參數

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- service – 目標 AWS API 命名空間。支援的值為 `ec2`。
- percentage – 將錯誤注入其中的呼叫百分比 (1-100)。

- operations – 插入錯誤的操作，以逗號分隔。如需 `ec2` 命名空間的 API 動作清單，請參閱《Amazon EC2 API 參考》中的動作。

許可

- `fis:InjectApiUnavailableError`

復原動作

執行復原動作是為了降低風險或在受損後保護應用程式。

AWS FIS 支援下列復原動作。

aws:arc:start-zonal-autoshift

自動將受支援資源的流量從可能受損的可用區域 (AZ) 轉移，並將其重新路由至相同 AWS 區域中運作狀態良好的 AZs。這允許透過 FIS 體驗區域自動轉移。區域自動轉移是 Amazon Application Recovery Controller (ARC) 的一項功能，允許 AWS 代表您將資源的流量移離 AZ，當 AWS 判斷存在可能影響 AZ 中客戶的損害時。

當您執行`aws:arc:start-zonal-autoshift`動作時， 會使用 StartZonalShift、UpdateZonalShift 和 CancelZonalShift APIs 來 AWS FIS 管理區域轉移，並將這些請求的 `expiresIn` 欄位設定為 1 分鐘做為安全機制。這可讓 在發生網路中斷或系統問題等意外事件時 AWS FIS ，快速復原區域轉移。在 ARC AWS FIS主控台中，過期時間欄位會顯示受管，而實際的預期過期取決於區域轉移動作中指定的持續時間。

資源類型

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

  區域轉移受管資源是可以啟用 ARC 區域自動轉移的資源類型，包括 Amazon EKS 叢集、Amazon EC2 應用程式和網路負載平衡器，以及 Amazon EC2 Auto Scaling 群組。如需詳細資訊，請參閱《ARC 開發人員指南》中的支援的資源和啟用區域自動轉移資源。

參數

- duration – 流量將轉移的時間長度。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

- availabilityZoneIdentifier – 流量從此可用區域移開。這可以是 AZ 名稱 (us-east-1a) 或 AZ ID (use1-az1)。
- managedResourceTypes – 流量將從中轉移的資源類型，以逗號分隔。可能的選項包括 `ASG`(Auto Scaling 群組）、 `ALB` (Application Load Balancer)、 `NLB`(Network Load Balancer) 和 `EKS`(Amazon EKS)。
- zonalAutoshiftStatus – 您要鎖定的資源`zonalAutoshiftStatus`狀態。可能的選項為 `ENABLED` `DISABLED`、 和 `ANY`。預設值為 `ENABLED`。

許可

- arc-zonal-shift：StartZonalShift
- arc-zonal-shift：GetManagedResource
- arc-zonal-shift：UpdateZonalShift
- arc-zonal-shift：CancelZonalShift
- arc-zonal-shift：ListManagedResources
- autoscaling：DescribeTags
- tag:GetResources

等待動作

AWS FIS 支援下列等待動作。

aws:fis:wait

執行 AWS FIS 等待動作。

參數

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

- 無

Amazon CloudWatch 動作

AWS FIS 支援下列 Amazon CloudWatch 動作。

aws:cloudwatch:assert-alarm-state

驗證指定的警示是否處於其中一個指定的警示狀態。

資源類型

- 無

參數

- alarmArns – 警示ARNs，以逗號分隔。您最多可以指定五個警示。
- alarmStates – 警示狀態，以逗號分隔。可能的警示狀態為 `OK`、 `ALARM`和 `INSUFFICIENT_DATA`。

許可

- `cloudwatch:DescribeAlarms`

Amazon DynamoDB 動作

AWS FIS 支援下列 Amazon DynamoDB 動作。

aws:dynamodb:global-table-pause-replication

暫停 Amazon DynamoDB 全域資料表複寫至任何複本資料表。在動作開始後，資料表最多可繼續複寫 5 分鐘。

下列陳述式將動態附加至目標 DynamoDB 全域資料表的政策：

```
{
   "Statement":[
      {
         "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxx",
         "Effect":"Deny",
         "Principal":{
```

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

下列陳述式將動態附加到目標 DynamoDB 全域資料表的串流政策：

```
{
   "Statement":[
      {
         "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxx",
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
         },
         "Action":[
            "dynamodb:GetRecords",
            "dynamodb:DescribeStream",
            "dynamodb:GetShardIterator"
         ],
         "Resource":"arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable/
stream/2023-08-31T09:50:24.025",
         "Condition": {
```

```
            "DateLessThan": {
            "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
         }
      }
   ]
}
```

如果目標資料表或串流沒有任何連接的資源政策，則會在實驗期間建立資源政策，並在實驗結束時自動刪除。否則，錯誤陳述式會插入現有政策中，而不會對現有政策陳述式進行任何額外的修改。然後，故障陳述式會在實驗結束時從政策中移除。

資源類型

- aws:dynamodb:global-table

參數

- duration – 在 AWS FIS API 中， 值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

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

Amazon EBS 動作

AWS FIS 支援下列 Amazon EBS 動作。

aws:ebs:pause-volume-io

在目標 EBS 磁碟區上暫停 I/O 操作。目標磁碟區必須位於相同的可用區域，且必須連接至建置在 Nitro 系統的執行個體。磁碟區無法連接至 Outpost 上的執行個體。

若要使用 Amazon EC2 主控台啟動實驗，請參閱《Amazon Amazon EC2使用者指南》中的 Amazon EBS 故障測試。

資源類型

- aws:ec2:ebs-volume

參數

- duration – 持續時間，從一秒到 12 小時。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘，PT5S 代表五秒，PT6H 代表六小時。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。如果持續時間很短，例如 PT5S，則 I/O 會在指定的持續時間內暫停，但由於初始化實驗所需的時間，實驗可能需要更長的時間才能完成。

許可

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

Amazon EC2 動作

AWS FIS 支援下列 Amazon EC2 動作。

動作

- 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 也支援透過 AWS Systems Manager SSM Agent 的錯誤注入動作。Systems Manager 使用 SSM 文件來定義要在 EC2 執行個體上執行的動作。您可以使用自己的文件來插入自訂錯誤，也可以使用預先設定的 SSM 文件。如需詳細資訊，請參閱the section called “SSM 文件動作”。

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

對目標 IAM 角色提出的請求注入`InsufficientInstanceCapacity`錯誤回應。支援的操作包括 RunInstances、CreateCapacityReservation、StartInstances、CreateFleet 呼叫。不支援在多個可用區域中包含容量查詢的請求。此動作不支援使用資源標籤、篩選條件或參數來定義目標。

資源類型

- aws:iam:role

參數

- duration – 在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- availabilityZoneIdentifiers – 可用區域的逗號分隔清單。支援區域 IDs（例如 `"use1-az1, use1-az2"`) 和區域名稱（例如 `"us-east-1a"`)。
- percentage – 將錯誤注入其中的呼叫百分比 (1-100)。

許可

- `ec2:InjectApiError`條件索引鍵`ec2:FisActionId`值設定為 `aws:ec2:api-insufficient-instance-capacity-error`，`ec2:FisTargetArns`條件索引鍵設定為目標 IAM 角色。

如需政策範例，請參閱 範例：使用 的條件索引鍵 `ec2:InjectApiError`。

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

對目標 Auto Scaling 群組提出的請求注入`InsufficientInstanceCapacity`錯誤回應。此動作僅支援使用啟動範本的 Auto Scaling 群組。若要進一步了解執行個體容量不足的錯誤，請參閱 Amazon EC2 使用者指南。

資源類型

- aws:ec2:autoscaling-group

參數

- duration – 在 AWS FIS API 中， 值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- availabilityZoneIdentifiers – 可用區域的逗號分隔清單。支援區域 IDs（例如 `"use1-az1, use1-az2"`) 和區域名稱（例如 `"us-east-1a"`)。
- percentage - 選用。注入錯誤的目標 Auto Scaling 群組啟動請求的百分比 (1-100)。預設為 100。

許可

- `ec2:InjectApiError`條件索引鍵 ec2：FisActionId 值設為 `aws:ec2:asg-insufficient-instance-capacity-error` ，`ec2:FisTargetArns`條件索引鍵設為目標 Auto Scaling 群組。
- `autoscaling:DescribeAutoScalingGroups`

如需政策範例，請參閱 範例：使用 的條件索引鍵 `ec2:InjectApiError`。

aws:ec2:reboot-instances

在目標 EC2 執行個體上執行 Amazon EC2 API 動作 RebootInstances。

資源類型

- aws:ec2:instance

參數

- 無

許可

- `ec2:RebootInstances`
- `ec2:DescribeInstances`

AWS 受管政策

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:send-spot-instance-interruptions

中斷目標 Spot 執行個體。在中斷 Spot 執行個體的兩分鐘前，傳送 Spot 執行個體中斷通知至目標執行個體。中斷時間取決於指定的 durationBeforeInterruption 參數。中斷時間兩分鐘後，Spot 執行個體會終止或停止，視其中斷行為而定。在您重新啟動前， AWS FIS 停止的 Spot 執行個體會保持在停止狀態。

動作啟動後，目標執行個體會立即收到 EC2 執行個體重新平衡建議。如果您指定 durationBeforeInterruption，重新平衡建議與中斷通知之間可能會有延遲。

如需詳細資訊，請參閱the section called “測試 Spot 執行個體中斷”。或者，若要使用 Amazon EC2 主控台啟動實驗，請參閱《Amazon EC2 使用者指南》中的啟動 Spot 執行個體中斷。

資源類型

- aws:ec2:spot-instance

參數

- durationBeforeInterruption – 中斷執行個體前的等待時間，從 2 分鐘到 15 分鐘。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT2M 代表兩分鐘。在 AWS FIS 主控台中，您可以輸入分鐘數。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:stop-instances

在目標 EC2 執行個體上執行 Amazon EC2 API 動作 StopInstances。

資源類型

- aws:ec2:instance

參數

- startInstancesAfterDuration - 選用。啟動執行個體前的等待時間，從一分鐘到 12 小時。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。如果執行個體具有加密的 EBS 磁碟區，您必須將 AWS FIS 許可授予用於加密磁碟區的 KMS 金鑰，或將實驗角色新增至 KMS 金鑰政策。
- completeIfInstancesTerminated - 選用。如果為 true，且如果 `startInstancesAfterDuration` 也是 true，則當 FIS 外部的個別請求終止目標 EC2 執行個體且無法重新啟動時，此動作將不會失敗。例如，在此動作完成之前，Auto Scaling 群組可以在其控制下終止已停止的 EC2 執行個體。預設值為 false。

許可

- `ec2:StopInstances`
- `ec2:StartInstances`
- `ec2:DescribeInstances` - 選用。使用 completeIfInstancesTerminated 時為必要，以在動作結束時驗證執行個體狀態。
- `kms:CreateGrant` - 選用。使用 startInstancesAfterDuration 重新啟動具有加密磁碟區的執行個體時需要。

AWS 受管政策

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:terminate-instances

在目標 EC2 執行個體上執行 Amazon EC2 API 動作 TerminateInstances。

資源類型

- aws:ec2:instance

參數

- 無

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEC2Access

Amazon ECS 動作

AWS FIS 支援下列 Amazon ECS 動作。

動作

- 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

執行 Amazon ECS API 動作 UpdateContainerInstancesState，以耗盡目標叢集上基礎 Amazon EC2 執行個體的指定百分比。

資源類型

- aws:ecs:cluster

參數

- drainagePercentage – 百分比 (1-100)。

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:stop-task

執行 Amazon ECS API 動作 StopTask 以停止目標任務。

資源類型

- aws:ecs:task

參數

- 無

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:task-cpu-stress

對目標任務執行 CPU 應力。使用 AWSFIS-Run-CPU-Stress SSM 文件。任務必須由 管理 AWS Systems Manager。如需詳細資訊，請參閱ECS 任務動作。

資源類型

- aws:ecs:task

參數

- duration – 壓力測試的持續時間，採用 ISO 8601 格式。
- percent - 選用。目標負載百分比，從 0 （無負載） 到 100 （完全負載）。預設為 100。
- workers - 選用。要使用的應力器數量。預設值為 0，它使用所有壓力源。
- installDependencies - 選用。如果此值為 `True`，Systems Manager 會在尚未安裝 SSM 代理程式的附屬容器上安裝必要的相依性。預設值為 `True`。相依性為 stress-ng。

許可

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

aws:ecs:task-io-stress

對目標任務執行 I/O 壓力。使用 AWSFIS-Run-IO-Stress SSM 文件。任務必須由 管理 AWS Systems Manager。如需詳細資訊，請參閱ECS 任務動作。

資源類型

- aws:ecs:task

參數

- duration – 壓力測試的持續時間，採用 ISO 8601 格式。
- percent - 選用。檔案系統上要在壓力測試期間使用的可用空間百分比。預設值為 80%。

- workers - 選用。工作程序數量。工作者會執行循序、隨機和記憶體映射的讀取/寫入操作、強制同步和快取捨棄的混合。多個子程序會對相同的檔案執行不同的 I/O 操作。預設為 1。
- installDependencies - 選用。如果此值為 `True`，Systems Manager 會在尚未安裝 SSM 代理程式的附屬容器上安裝必要的相依性。預設值為 `True`。相依性為 stress-ng。

許可

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

aws:ecs:task-kill-process

使用 killall命令停止任務中指定的程序。使用 AWSFIS-Run-Kill-Process SSM 文件。任務定義必須`pidMode`設定為 `task`。任務必須由 管理 AWS Systems Manager。如需詳細資訊，請參閱ECS 任務動作。

資源類型

- aws:ecs:task

參數

- processName – 要停止的程序名稱。
- signal - 選用。隨 命令一起傳送的訊號。可能的值為 `SIGTERM`（接收者可以選擇忽略） 和 `SIGKILL`（無法忽略）。預設值為 `SIGTERM`。
- installDependencies – 選用。如果此值為 `True`，Systems Manager 會在尚未安裝 SSM 代理程式的附屬容器上安裝必要的相依性。預設值為 `True`。相依性為 killall。

許可

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

aws:ecs:task-network-blackhole-port

使用 Amazon ECS Fault Injection 端點，捨棄指定通訊協定和連接埠的傳入或傳出流量。使用 AWSFIS-Run-Network-Blackhole-Port-ECS SSM 文件。任務定義必須`pidMode`設定為 `task`。任務必須由 管理 AWS Systems Manager。您無法在任務定義`bridge`中`networkMode`將 設定為 。如需詳細資訊，請參閱ECS 任務動作。

當 `useEcsFaultInjectionEndpoints` 設定為 時`false`，故障會使用 `iptables`工具，並使用 AWSFIS-Run-Network-Blackhole-Port SSM 文件。

資源類型

- aws:ecs:task

參數

- duration – 測試持續時間，採用 ISO 8601 格式。
- port – 連接埠號碼。
- trafficType – 流量的類型。可能的值為 `ingress` 和 `egress`。
- protocol - 選用。通訊協定。可能的值為 `tcp` 和 udp。預設值為 `tcp`。
- installDependencies – 選用。如果此值為 `True`，Systems Manager 會在尚未安裝 SSM 代理程式的附屬容器上安裝必要的相依性。預設值為 `True`。相依性為 atd、 curl-minimaldig和 jq。
- useEcsFaultInjectionEndpoints - 選用。如果設定為 true，則會使用 Amazon ECS 故障注入 APIs。預設值為 false。

許可

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

aws:ecs:task-network-latency

使用 Amazon ECS 故障注入端點 ，將輸出流量的延遲和抖動新增至網路界面，以傳送至特定來源。使用 AWSFIS-Run-Network-Latency-ECS SSM 文件。任務定義必須`pidMode`設定為 `task`。任務必須

由 管理 AWS Systems Manager。您無法在任務定義`bridge`中`networkMode`將 設定為 。如需詳細資訊，請參閱ECS 任務動作。

當 `useEcsFaultInjectionEndpoints` 設定為 時`false`，故障會使用 `tc`工具，並使用 AWSFIS-Run-Network-Latency-Sources SSM 文件。

資源類型

- aws:ecs:task

參數

- duration – 測試持續時間，採用 ISO 8601 格式。
- delayMilliseconds - 選用。延遲，以毫秒為單位。預設值為 200。
- jitterMilliseconds - 選用。抖動，以毫秒為單位。預設為 10。
- sources - 選用。來源，以逗號分隔，不含空格。可能的值為：IPv4 地址、IPv4 CIDR 區塊、網域名稱、 `DYNAMODB`和 `S3`。如果您指定 `DYNAMODB`或 `S3`，這僅適用於目前區域中的區域端點。預設值為 0.0.0.0/0，符合所有 IPv4 流量。
- installDependencies - 選用。如果此值為 `True`，Systems Manager 會在尚未安裝 SSM 代理程式的附屬容器上安裝必要的相依性。預設值為 `True`。相依性為 atd、dig、 curl-minimaljq和 lsof。
- useEcsFaultInjectionEndpoints - 選用。如果設定為 true，則會使用 Amazon ECS 故障注入 APIs。預設值為 false。

許可

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

aws:ecs:task-network-packet-loss

使用 Amazon ECS 故障注入端點，將封包遺失新增至網路界面，以將流量輸出至特定來源。使用 AWSFIS-Run-Network-Packet-Loss-ECS SSM 文件。任務定義必須`pidMode`設定為 `task`。任務必須由 管理 AWS Systems Manager。您無法在任務定義`bridge`中`networkMode`將 設定為 。如需詳細資訊，請參閱ECS 任務動作。

當 `useEcsFaultInjectionEndpoints` 設定為 時`false`，故障會使用 `tc`工具，並使用 AWSFIS-Run-Network-Packet-Loss-Sources SSM 文件。

資源類型

- aws:ecs:task

參數

- duration – 測試持續時間，採用 ISO 8601 格式。
- lossPercent - 選用。封包遺失的百分比。預設值為 7%。
- sources - 選用。來源，以逗號分隔，不含空格。可能的值為：IPv4 地址、IPv4 CIDR 區塊、網域名稱、 `DYNAMODB`和 `S3`。如果您指定 `DYNAMODB`或 `S3`，這僅適用於目前區域中的區域端點。預設值為 0.0.0.0/0，符合所有 IPv4 流量。
- installDependencies - 選用。如果此值為 `True`，Systems Manager 會在尚未安裝 SSM 代理程式的附屬容器上安裝必要的相依性。預設值為 `True`。相依性為 atd、dig、 curl-minimaljq和 lsof。
- useEcsFaultInjectionEndpoints - 選用。如果設定為 true，則會使用 Amazon ECS 故障注入 APIs。預設值為 false。

許可

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

Amazon EKS 動作

AWS FIS 支援下列 Amazon EKS 動作。

動作

- 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

在單一目標叢集上執行 ChaosMesh 或 Litmus 實驗。您必須在目標叢集上安裝 ChaosMesh 或 Litmus。

當您建立實驗範本並定義類型 的目標時`aws:eks:cluster`，您必須將此動作設為單一 Amazon Resource Name (ARN)。此動作不支援使用資源標籤、篩選條件或參數來定義目標。

安裝 ChaosMesh 時，您必須指定適當的容器執行時間。從 Amazon EKS 1.23 版開始，預設執行時間從 Docker 變更為 containerd。從 1.24 版開始，Docker 已移除。

資源類型

- aws:eks:cluster

參數

- kubernetesApiVersion – Kubernetes 自訂資源的 API 版本。可能的值為 `chaos-mesh.org/v1alpha1` | `litmuschaos.io/v1alpha1`。
- kubernetesKind – Kubernetes 自訂資源類型。值取決於 API 版本。
  - `chaos-mesh.org/v1alpha1` – 可能的值為 `AWSChaos` | `DNSChaos` | `GCPChaos` | `HTTPChaos` | `IOChaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodIOChaos` | `PodNetworkChaos` | `Schedule` | | `StressChaos` | | `TimeChaos` |
  - `litmuschaos.io/v1alpha1` – 可能的值為 `ChaosEngine`。
- kubernetesNamespace – Kubernetes 命名空間。
- kubernetesSpec – Kubernetes 自訂資源的 `spec`區段，採用 JSON 格式。
- maxDuration – 自動化執行允許的完成時間上限，從一分鐘到 12 小時。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

此動作不需要 AWS Identity and Access Management (IAM) 許可。使用此動作所需的許可是由 Kubernetes 使用 RBAC 授權來控制。如需詳細資訊，請參閱官方 Kubernetes 文件中的使用 RBAC 授權。如需 Chaos Mesh 的詳細資訊，請參閱官方的 Chaos Mesh 文件。如需 Litmus 的詳細資訊，請參閱官方 Litmus 文件。

aws:eks:pod-cpu-stress

在目標 Pod 上執行 CPU 壓力。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- duration – 壓力測試的持續時間，採用 ISO 8601 格式。
- percent - 選用。目標負載百分比，從 0 （無負載） 到 100 （完全負載）。預設為 100。
- workers - 選用。要使用的應力器數量。預設值為 0，它使用所有壓力源。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。
- fisPodSecurityPolicy - 選用。用於 FIS 和暫時性容器所建立之故障協調 Pod 的 Kubernetes 安全標準政策。可能的值為 `privileged`、 `baseline`和 `restricted`。此動作與所有政策層級相容。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-delete

刪除目標 Pod。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- gracePeriodSeconds - 選用。等待 Pod 正常終止的持續時間，以秒為單位。如果值為 0，我們會立即執行動作。如果值為 nil，我們使用 Pod 的預設寬限期。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。
- fisPodSecurityPolicy - 選用。用於 FIS 和暫時性容器所建立之故障協調 Pod 的 Kubernetes 安全標準政策。可能的值為 `privileged`、 `baseline`和 `restricted`。此動作與所有政策層級相容。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-io-stress

在目標 Pod 上執行 I/O 壓力。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- duration – 壓力測試的持續時間，採用 ISO 8601 格式。
- workers - 選用。工作程序數量。工作者會執行循序、隨機和記憶體映射的讀取/寫入操作、強制同步和快取捨棄的混合。多個子程序會在同一檔案上執行不同的 I/O 操作。預設為 1。
- percent - 選用。檔案系統上要在壓力測試期間使用的可用空間百分比。預設值為 80%。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。
- fisPodSecurityPolicy - 選用。用於 FIS 和暫時性容器所建立之故障協同運作 Pod 的 Kubernetes 安全標準政策。可能的值為 `privileged`、 `baseline`和 `restricted`。此動作與所有政策層級相容。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-memory-stress

在目標 Pod 上執行記憶體壓力。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- duration – 壓力測試的持續時間，採用 ISO 8601 格式。
- workers - 選用。要使用的應力器數量。預設為 1。
- percent - 選用。壓力測試期間要使用的虛擬記憶體百分比。預設值為 80%。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。
- fisPodSecurityPolicy - 選用。用於 FIS 和暫時性容器所建立之故障協調 Pod 的 Kubernetes 安全標準政策。可能的值為 `privileged`、 `baseline`和 `restricted`。此動作與所有政策層級相容。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-blackhole-port

捨棄指定通訊協定和連接埠的傳入或傳出流量。僅與 Kubernetes 安全標準`privileged`政策相容。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- duration – 測試持續時間，採用 ISO 8601 格式。
- protocol – 通訊協定。可能的值為 `tcp` 和 `udp`。
- trafficType – 流量的類型。可能的值為 `ingress` 和 `egress`。
- port – 連接埠號碼。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-latency

針對進出特定來源的流量，使用 tc工具將延遲和抖動新增至網路界面。僅與 Kubernetes 安全標準`privileged`政策相容。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- duration – 測試持續時間，採用 ISO 8601 格式。
- interface - 選用。網路介面。預設值為 `eth0`。
- delayMilliseconds – 選用。延遲，以毫秒為單位。預設值為 200。
- jitterMilliseconds - 選用。抖動，以毫秒為單位。預設為 10。
- sources - 選用。來源，以逗號分隔，不含空格。可能的值為：IPv4 地址、IPv4 CIDR 區塊、網域名稱、 `DYNAMODB`和 `S3`。如果您指定 `DYNAMODB`或 `S3`，這僅適用於目前區域中的區域端點。預設值為 0.0.0.0/0，符合所有 IPv4 流量。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-packet-loss

使用 tc工具將封包遺失新增至網路介面。僅與 Kubernetes 安全標準`privileged`政策相容。如需詳細資訊，請參閱EKS Pod 動作。

資源類型

- aws:eks:pod

參數

- duration – 測試持續時間，採用 ISO 8601 格式。
- interface - 選用。網路介面。預設值為 `eth0`。
- lossPercent – 選用。封包遺失的百分比。預設值為 7%。
- sources - 選用。來源，以逗號分隔，不含空格。可能的值為：IPv4 地址、IPv4 CIDR 區塊、網域名稱、 `DYNAMODB`和 `S3`。如果您指定 `DYNAMODB`或 `S3`，這僅適用於目前區域中的區域端點。預設值為 0.0.0.0/0，符合所有 IPv4 流量。
- kubernetesServiceAccount – Kubernetes 服務帳戶。如需必要許可的詳細資訊，請參閱the section called “設定 Kubernetes 服務帳戶”。
- fisPodContainerImage - 選用。用來建立故障注入器 Pod 的容器映像。預設為使用 提供的映像 AWS FIS。如需詳細資訊，請參閱the section called “Pod 容器映像”。
- maxErrorsPercent – 選用。在故障注入失敗之前，可能失敗的目標百分比。預設值為 0。
- fisPodLabels - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 標籤。
- fisPodAnnotations - 選用。連接至 FIS 建立之錯誤協同運作 Pod 的 Kubernetes 註釋。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:terminate-nodegroup-instances

在目標節點群組上執行 Amazon EC2 API 動作 TerminateInstances。

資源類型

- aws:eks:nodegroup

參數

- instanceTerminationPercentage – 要終止的執行個體百分比 (1-100)。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEKSAccess

Amazon ElastiCache 動作

AWS FIS 支援下列 ElastiCache 動作。

aws:elasticache:replicationgroup-interrupt-az-power

針對已啟用異地同步備份的目標 ElastiCache 複寫群組，中斷指定可用區域中節點的電源。每個複寫群組一次只能影響一個可用區域。當主要節點成為目標時，具有最小複寫延遲的對應僅供讀取複本會提升為主要節點。在此動作期間，指定可用區域中的僅供讀取複本替換會遭到封鎖，這表示目標複寫群組的運作容量會降低。此動作的目標同時支援 Redis 和 Valkey 引擎。動作不支援「無伺服器」部署選項。

資源類型

- aws:elasticache:replicationgroup

參數

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中， 值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

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

> **Note**
>
> ElastiCache 中斷 AZ 電源動作現在支援所有複寫群組類型，包括 Valkey 和 Redis。為了更好地代表此功能，已重新命名 動作。如果您目前正在使用 `aws:elasticache:interrupt-cluster-az-power`，我們建議您遷移至新動作`aws:elasticache:replicationgroup-interrupt-az-power`，以利用最新的功能。

AWS Lambda 動作

AWS Lambda 支援下列 Lambda 動作

動作

- aws:lambda:invocation-add-delay
- aws:lambda:invocation-error
- aws:lambda:invocation-http-integration-response

aws:lambda:invocation-add-delay

在您指定的幾毫秒內延遲啟動函數。此動作的效果類似於 Lambda 冷啟動，但額外的時間會花費在計費持續時間中，並套用至所有執行環境，而不是只影響新的執行環境。這表示您可能會同時遇到 Lambda 冷啟動和此延遲。透過將延遲值設定為高於 Lambda 函數上設定的逾時，此動作也會提供高保真度逾時事件的存取權。

資源類型

- aws:lambda:function

參數

- duration – 動作持續的時間長度。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- invocationPercentage – 選用。要注入故障的函數叫用百分比 (1-100)。預設為 100。
- startupDelayMilliseconds – 選用。在叫用和執行函數程式碼之間等待的時間，以毫秒 (0-900，000) 為單位。預設為 1000。

許可

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

aws:lambda:invocation-error

將 Lambda 函數叫用標示為失敗。此動作適用於測試錯誤處理機制，例如警示和重試組態。使用此動作時，您可以選擇是否要在傳回錯誤之前執行函數程式碼。

資源類型

- aws:lambda:function

參數

- duration – 動作持續的時間長度。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- invocationPercentage – 選用。要注入故障的函數叫用百分比 (1-100)。預設為 100。
- preventExecution – 如果值為 true，則動作會傳回錯誤而不執行函數。

許可

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

aws:lambda:invocation-http-integration-response

修改 函數的行為。您可以選擇內容類型和 HTTP 回應程式碼，以支援與 ALB、API-GW 和 VPC Lattice 的整合。若要啟用選擇性地影響上游或下游整合，您可以選擇是否直接傳回修改後的回應，還是在函數完成執行後執行函數並取代回應。

資源類型

- aws:lambda:function

參數

- contentTypeHeader – 從 Lambda 函數傳回的 HTTP 內容類型標頭字串值。
- duration – 動作持續的時間長度。在 AWS FIS API 中，該值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- invocationPercentage – 選用。要注入故障的函數叫用百分比 (1-100)。預設為 100。
- preventExecution – 如果值為 true，動作將傳回回應而不執行函數。
- statusCode – 從 Lambda 函數傳回的 HTTP 狀態碼 (000-999) 值。

許可

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

網路動作

AWS FIS 支援下列網路動作。

動作

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

aws:network:disrupt-connectivity

透過暫時複製與目標子網路相關聯的原始網路存取控制清單（網路 ACL)，拒絕至目標子網路的指定流量。FIS 會將拒絕規則新增至複製的網路 ACL，其具有標記 managedbyFIS=true，並在動作期間將其與子網路建立關聯。動作完成時，FIS 會刪除複製的網路 ACL，並還原原始網路 ACL 關聯。

資源類型

- aws:ec2:subnet

參數

- scope – 要拒絕的流量類型。當範圍不是 時`all`，網路 ACLs中的項目數目上限為 20。可能值如下：
  - `all` – 拒絕所有進出子網路的流量。請注意，此選項允許子網路內流量，包括往返子網路中網路介面的流量。
  - `availability-zone` – 拒絕在其他可用區域中往返子網路的 VPC 內流量。VPC 中可設為目標的子網路數目上限為 30。
  - `dynamodb` – 拒絕流量往返目前區域中 DynamoDB 的區域端點。
  - `prefix-list` – 拒絕往返指定字首清單的流量。
  - `s3` – 拒絕往返目前區域中 Amazon S3 區域端點的流量。
  - `vpc` – 拒絕進出 VPC 的流量。
- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中， 值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- prefixListIdentifier – 如果範圍為 `prefix-list`，則這是客戶受管字首清單的識別符。您可以指定名稱、ID 或 ARN。字首清單最多可以有 10 個項目。

許可

- `ec2:CreateNetworkAcl` – 使用標記 managedByFIS=true 建立網路 ACL。
- `ec2:CreateNetworkAclEntry` – 網路 ACL 必須具有標記 managedByFIS=true。
- `ec2:CreateTags`
- `ec2:DeleteNetworkAcl` – 網路 ACL 必須具有標記 managedByFIS=true。
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:GetManagedPrefixListEntries`
- `ec2:ReplaceNetworkAclAssociation`

AWS 受管政策

- AWSFaultInjectionSimulatorNetworkAccess

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

封鎖源自目標子網路且目的地為指定區域的流量。建立路由表，其中包含要隔離的區域的所有路由。若要允許 FIS 建立這些路由表，請將 的 Amazon VPC 配額提高`routes per route table`到 250 加上現有路由表中的路由數量。

資源類型

- aws:ec2:subnet

參數

- `region` – 要隔離的區域程式碼（例如 eu-west-1)。
- `duration` – 動作持續的時間長度。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

- `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`

† 使用標籤 的範圍managedByFIS=true。您不需要管理此標籤。在實驗期間 AWS FIS 新增和移除此標籤。

AWS 受管政策

- AWSFaultInjectionSimulatorNetworkAccess

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

封鎖來自目的地為指定區域之目標傳輸閘道對等互連附件的流量。

資源類型

- aws:ec2:transit-gateway

參數

- `region` – 要隔離的區域程式碼（例如 eu-west-1)。
- `duration` – 動作持續的時間長度。在 AWS FIS API 中， 值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorNetworkAccess

Amazon RDS 動作

AWS FIS 支援下列 Amazon RDS 動作。

動作

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

aws:rds:failover-db-cluster

在目標 Aurora 資料庫叢集上執行 Amazon RDS API 動作 FailoverDBCluster。

資源類型

- aws:rds:cluster

參數

- 無

許可

- `rds:FailoverDBCluster`
- `rds:DescribeDBClusters`
- `tag:GetResources`

AWS 受管政策

- AWSFaultInjectionSimulatorRDSAccess

aws:rds:reboot-db-instances

在目標資料庫執行個體上執行 Amazon RDS API 動作 RebootDBInstance。

資源類型

- aws:rds:db

參數

- forceFailover - 選用。如果值為 true，且執行個體為異地同步備份，則 會強制從一個可用區域容錯移轉到另一個可用區域。預設值為 false。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorRDSAccess

Amazon S3 動作

AWS FIS 支援下列 Amazon S3 動作。

動作

- aws:s3:bucket-pause-replication

aws:s3:bucket-pause-replication

暫停從目標來源儲存貯體到目的地儲存貯體的複寫。目的地儲存貯體可以位於不同的 AWS 區域，也可以位於與來源儲存貯體相同的區域內。在動作開始後，現有物件最多可繼續複寫一小時。此動作僅支援以標籤為目標。若要進一步了解 Amazon S3 複寫，請參閱 Amazon S3 使用者指南。

資源類型

- aws:s3:bucket

參數

- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。
- region – 目的地儲存貯體所在的 AWS 區域。
- destinationBuckets - 選用。以逗號分隔的目的地 S3 儲存貯體清單 (S3 儲存貯體）。
- prefixes - 選用。以逗號分隔的 S3 物件金鑰字首清單與複寫規則篩選條件。目標儲存貯體的複寫規則（根據字首的篩選條件）將會暫停。

許可

- `S3:PutReplicationConfiguration` 條件索引鍵`S3:IsReplicationPauseRequest`設定為`True`
- `S3:GetReplicationConfiguration` 條件索引鍵`S3:IsReplicationPauseRequest`設定為`True`

- `S3:PauseReplication`
- `S3:ListAllMyBuckets`
- `tag:GetResources`

如需政策範例，請參閱 範例：使用 的條件索引鍵 `aws:s3:bucket-pause-replication`。

Systems Manager 動作

AWS FIS 支援下列 Systems Manager 動作。

動作

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

aws:ssm:send-command

在目標 EC2 執行個體上執行 Systems Manager API 動作 SendCommand。Systems Manager 文件 (SSM 文件） 定義 Systems Manager 在執行個體上執行的動作。如需詳細資訊，請參閱使用 aws:ssm:send-command動作。

資源類型

- aws:ec2:instance

參數

- documentArn – 文件的 Amazon Resource Name (ARN)。在 主控台中，如果您從動作類型中選擇對應於其中一個預先設定 AWS FIS SSM 文件的值，則會為您完成此參數。
- documentVersion - 選用。文件的版本。如果為空，則預設版本會執行。
- documentParameters – 條件式。文件接受的必要和選用參數。格式是 JSON 物件，其金鑰為字串，值為字串或字串陣列。
- duration – 持續時間，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

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

AWS 受管政策

- AWSFaultInjectionSimulatorEC2Access

aws:ssm:start-automation-execution

執行 Systems Manager API 動作 StartAutomationExecution。

資源類型

- 無

參數

- documentArn – 自動化文件的 Amazon Resource Name (ARN)。
- documentVersion - 選用。文件的版本。如果為空，則預設版本會執行。
- documentParameters – 條件式。文件接受的必要和選用參數。格式是 JSON 物件，其金鑰為字串，值為字串或字串陣列。
- maxDuration – 自動化執行允許的完成時間上限，從一分鐘到 12 小時。在 AWS FIS API 中，值是 ISO 8601 格式的字串。例如，PT1M 代表一分鐘。在 AWS FIS 主控台中，您可以輸入秒數、分鐘數或小時數。

許可

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `iam:PassRole` - 選用。如果自動化文件擔任角色，則為必要。

AWS 受管政策

- AWSFaultInjectionSimulatorSSMAccess

搭配 AWS FIS 使用 Systems Manager SSM 文件

AWS FIS 透過 AWS Systems Manager SSM Agent 和 AWS FIS 動作 支援自訂錯誤類型aws:ssm:send-command。可用於建立常見錯誤注入動作的預先設定 Systems Manager SSM 文件 (SSM 文件），可作為開頭為 AWSFIS- 字首的公有 AWS 文件。

SSM Agent 是可在 Amazon EC2 執行個體、內部部署伺服器或虛擬機器 (VM) 上安裝和設定的 Amazon 軟體。 VMs 這可讓 Systems Manager 管理這些資源。代理程式會處理 Systems Manager 的請求，然後依照請求中的指定執行它們。您可以包含自己的 SSM 文件來插入自訂錯誤，或參考其中一個公有 Amazon 擁有的文件。

要求

對於需要 SSM Agent 在目標上執行動作的動作，您必須確保下列事項：

- 代理程式安裝在目標上。根據預設，某些 Amazon Machine Image (AMIs) 會安裝 SSM Agent。否則，您可以在執行個體上安裝 SSM 代理程式。如需詳細資訊，請參閱AWS Systems Manager 《 使用者指南》中的手動安裝 EC2 執行個體的 SSM Agent。
- Systems Manager 具有在您的執行個體上執行動作的許可。您可以使用 IAM 執行個體描述檔授予存取權。如需詳細資訊，請參閱《 使用者指南》中的建立 Systems Manager 的 IAM 執行個體描述檔和將 IAM 執行個體描述檔連接至 EC2 執行個體。 AWS Systems Manager

使用 aws:ssm:send-command動作

SSM 文件定義 Systems Manager 在受管執行個體上執行的動作。Systems Manager 包含許多預先設定的文件，您也可以建立自己的文件。如需建立自己的 SSM 文件的詳細資訊，請參閱AWS Systems Manager 《 使用者指南》中的建立 Systems Manager 文件。如需 SSM 文件的一般詳細資訊，請參閱AWS Systems Manager 《 使用者指南》中的AWS Systems Manager 文件。

AWS FIS 提供預先設定的 SSM 文件。您可以在 AWS Systems Manager 主控台的文件下檢視預先設定的 SSM 文件：https：//https://console.aws.amazon.com/systems-manager/documents。您也可以在 AWS FIS 主控台中選擇預先設定的文件。如需詳細資訊，請參閱預先設定的 AWS FIS SSM 文件。

若要在 AWS FIS 實驗中使用 SSM 文件，您可以使用 aws:ssm:send-command動作。此動作會在您的目標執行個體上擷取並執行指定的 SSM 文件。

當您在實驗範本中使用 `aws:ssm:send-command`動作時，您必須為 動作指定其他參數，包括下列項目：

- documentArn - 必要。SSM 文件的 Amazon Resource Name (ARN)。
- documentParameters – 條件式。SSM 文件接受的必要和選用參數。格式是 JSON 物件，其金鑰為字串，值為字串或字串陣列。
- documentVersion - 選用。要執行的 SSM 文件版本。

您可以使用 Systems Manager 主控台或命令列來檢視 SSM 文件的資訊（包括文件的參數）。

使用主控台檢視 SSM 文件的相關資訊

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，選擇 Documents (文件)。
3. 選取文件，然後選擇詳細資訊索引標籤。

使用命令列檢視 SSM 文件的相關資訊

使用 SSM describe-document 命令。

預先設定的 AWS FIS SSM 文件

您可以在實驗範本中使用預先設定的 AWS FIS SSM 文件搭配 `aws:ssm:send-command`動作。

要求

- AWS FIS 提供的預先設定 SSM 文件僅支援下列作業系統：
  - Amazon Linux 2023、Amazon Linux 2
  - Ubuntu
  - RHEL 8、9
  - CentOS 9
- AWS FIS 提供的預先設定 SSM 文件僅支援 EC2 執行個體。其他類型的受管節點不支援這些節點，例如內部部署伺服器。

若要在 ECS 任務的實驗中使用這些 SSM 文件，請使用對應的 the section called “Amazon ECS 動作”。例如， aws:ecs:task-cpu-stress動作會使用 AWSFIS-Run-CPU-Stress 文件。

Documents

- 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

AWS FIS SSM 文件中動作持續時間和 DurationSeconds 之間的差異

有些 SSM 文件會限制自己的執行時間，例如，某些預先設定的 AWS FIS SSM 文件會使用 DurationSeconds 參數。因此，您需要在 AWS FIS 動作定義中指定兩個獨立的持續時間：

- Action duration：對於具有單一動作的實驗，動作持續時間等同於實驗持續時間。使用多個動作時，實驗持續時間取決於個別動作持續時間及其執行順序。 AWS FIS 會監控每個動作，直到其動作持續時間經過為止。
- 文件參數 DurationSeconds：SSM 文件將執行的持續時間，以秒為單位。

您可以為兩種持續時間類型選擇不同的值：

- Action duration exceeds DurationSeconds：SSM 文件執行會在動作完成之前完成。 AWS FIS 會等到動作持續時間經過之後，再啟動後續動作。
- Action duration is shorter than DurationSeconds：SSM 文件會在動作完成後繼續執行。如果 SSM 文件執行仍在進行中，且動作持續時間已過，則動作狀態會設為已完成。 AWS FIS 只會監控執行，直到動作持續時間過了為止。

請注意，某些 SSM 文件具有可變持續時間。例如 AWS ，FIS SSM 文件可以選擇安裝先決條件，這可以將整體執行持續時間延長到超過指定的 DurationSeconds 參數。因此，如果您將動作持續時間和 DurationSeconds 設定為相同的值，SSM 指令碼的執行時間可能會超過動作持續時間。

AWSFIS-Run-CPU-Stress

使用 stress-ng工具在執行個體上執行 CPU 應力。使用 AWSFIS-Run-CPU-Stress SSM 文件。

動作類型（僅限主控台）

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

ARN

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

文件參數

- DurationSeconds - 必要。CPU 壓力測試的持續時間，以秒為單位。
- CPU - 選用。要使用的 CPU 壓力器數量。預設值為 0，它使用所有 CPU 壓力器。
- LoadPercent - 選用。目標 CPU 負載百分比，從 0（無負載）到 100（完全負載）。預設為 100。
- InstallDependencies - 選用。如果值為 `True`，則 Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 stress-ng。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Disk-Fill

配置執行個體根磁碟區上的磁碟空間，以模擬磁碟完全故障。使用 AWSFIS-Run-Disk-Fill SSM 文件。

如果手動或透過停止條件來停止注入此故障的實驗， AWS FIS 會取消執行中的 SSM 文件來嘗試轉返。不過，如果磁碟已滿 100%，可能是由於故障或故障加上應用程式活動，Systems Manager 可能無法完成取消操作。因此，如果您可能需要停止實驗，請確保磁碟不會 100% 已滿。

動作類型（僅限主控台）

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

ARN

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

文件參數

- DurationSeconds - 必要。磁碟填充測試的持續時間，以秒為單位。
- Percent - 選用。在磁碟填充測試期間要配置的磁碟百分比。預設值為 95%。
- InstallDependencies - 選用。如果值為 `True`，則 Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 atd、 kmod和 fallocate。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-IO-Stress

使用 stress-ng工具在執行個體上執行 IO 應力。使用 AWSFIS-Run-IO-Stress SSM 文件。

動作類型 （僅限主控台）

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

ARN

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

文件參數

- DurationSeconds - 必要。IO 壓力測試的持續時間，以秒為單位。
- Workers - 選用。執行循序、隨機和記憶體映射讀取/寫入操作、強制同步和快取捨棄的混合工作者數量。多個子程序會在同一檔案上執行不同的 I/O 操作。預設為 1。
- Percent - 選用。檔案系統上要在 IO 壓力測試期間使用的可用空間百分比。預設值為 80%。
- InstallDependencies - 選用。如果值為 `True`，則 Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 stress-ng。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Kill-Process

使用 killall命令停止執行個體中指定的程序。使用 AWSFIS-Run-Kill-Process SSM 文件。

動作類型（僅限主控台）

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

ARN

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

文件參數

- ProcessName - 必要。要停止的程序名稱。
- Signal - 選用。隨 命令一起傳送的訊號。可能的值為 `SIGTERM`（接收者可以選擇忽略）和 `SIGKILL`（無法忽略）。預設值為 `SIGTERM`。
- InstallDependencies – 選用。如果值為 `True`，則 Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 killall。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Memory-Stress

使用 stress-ng工具在執行個體上執行記憶體壓力。使用 AWSFIS-Run-Memory-Stress SSM 文件。

動作類型（僅限主控台）

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

ARN

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

文件參數

- DurationSeconds - 必要。記憶體壓力測試的持續時間，以秒為單位。
- Workers - 選用。虛擬記憶體壓力源的數量。預設為 1。
- Percent - 必要。在記憶體壓力測試期間要使用的虛擬記憶體百分比。
- InstallDependencies - 選用。如果值為 `True`，則 Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 stress-ng。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Network-Blackhole-Port

使用 iptables工具捨棄通訊協定和連接埠的傳入或傳出流量。使用 AWSFIS-Run-Network-Blackhole-Port SSM 文件。

動作類型 （僅限主控台）

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

ARN

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

文件參數

- Protocol - 必要。通訊協定。可能的值為 `tcp` 和 udp。
- Port - 必要。連接埠號碼。
- TrafficType - 選用。流量類型。可能的值為 `ingress` 和 `egress`。預設值為 `ingress`。
- DurationSeconds - 必要。網路黑洞測試的持續時間，以秒為單位。
- InstallDependencies - 選用。如果值為 `True`，則 Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 atd、lsof、 dig和 iptables。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Network-Latency

使用 tc工具將延遲新增至網路界面。使用 AWSFIS-Run-Network-Latency SSM 文件。

動作類型 （僅限主控台）

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

ARN

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

文件參數

- Interface - 選用。網路介面。預設值為 `eth0`。
- DelayMilliseconds – 選用。延遲，以毫秒為單位。預設值為 200。
- DurationSeconds - 必要。網路延遲測試的持續時間，以秒為單位。
- InstallDependencies - 選用。如果值為 `True`，Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 atd、 dig和 tc。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Network-Latency-Sources

針對進出特定來源的流量，使用 tc工具將延遲和抖動新增至網路界面。使用 AWSFIS-Run-Network-Latency-Sources SSM 文件。

動作類型 （僅限主控台）

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

ARN

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

文件參數

- Interface - 選用。網路介面。預設值為 `eth0`。
- DelayMilliseconds – 選用。延遲，以毫秒為單位。預設值為 200。
- JitterMilliseconds - 選用。抖動，以毫秒為單位。預設為 10。
- Sources - 必要。來源，以逗號分隔，不含空格。可能的值為：IPv4 地址、IPv4 CIDR 區塊、網域名稱、 `DYNAMODB`和 `S3`。如果您指定 `DYNAMODB`或 `S3`，這僅適用於目前區域中的區域端點。
- TrafficType - 選用。流量類型。可能的值為 `ingress` 和 `egress`。預設值為 `ingress`。
- DurationSeconds - 必要。網路延遲測試的持續時間，以秒為單位。

- InstallDependencies - 選用。如果值為 `True`，Systems Manager 會在目標執行個體上安裝所需的相依性，如果這些執行個體尚未安裝。預設值為 `True`。相依性為 atd、dig、lsof、 jq和 tc。

以下是您可以在 主控台中輸入的字串範例。

```
{"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工具將封包遺失新增至網路介面。使用 AWSFIS-Run-Network-Packet-Loss SSM 文件。

動作類型 （僅限主控台）

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

ARN

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

文件參數

- Interface - 選用。網路介面。預設值為 `eth0`。
- LossPercent – 選用。封包遺失的百分比。預設值為 7%。
- DurationSeconds - 必要。網路封包遺失測試的持續時間，以秒為單位。
- InstallDependencies - 選用。如果值為 `True`，Systems Manager 會在目標執行個體上安裝所需的相依性。預設值為 `True`。相依性為 atd、dig、 lsof和 tc。

以下是您可以在 主控台中輸入的字串範例。

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

AWSFIS-Run-Network-Packet-Loss-Sources

針對進出特定來源的流量，使用 tc工具將封包遺失新增至網路介面。使用 AWSFIS-Run-Network-Packet-Loss-Sources SSM 文件。

動作類型（僅限主控台）

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

ARN

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

文件參數

- Interface - 選用。網路介面。預設值為 `eth0`。
- LossPercent – 選用。封包遺失的百分比。預設值為 7%。
- Sources - 必要。來源，以逗號分隔，不含空格。可能的值為：IPv4 地址、IPv4 CIDR 區塊、網域名稱、 `DYNAMODB`和 `S3`。如果您指定 `DYNAMODB`或 `S3`，這僅適用於目前區域中的區域端點。
- TrafficType - 選用。流量類型。可能的值為 `ingress` 和 `egress`。預設值為 `ingress`。
- DurationSeconds - 必要。網路封包遺失測試的持續時間，以秒為單位。
- InstallDependencies - 選用。如果值為 `True`，Systems Manager 會在目標執行個體上安裝所需的相依性。預設值為 `True`。相依性為 atd、dig、lsof、 jq和 tc。

以下是您可以在 主控台中輸入的字串範例。

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

範例

如需範例實驗範本，請參閱 the section called “執行預先設定的 AWS FIS SSM 文件”。

如需教學課程範例，請參閱在執行個體上執行 CPU 應力。

故障診斷

使用下列程序對問題進行疑難排解。

疑難排解 SSM 文件的問題

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，選擇節點管理、執行命令。

3. 在命令歷史記錄索引標籤上，使用篩選條件來尋找文件的執行。
4. 選擇命令的 ID 以開啟其詳細資訊頁面。
5. 選擇執行個體的 ID。檢閱每個步驟的輸出和錯誤。

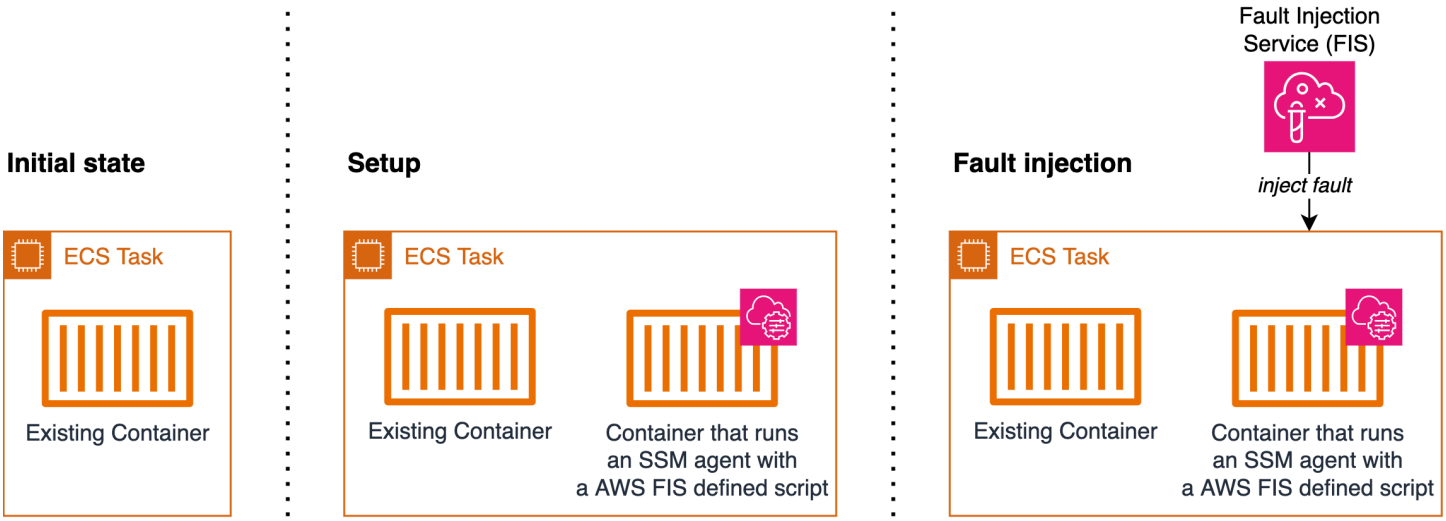
使用 AWS FIS aws：ecs：task 動作

您可以使用 aws：ecs：task 動作，將錯誤注入 Amazon ECS 任務。支援 Amazon EC2 和 Fargate 容量類型。

這些動作會使用 AWS Systems Manager (SSM) 文件來注入錯誤。若要使用 `aws:ecs:task` 動作，您需要將具有 SSM 代理程式的容器新增至 Amazon Elastic Container Service (Amazon ECS) 任務定義。容器會執行 AWS FIS 定義的指令碼，在 SSM 服務中將 Amazon ECS 任務註冊為受管執行個體。此外，指令碼會擷取任務中繼資料，將標籤新增至受管執行個體。設定將允許 AWS FIS 解析目標任務。此段落是指下圖中的設定。

當您執行以 為目標的 AWS FIS 實驗時`aws:ecs:task`， AWS FIS 會使用資源標籤 ，將您在 AWS FIS 實驗範本中指定的目標 Amazon ECS 任務映射至一組 SSM 受管執行個體`ECS_TASK_ARN`。標籤值是應執行 SSM 文件之相關聯 Amazon ECS 任務的 ARN。此段落是指下圖中的 Fault Injection。

下圖示範具有一個現有容器的任務的設定和錯誤注入。



動作

- 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

- 下列動作無法平行執行：
  - aws:ecs:task-network-blackhole-port
  - aws:ecs:task-network-latency
  - aws:ecs:task-network-packet-loss
- 如果您啟用 Amazon ECS Exec，您必須先停用它，才能使用這些動作。
- 即使實驗狀態為已完成，SSM 文件執行仍可能已取消狀態。執行 Amazon ECS 動作時，會在實驗中的動作持續時間和 Amazon EC2 Systems Manager (SSM) 文件持續時間中使用客戶提供的持續時間。動作啟動後，SSM 文件需要一些時間才能開始執行。因此，在達到指定的動作持續時間時，SSM 文件可能仍有幾秒鐘的時間來完成其執行。達到實驗動作持續時間時，動作會停止，SSM 文件執行也會取消。故障注入成功。

要求

- 將下列許可新增至 AWS FIS 實驗角色：
  - `ssm:SendCommand`
  - `ssm:ListCommands`
  - `ssm:CancelCommand`
- 將下列許可新增至 Amazon ECS 任務 IAM 角色：
  - `ssm:CreateActivation`
  - `ssm:AddTagsToResource`
  - `iam:PassRole`

  請注意，您可以將受管執行個體角色的 ARN 指定為 的資源`iam:PassRole`。
- 建立 Amazon ECS 任務執行 IAM 角色，並新增 AmazonECSTaskExecutionRolePolicy 受管政策。
- 在任務定義中，將環境變數`MANAGED_INSTANCE_ROLE_NAME`設定為受管執行個體角色的名稱。這是將連接到在 SSM 中註冊為受管執行個體之任務的角色。
- 將下列許可新增至受管執行個體角色：

  - `ssm:DeleteActivation`
  - `ssm:DeregisterManagedInstance`
- 將 AmazonSSMManagedInstanceCore 受管政策新增至受管執行個體角色。
- 將 SSM 代理程式容器新增至 Amazon ECS 任務定義。命令指令碼會將 Amazon ECS 任務註冊為受管執行個體。

```
{
    "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 procps awscli -y; term_handler()
 { echo \"Deleting SSM activation $ACTIVATION_ID\"; if ! aws ssm delete-
activation --activation-id $ACTIVATION_ID --region $ECS_TASK_REGION; then
 echo \"SSM activation $ACTIVATION_ID failed to be deleted\" 1>&2; fi;
 MANAGED_INSTANCE_ID=$(jq -e -r .ManagedInstanceID /var/lib/amazon/ssm/registration);
 echo \"Deregistering SSM Managed Instance $MANAGED_INSTANCE_ID\"; if ! aws
 ssm deregister-managed-instance --instance-id $MANAGED_INSTANCE_ID --region
 $ECS_TASK_REGION; then echo \"SSM Managed Instance $MANAGED_INSTANCE_ID
 failed to be deregistered\" 1>&2; fi; kill -SIGTERM $SSM_AGENT_PID; }; trap
 term_handler SIGTERM SIGINT; if [[ -z $MANAGED_INSTANCE_ROLE_NAME ]]; then
 echo \"Environment variable MANAGED_INSTANCE_ROLE_NAME not set, exiting\"
 1>&2; exit 1; fi; if ! ps ax | grep amazon-ssm-agent | grep -v grep > /dev/
null; then if [[ -n $ECS_CONTAINER_METADATA_URI_V4 ]] ; then echo \"Found ECS
 Container Metadata, running activation with metadata\"; TASK_METADATA=$(curl
 \"${ECS_CONTAINER_METADATA_URI_V4}/task\"); ECS_TASK_AVAILABILITY_ZONE=$(echo
 $TASK_METADATA | jq -e -r '.AvailabilityZone'); ECS_TASK_ARN=$(echo $TASK_METADATA
 | jq -e -r '.TaskARN'); ECS_TASK_REGION=$(echo $ECS_TASK_AVAILABILITY_ZONE | sed
 's/.$//'); ECS_TASK_AVAILABILITY_ZONE_REGEX='^(af|ap|ca|cn|eu|me|sa|us|us-gov)-
(central|north|(north(east|west))|south|south(east|west)|east|west)-[0-9]{1}[a-z]
{1}$'; if ! [[ $ECS_TASK_AVAILABILITY_ZONE =~ $ECS_TASK_AVAILABILITY_ZONE_REGEX ]];
 then echo \"Error extracting Availability Zone from ECS Container Metadata,
 exiting\" 1>&2; exit 1; fi; ECS_TASK_ARN_REGEX='^arn:(aws|aws-cn|aws-us-gov):ecs:
[a-z0-9-]+:[0-9]{12}:task/[a-zA-Z0-9_-]+/[a-zA-Z0-9]+$'; if ! [[ $ECS_TASK_ARN
 =~ $ECS_TASK_ARN_REGEX ]]; then echo \"Error extracting Task ARN from ECS
 Container Metadata, exiting\" 1>&2; exit 1; fi; CREATE_ACTIVATION_OUTPUT=
```

```
$(aws ssm create-activation --iam-role $MANAGED_INSTANCE_ROLE_NAME --
tags Key=ECS_TASK_AVAILABILITY_ZONE,Value=$ECS_TASK_AVAILABILITY_ZONE
 Key=ECS_TASK_ARN,Value=$ECS_TASK_ARN Key=FAULT_INJECTION_SIDECAR,Value=true --
region $ECS_TASK_REGION); ACTIVATION_CODE=$(echo $CREATE_ACTIVATION_OUTPUT | jq
 -e -r .ActivationCode); ACTIVATION_ID=$(echo $CREATE_ACTIVATION_OUTPUT | jq -e
 -r .ActivationId); if ! amazon-ssm-agent -register -code $ACTIVATION_CODE -id
 $ACTIVATION_ID -region $ECS_TASK_REGION; then echo \"Failed to register with AWS
 Systems Manager (SSM), exiting\" 1>&2; exit 1; fi; amazon-ssm-agent & SSM_AGENT_PID=
$!; wait $SSM_AGENT_PID; else echo \"ECS Container Metadata not found, exiting\"
 1>&2; exit 1; fi; else echo \"SSM agent is already running, exiting\" 1>&2; exit 1;
 fi"
    ],
    "environment": [
        {
            "name": "MANAGED_INSTANCE_ROLE_NAME",
            "value": "SSMManagedInstanceRole"
        }
    ],
    "environmentFiles": [],
    "mountPoints": [],
    "volumesFrom": [],
    "secrets": [],
    "dnsServers": [],
    "dnsSearchDomains": [],
    "extraHosts": [],
    "dockerSecurityOptions": [],
    "dockerLabels": {},
    "ulimits": [],
    "logConfiguration": {},
    "systemControls": []
}
```

如需指令碼更易讀的版本，請參閱 the section called “指令碼的參考版本”。

- 透過在 Amazon ECS 任務定義中設定 `enableFaultInjection` 欄位，啟用 Amazon ECS Fault Injection APIs：

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

- 在 Fargate 任務上使用 `aws:ecs:task-network-blackhole-portaws:ecs:task-network-latency`、 或 `aws:ecs:task-network-packet-loss`動作時，動作必須將 `useEcsFaultInjectionEndpoints` 參數設定為 `true`。

- 使用 `aws:ecs:task-kill-process`、`aws:ecs:task-network-latency`、`aws:ecs:task-network-blackhole-port`和 `aws:ecs:task-network-packet-loss`動作時，Amazon ECS 任務定義必須`pidMode`設定為 `task`。
- 使用 `aws:ecs:task-network-blackhole-port`、 `aws:ecs:task-network-latency`和 `aws:ecs:task-network-packet-loss`動作時，Amazon ECS 任務定義必須`networkMode`設定為 以外的值`bridge`。

指令碼的參考版本

以下是需求區段中可供讀取的指令碼版本，供您參考。

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

# This is the activation script used to register ECS tasks as Managed Instances in SSM
# The script retrieves information form the ECS task metadata endpoint to add three
 tags to the Managed Instance
#  - ECS_TASK_AVAILABILITY_ZONE: To allow customers to target Managed Instances / Tasks
 in a specific Availability Zone
#  - ECS_TASK_ARN: To allow customers to target Managed Instances / Tasks by using the
 Task ARN
#  - FAULT_INJECTION_SIDECAR: To make it clear that the tasks were registered as
 managed instance for fault injection purposes. Value is always 'true'.
# The script will leave the SSM Agent running in the background
# When the container running this script receives a SIGTERM or SIGINT signal, it will
 do the following cleanup:
#  - Delete SSM activation
#  - Deregister SSM managed instance

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

dnf upgrade -y
dnf install jq procps awscli -y

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

  MANAGED_INSTANCE_ID=$(jq -e -r .ManagedInstanceID /var/lib/amazon/ssm/registration)
```

```
  echo "Deregistering SSM Managed Instance $MANAGED_INSTANCE_ID"
  if ! aws ssm deregister-managed-instance --instance-id $MANAGED_INSTANCE_ID --region
 $ECS_TASK_REGION; then
    echo "SSM Managed Instance $MANAGED_INSTANCE_ID failed to be deregistered" 1>&2
  fi

  kill -SIGTERM $SSM_AGENT_PID
}
trap term_handler SIGTERM SIGINT

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

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

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

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

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

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

```
    fi

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

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

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

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

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

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

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

範例實驗範本

以下是 the section called “aws:ecs:task-cpu-stress”動作的實驗範本範例。

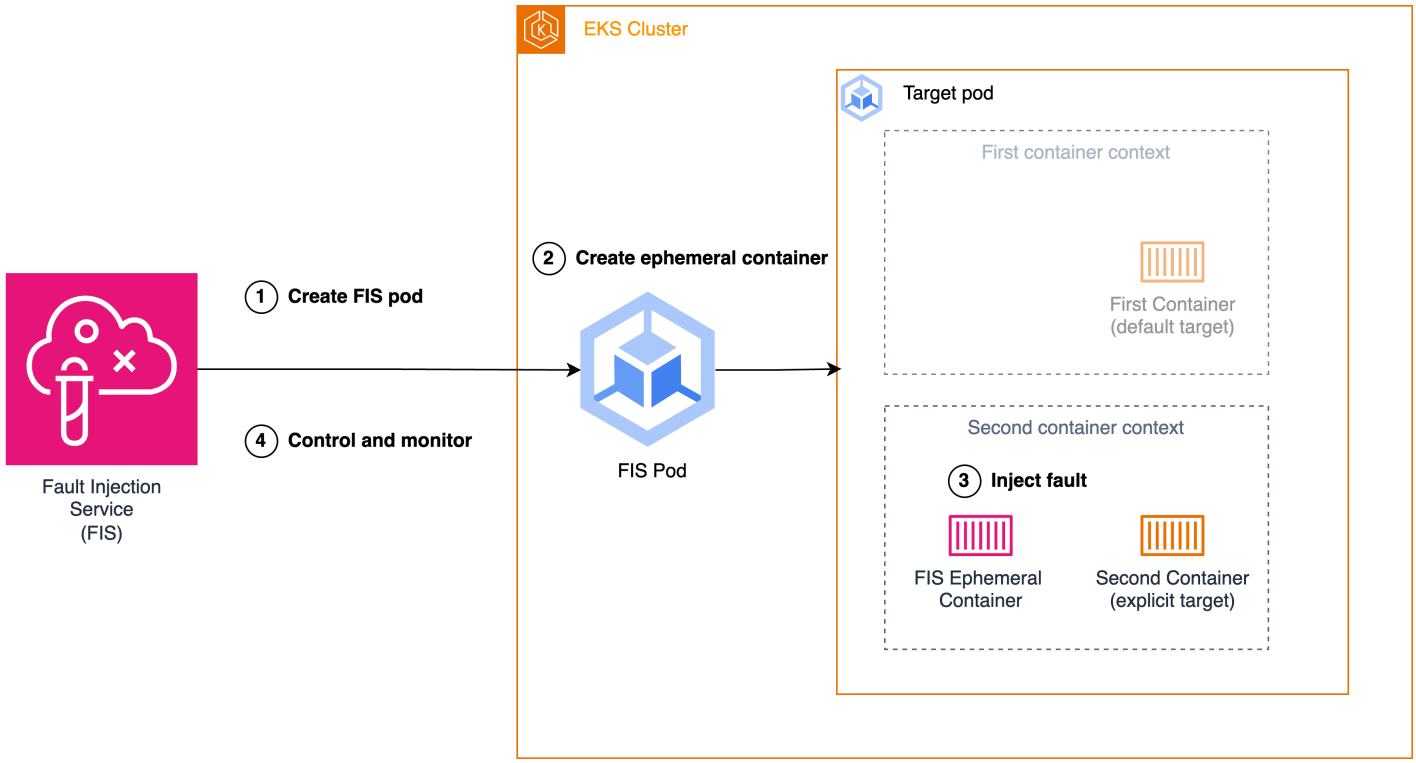
```
{
    "description": "Run CPU stress on the target ECS tasks",
```

```
    "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 AWS FIS aws：eks：pod 動作

您可以使用 aws：eks：pod 動作，將故障注入 EKS 叢集中執行的 Kubernetes Pod。

啟動動作時，FIS 會擷取 FIS Pod 容器映像。然後，此影像用於在目標 EKS 叢集中建立 Pod。新建立的 Pod 負責注入、控制和監控錯誤。對於除 aws：eks：pod-delete 以外的所有 FIS EKS 動作，故障注入是透過使用暫時性容器來實現，這是允許在現有 Pod 中建立暫時容器的 Kubernetes 功能。暫時性容器會在與目標容器相同的命名空間中啟動，並執行所需的錯誤注入任務。如果未指定目標容器，則會選取 Pod 規格中的第一個容器做為目標。



1. FIS 會在實驗範本中指定的目標叢集中建立 FIS Pod。
2. FIS Pod 會在與目標容器位於相同命名空間的目標 Pod 中建立暫時性容器。
3. 暫時性容器會在目標容器的命名空間中注入錯誤。
4. FIS Pod 控制和監控暫時性容器的錯誤注入，以及 FIS 控制和監控 FIS Pod。

完成實驗或發生錯誤時，會移除暫時性容器和 FIS Pod。

動作

- 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

- 下列動作不適用於 AWS Fargate：
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- 下列動作不支援 `bridge` 網路模式：
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- 下列動作需要暫時性容器內的根許可。
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss

  暫時性容器會從目標 Pod 的安全內容繼承其許可。如果您需要以非根使用者身分執行 Pod 中的容器，您可以為目標 Pod 中的容器設定個別的安全內容。
- 您無法在實驗範本中使用資源 ARNs或資源標籤來識別 aws：eks：pod 類型的目標。您必須使用所需的資源參數來識別目標。
- 動作 aws：eks：pod-network-latency 和 aws：eks：pod-network-packet-loss 不應平行執行，並以相同的 Pod 為目標。根據您指定的`maxErrors`參數值，動作可能會結束為已完成或失敗狀態：
  - 如果 `maxErrorsPercent`為 0（預設），動作將結束為失敗狀態。
  - 否則，失敗會加總到`maxErrorsPercent`預算。如果失敗的注入次數未達到提供的 `maxErrors`，則動作最終會處於完成狀態。
  - 您可以從目標 Pod 中注入暫時性容器的日誌中識別這些失敗。它會因 而失敗`Exit Code: 16`。
- 動作 aws：eks：pod-network-blackhole-port 不應與以相同 Pod 為目標並使用相同 的其他動作平行執行`trafficType`。支援使用不同流量類型的平行動作。
- FIS 只能在目標 Pod `securityContext`的 設定為 時監控故障注入的狀態`readOnlyRootFilesystem: false`。如果沒有此組態，所有 EKS Pod 動作都會失敗。

要求

- AWS CLI 在電腦上安裝 。只有在您將使用 AWS CLI 建立 IAM 角色時才需要此操作。如需詳細資訊，請參閱安裝或更新 AWS CLI。
- 在您電腦上安裝 kubectl。這僅適用於與 EKS 叢集互動，以設定或監控目標應用程式。如需詳細資訊，請參閱 https：//https://kubernetes.io/docs/tasks/tools/。
- EKS 的最低支援版本為 1.23。

建立實驗角色

若要執行實驗，您需要為實驗設定 IAM 角色。如需詳細資訊，請參閱the section called “實驗角色”。此角色的必要許可取決於您正在使用的動作。請參閱以 動作為AWS 目標的 FIS `aws:eks:pod` 動作，以尋找動作的必要許可。

設定 Kubernetes 服務帳戶

設定 Kubernetes 服務帳戶，以對指定 Kubernetes 命名空間中的目標執行實驗。在下列範例中，服務帳戶是 *myserviceaccount*，命名空間是*##*的。請注意， default是標準 Kubernetes 命名空間之一。

設定 Kubernetes 服務帳戶

1. 建立名為 的檔案`rbac.yaml`，並新增下列項目。

```
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. 執行下列命令。

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

授予 IAM 使用者和角色對 Kubernetes APIs存取權

請遵循EKS文件中將 IAM 身分與 Kubernetes 許可建立關聯中說明的步驟。

選項 1：建立存取項目

我們建議您Access Entries遵循授予 IAM 使用者使用 EKS 存取項目存取 Kubernetes 中說明的步驟來使用 。

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

> **Important**
>
> 為了利用存取項目，EKS 叢集的身分驗證模式必須設定為 `API_AND_CONFIG_MAP`或 `API` 模式。

選項 2：將項目新增至 aws-auth ConfigMap

您也可以使用下列命令來建立身分映射。如需詳細資訊，請參閱 eksctl 文件中的 Manage IAM users and roles 一節。

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

> **Important**
>
> 利用 eksctl 工具組設定身分映射將導致在 ConfigMap `aws-auth` 中建立項目。請務必注意，這些產生的項目不支援包含路徑元件。因此，提供做為輸入的 ARN 不得包含路徑區段（例如 `arn:aws:iam::123456789012:role/service-role/fis-experiment-role`)。

Pod 容器映像

AWS FIS 提供的 Pod 容器映像託管在 Amazon ECR 中。當您從 Amazon ECR 參考映像時，必須使用完整映像 URI。

Pod 容器映像也可以在 AWS ECR Public Gallery 中使用。

AWS 區域	映像 URI
美國東部 (俄亥俄)	051821878176.dkr.ecr.us-east-2.amazonaws.com/aws-fis-pod:0.1
美國東部 (維吉尼亞北部)	731367659002.dkr.ecr.us-east-1.amazonaws.com/aws-fis-pod:0.1
美國西部 (加利佛尼亞北部)	080694859247.dkr.ecr.us-west-1.amazonaws.com/aws-fis-pod:0.1
美國西部 (奧勒岡)	864386544765.dkr.ecr.us-west-2.amazonaws.com/aws-fis-pod:0.1
非洲 (開普敦)	056821267933.dkr.ecr.af-south-1.amazonaws.com/aws-fis-pod:0.1
亞太區域 (香港)	246405402639.dkr.ecr.ap-east-1.amazonaws.com/aws-fis-pod:0.1
亞太區域 (孟買)	524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1
亞太區域 (首爾)	526524659354.dkr.ecr.ap-northeast-2.amazonaws.com/aws-fis-pod:0.1
亞太區域 (新加坡)	316401638346.dkr.ecr.ap-southeast-1.amazonaws.com/aws-fis-pod:0.1
亞太區域 (雪梨)	488104106298.dkr.ecr.ap-southeast-2.amazonaws.com/aws-fis-pod:0.1
亞太區域 (東京)	635234321696.dkr.ecr.ap-northeast-1.amazonaws.com/aws-fis-pod:0.1
加拿大 (中部)	490658072207.dkr.ecr.ca-central-1.amazonaws.com/aws-fis-pod:0.1

AWS 區域	映像 URI
歐洲 (法蘭克福)	713827034473.dkr.ecr.eu-central-1.amazonaws.com/aws-fis-pod:0.1
歐洲 (愛爾蘭)	205866052826.dkr.ecr.eu-west-1.amazonaws.com/aws-fis-pod:0.1
歐洲 (倫敦)	327424803546.dkr.ecr.eu-west-2.amazonaws.com/aws-fis-pod:0.1
歐洲 (米蘭)	478809367036.dkr.ecr.eu-south-1.amazonaws.com/aws-fis-pod:0.1
Europe (Paris)	154605889247.dkr.ecr.eu-west-3.amazonaws.com/aws-fis-pod:0.1
歐洲 (西班牙)	395402409451.dkr.ecr.eu-south-2.amazonaws.com/aws-fis-pod:0.1
歐洲 (斯德哥爾摩)	263175118295.dkr.ecr.eu-north-1.amazonaws.com/aws-fis-pod:0.1
Middle East (Bahrain)	065825543785.dkr.ecr.me-south-1.amazonaws.com/aws-fis-pod:0.1
南美洲 (聖保羅)	767113787785.dkr.ecr.sa-east-1.amazonaws.com/aws-fis-pod:0.1
AWS GovCloud（美國東部）	246533647532.dkr.ecr.us-gov-east-1.amazonaws.com/aws-fis-pod:0.1
AWS GovCloud（美國西部）	246529956514.dkr.ecr.us-gov-west-1.amazonaws.com/aws-fis-pod:0.1

範例實驗範本

以下是 the section called “aws:eks:pod-network-latency”動作的實驗範本範例。

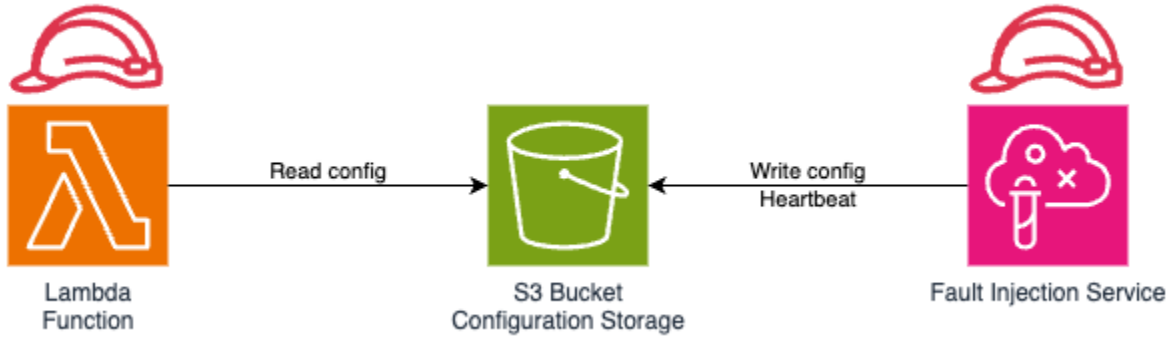
```
{
    "description": "Add latency and jitter to the network interface for the target EKS
 Pods",
    "targets": {
        "myPods": {
            "resourceType": "aws:eks:pod",
            "parameters": {
                "clusterIdentifier": "mycluster",
                "namespace": "default",
                "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：function 動作

您可以使用 aws：lambda：function 動作，將錯誤注入 AWS Lambda 函數的叫用中。

這些動作使用 AWS FIS 受管擴充功能來注入錯誤。若要使用 aws：lambda：function 動作，您需要將擴充功能做為層連接至 Lambda 函數，並設定 Amazon S3 儲存貯體以在 AWS FIS 和擴充功能之間通訊。

當您執行以 aws：lambda：function 為目標的 AWS FIS 實驗時， 會從 Lambda 函數 AWS FIS 讀取 Amazon S3 組態，並將錯誤注入資訊寫入指定的 Amazon S3 位置，如下圖所示。



動作

- 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

- AWS FIS Lambda 延伸模組無法與使用回應串流的函數搭配使用。即使未套用錯誤， AWS FIS Lambda 延伸模組也會抑制串流組態。如需詳細資訊，請參閱《 AWS Lambda 使用者指南》中的 Lambda 函數的回應串流。

先決條件

在使用 AWS FIS Lambda 動作之前，請確定您已完成下列一次性任務：

- 在您計劃從中開始實驗的區域中建立 Amazon S3 儲存貯體 - 您可以使用單一 Amazon S3 儲存貯體進行多個實驗，並在多個 AWS 帳戶之間共用儲存貯體。不過，每個儲存貯體都必須有個別的儲存貯體 AWS 區域。
- 建立 IAM 政策，將 Lambda 延伸的讀取存取權授予 Amazon S3 儲存貯體 - 在下列範本中，將 取代`my-config-distribution-bucket`為您在上方建立的 Amazon S3 儲存貯體名稱，並將 `FisConfigs` 取代為您要使用的 Amazon S3 儲存貯體中的資料夾名稱。

```
{
  "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/*"]
      }
  ]
}
```

- 建立 IAM 政策，將 AWS FIS 實驗的寫入存取權授予 Amazon S3 儲存貯體 - 在下列範本中，將取代`my-config-distribution-bucket`為您在上方建立的 Amazon S3 儲存貯體名稱，並將 `FisConfigs` 取代為您要使用的 Amazon S3 儲存貯體中的資料夾名稱。

```
{
  "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": "*"
        }
    ]
}
```

設定 Lambda 函數

針對您要影響的每個 Lambda 函數，請遵循下列步驟：

1. 將上述建立的 Amazon S3 讀取存取政策連接至 Lambda 函數。
2. 將 AWS FIS 延伸模組做為 layer 連接至 函數。如需 layer ARNs的詳細資訊，請參閱 Lambda 擴充 AWS FIS 功能的可用版本。
3. 將`AWS_FIS_CONFIGURATION_LOCATION`變數設定為 Amazon S3 組態資料夾的 ARN，例如 `arn:aws:s3:::my-config-distribution-bucket/FisConfigs/`。
4. 將 `AWS_LAMBDA_EXEC_WRAPPER` 變數設定為 `/opt/aws-fis/bootstrap`。

設定 AWS FIS 實驗

在執行實驗之前，請確定您已將您在先決條件中建立的 Amazon S3 寫入存取政策連接到將使用 AWS FIS Lambda 動作的實驗角色。如需如何設定 AWS FIS 實驗的詳細資訊，請參閱 管理 AWS FIS 實驗範本。

日誌

AWS FIS Lambda 擴充功能會將日誌寫入主控台和 CloudWatch 日誌。您可以使用 `AWS_FIS_LOG_LEVEL`變數來設定記錄。支援的值為 `INFO`、`WARN` 和 `ERROR`。日誌將以為 Lambda 函數設定的日誌格式撰寫。

以下是文字格式的日誌範例：

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

以下是 JSON 格式的日誌範例：

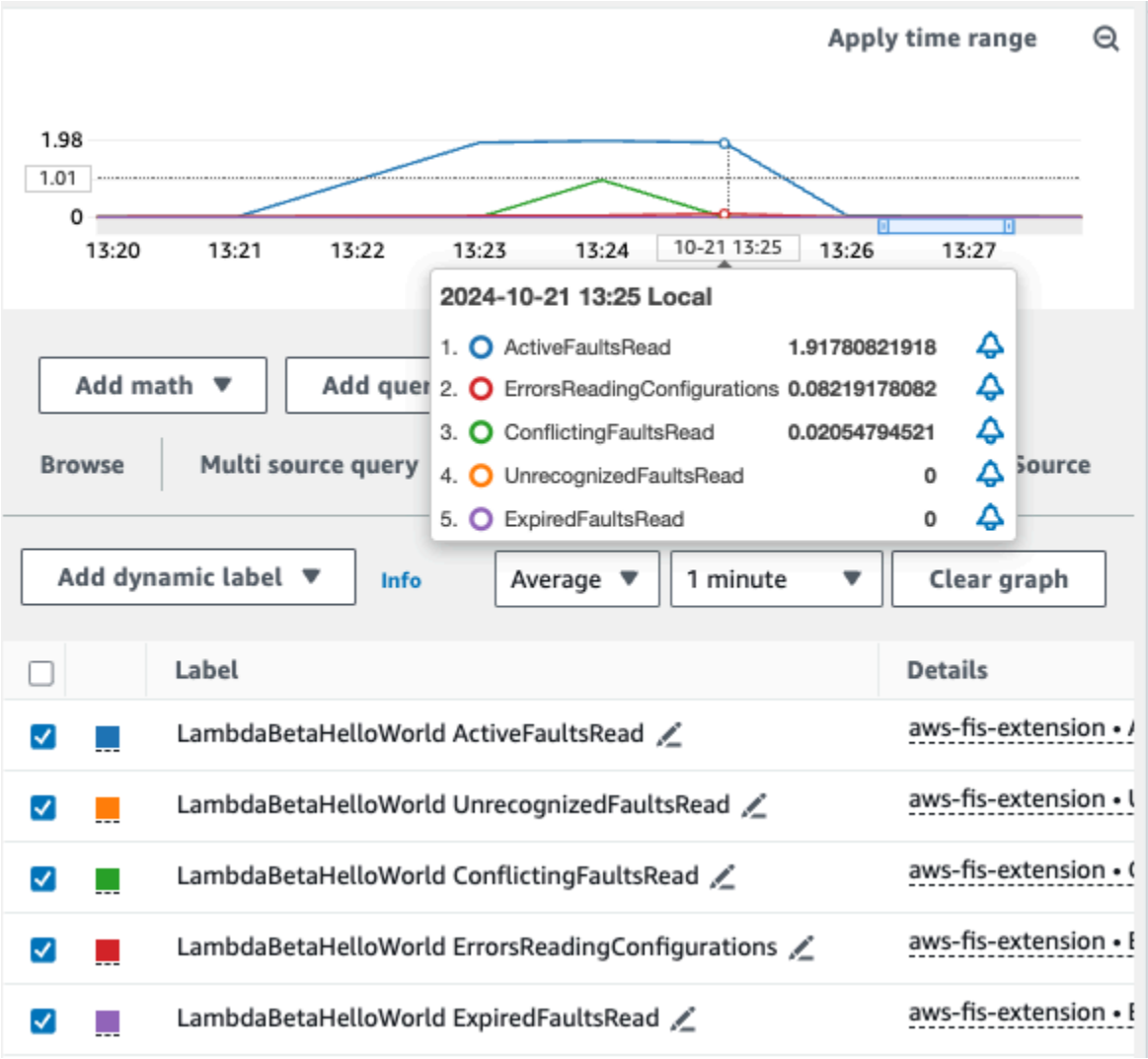
```
{
  "timestamp": "2024-10-08T17:15:36.953905Z",
  "level": "INFO",
  "fields": {
    "message": "AWS FIS EXTENSION - adding 5000 milliseconds of latency to function
 invocation",
    "requestId":"0608bf70-908f-4a17-bbfe-3782cd783d8b"
  }
}
```

發出的日誌可與 Amazon CloudWatch 指標篩選條件搭配使用，以產生自訂指標。如需指標篩選條件的詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的使用篩選條件從日誌事件建立指標。

使用 CloudWatch Embedded Metric Format (EMF)

您可以將 `AWS_FIS_EXTENSION_METRICS`變數設定為 ，以設定 AWS FIS Lambda 延伸來發出 EMF 日誌`all`。根據預設，擴充功能不會發出 EMF 日誌，且`AWS_FIS_EXTENSION_METRICS`預設為 `none`。EMF 日誌會在 CloudWatch 主控台`aws-fis-extension namespace`的 中發佈。

在`aws-fis-extension`命名空間中，您可以選取要在圖形中顯示的特定指標。以下範例顯示 `aws-fis-extension` 命名空間中的一些可用指標。



進階主題

本節提供如何使用 Lambda AWS FIS 延伸模組和特殊使用案例的其他資訊。

主題

- 了解輪詢
- 了解並行
- 了解調用百分比
- SnapStart 的特殊考量
- 快速不常函數的特殊考量
- 使用 Lambda 執行期 API 代理設定多個延伸模組
- 使用 AWS FIS 搭配容器執行時間

- AWS FIS Lambda 環境變數

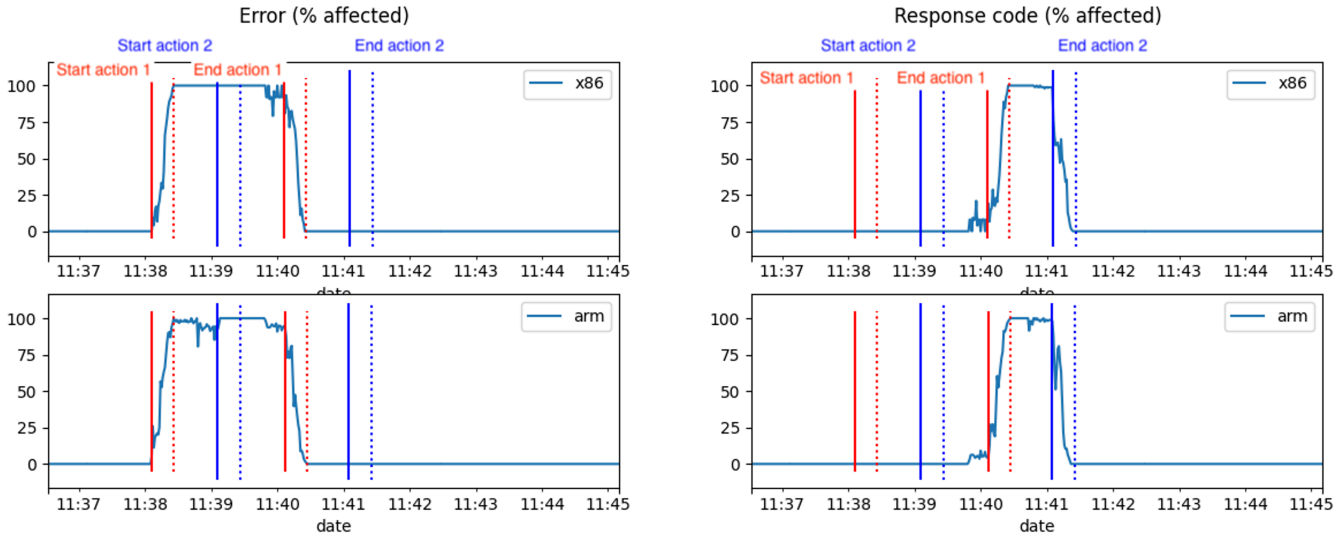
了解輪詢

您可能會注意到在故障開始影響所有調用之前，緩衝期最多為 60 秒。這是因為 Lambda 延伸模組在等待實驗啟動時不常輪詢組態資訊。您可以設定`AWS_FIS_SLOW_POLL_INTERVAL_SECONDS`環境變數（預設 60 秒）來調整輪詢間隔。較低的值將更頻繁輪詢，但對效能和成本的影響更大。在插入錯誤之後，您可能也會注意到延遲時間最多 20 秒。這是因為延伸模組會在實驗執行時更頻繁地輪詢。

了解並行

您可以同時以具有多個動作的相同 Lambda 函數為目標。如果動作彼此不同，則會套用所有動作。例如，您可以在傳回錯誤之前新增初始延遲。如果將兩個相同或衝突的動作套用至相同的函數，則只會套用最早開始日期的動作。

下圖顯示兩個衝突的動作：aws：lambda：invocation-error 和 aws：lambda：invocation-http-integration-response，重疊。一開始，aws：lambda：invocation-error 會在 11：38 上升並執行 2 分鐘。然後，aws：lambda：invocation-http-integration-response 會嘗試從 11：39 開始，但在第一個動作結束之後的 11：40 才會生效。為了維持實驗時間，aws：lambda：invocation-http-integration-response 仍會在最初的預定時間 11：41 完成。



了解調用百分比

AWS Fault Injection Service Lambda 動作使用 aws：lambda：function 目標，可讓您選取一或多個 AWS Lambda 函數 ARNs。使用這些 ARNs， AWS Fault Injection Service Lambda 動作可以在每次叫用選取的 Lambda 函數時注入錯誤。若要只允許您將故障注入一小部分的調用中，每個動作都允許

您指定值為 0 到 100 的`invocationPercentage`參數。使用 `invocationPercentage` 參數，即使調用百分比低於 100%，您也可以確保動作是並行的。

SnapStart 的特殊考量

AWS Lambda 啟用 SnapStart 的 函數在取得第一個故障組態`AWS_FIS_SLOW_POLL_INTERVAL_SECONDS`之前，有較高的可能性會等待 的完整持續時間，即使實驗已在執行中。這是因為 Lambda SnapStart 使用單一快照做為多個執行環境的初始狀態，並保留暫時儲存。對於 AWS Fault Injection Service Lambda 延伸，它會保留輪詢頻率，並略過初始化執行環境時的初始組態檢查。如需 Lambda SnapStart 的詳細資訊，請參閱《 使用者指南》中的使用 Lambda SnapStart 改善啟動效能。 AWS Lambda

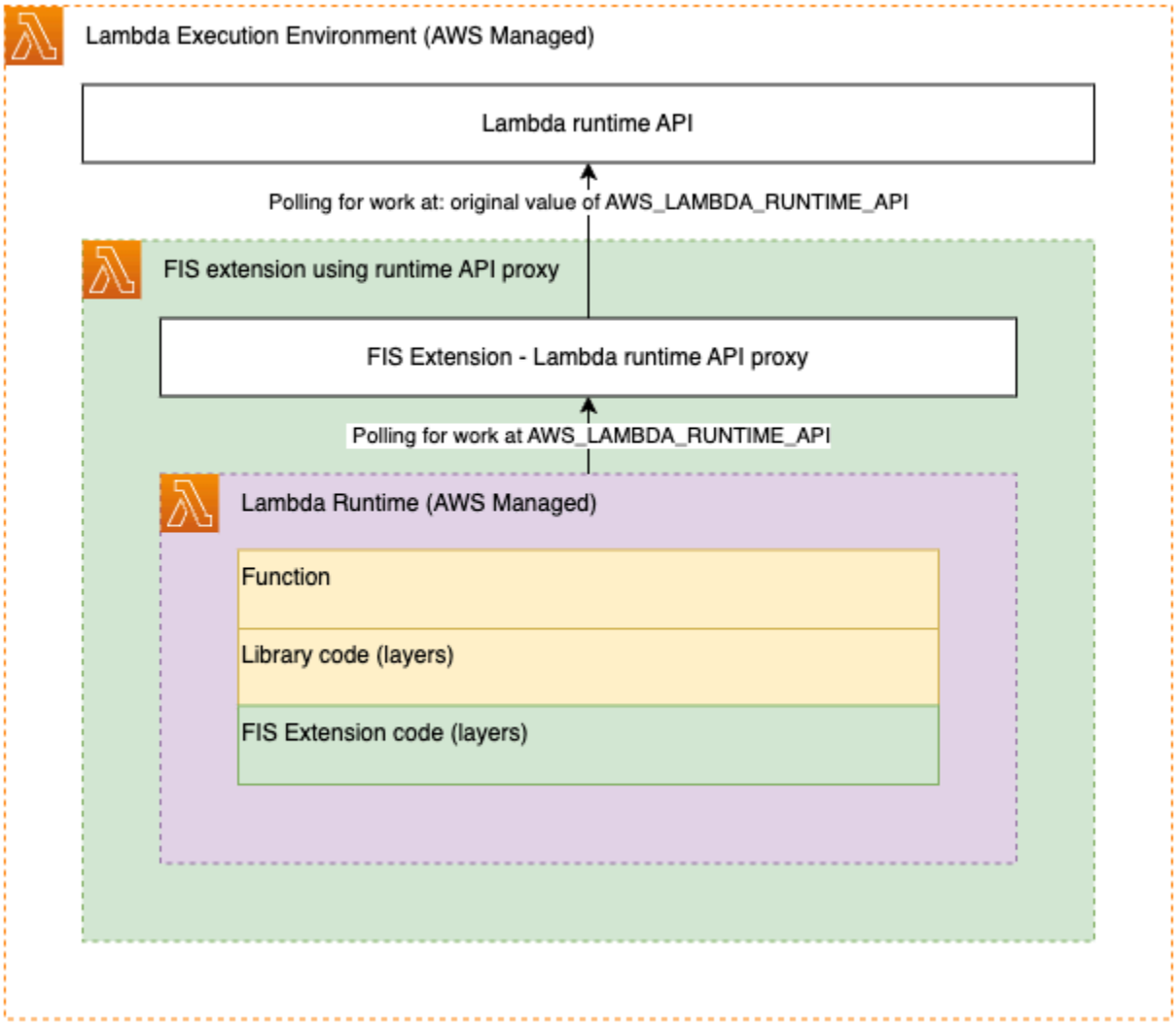
快速不常函數的特殊考量

如果您的 Lambda 函數執行時間少於平均輪詢持續時間 70 毫秒，則輪詢執行緒可能需要多次調用才能取得錯誤組態。如果函數不常執行，例如每 15 分鐘執行一次，則永遠不會完成輪詢。若要確保輪詢執行緒可以完成，請設定 `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` 參數。延伸項目會等到您為傳輸中輪詢設定的持續時間結束，再啟動函數。請注意，這將增加計費的函數持續時間，並導致某些調用的額外延遲。

使用 Lambda 執行期 API 代理設定多個延伸模組

Lambda 延伸模組會使用 AWS Lambda 執行期 API 代理來攔截函數呼叫，再到達執行期。其作法是將 AWS Lambda 執行時間 API 的代理公開至執行時間，並在 `AWS_LAMBDA_RUNTIME_API`變數中公告其位置。

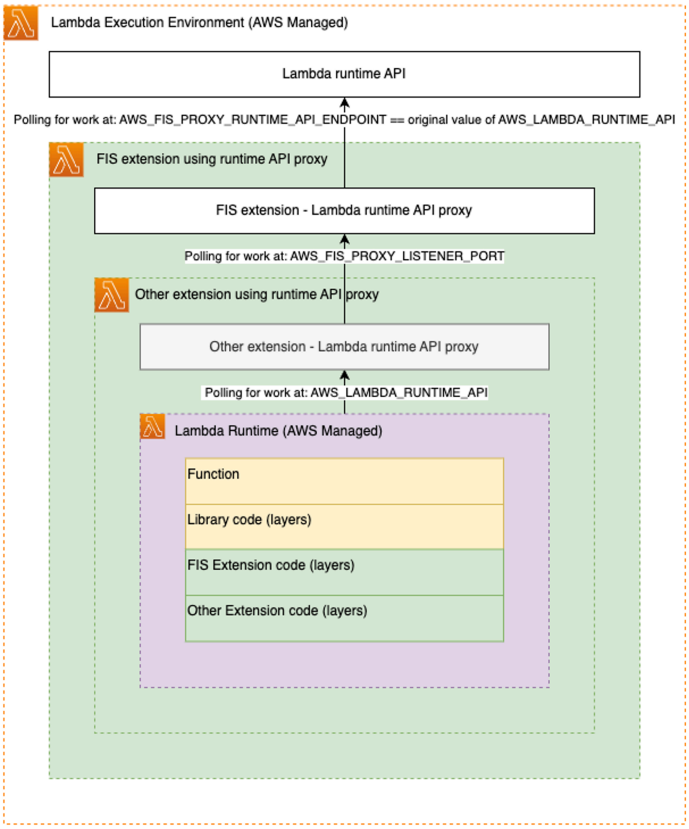
下圖顯示使用 Lambda 執行期 API 代理的單一延伸模組的組態：

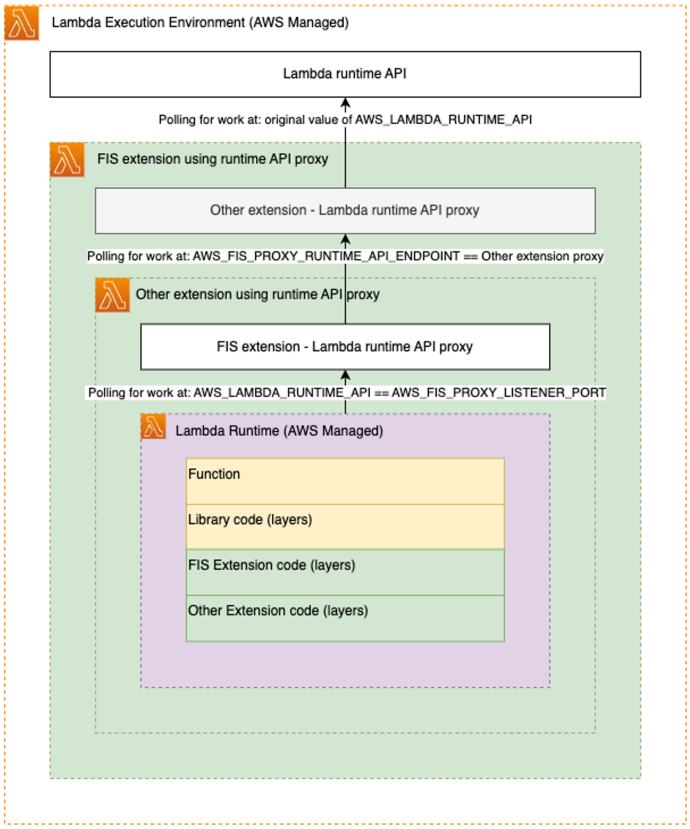


若要使用 AWS Lambda 執行期 API 代理模式將 AWS FIS Lambda 延伸模組與另一個延伸模組搭配使用，您將需要使用自訂引導指令碼鏈結代理。 AWS FIS Lambda 延伸模組接受下列環境變數：

- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT` - 採用`127.0.0.1:9876`代表 AWS Lambda 執行時間 API 的本機 IP 和接聽程式連接埠格式的字串。這可以是 的原始值`AWS_LAMBDA_RUNTIME_API`或其他代理的位置。
- `AWS_FIS_PROXY_LISTENER_PORT` - 根據預設， 會取得 AWS FIS 擴充功能應啟動其代理的連接埠號碼`9100`。

透過這些設定，您可以使用 Lambda 執行期 API 代理，以兩個不同的順序將 AWS FIS 延伸項目與另一個延伸項目鏈結。





如需 AWS Lambda 執行期 API 代理的詳細資訊，請參閱《 AWS Lambda 使用者指南》中的使用 AWS Lambda 執行期 API 代理延伸來增強執行期安全性和管理，以及針對自訂執行期使用 Lambda 執行期 API。

使用 AWS FIS 搭配容器執行時間

對於使用接受`AWS_LAMBDA_RUNTIME_API`環境變數的容器映像的 AWS Lambda 函數，您可以依照下列步驟，將 AWS FIS Lambda 延伸模組封裝到容器映像中：

1. 決定要從中擷取延伸模組的 layer ARN。如需如何尋找 ARN 的詳細資訊，請參閱 設定 Lambda 函數。
2. 使用 AWS Command Line Interface (CLI) 來請求有關延伸模組 的詳細資訊`aws lambda get-layer-version-by-arn --arn fis-extension-arn`。回應將包含一個`Location`欄位，其中包含預先簽章的 URL，您可以從中將 FIS 擴充功能下載為 ZIP 檔案。
3. 將延伸模組的內容解壓縮至 `/opt` Docker 檔案系統。以下是以 NodeJS Lambda 執行時間為基礎的 Dockerfile 範例：

```
# 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" ]
```

如需容器映像的詳細資訊，請參閱AWS Lambda 《 使用者指南》中的使用容器映像建立 Lambda 函數。

AWS FIS Lambda 環境變數

以下是 AWS FIS Lambda 延伸模組的環境變數清單

- `AWS_FIS_CONFIGURATION_LOCATION` - 必要。 AWS FIS 將寫入作用中錯誤組態且延伸模組將讀取錯誤組態的位置。位置應為 Amazon S3 ARN 格式，包括儲存貯體和路徑。例如 `arn:aws:s3:::my-fis-config-bucket/FisConfigs/`。
- `AWS_LAMBDA_EXEC_WRAPPER` - 必要。用於設定 AWS FIS Lambda 延伸模組的 AWS Lambda 包裝函式指令碼位置。這應該設定為 延伸模組隨附的`/opt/aws-fis/bootstrap`指令碼。
- `AWS_FIS_LOG_LEVEL` - 選用。 AWS FIS Lambda 延伸所發出訊息的日誌層級。支援的值為 `INFO`、`WARN` 和 `ERROR`。如果未設定， AWS FIS 擴充會預設為 `INFO`。
- `AWS_FIS_EXTENSION_METRICS` - 選用。可能值為 `all` 和 `none`。如果設定為 延伸`all`模組，則會在 下發出 EMF 指標`aws-fis-extension namespace`。
- `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` - 選用。如果設定 將覆寫輪詢間隔 （以秒為單位），而延伸模組未注入錯誤並等待將錯誤組態新增至組態位置。預設為 `60`。
- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT` - 選用。如果設定 會覆寫 的值`AWS_LAMBDA_RUNTIME_API`，以定義 AWS FIS 延伸模組與 AWS Lambda 執行時間 API 互動的位置，以控制函數叫用。預期 IP：PORT，例如 `127.0.0.1:9000`。如需 的詳細資訊`AWS_LAMBDA_RUNTIME_API`，請參閱《 使用者指南》中的將 Lambda 執行時間 API 用於自訂執行時間。 AWS Lambda

- `AWS_FIS_PROXY_LISTENER_PORT` - 選用。定義 AWS FIS Lambda 延伸模組公開 AWS Lambda 執行時間 API 代理的連接埠，可供另一個延伸模組或執行時間使用。預設為 `9100`。
- `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` - 選用。如果設定為非零值，此變數會定義延伸模組在評估故障組態並開始調用執行時間之前，等待傳輸中非同步輪詢完成的毫秒數。預設為 `0`。

Lambda 擴充 AWS FIS 功能的可用版本

本節包含 AWS FIS Lambda 延伸版本的相關資訊。延伸支援針對 x86-64 和 ARM64 (Graviton2) 平台開發的 Lambda 函數。您的 Lambda 函數必須設定為針對目前託管 AWS 區域 的 使用特定的 Amazon Resource Name (ARN)。您可以在下面檢視 AWS 區域 和 ARN 詳細資訊。

主題

- AWS FIS Lambda 延伸模組版本備註
- Lambda 延伸模組 ARNs的存取指南

AWS FIS Lambda 延伸模組版本備註

下表說明對 AWS FIS Lambda 延伸模組最新版本所做的變更

版本	啟動日期	備註
1.0.0	2024-10-29	初始版本

Lambda 延伸模組 ARNs的存取指南

您在 AWS 帳戶 和 AWS 區域 中必須至少有一個參數，才能使用 主控台搜尋公有參數。若要探索公有參數，請參閱在參數存放區中探索公有參數。

主控台存取：

1. 在 https：//https://console.aws.amazon.com/systems-manager/ 開啟 AWS Systems Manager 主控台。
2. 在導覽窗格中，選擇 Parameter Store (參數存放區)。
3. 選擇 Public parameters (公有參數) 索引標籤。
4. 選擇 Select a service (選取服務) 下拉式選單。從下拉式清單選項中，選擇 `fis`。

5. （選用）透過在搜尋列中輸入更多資訊來篩選您選取的參數。對於 arm64 架構，輸入 "arm64" 來篩選參數。對於 x86_64 架構，輸入 "x86_64" 來篩選參數。
6. 選擇您要使用的公有參數。
7. 從參數詳細資訊中，找到 ARN 值。複製 ARN 以用於設定目標 Lambda 函數上的層延伸。

AWS CLI 存取：

SSM 參數名稱

下列 SSM 參數名稱適用於不同的架構：

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

若要擷取擴充功能 ARNs，請使用下列 AWS CLI 命令格式，其中 parameterName 是架構的名稱，而 region 是目標 AWS 區域：

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

使用範例

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

回應格式

命令會傳回包含參數詳細資訊的 JSON 物件，如下所示。Lambda 層的 ARN 包含在參數物件的值欄位中。複製 ARN 以用於設定目標 Lambda 函數上的層延伸。

```
 {
     "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"
        }
 }
```

程式設計存取：

使用基礎設施即程式碼 (IaC) 建置或設定 Lambda 函數時，以程式設計方式擷取這些公有參數。此方法有助於使用最新的層版本 ARN 來維護 Lambda 函數，而不需要在 AWS FIS 延伸層 ARN 經過硬式編碼時進行手動程式碼更新。下列資源示範如何使用常見的 IaC 平台擷取公有參數：

- 使用 AWS SDK 取得公有參數
- 從 AWS Systems Manager 參數存放區取得公有參數
- 使用 Terraform 取得公有參數

管理 AWS FIS 實驗範本

您可以使用 AWS FIS 主控台或命令列來建立和管理實驗範本。實驗範本包含一或多個在實驗期間於指定目標上執行的動作。它還包含阻止實驗超出界限的停止條件。如需實驗範本元件的詳細資訊，請參閱實驗範本元件。建立實驗範本之後，您可以使用它來執行實驗。

任務

- 建立實驗範本
- 檢視實驗範本
- 從實驗範本產生目標預覽
- 從範本開始實驗
- 更新實驗範本
- 標記實驗範本
- 刪除實驗範本
- AWS FIS 實驗範本範例

建立實驗範本

開始之前，請先完成以下任務：

- 規劃您的實驗。
- 建立 IAM 角色，授予 AWS FIS 服務代表您執行動作的許可。如需詳細資訊，請參閱AWS FIS 實驗的 IAM 角色。
- 確保您可存取 AWS FIS。如需詳細資訊，請參閱 AWS FIS 政策範例。

使用主控台建立實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選擇建立實驗範本。
4. 針對步驟 1，指定範本詳細資訊，執行下列動作：

    a. 針對描述和名稱，輸入範本的描述，例如 `Amazon S3 Network Disrupt Connectivity`。

b. （選用）針對帳戶目標，選擇多個帳戶以設定多帳戶實驗範本。

c. 選擇下一步，然後移至步驟 2，指定動作和目標。

5. 針對動作，指定範本的動作集。針對每個動作，選擇新增動作並完成下列動作：

   - 針對名稱，輸入動作的名稱。

     允許字元為英數字元、連字號 (-) 和底線 (_)。名稱必須以字母開頭。不可使用空格。此範本中的每個動作名稱都必須是唯一的。

   - （選用）針對描述，輸入動作的描述。長度上限為 512 個字元。
   - （選用）對於 之後開始，選取此範本中定義的另一個動作，必須在目前動作開始之前完成。否則，動作會在實驗開始時執行。
   - 針對動作類型，選擇 AWS FIS 動作。
   - 針對目標，選擇您在目標區段中定義的目標。如果您尚未定義此動作的目標， AWS FIS 會為您建立新的目標。
   - 針對動作參數，指定動作的參數。只有在 AWS FIS 動作具有參數時，才會顯示本節。
   - 選擇 Save (儲存)。

6. 針對目標，定義要在其中執行動作的目標資源。您必須指定至少一個資源 ID 或一個資源標籤做為目標。選擇編輯以編輯 AWS FIS 在上一個步驟中為您建立的目標，或選擇新增目標。針對每個目標，執行下列動作：

   - 針對名稱，輸入目標的名稱。

     允許字元為英數字元、連字號 (-) 和底線 (_)。名稱必須以字母開頭。不可使用空格。每個目標名稱在此範本中必須是唯一的。

   - 針對資源類型，選擇 動作支援的資源類型。
   - 針對目標方法，執行下列其中一項：
     - 選擇資源 IDs，然後選擇或新增資源 IDs。
     - 選擇資源標籤、篩選條件和參數，然後新增您需要的標籤和篩選條件。如需詳細資訊，請參閱the section called “識別目標資源”。
   - 對於選取模式，選擇計數以在指定數量的已識別目標上執行動作，或選擇百分比在指定百分比的已識別目標上執行動作。根據預設， 動作會在所有已識別的目標上執行。
   - 選擇 Save (儲存)。

7. 若要使用您建立的目標更新動作，請在動作下尋找動作，選擇編輯，然後更新目標。您可以針對多個動作使用相同的目標。

8. （選用）針對實驗選項，選取空白目標解析模式的行為。
9. 選擇下一步以移至步驟 3，設定服務存取。
10. 針對服務存取，選擇使用現有的 IAM 角色，然後選擇您建立的 IAM 角色，如本教學課程的先決條件所述。如果您的角色未顯示，請確認其具有所需的信任關係。如需詳細資訊，請參閱the section called “實驗角色”。
11. （僅限多帳戶實驗）針對目標帳戶組態，為每個目標帳戶新增角色 ARN 和選用描述。若要使用 CSV 檔案上傳目標帳戶角色 ARNs，請選擇為所有目標帳戶上傳角色 ARNs，然後選擇選擇 .CSV 檔案
12. 選擇下一步以移至步驟 4，設定選用設定。
13. （選用）針對停止條件，選取停止條件的 Amazon CloudWatch 警示。如需詳細資訊，請參閱AWS FIS 的停止條件。
14. （選用）對於日誌，設定目的地選項。若要將日誌傳送至 S3 儲存貯體，請選擇傳送至 Amazon S3 儲存貯體，然後輸入儲存貯體名稱和字首。若要將日誌傳送至 CloudWatch Logs，請選擇傳送至 CloudWatch Logs，然後輸入日誌群組。
15. （選用）針對標籤，選擇新增標籤，並指定標籤索引鍵和標籤值。您新增的標籤會套用至實驗範本，而不是使用範本執行的實驗。
16. 選擇下一步以移至步驟 5，檢閱並建立。
17. 檢閱範本，然後選擇建立實驗範本。出現確認提示時，輸入 `create`，然後選擇建立實驗範本。

使用 CLI 建立實驗範本

使用 create-experiment-template 命令。

您可以從 JSON 檔案載入實驗範本。

使用 `--cli-input-json` 參數。

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

如需詳細資訊，請參閱AWS Command Line Interface 《使用者指南》中的產生 CLI 骨架範本。如需範本範例，請參閱 AWS FIS 實驗範本範例。

檢視實驗範本

您可以檢視您建立的實驗範本。

使用主控台檢視實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 若要檢視特定範本的相關資訊，請選取實驗範本 ID。
4. 在詳細資訊區段中，您可以檢視範本的描述和停止條件。
5. 若要檢視實驗範本的動作，請選擇動作。
6. 若要檢視實驗範本的目標，請選擇目標。
7. 若要檢視實驗範本的標籤，請選擇標籤。

使用 CLI 檢視實驗範本

使用 list-experiment-templates 命令取得實驗範本清單，並使用 get-experiment-template 命令取得特定實驗範本的相關資訊。

從實驗範本產生目標預覽

開始實驗之前，您可以產生目標預覽，以確認您的實驗範本已設定為以預期資源為目標。當您開始實際實驗時，目標資源可能與預覽中的資源不同，因為資源可能會隨機移除、更新或取樣。當您產生目標預覽時，您會開始略過所有動作的實驗。

> **Note**
>
> 產生目標預覽無法讓您驗證您是否具有對資源採取動作的必要許可。

使用主控台啟動目標預覽

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 若要檢視實驗範本的目標，請選擇目標。
4. 若要驗證實驗範本的目標資源，請選擇產生預覽。當您執行實驗時，此目標預覽會自動更新為來自最新實驗的目標。

使用 CLI 啟動目標預覽

- 執行下列 start-experiment 命令。將斜體值取代為您自己的值。

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

從範本開始實驗

建立實驗範本之後，您可以使用該範本開始實驗。

當您開始實驗時，我們會建立指定範本的快照，並使用該快照來執行實驗。因此，如果實驗範本在實驗執行時更新或刪除，這些變更不會影響執行中的實驗。

當您開始實驗時， AWS FIS 會代表您建立服務連結角色。如需詳細資訊，請參閱使用 Fault Injection Service AWS 的服務連結角色。

開始實驗後，您可以隨時停止實驗。如需詳細資訊，請參閱停止實驗。

使用主控台開始實驗

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇開始實驗。
4. （選用）若要將標籤新增至實驗，請選擇新增標籤，然後輸入標籤索引鍵和標籤值。
5. 選擇 Start experiment (開始實驗)。出現確認提示時，輸入 **start** 並選擇開始實驗。

使用 CLI 開始實驗

使用 start-experiment 命令。

更新實驗範本

您可以更新現有的實驗範本。當您更新實驗範本時，變更不會影響任何使用範本的執行中實驗。

使用主控台更新實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。

2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、更新實驗範本。
4. 視需要修改範本詳細資訊，然後選擇更新實驗範本。

使用 CLI 更新實驗範本

使用 update-experiment-template 命令。

標記實驗範本

您可以將自己的標籤套用至實驗範本，以協助您整理範本。您也可以實作標籤型 IAM 政策，以控制對實驗範本的存取。

使用主控台標記實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、管理標籤。
4. 若要新增標籤，請選擇新增標籤，然後指定金鑰和值。

   若要移除標籤，請選擇移除標籤。
5. 選擇 Save (儲存)。

使用 CLI 標記實驗範本

使用 tag-resource 命令。

刪除實驗範本

如果您不再需要實驗範本，則可以將其刪除。當您刪除實驗範本時，任何使用範本的執行中實驗都不會受到影響。實驗會持續執行，直到完成或停止為止。不過，已刪除的實驗範本無法從 主控台的實驗頁面檢視。

使用主控台刪除實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。

3. 選取實驗範本，然後選擇動作、刪除實驗範本。
4. 出現確認提示時，輸入 **delete** 並選擇刪除實驗範本。

使用 CLI 刪除實驗範本

使用 delete-experiment-template 命令。

AWS FIS 實驗範本範例

如果您使用 AWS FIS API 或命令列工具來建立實驗範本，您可以在 JavaScript 物件標記 (JSON) 中建構範本。如需實驗範本元件的詳細資訊，請參閱 AWS FIS 實驗範本元件。

若要使用其中一個範例範本建立實驗，請將其儲存到 JSON 檔案（例如，`my-template.json`），以您自己的值取代##的預留位置值，然後執行下列 create-experiment-template 命令。

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

範例範本

- 根據篩選條件停止 EC2 執行個體
- 停止指定數量的 EC2 執行個體
- 執行預先設定的 AWS FIS SSM 文件
- 執行預先定義的 Automation Runbook
- 具有目標 IAM 角色之 EC2 執行個體上的調節 API 動作
- Kubernetes 叢集中 Pod 的壓力測試 CPU

根據篩選條件停止 EC2 執行個體

下列範例會停止指定區域中所有執行中的 Amazon EC2 執行個體，並在指定的 VPC 中具有指定的標籤。它會在兩分鐘後重新啟動它們。

```
{
    "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"
}
```

停止指定數量的 EC2 執行個體

下列範例會停止具有指定標籤的三個執行個體。 AWS FIS 會選取要隨機停止的特定執行個體。它會在兩分鐘後重新啟動這些執行個體。

```
{
    "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"
}
```

執行預先設定的 AWS FIS SSM 文件

下列範例使用預先設定的 AWS FIS SSM 文件，在指定的 EC2 執行個體上執行 CPU 故障注入 60 秒，AWSFIS-Run-CPU-Stress. AWS FIS 會監控實驗 2 分鐘。

```
{
    "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"
}
```

執行預先定義的 Automation Runbook

下列範例會使用 Systems Manager、AWS-PublishSNSNotification 提供的 Runbook 將通知發佈至 Amazon SNS。 PublishSNSNotification 角色必須具有將通知發佈至指定 SNS 主題的許可。

```
{
    "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"
}
```

具有目標 IAM 角色之 EC2 執行個體上的調節 API 動作

下列範例會針對由目標定義中指定的 IAM 角色 (IAM) 發出的 API 呼叫，調節 動作定義中指定的 100% API 呼叫。

> **Note**
>
> 如果您想要將屬於 Auto Scaling 群組成員的 EC2 執行個體設為目標，請改用 aws：ec2：asg-insufficient-instance-capacity-error 動作，並使用 Auto Scaling 群組的目標。如需詳細資訊，請參閱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"
}
```

Kubernetes 叢集中 Pod 的壓力測試 CPU

下列範例使用 Chaos Mesh 來對 Amazon EKS Kubernetes 叢集中的 Pod CPU 進行壓力測試一分鐘。

```
{
    "description": "ChaosMesh StressChaos example",
    "targets": {
        "Cluster-Target-1": {
            "resourceType": "aws:eks:cluster",
            "resourceArns": [
                "arn:aws:eks:arn:aws::111122223333:cluster/cluster-id"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "TestCPUStress": {
            "actionId": "aws:eks:inject-kubernetes-custom-resource",
            "parameters": {
                "maxDuration": "PT2M",
                "kubernetesApiVersion": "chaos-mesh.org/v1alpha1",
                "kubernetesKind": "StressChaos",
                "kubernetesNamespace": "default",
                "kubernetesSpec": "{\"selector\":{\"namespaces\":[\"default\"],
\"labelSelectors\":{\"run\":\"nginx\"}},\"mode\":\"all\",\"stressors\": {\"cpu\":
{\"workers\":1,\"load\":50}},\"duration\":\"1m\"}"
            },
            "targets": {
                "Cluster": "Cluster-Target-1"
            }
        }
    },
    "stopConditions": [{
        "source": "none"
    }],
    "roleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {}
}
```

下列範例使用 Litmus 來對 Amazon EKS Kubernetes 叢集中的 Pod CPU 進行壓力測試一分鐘。

```
{
    "description": "Litmus CPU Hog",
```

```
    "targets": {
        "MyCluster": {
            "resourceType": "aws:eks:cluster",
            "resourceArns": [
                "arn:aws:eks:arn:aws::111122223333:cluster/cluster-id"
            ],
            "selectionMode": "ALL"
        }
    },
    "actions": {
        "MyAction": {
            "actionId": "aws:eks:inject-kubernetes-custom-resource",
            "parameters": {
                "maxDuration": "PT2M",
                "kubernetesApiVersion": "litmuschaos.io/v1alpha1",
                "kubernetesKind": "ChaosEngine",
                "kubernetesNamespace": "litmus",
                "kubernetesSpec": "{\"engineState\":\"active\",\"appinfo\":
{\"appns\":\"default\",\"applabel\":\"run=nginx\",\"appkind\":\"deployment\"},
\"chaosServiceAccount\":\"litmus-admin\",\"experiments\":[{\"name\":\"pod-cpu-hog
\",\"spec\":{\"components\":{\"env\":[{\"name\":\"TOTAL_CHAOS_DURATION\",\"value\":
\"60\"},{\"name\":\"CPU_CORES\",\"value\":\"1\"},{\"name\":\"PODS_AFFECTED_PERC\",
\"value\":\"100\"},{\"name\":\"CONTAINER_RUNTIME\",\"value\":\"docker\"},{\"name\":
\"SOCKET_PATH\",\"value\":\"/var/run/docker.sock\"}]},\"probe\":[]}}],\"annotationCheck
\":\"false\"}"
            },
            "targets": {
                "Cluster": "MyCluster"
            }
        }
    },
    "stopConditions": [{
        "source": "none"
    }],
    "roleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {}
}
```

管理您的 AWS FIS 實驗

AWS FIS 可讓您在 AWS 工作負載上執行故障注入實驗。若要開始使用，請建立實驗範本。建立實驗範本之後，您可以使用它來開始實驗。

發生下列其中一種情況時，實驗即完成：

- 範本中的所有動作都已成功完成。
- 會觸發停止條件。
- 由於發生錯誤，無法完成動作。例如，如果找不到目標。
- 實驗會手動停止。

您無法繼續已停止或失敗的實驗。您也無法重新執行已完成的實驗。不過，您可以從相同的實驗範本開始新的實驗。您可以選擇性地更新實驗範本，然後再於新實驗中再次指定。

任務

- 開始實驗
- 檢視您的實驗
- 標記實驗
- 停止實驗
- 列出已解析的目標

開始實驗

您可以從實驗範本開始實驗。如需詳細資訊，請參閱從範本開始實驗。

您可以使用 將實驗排程為一次性任務或重複性任務 Amazon EventBridge。如需詳細資訊，請參閱教學課程：排程重複實驗。

您可以使用下列任何功能來監控實驗：

- 在 AWS FIS 主控台中檢視您的實驗。如需詳細資訊，請參閱檢視您的實驗。
- 檢視實驗中目標資源的 Amazon CloudWatch 指標，或檢視 AWS FIS 用量指標。如需詳細資訊，請參閱使用 CloudWatch 監控 。

- 啟用實驗記錄，以在實驗執行時擷取實驗的詳細資訊。如需更多資訊，請參閱實驗記錄。

檢視您的實驗

您可以檢視執行中實驗的進度，也可以檢視已完成、已停止或失敗的實驗。

停止、完成和失敗的實驗會在 120 天後自動從您的帳戶中移除。

使用主控台檢視實驗

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗。
3. 選擇實驗的實驗 ID 以開啟其詳細資訊頁面。
4. 執行下列其中一項或多項：
   - 檢查詳細資訊、實驗狀態的狀態。 ???
   - 選擇動作索引標籤以取得實驗動作的相關資訊。
   - 選擇目標索引標籤以取得實驗目標的相關資訊。
   - 選擇時間軸索引標籤，根據動作的開始和結束時間以視覺化方式呈現動作。

使用 CLI 檢視實驗

使用 list-experiments 命令取得實驗清單，並使用 get-experiment 命令取得特定實驗的相關資訊。

實驗狀態

實驗可以處於下列其中一種狀態：

- 待定 – 實驗待定。
- 啟動 – 實驗正在準備開始。
- 執行中 – 實驗正在執行中。
- 已完成 – 實驗中的所有動作都已成功完成。
- 停止 – 觸發停止條件或手動停止實驗。
- 已停止 – 實驗中的所有執行中或待定動作都會停止。
- 失敗 – 實驗因錯誤而失敗，例如許可不足或語法不正確。

- 已取消 – 由於已參與的安全控制桿，實驗已停止或無法啟動。

動作狀態

動作可以處於下列其中一種狀態：

- 待定 – 動作處於待定狀態，因為實驗尚未開始，或動作稍後會在實驗中開始。
- 啟動 – 動作正在準備啟動。
- 執行中 – 動作正在執行中。
- 已完成 – 動作已成功完成。
- 已取消 – 實驗在動作開始之前停止。
- 已略過 – 動作已略過。
- 停止 – 動作正在停止。
- 已停止 – 實驗中的所有執行中或待定動作都會停止。
- 失敗 – 動作因用戶端錯誤而失敗，例如許可不足或語法不正確。

標記實驗

您可以將標籤套用至實驗，以協助您整理它們。您也可以實作標籤型 IAM 政策來控制對實驗的存取。

使用主控台標記實驗

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗。
3. 選取實驗，然後選擇動作、管理標籤。
4. 若要新增標籤，請選擇新增標籤，然後指定索引鍵和值。

   若要移除標籤，請為標籤選擇移除。
5. 選擇儲存。

使用 CLI 標記實驗

使用 tag-resource 命令。

停止實驗

您可以隨時停止執行中的實驗。當您停止實驗時，任何尚未完成動作的後置動作都會在實驗停止之前完成。您無法繼續已停止的實驗。

使用主控台停止實驗

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗。
3. 選取實驗，然後選擇停止實驗。
4. 在確認對話方塊中，選擇停止實驗。

使用 CLI 停止實驗

使用 stop-experiment 命令。

列出已解析的目標

您可以在目標解析結束後檢視實驗已解析目標的資訊。

使用主控台檢視已解析的目標

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗。
3. 選取實驗，然後選擇報告。
4. 在 資源下檢視已解析的目標資訊。

使用 CLI 檢視已解析的目標

使用 list-experiment-resolved-targets 命令。

Fault Injection Service AWS 教學課程

下列教學課程說明如何使用 Fault Injection Service (AWS FIS) AWS 建立和執行實驗。

教學課程

- 教學課程：測試執行個體停止和開始使用 AWS FIS
- 教學課程：使用 AWS FIS 在執行個體上執行 CPU 應力
- 教學課程：使用 AWS FIS 測試 Spot 執行個體中斷
- 教學課程：模擬連線事件
- 教學課程：排程重複實驗

教學課程：測試執行個體停止和開始使用 AWS FIS

您可以使用 AWS Fault Injection Service (AWS FIS) 來測試應用程式如何處理執行個體停止和啟動。使用此教學課程來建立實驗範本，該範本使用 AWS FIS `aws:ec2:stop-instances`動作來停止一個執行個體，然後停止第二個執行個體。

先決條件

若要完成本教學課程，請確定您執行下列動作：

- 在帳戶中啟動兩個測試 EC2 執行個體。啟動執行個體後，請注意兩個執行個體IDs。
- 建立可讓 AWS FIS 服務代表您執行`aws:ec2:stop-instances`動作的 IAM 角色。如需詳細資訊，請參閱AWS FIS 實驗的 IAM 角色。
- 確保您可存取 AWS FIS。如需詳細資訊，請參閱 AWS FIS 政策範例。

步驟 1：建立實驗範本

使用 AWS FIS 主控台建立實驗範本。在範本中，您可以指定兩個動作，每個動作會依序執行三分鐘。第一個動作會停止其中一個測試執行個體，由 AWS FIS 隨機選擇。第二個動作會停止兩個測試執行個體。

建立實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。

2. 在導覽窗格中，選擇實驗範本。
3. 選擇建立實驗範本。
4. 針對步驟 1，指定範本詳細資訊，執行下列動作：
   a. 針對描述和名稱，輸入範本的描述，例如 `Amazon S3 Network Disrupt Connectivity`。
   b. 選擇下一步，然後移至步驟 2，指定動作和目標。
5. 對於 Actions (動作)，執行下列動作：
   a. 選擇新增動作。
   b. 輸入動作的名稱。例如，輸入 **`stopOneInstance`**。
   c. 針對動作類型，選擇 aws：ec2：stop-instances。
   d. 對於目標，保留 AWS FIS 為您建立的目標。
   e. 對於動作參數，在持續時間後啟動執行個體，請指定 3 分鐘 (PT3M)。
   f. 選擇 Save (儲存)。
6. 對於 Targets (目標)，執行下列動作：
   a. 針對 AWS 上一個步驟中 FIS 自動為您建立的目標，選擇編輯。
   b. 以更描述性的名稱取代預設名稱。例如，輸入 **`oneRandomInstance`**。
   c. 確認資源類型為 aws：ec2：instance。
   d. 針對目標方法，選擇資源 IDs，然後選擇兩個測試執行個體的 IDs。
   e. 針對選取模式，選擇計數。針對資源數量，輸入 **`1`**。
   f. 選擇 Save (儲存)。
7. 選擇新增目標並執行下列動作：
   a. 輸入目標的名稱。例如，輸入 **`bothInstances`**。
   b. 針對資源類型，選擇 aws：ec2：instance。
   c. 針對目標方法，選擇資源 IDs，然後選擇兩個測試執行個體的 IDs。
   d. 針對選取模式，選擇全部。
   e. 選擇 Save (儲存)。
8. 從動作區段中，選擇新增動作。請執行下列操作：
   a. 針對名稱，輸入動作的名稱。例如，輸入 **`stopBothInstances`**。
   b. 針對動作類型，選擇 aws：ec2：stop-instances。

c. 針對之後開始，選擇您新增的第一個動作 (**stopOneInstance**)。

d. 針對目標，選擇您新增的第二個目標 (**bothInstances**)。

e. 對於動作參數，在持續時間後啟動執行個體，請指定 3 分鐘 (PT3M)。

f. 選擇 Save (儲存)。

9. 選擇下一步以移至步驟 3，設定服務存取。
10. 針對服務存取，選擇使用現有的 IAM 角色，然後選擇您建立的 IAM 角色，如本教學課程的先決條件所述。如果您的角色未顯示，請確認其具有所需的信任關係。如需詳細資訊，請參閱the section called “實驗角色”。
11. 選擇下一步以移至步驟 4，設定選用設定。
12. （選用）針對標籤，選擇新增標籤，並指定標籤索引鍵和標籤值。您新增的標籤會套用至實驗範本，而不是使用範本執行的實驗。
13. 選擇下一步以移至步驟 5，檢閱並建立。
14. 檢閱範本，然後選擇建立實驗範本。出現確認提示時，輸入 `create`，然後選擇建立實驗範本。

（選用）檢視實驗範本 JSON

選擇匯出索引標籤。以下是上述主控台程序所建立 JSON 的範例。

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

步驟 2：開始實驗

完成建立實驗範本後，您可以使用它來開始實驗。

開始實驗

1. 您應該位於您剛建立之實驗範本的詳細資訊頁面。否則，請選擇實驗範本，然後選擇實驗範本的 ID 以開啟詳細資訊頁面。
2. 選擇 Start experiment (開始實驗)。
3. （選用）若要將標籤新增至實驗，請選擇新增標籤，然後輸入標籤索引鍵和標籤值。

4. 選擇 Start experiment (開始實驗)。出現確認提示時，輸入 **start** 並選擇開始實驗。

步驟 3：追蹤實驗進度

您可以追蹤執行中實驗的進度，直到實驗完成、停止或失敗為止。

追蹤實驗的進度

1. 您應該位於您剛開始之實驗的詳細資訊頁面。否則，請選擇實驗，然後選擇實驗的 ID 以開啟詳細資訊頁面。
2. 若要檢視實驗狀態，請在詳細資訊窗格中檢查狀態。如需詳細資訊，請參閱實驗狀態。
3. 當實驗的狀態正在執行時，請前往下一個步驟。

步驟 4：驗證實驗結果

您可以驗證執行個體是否如預期由實驗停止和啟動。

驗證實驗的結果

1. 在新的瀏覽器索引標籤或視窗中開啟 Amazon EC2 主控台，網址為 https://console.aws.amazon.com/ec2/：//。這可讓您繼續在 AWS FIS 主控台中追蹤實驗進度，同時在 Amazon EC2 主控台中檢視實驗結果。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 當第一個動作的狀態從待定變更為執行中 (AWS FIS 主控台） 時，其中一個目標執行個體的狀態會從執行中變更為已停止 (Amazon EC2 主控台）。
4. 三分鐘後，第一個動作的狀態會變更為已完成、第二個動作的狀態會變更為執行中，而其他目標執行個體的狀態會變更為已停止。
5. 三分鐘後，第二個動作的狀態會變更為已完成，目標執行個體的狀態會變更為正在執行，而實驗的狀態會變更為已完成。

步驟 5：清除

如果您不再需要您為此實驗建立的測試 EC2 執行個體，您可以終止它們。

終止執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取兩個測試執行個體，然後選取 Instance state (執行個體狀態)、Terminate instance (終止執行個體)。
4. 出現確認提示時，請選擇終止。

如果您不再需要實驗範本，則可以將其刪除。

使用 AWS FIS 主控台刪除實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、刪除實驗範本。
4. 出現確認提示時，輸入 **delete**，然後選擇刪除實驗範本。

教學課程：使用 AWS FIS 在執行個體上執行 CPU 應力

您可以使用 AWS Fault Injection Service (AWS FIS) 來測試應用程式如何處理 CPU 壓力。使用此教學課程來建立實驗範本，該範本使用 AWS FIS 來執行在執行個體上執行 CPU 應力的預先設定 SSM 文件。當執行個體的 CPU 使用率超過設定的閾值時，教學課程會使用停止條件來停止實驗。

如需詳細資訊，請參閱the section called “預先設定的 AWS FIS SSM 文件”。

先決條件

在使用 AWS FIS 執行 CPU 應力之前，請先完成下列先決條件。

建立 IAM 角色

建立角色並連接政策，讓 AWS FIS 代表您使用 `aws:ssm:send-command`動作。如需詳細資訊，請參閱AWS FIS 實驗的 IAM 角色。

驗證對 AWS FIS 的存取

確保您可存取 AWS FIS。如需詳細資訊，請參閱 AWS FIS 政策範例。

準備測試 EC2 執行個體

- 根據預先設定的 SSM 文件，使用 Amazon Linux 2 或 Ubuntu 啟動 EC2 執行個體。

- 執行個體必須由 SSM 管理。若要確認執行個體是由 SSM 管理，請開啟 Fleet Manager 主控台。如果執行個體不是由 SSM 管理，請確認 SSM Agent 已安裝，且執行個體具有與 AmazonSSMManagedInstanceCore 政策連接的 IAM 角色。若要驗證已安裝的 SSM Agent，請連線至您的執行個體並執行下列命令。

  Amazon Linux 2

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

  Ubuntu

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

- 啟用執行個體的詳細監控。這會在 1 分鐘內提供資料，需額外付費。選取執行個體，然後選擇動作、監控和故障診斷、管理詳細監控。

步驟 1：為停止條件建立 CloudWatch 警示

設定 CloudWatch 警示，以便在 CPU 使用率超過您指定的閾值時停止實驗。下列程序會將目標執行個體的閾值設定為 50% CPU 使用率。如需詳細資訊，請參閱停止條件。

建立警示，指出 CPU 使用率何時超過閾值

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取目標執行個體，然後選擇動作、監控和故障診斷、管理 CloudWatch 警示。
4. 對於警示通知，請使用切換來關閉 Amazon SNS 通知。
5. 對於警示閾值，請使用下列設定：
   - 分組範例依據：上限
   - 要範例的資料類型：CPU 使用率
   - 百分比：**`50`**
   - 期間：**`1 Minute`**
6. 當您完成設定警示時，請選擇建立。

步驟 2：建立實驗範本

使用 AWS FIS 主控台建立實驗範本。在範本中，您可以指定要執行的下列動作：aws：ssm：send-command/AWSFIS-Run-CPU-Stress。

建立實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選擇建立實驗範本。
4. 針對步驟 1，指定範本詳細資訊，執行下列動作：
   a. 針對描述和名稱，輸入範本的描述。
   b. 選擇下一步，然後移至步驟 2，指定動作和目標。
5. 對於 Actions (動作)，執行下列動作：
   a. 選擇新增動作。
   b. 輸入動作的名稱。例如，輸入 **runCpuStress**。
   c. 針對動作類型，選擇 aws：ssm：send-command/AWSFIS-Run-CPU-Stress。這會自動將 SSM 文件的 ARN 新增至文件 ARN。
   d. 對於目標，保留 AWS FIS 為您建立的目標。
   e. 針對動作參數、文件參數，輸入下列內容：

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

   f. 針對動作參數、持續時間，指定 5 分鐘 (PT5M)。
   g. 選擇 Save (儲存)。
6. 對於 Targets (目標)，執行下列動作：
   a. 針對 AWS 上一個步驟中 FIS 自動為您建立的目標，選擇編輯。
   b. 以更描述性的名稱取代預設名稱。例如，輸入 **testInstance**。
   c. 確認資源類型為 aws：ec2：instance。
   d. 針對目標方法，選擇資源 IDs，然後選擇測試執行個體的 ID。
   e. 針對選取模式，選擇全部。
   f. 選擇 Save (儲存)。

7. 選擇下一步以移至步驟 3，設定服務存取。
8. 針對服務存取，選擇使用現有的 IAM 角色，然後選擇您建立的 IAM 角色，如本教學課程的先決條件所述。如果您的角色未顯示，請確認其具有所需的信任關係。如需詳細資訊，請參閱the section called “實驗角色”。
9. 選擇下一步以移至步驟 4，設定選用設定。
10. 針對停止條件，選取您在步驟 1 中建立的 CloudWatch 警示。
11. （選用）針對標籤，選擇新增標籤並指定標籤索引鍵和標籤值。您新增的標籤會套用至實驗範本，而不是使用範本執行的實驗。
12. 選擇下一步以移至步驟 5，檢閱並建立。
13. 檢閱範本，然後選擇建立實驗範本。出現確認提示時，輸入 `create`，然後選擇建立實驗範本。

（選用）檢視實驗範本 JSON

選擇匯出索引標籤。以下是上述主控台程序所建立 JSON 的範例。

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

```
        {
            "source": "aws:cloudwatch:alarm",
            "value": "arn:aws:cloudwatch:region:123456789012:alarm:awsec2-instance_id-
GreaterThanOrEqualToThreshold-CPUUtilization"
        }
    ],
    "roleArn": "arn:aws:iam::123456789012:role/AllowFISSSMActions",
    "tags": {}
}
```

步驟 3：開始實驗

完成建立實驗範本後，您可以使用它來開始實驗。

開始實驗

1. 您應該位於您剛建立之實驗範本的詳細資訊頁面。否則，請選擇實驗範本，然後選擇實驗範本的 ID 以開啟詳細資訊頁面。
2. 選擇 Start experiment (開始實驗)。
3. （選用）若要將標籤新增至實驗，請選擇新增標籤並輸入標籤索引鍵和標籤值。
4. 選擇 Start experiment (開始實驗)。出現確認提示時，請按一下 **start**。選擇 Start experiment (開始實驗)。

步驟 4：追蹤實驗進度

您可以追蹤執行中實驗的進度，直到實驗完成、停止或失敗為止。

追蹤實驗的進度

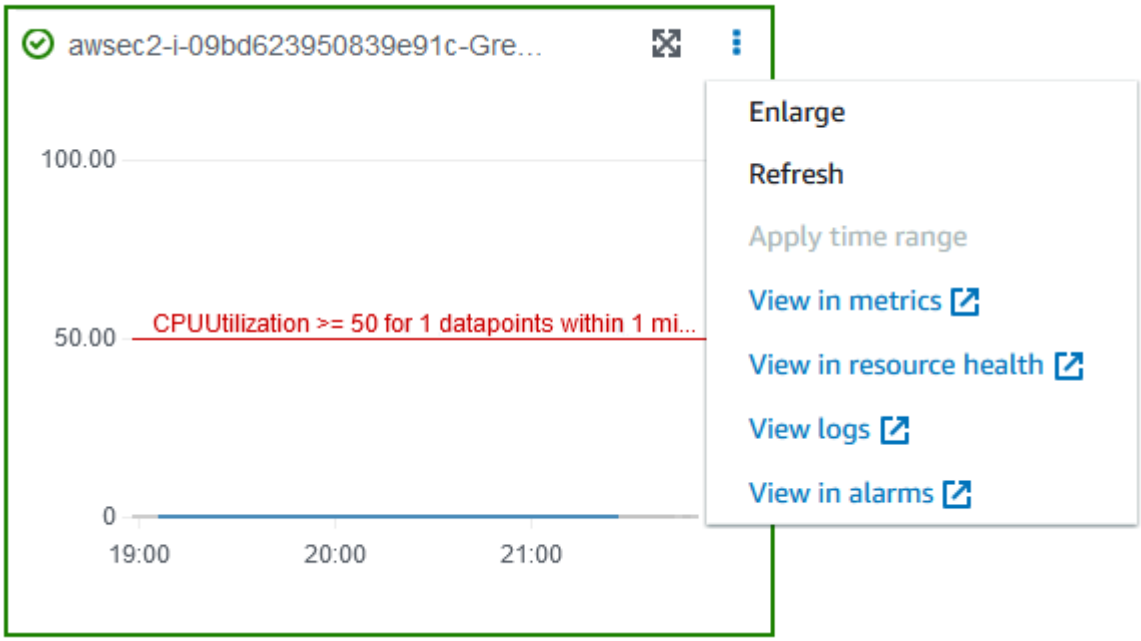
1. 您應該位於您剛開始之實驗的詳細資訊頁面。否則，請選擇實驗，然後選擇實驗的 ID 以開啟實驗的詳細資訊頁面。
2. 若要檢視實驗狀態，請在詳細資訊窗格中檢查狀態。如需詳細資訊，請參閱實驗狀態。
3. 當實驗狀態正在執行時，請前往下一個步驟。

步驟 5：驗證實驗結果

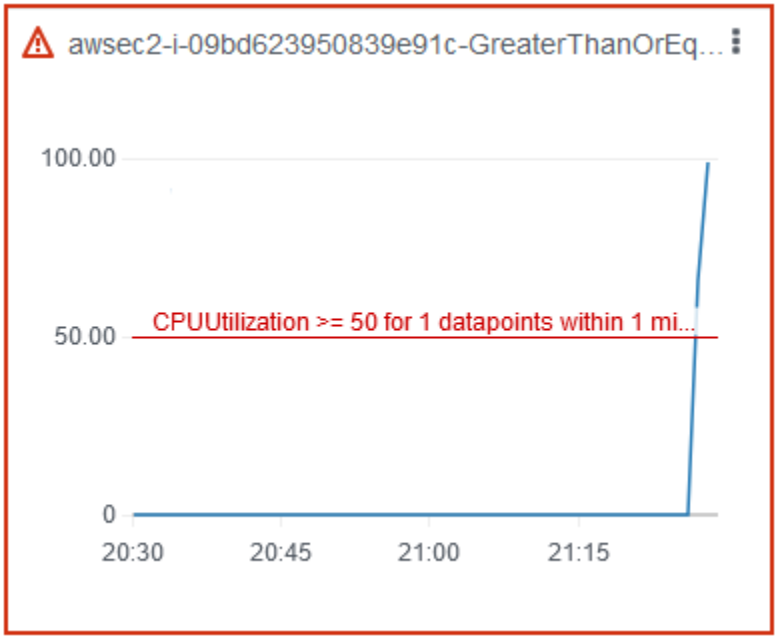
您可以在實驗執行時監控執行個體的 CPU 使用率。當 CPU 使用率達到閾值時，會觸發警示，且實驗會因停止條件而停止。

驗證實驗的結果

1. 選擇停止條件索引標籤。綠色邊界和綠色核取記號圖示表示警示的初始狀態為 OK。紅線表示警示閾值。如果您偏好更詳細的圖形，請從小工具功能表中選擇放大。



2. 當 CPU 使用率超過閾值時，停止條件索引標籤中的紅色邊界和紅色驚嘆號圖示表示警示狀態已變更為 ALARM。在詳細資訊窗格中，實驗的狀態為停止。如果您選取 狀態，則顯示的訊息為「停止條件停止實驗」。



3. 當 CPU 使用率低於閾值時，綠色邊界和綠色核取記號圖示表示警示狀態已變更為 OK。

4. （選用）從小工具功能表中選擇在警示中檢視。這會在 CloudWatch 主控台中開啟警示詳細資訊頁面，您可以在其中取得警示的詳細資訊或編輯警示設定。

步驟 6：清除

如果您不再需要您為此實驗建立的測試 EC2 執行個體，您可以將其終止。

若要終止執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取測試執行個體，然後選擇執行個體狀態、終止執行個體。
4. 出現確認提示時，請選擇終止。

如果您不再需要實驗範本，則可以將其刪除。

使用 AWS FIS 主控台刪除實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、刪除實驗範本。
4. 出現確認提示時，輸入 **`delete`**，然後選擇刪除實驗範本。

教學課程：使用 AWS FIS 測試 Spot 執行個體中斷

Spot 執行個體使用可用的備用 EC2 容量，相較於隨需定價，最多可獲得 90% 的折扣。不過，Amazon EC2 可以在需要恢復容量時中斷 Spot 執行個體。使用 Spot 執行個體時，您必須準備好解決潛在的中斷。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 Spot 執行個體中斷。

您可以使用 AWS Fault Injection Service (AWS FIS) 來測試應用程式如何處理 Spot 執行個體中斷。使用此教學課程來建立實驗範本，該範本使用 AWS FIS `aws:ec2:send-spot-instance-interruptions`動作來中斷其中一個 Spot 執行個體。

或者，若要使用 Amazon EC2 主控台啟動實驗，請參閱《Amazon EC2 使用者指南》中的啟動 Spot 執行個體中斷。

先決條件

您必須先完成下列先決條件，才能使用 AWS FIS 中斷 Spot 執行個體。

1. 建立 IAM 角色

建立角色並連接政策，讓 AWS FIS 代表您執行`aws:ec2:send-spot-instance-interruptions`動作。如需詳細資訊，請參閱AWS FIS 實驗的 IAM 角色。

2. 驗證對 AWS FIS 的存取

確保您可存取 AWS FIS。如需詳細資訊，請參閱 AWS FIS 政策範例。

3. （選用）建立 Spot 執行個體請求

如果您希望在此實驗中使用新的 Spot 執行個體，請使用 run-instances 命令來請求 Spot 執行個體。預設為終止中斷的 Spot 執行個體。如果您將中斷行為設定為 `stop`，您還必須將 類型設定為 `persistent`。在本教學課程中，請勿將中斷行為設定為 `hibernate`，因為休眠程序會立即開始。

```
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` 檔案的範例。

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

範例命令中的 `--query`選項使其能夠讓命令只傳回 Spot 執行個體的執行個體 ID。下列為範例輸出。

```
[
```

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

4. 新增標籤，讓 AWS FIS 可以識別目標 Spot 執行個體

使用 create-tags 命令將標籤新增至Name=interruptMe您的目標 Spot 執行個體。

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

步驟 1：建立實驗範本

使用 AWS FIS 主控台建立實驗範本。在 範本中，您可以指定要執行的動作。動作會中斷具有指定標籤的 Spot 執行個體。如果有一個以上的 Spot 執行個體具有標籤， AWS FIS 會隨機選擇其中一個。

建立實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選擇建立實驗範本。
4. 針對步驟 1，指定範本詳細資訊，執行下列動作：
   a. 針對描述和名稱，輸入範本的描述和名稱。
   b. 選擇下一步，然後移至步驟 2，指定動作和目標。
5. 對於 Actions (動作)，執行下列動作：
   a. 選擇新增動作。
   b. 輸入動作的名稱。例如，輸入 **`interruptSpotInstance`**。
   c. 針對動作類型，選擇 aws：ec2：send-spot-instance-interruptions。
   d. 對於目標，保留 AWS FIS 為您建立的目標。
   e. 對於動作參數，中斷之前的持續時間，指定 2 分鐘 (PT2M)。
   f. 選擇 Save (儲存)。
6. 對於 Targets (目標)，執行下列動作：
   a. 針對 AWS FIS 在上一個步驟中自動為您建立的目標，選擇編輯。
   b. 以更描述性的名稱取代預設名稱。例如，輸入 **`oneSpotInstance`**。

c. 確認資源類型為 aws：ec2：spot-instance。

d. 針對目標方法，選擇資源標籤、篩選條件和參數。

e. 針對資源標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。使用您新增至 Spot 執行個體的標籤來中斷，如本教學課程的先決條件所述。

f. 針對資源篩選條件，選擇新增篩選條件，然後輸入 **`State.Name`** 做為路徑，輸入 **`running`**做為值。

g. 針對選取模式，選擇計數。針對資源數量，輸入 **1**。

h. 選擇 Save (儲存)。

7. 選擇下一步以移至步驟 3，設定服務存取。
8. 針對服務存取，選擇使用現有的 IAM 角色，然後選擇您建立的 IAM 角色，如本教學課程的先決條件所述。如果您的角色未顯示，請確認其具有所需的信任關係。如需詳細資訊，請參閱the section called “實驗角色”。
9. 選擇下一步以移至步驟 4，設定選用設定。
10. （選用）針對標籤，選擇新增標籤，並指定標籤索引鍵和標籤值。您新增的標籤會套用至實驗範本，而不是使用範本執行的實驗。
11. 選擇下一步以移至步驟 5，檢閱並建立。
12. 檢閱範本，然後選擇建立實驗範本。出現確認提示時，輸入 `create`，然後選擇建立實驗範本。

（選用）檢視實驗範本 JSON

選擇匯出索引標籤。以下是上述主控台程序所建立 JSON 的範例。

```
{
    "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"
    }
}
```

步驟 2：開始實驗

完成建立實驗範本後，您可以使用它來開始實驗。

開始實驗

1. 您應該位於您剛建立之實驗範本的詳細資訊頁面。否則，請選擇實驗範本，然後選擇實驗範本的 ID 以開啟詳細資訊頁面。
2. 選擇 Start experiment (開始實驗)。
3. （選用）若要將標籤新增至實驗，請選擇新增標籤並輸入標籤索引鍵和標籤值。
4. 選擇 Start experiment (開始實驗)。出現確認提示時，輸入 **start** 並選擇開始實驗。

步驟 3：追蹤實驗進度

您可以追蹤執行中實驗的進度，直到實驗完成、停止或失敗為止。

追蹤實驗的進度

1. 您應該位於您剛開始之實驗的詳細資訊頁面。否則，請選擇實驗，然後選擇實驗的 ID 以開啟詳細資訊頁面。
2. 若要檢視實驗狀態，請在詳細資訊窗格中檢查狀態。如需詳細資訊，請參閱實驗狀態。
3. 當實驗的狀態正在執行時，請前往下一個步驟。

步驟 4：驗證實驗結果

此實驗的動作完成時，會發生下列情況：

- 目標 Spot 執行個體會收到執行個體重新平衡建議。
- Spot 執行個體中斷通知會在 Amazon EC2 終止或停止執行個體的兩分鐘前發出。
- 兩分鐘後，Spot 執行個體會終止或停止。
- FIS AWS 停止的 Spot 執行個體會保持停止狀態，直到您重新啟動為止。

驗證執行個體是否被實驗中斷

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 從導覽窗格中，在單獨的瀏覽器索引標籤或視窗中開啟 Spot Requests (Spot 請求) 和 Instances (執行個體)。
3. 對於 Spot Requests (Spot 請求)，選取 Spot 執行個體請求。起始狀態為 `fulfilled`。實驗完成後，狀態會變更，如下所示：
   - `terminate` - 狀態變更為 `instance-terminated-by-experiment`。
   - `stop` - 狀態會變更為 `marked-for-stop-by-experiment`，然後變更為 `instance-stopped-by-experiment`。
4. 對於 Instances (執行個體)，選取 Spot 執行個體。起始狀態為 `Running`。收到 Spot 執行個體中斷通知後兩分鐘，狀態會變更，如下所示：
   - `stop` - 狀態會變更為 `Stopping`，然後變更為 `Stopped`。
   - `terminate` - 狀態會變更為 `Shutting-down`，然後變更為 `Terminated`。

步驟 5：清除

如果您為此實驗使用 的中斷行為建立測試 Spot 執行個體，`stop`且不再需要它，您可以取消 Spot 執行個體請求並終止 Spot 執行個體。

使用 取消請求並終止執行個體 AWS CLI

1. 使用 cancel-spot-instance-requests 命令來取消 Spot 執行個體請求。

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

2. 使用 terminate-instances 命令來終止執行個體。

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

如果您不再需要實驗範本，則可以將其刪除。

使用 AWS FIS 主控台刪除實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、刪除實驗範本。
4. 出現確認提示時，輸入 **`delete`**，然後選擇刪除實驗範本。

教學課程：模擬連線事件

您可以使用 AWS Fault Injection Service (AWS FIS) 來模擬各種連線事件。 AWS FIS 透過下列其中一種方式封鎖網路連線來模擬連線事件：

- `all` – 拒絕所有進出子網路的流量。請注意，此選項允許子網路內流量，包括往返子網路中網路介面的流量。
- `availability-zone` – 拒絕其他可用區域中往返子網路的 VPC 內流量。
- `dynamodb` – 拒絕目前區域中 DynamoDB 區域端點的往返流量。
- `prefix-list` – 拒絕往返指定字首清單的流量。
- `s3` – 拒絕目前區域中 Amazon S3 區域端點的往返流量。
- `vpc` – 拒絕傳入和離開 VPC 的流量。

使用此教學課程來建立實驗範本，該範本使用 AWS FIS `aws:network:disrupt-connectivity`動作在目標子網路中引入與 Amazon S3 的連線中斷。

主題

- 先決條件
- 步驟 1：建立 AWS FIS 實驗範本
- 步驟 2：Ping Amazon S3 端點
- 步驟 3：啟動您的 AWS FIS 實驗
- 步驟 4：追蹤您的 AWS FIS 實驗進度
- 步驟 5：驗證 Amazon S3 網路中斷
- 步驟 5：清除

先決條件

開始本教學課程之前，您需要在 中具有適當許可的角色 AWS 帳戶，以及測試 Amazon EC2 執行個體：

在您的 中具有許可的角色 AWS 帳戶

建立角色並連接政策，讓 AWS FIS 代表您執行`aws:network:disrupt-connectivity`動作。

您的 IAM 角色需要下列政策：

- AWSFaultInjectionSimulatorNetworkAccess – 在 Amazon EC2 網路和其他必要服務中授予 AWS FIS 服務許可，以執行與網路基礎設施相關的 AWS FIS 動作。

> **Note**
>
> 為了簡化，本教學課程使用 AWS 受管政策。對於生產用途，我們建議您改為僅授予使用案例所需的最低許可。
> 如需如何建立 IAM 角色的詳細資訊，請參閱《IAM 使用者指南》中的 AWS FIS 實驗的 IAM 角色 (AWS CLI) 或建立 IAM 角色（主控台）。

測試 Amazon EC2 執行個體

啟動並連線至測試 Amazon EC2 執行個體。您可以使用下列教學課程來啟動並連線至 Amazon EC2 執行個體：教學課程：Amazon EC2 使用者指南中的 Amazon EC2 Linux 執行個體入門。 Amazon EC2

步驟 1：建立 AWS FIS 實驗範本

使用 AWS FIS 建立實驗範本 AWS Management Console。 AWS FIS 範本是由動作、目標、停止條件和實驗角色所組成。如需範本運作方式的詳細資訊，請參閱 AWS FIS 的實驗範本。

開始之前，請確定您已備妥下列項目：

- 具有正確許可的 IAM 角色。
- Amazon EC2 執行個體。
- Amazon EC2 執行個體的子網路 ID。

建立實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在左側導覽窗格中，選擇實驗範本。
3. 選擇建立實驗範本。
4. 針對步驟 1，指定範本詳細資訊，執行下列動作：
   a. 針對描述和名稱，輸入範本的描述，例如 `Amazon S3 Network Disrupt Connectivity`。
   b. 選擇下一步，然後移至步驟 2，指定動作和目標。
5. 在動作下，選擇新增動作。
   a. 針對名稱，輸入 `disruptConnectivity`。
   b. 針對動作類型，選取 aws：network：disrupt-connectivity。
   c. 在動作參數下，將持續時間設定為 `2 minutes`。
   d. 在範圍下，選取 s3。
   e. 在頂端，選擇儲存。
6. 在目標下，您應該會看到自動建立的目標。選擇編輯。
   a. 確認資源類型為 `aws:ec2:subnet`。
   b. 在目標方法下，選取資源 IDs，然後在先決條件步驟中選擇您在建立 Amazon EC2 執行個體時使用的子網路。

c. 確認選取模式為全部。

d. 選擇 Save (儲存)。

7. 選擇下一步以移至步驟 3，設定服務存取。
8. 在服務存取下，選取您建立的 IAM 角色，如本教學課程的先決條件所述。如果您的角色未顯示，請確認其具有所需的信任關係。如需詳細資訊，請參閱the section called “實驗角色”。
9. 選擇下一步以移至步驟 4，設定選用設定。
10. （選用）在停止條件下，您可以選取 CloudWatch 警示，以在條件發生時停止實驗。如需詳細資訊，請參閱 FIS AWS 的停止條件。
11. （選用）在日誌下，您可以選取 Amazon S3 儲存貯體，或將日誌傳送至 CloudWatch 進行實驗。
12. 選擇下一步以移至步驟 5，檢閱並建立。
13. 檢閱範本，然後選擇建立實驗範本。出現確認提示時，輸入 `create`，然後選擇建立實驗範本。

步驟 2：Ping Amazon S3 端點

確認您的 Amazon EC2 執行個體能夠到達 Amazon S3 端點。

1. 連線至您在先決條件步驟中建立的 Amazon EC2 執行個體。

   如需故障診斷，請參閱《Amazon EC2 使用者指南》中的連線至執行個體的故障診斷。

2. 檢查 以查看 AWS 區域 執行個體所在的 。您可以在 Amazon EC2 主控台或執行下列命令來執行此操作。

```
hostname
```

   例如，如果您在 中啟動 Amazon EC2 執行個體`us-west-2`，您會看到下列輸出。

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

3. 在 中 Ping Amazon S3 端點 AWS 區域。將 *AWS ##* 取代為您的區域。

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

   對於輸出，您應該會看到封包遺失 0% 的成功 ping，如下列範例所示。

```
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
```

步驟 3：啟動您的 AWS FIS 實驗

使用您剛建立的實驗範本開始實驗。

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在左側導覽窗格中，選擇實驗範本。
3. 選取您建立的實驗範本 ID，以開啟其詳細資訊頁面。
4. 選擇 Start experiment (開始實驗)。
5. （選用）在確認頁面中，為您的實驗新增標籤。
6. 在確認頁面中，選擇開始實驗。

步驟 4：追蹤您的 AWS FIS 實驗進度

您可以追蹤執行中實驗的進度，直到實驗完成、停止或失敗為止。

1. 您應該位於您剛開始之實驗的詳細資訊頁面。如果不是，請選擇實驗，然後選擇實驗的 ID 以開啟其詳細資訊頁面。
2. 若要檢視實驗狀態，請在詳細資訊窗格中檢查狀態。如需詳細資訊，請參閱實驗狀態。
3. 當實驗的狀態正在執行時，請移至下一個步驟。

步驟 5：驗證 Amazon S3 網路中斷

您可以透過 ping Amazon S3 端點來驗證實驗進度。

- 從您的 Amazon EC2 執行個體中，ping 您 中的 Amazon S3 端點 AWS 區域。將 *AWS ##* 取代為您的區域。

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

  對於輸出，您應該會看到封包遺失 100% 的失敗 ping，如下列範例所示。

```
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
```

步驟 5：清除

如果您不再需要您為此實驗或 AWS FIS 範本建立的 Amazon EC2 執行個體，則可以將其移除。

移除 Amazon EC2 執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Instances (執行個體)。
3. 選取測試執行個體，選擇執行個體狀態，然後選擇終止執行個體。
4. 出現確認提示時，請選擇終止。

使用 AWS FIS 主控台刪除實驗範本

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、刪除實驗範本。
4. 出現確認提示時，輸入 `delete`，然後選擇刪除實驗範本。

教學課程：排程重複實驗

透過 Fault Injection Service AWS (AWS FIS)，您可以對 AWS 工作負載執行故障注入實驗。這些實驗會在範本上執行，其中包含一或多個要在指定目標上執行的動作。當您也使用 時 Amazon EventBridge，您可以將實驗排程為一次性任務或重複性任務。

使用此教學課程來建立 EventBridge 排程，每 5 分鐘執行一次 AWS FIS 實驗範本。

任務

- 先決條件

- 步驟 1：建立 IAM 角色和政策
- 步驟 2：建立 Amazon EventBridge 排程器
- 步驟 3：驗證您的實驗
- 步驟 4：清理

先決條件

開始本教學課程之前， 必須有您要排程執行的 AWS FIS 實驗範本。如果您已有工作實驗範本，請記下範本 ID 和 AWS 區域。否則，您可以按照 中的指示建立範本the section called “測試執行個體停止和啟動”，然後返回本教學課程。

步驟 1：建立 IAM 角色和政策

建立 IAM 角色和政策

1. 開啟位於 https://console.aws.amazon.com/iam/ 的 IAM 主控台。
2. 在左側導覽窗格中，選擇角色，然後選擇建立角色。
3. 選擇自訂信任政策，然後插入下列程式碼片段，以允許 Amazon EventBridge Scheduler 代表您擔任該角色。

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

   選擇 Next (下一步)。
4. 在新增許可下，選擇建立政策。
5. 選擇 JSON，然後插入下列政策。使用先決條件步驟中的實驗範本 ID *your-experiment-template-id*。

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

您可以限制排程器只執行具有特定標籤值的 AWS FIS 實驗範本。例如，下列政策會授予所有 AWS FIS 實驗的`StartExperiment`許可，但限制排程器只執行已標記 的實驗範本`Purpose=Schedule`。

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

選擇下一步：標籤。

6. 選擇下一步：檢閱。
7. 在檢閱政策下，為您的政策命名 `FIS_RecurringExperiment`，然後選擇建立政策。
8. 在新增許可下，將新`FIS_RecurringExperiment`政策新增至您的角色，然後選擇下一步。
9. 在名稱下，檢閱和建立角色 `FIS_RecurringExperiment_role`，然後選擇建立角色。

步驟 2：建立 Amazon EventBridge 排程器

建立 Amazon EventBridge 排程器

1. 前往 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。
2. 在左側導覽窗格中，選擇排程。
3. 確認您位於與 AWS FIS 實驗範本 AWS 區域 相同的 中。
4. 選擇建立排程，然後填入以下內容：
   - 在排程名稱下，插入 `FIS_recurring_experiment_tutorial`。
   - 在排程模式下，選取週期性排程。
   - 在排程類型下，選取以速率為基礎的排程。
   - 在速率表達式下，選擇 5 分鐘。
   - 在彈性時段下，選取關閉。
   - （選用）在時間範圍下，選取您的時區。
   - 選擇 Next (下一步)。
5. 在選取目標下，選擇所有 APIs，然後搜尋 AWS FIS。
6. 選擇 AWS FIS，然後選擇 StartExperiment。
7. 在輸入下，插入下列 JSON 承載。將 *`your-experiment-template-id`* 值取代為實驗的範本 ID。`ClientToken` 是排程器的唯一識別符。在本教學課程中，我們使用 Amazon EventBridge 排程器允許的內容關鍵字。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的新增內容屬性。

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

   選擇 Next (下一步)。

8. （選用）在設定下，您可以設定重試政策、無效字母佇列 (DLQ) 和加密設定。或者，您可以保留預設值。
9. 在許可下，選取使用現有角色，然後搜尋 `FIS_RecurringExperiment_role`。
10. 選擇 Next (下一步)。
11. 在檢閱和建立排程下，檢閱排程器詳細資訊，然後選擇建立排程。

步驟 3：驗證您的實驗

驗證您的 AWS FIS 實驗是否按排程執行

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在左側導覽窗格中，選擇實驗。
3. 建立排程後五分鐘，您應該會看到實驗正在執行。

步驟 4：清理

停用 Amazon EventBridge 排程器

1. 前往 https://console.aws.amazon.com/events/ 開啟 Amazon EventBridge 主控台。
2. 在左側導覽窗格中，選擇排程。
3. 選取您新建立的排程器，然後選擇停用。

使用 AWS FIS 案例程式庫

案例定義客戶可以套用的事件或條件，以測試其應用程式的彈性，例如應用程式執行所在的運算資源中斷。案例由 AWS 建立和擁有，並透過為您提供一組預先定義的目標和故障動作（例如，停止自動擴展群組中的 30% 執行個體）來減少常見的應用程式受損，從而將未區分的繁重工作降至最低。

案例是透過僅限主控台的案例程式庫提供， AWS FIS 並使用實驗範本執行。為了使用案例執行實驗，您將從程式庫中選取案例，指定符合您工作負載詳細資訊的參數，並將其儲存為帳戶中的實驗範本。

主題

- 檢視案例
- 使用案例
- 匯出案例
- 案例參考

檢視案例

若要使用主控台檢視案例：

1. 在 開啟 AWS FIS 主控台https://console.aws.amazon.com/fis/。
2. 在導覽窗格中，選擇案例程式庫。
3. 若要檢視特定案例的相關資訊，請選取案例卡片以顯示分割面板。
   - 在頁面底部的分割面板中的描述索引標籤中，您可以檢視案例的簡短描述。您也可以找到先決條件的簡短摘要，其中包含所需的目標資源摘要，以及準備資源以用於案例所需的任何動作。最後，您也可以查看有關案例中目標和動作的其他資訊，以及實驗以預設設定成功執行時的預期持續時間。
   - 在頁面底部的分割面板的內容索引標籤中，您可以預覽從案例建立的實驗範本部分填入版本。
   - 在頁面底部的分割面板中的詳細資訊索引標籤中，您可以找到實作案例的詳細說明。這可能包含如何近似案例個別層面的詳細資訊。在適用的情況下，您也可以閱讀要用作停止條件的指標，並提供從實驗中學習的可觀測性。最後，您會找到如何展開產生的實驗範本的建議。

使用案例

若要使用主控台使用案例：

1. 在 開啟 AWS FIS 主控台https://console.aws.amazon.com/fis/。
2. 在導覽窗格中，選擇案例程式庫。
3. 若要檢視特定案例的相關資訊，請選取案例卡片以顯示分割面板
4. 若要使用案例，請選取案例卡，然後選擇使用案例建立範本。
5. 在建立實驗範本檢視中，填入任何遺失的項目。
   a. 有些案例可讓您大量編輯在多個動作或目標之間共用的參數。此功能會在您對案例進行任何變更後停用，包括透過大量參數編輯進行變更。若要使用此功能，請選取編輯大量參數按鈕。在模態中編輯參數，然後選取儲存按鈕。
   b. 有些實驗範本可能缺少動作或目標參數，在每個動作和目標卡上反白顯示。選取每張卡片的編輯按鈕，新增缺少的資訊，然後選取卡片上的儲存按鈕。
   c. 所有 範本都需要服務存取執行角色。您可以選擇現有的角色，或為此實驗範本建立新的角色。
   d. 建議您選取現有的 AWS CloudWatch 警示，以定義一或多個選用的停止條件。進一步了解 AWS FIS 的停止條件。如果您尚未設定警示，您可以遵循使用 Amazon CloudWatch 警示中的指示，稍後再更新實驗範本。
   e. 我們建議啟用選用實驗日誌到 Amazon CloudWatch logs或 Amazon S3 儲存貯體。進一步了解 AWS FIS 的實驗記錄。如果您尚未設定適當的資源，您可以稍後更新實驗範本。
6. 在建立實驗範本中，選取建立實驗範本。
7. 從 AWS FIS 主控台的實驗範本檢視中，選取開始實驗。進一步了解 管理 AWS FIS 實驗範本。

匯出案例

案例是僅限主控台的體驗。雖然與實驗範本類似，但案例並非完整的實驗範本，且無法直接匯入 AWS FIS。如果您想要使用案例做為自己的自動化的一部分，您可以使用兩個路徑之一：

1. 請依照 中的步驟使用案例建立有效的 AWS FIS 實驗範本，並匯出該範本。
2. 依照步驟 3 檢視案例中的步驟，從內容索引標籤複製並儲存案例內容，然後手動新增缺少的參數，以建立有效的實驗範本。

案例參考

案例庫中包含的案例旨在盡可能使用標籤，每個案例都會描述案例描述的先決條件和運作方式章節中的必要標籤。您可以使用這些預先定義的標籤來標記資源，也可以使用大量參數編輯體驗來設定自己的標籤 （請參閱 使用案例)。

此參考說明 AWS FIS 案例庫中的常見案例。您也可以使用 AWS FIS 主控台列出支援的案例。

如需詳細資訊，請參閱使用 AWS FIS 案例程式庫。

AWS FIS 支援下列 Amazon EC2 案例。這些案例以使用標籤的執行個體為目標。您可以使用自己的標籤，或使用案例中包含的預設標籤。其中一些案例使用 SSM 文件。

- EC2 壓力：執行個體故障 - 透過停止一或多個 EC2 執行個體，探索執行個體故障的影響。

  目前區域中已連接特定標籤的目標執行個體。在這種情況下，我們將停止這些執行個體，並在動作持續時間結束時重新啟動它們，預設為 5 分鐘。
- EC2 壓力：磁碟 - 探索磁碟使用率增加對以 EC2 為基礎的應用程式的影響。

  在此案例中，我們將目前區域中已連接特定標籤的 EC2 執行個體設為目標。在此案例中，您可以自訂在動作持續時間內，在目標 EC2 執行個體上插入的磁碟使用率增加，每個磁碟壓力動作預設為 5 分鐘。
- EC2 壓力：CPU - 探索增加的 CPU 對以 EC2 為基礎的應用程式的影響。

  在此案例中，我們將目前區域中已連接特定標籤的 EC2 執行個體設為目標。在此案例中，您可以自訂在動作持續時間內，在目標 EC2 執行個體上注入的 CPU 應力量增加，每個 CPU 應力動作預設為 5 分鐘。
- EC2 壓力：記憶體 - 探索記憶體使用率增加對以 EC2 為基礎的應用程式的影響。

  在此案例中，我們將目前區域中已連接特定標籤的 EC2 執行個體設為目標。在此案例中，您可以自訂在動作持續時間內，在目標 EC2 執行個體上注入的記憶體壓力量增加，每個記憶體壓力動作預設為 5 分鐘。
- EC2 壓力：網路延遲 - 探索網路延遲增加對以 EC2 為基礎的應用程式的影響。

  在此案例中，我們將目前區域中已連接特定標籤的 EC2 執行個體設為目標。在此案例中，您可以自訂在動作持續時間內注入目標 EC2 執行個體的網路延遲量增加，每個延遲動作預設為 5 分鐘。

AWS FIS 支援下列 Amazon EKS 案例。這些案例以使用 Kubernetes 應用程式標籤的 EKS Pod 為目標。您可以使用自己的標籤，或使用案例中包含的預設標籤。如需使用 FIS 之 EKS 的詳細資訊，請參閱 EKS Pod 動作。

- EKS 壓力：Pod 刪除 - 透過刪除一或多個 Pod 探索 EKS Pod 失敗的影響。

  在此案例中，我們將鎖定目前區域中與應用程式標籤相關聯的 Pod。在此案例中，我們將終止所有相符的 Pod。重新建立 Pod 將由 kubernetes 組態控制。

- EKS 壓力：CPU - 探索增加的 CPU 對以 EKS 為基礎的應用程式的影響。

  在此案例中，我們將鎖定目前區域中與應用程式標籤相關聯的 Pod。在此案例中，您可以在動作持續時間內，自訂在目標 EKS Pod 上注入的 CPU 應力增加量，每個 CPU 應力動作預設為 5 分鐘。
- EKS 壓力：磁碟 - 探索磁碟使用率增加對以 EKS 為基礎的應用程式的影響。

  在此案例中，我們將鎖定目前區域中與應用程式標籤相關聯的 Pod。在此案例中，您可以自訂動作持續時間內在目標 EKS Pod 上注入的磁碟壓力增加量，每個 CPU 壓力動作預設為 5 分鐘。
- EKS 壓力：記憶體 - 探索記憶體使用率增加對以 EKS 為基礎的應用程式的影響。

  在此案例中，我們將鎖定目前區域中與應用程式標籤相關聯的 Pod。在此案例中，您可以自訂目標 EKS Pod 在動作持續時間內注入的記憶體壓力量增加，每個記憶體壓力動作預設為 5 分鐘。
- EKS 壓力：網路延遲 - 探索網路延遲增加對以 EKS 為基礎的應用程式的影響。

  在此案例中，我們將鎖定目前區域中與應用程式標籤相關聯的 Pod。在此案例中，您可以自訂目標 EKS Pod 在動作持續時間內注入的網路延遲量增加，每個延遲動作預設為 5 分鐘。

AWS FIS 支援多可用區域和多區域應用程式的下列案例。這些案例以多種資源類型為目標。

- AZ Availability: Power Interruption - 在可用區域 (AZ) 中注入電源完全中斷的預期症狀。進一步了解 AZ Availability: Power Interruption。
- Cross-Region: Connectivity - 封鎖從實驗區域到目的地區域的應用程式網路流量，並暫停跨區域資料複寫。進一步了解如何使用 Cross-Region: Connectivity。

AZ Availability: Power Interruption

您可以使用 AZ Availability: Power Interruption案例來引發可用區域 (AZ) 中電源完全中斷的預期症狀。

此案例可用來示範多可用區域應用程式在單一完整可用區域電源中斷期間如預期般運作。其中包括區域運算 (Amazon EC2、EKS 和 ECS) 的遺失、可用區域中運算的重新擴展、子網路連線中斷、RDS 容錯移轉、ElastiCache 容錯移轉，以及無回應的 EBS 磁碟區。預設會略過找不到目標的動作。

動作

下列動作共同在單一 AZ 中產生完整電源中斷的許多預期症狀。可用區域可用性：電源中斷只會影響預期會在單一可用區域電源中斷期間看到影響的服務。根據預設，案例會注入電源中斷症狀 30 分鐘，然後額外注入復原期間可能發生的症狀 30 分鐘。

Stop-Instances

在 AZ 電源中斷期間，受影響 AZ 中的 EC2 執行個體將會關閉。電源還原後，執行個體將重新啟動。AZ Availability: Power Interruption包含 aws：ec2：stop-instances，以在中斷期間停止受影響 AZ 中的所有執行個體。持續時間過後，執行個體會重新啟動。停止由 Amazon EKS 管理的 EC2 執行個體會導致刪除相依的 EKS Pod。停止由 Amazon ECS 管理的 EC2 執行個體會導致相依 ECS 任務停止。

此動作以受影響 AZ 中執行的 EC2 執行個體為目標。根據預設，它會以標籤名為 `AzImpairmentPower` 且值為 的執行個體為目標`StopInstances`。您可以在實驗範本中將此標籤新增至執行個體，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的執行個體，則會略過此動作。

Stop-ASG-Instances

在 AZ 電源中斷期間，由受影響 AZ 中的 Auto Scaling 群組管理的 EC2 執行個體將會關閉。電源還原後，執行個體將重新啟動。 AZ Availability: Power Interruption包含 aws：ec2：stop-instances，以在中斷期間停止所有執行個體，包括 Auto Scaling 管理的執行個體。持續時間過後，執行個體會重新啟動。

此動作以受影響 AZ 中執行的 EC2 執行個體為目標。根據預設，它會以標籤名為 `AzImpairmentPower`且值為 的執行個體為目標`IceAsg`。您可以在實驗範本中將此標籤新增至執行個體，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的執行個體，則會略過此動作。

暫停執行個體啟動

在 AZ 電源中斷期間，在 AZ 中佈建容量的 EC2 API 呼叫將會失敗。特別是，下列 APIs 會受到影響：`ec2:StartInstances`、 `ec2:CreateFleet`和 `ec2:RunInstances`。 AZ Availability: Power Interruption includes包含 aws：ec2：api-insufficient-instance-capacity-error，以防止在受影響的可用區域中佈建新的執行個體。

此動作以用來佈建執行個體的 IAM 角色為目標。這些必須使用 ARN 作為目標。根據預設，如果找不到有效的 IAM 角色，則會略過此動作。

暫停 ASG Scaling

在 AZ 電源中斷期間，Auto Scaling 控制平面發出的 EC2 API 呼叫會失敗，以復原 AZ 中遺失的容量。特別是，下列 APIs 將受到影響：`ec2:StartInstances`、 `ec2:CreateFleet`和 `ec2:RunInstances`。 AZ Availability: Power Interruption包含 aws：ec2：asg-insufficient-instance-capacity-error，以防止在受影響的可用區域中佈建新的執行個體。這也可防止 Amazon EKS 和 Amazon ECS 在受影響的 AZ 中擴展。

此動作以 Auto Scaling 群組為目標。根據預設，它會以標籤名為 且值`AzImpairmentPower`為 的 Auto Scaling 群組為目標`IceAsg`。您可以在實驗範本中將此標籤新增至 Auto Scaling 群組，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的 Auto Scaling 群組，則會略過此動作。

暫停網路連線

在 AZ 電源中斷期間，AZ 中的網路將無法使用。發生這種情況時，某些 AWS 服務可能需要幾分鐘的時間來更新 DNS，以反映受影響 AZ 中的私有端點無法使用。在此期間，DNS 查詢可能會傳回無法存取的 IP 地址。 AZ Availability: Power Interruption包含 aws：network：disrupt-connectivity，以封鎖受影響 AZ 中所有子網路的所有網路連線 2 分鐘。這將強制大多數應用程式逾時和 DNS 重新整理。在 2 分鐘後結束動作可允許區域服務 DNS 的後續復原，同時 AZ 繼續無法使用。

此動作以子網路為目標。根據預設，它會以標籤名為 `AzImpairmentPower`且值為 的叢集為目標`DisruptSubnet`。您可以在實驗範本中將此標籤新增至子網路，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的子網路，則會略過此動作。

容錯移轉 RDS

在 AZ 電源中斷期間，受影響 AZ 中的 RDS 節點將會關閉。受影響 AZ 中的單一 AZ RDS 節點將完全無法使用。對於多可用區域叢集，寫入器節點將容錯移轉至不受影響的可用區域，且受影響的可用區域中的讀取器節點將無法使用。對於多可用區域叢集，如果寫入器位於受影響的可用區域， 會將 aws：rds：failover-db-cluster AZ Availability: Power Interruption納入容錯移轉。

此動作以 RDS 叢集為目標。根據預設，它會以標籤名為 `AzImpairmentPower`且值為 的叢集為目標`DisruptRds`。您可以在實驗範本中將此標籤新增至叢集，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的叢集，則會略過此動作。

暫停 ElastiCache 複寫群組

在 AZ 電源中斷期間，AZ 中的 ElastiCache 節點無法使用。 AZ Availability: Power Interruption包含 aws：elasticache：replicationgroup-interrupt-az-power，以終止受影響 AZ 中的 ElastiCache 節點。在中斷期間，不會在受影響的 AZ 中佈建新的執行個體，因此複寫群組將保持容量減少。

此動作以 ElastiCache 複寫群組為目標。根據預設，它會以名為 標籤的複寫群組為目標`AzImpairmentPower`，其值為 `ElasticacheImpact`。您可以在實驗範本中將此標籤新增至複寫群組，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的複寫群組，則會略過此動作。請注意，只有受影響 AZ 中寫入器節點的複寫群組才會被視為有效目標。

啟動 ARC 區域自動轉移

AZ 電源中斷開始的五分鐘後，復原動作`aws:arc:start-zonal-autoshift`會自動將資源流量從指定的 AZ 移出電源中斷的剩餘 25 分鐘。在該持續時間之後，流量會移回原始 AZ。請注意，如果啟用

自動轉移，在實際的可用區域電源中斷期間， AWS 會偵測中斷和轉移資源流量。雖然此輪班的時間會有所不同，但預估會在損害開始的五分鐘後發生。

此動作以啟用 Amazon Application Recovery Controller (ARC) Autoshift 的資源為目標。根據預設，它會以具有標籤索引鍵 `AzImpairmentPower` 和值 的資源為目標`RecoverAutoshiftResources`。您可以在實驗範本中將此標籤新增至資源，或以您自己的標籤取代預設標籤。例如，您可能想要使用應用程式特定的標籤。根據預設，如果找不到有效的資源，則會略過此動作。

暫停 EBS I/O

AZ 電源中斷後，一旦電源恢復，極少部分的執行個體可能會遇到無回應的 EBS 磁碟區。 AZ Availability: Power Interruption包含 aws：ebs：pause-io，讓 1 個 EBS 磁碟區處於無回應狀態。

根據預設，只有設定為在執行個體終止後保留的磁碟區才會成為目標。此動作以具有標籤名為 `AzImpairmentPower`且值為 的磁碟區為目標`APIPauseVolume`。您可以在實驗範本中將此標籤新增至磁碟區，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的磁碟區，則會略過此動作。

限制

- 此案例不包含停止條件。應用程式正確的停止條件應新增至實驗範本。
- 在目標 AZ 中，在 EC2 上執行的 Amazon EKS Pod 將以 EC2 工作者節點終止，並封鎖開始新的 EC2 節點。不過，不支援在 AWS Fargate 上執行的 Amazon EKS Pod。
- 在目標 AZ 中，在 EC2 上執行的 Amazon ECS 任務將以 EC2 工作者節點終止，並封鎖開始新的 EC2 節點。不過，不支援在 AWS Fargate 上執行的 Amazon ECS 任務。
- 不支援具有兩個可讀取待命資料庫執行個體的 Amazon RDS Multi-AZ。在此情況下，執行個體將終止、RDS 將容錯移轉，且容量將立即佈建回受影響的 AZ。受影響 AZ 中的可讀取待命將保持可用。

要求

- 將必要的許可新增至 AWS FIS 實驗角色。
- 資源標籤必須套用至實驗目標的資源。這些可以使用您自己的標記慣例或案例中定義的預設標籤。

許可

ARC 區域自動轉移使用 IAM 服務連結角色`AWSServiceRoleForZonalAutoshiftPracticeRun`代表您執行區域轉移。此角色使用 IAM 受管政策 `AWSZonalAutoshiftPracticeRunSLRPolicy`。您不需要手動建立角色。當您從 AWS CLI、 或 AWS SDK 中的 AZ 電力中斷案例建立實驗範本 AWS

Management Console時，ARC 會為您建立服務連結角色。如需詳細資訊，請參閱在 ARC 中使用服務連結角色進行區域自動轉移。

下列政策授予 AWS FIS 必要的許可，以執行具有 AZ Availability: Power Interruption案例的實驗。此政策必須連接到實驗角色。

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

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

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

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": [
                "arn:aws:kms:*:*:key/*"
            ],
            "Condition": {
                "StringLike": {
                    "kms:ViaService": "ec2.*.amazonaws.com"
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:PauseVolumeIO"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*"
        },
        {
            "Sid": "AllowInjectAPI",
            "Effect": "Allow",
            "Action": [
                "ec2:InjectApiError"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:api-insufficient-instance-capacity-error",
```

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

案例內容

下列內容定義了案例。此 JSON 可用來儲存，並使用 AWS 命令列界面 (AWS CLI) 中的 create-experiment-template 命令來建立實驗範本。如需最新版本的案例，請造訪 FIS 主控台中的案例程式庫。

```
{
    "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-IO": {
            "actionId": "aws:ebs:pause-volume-io",
            "parameters": {
                "duration": "PT30M"
            },
            "targets": {
                "Volumes": "EBS-Volumes"
            },
            "startAfter": [
                "Stop-Instances",
                "Stop-ASG-Instances"
            ]
        },
        "Stop-Instances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "completeIfInstancesTerminated": "true",
                "startInstancesAfterDuration": "PT30M"
            },
            "targets": {
                "Instances": "EC2-Instances"
            }
        },
        "Pause-ASG-Scaling": {
            "actionId": "aws:ec2:asg-insufficient-instance-capacity-error",
            "parameters": {
                "availabilityZoneIdentifiers": "us-east-1a",
```

```
                "duration": "PT30M",
                "percentage": "100"
            },
            "targets": {
                "AutoScalingGroups": "ASG"
            }
        },
        "Stop-ASG-Instances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "completeIfInstancesTerminated": "true",
                "startInstancesAfterDuration": "PT30M"
            },
            "targets": {
                "Instances": "ASG-EC2-Instances"
            }
        },
        "Pause-network-connectivity": {
            "actionId": "aws:network:disrupt-connectivity",
            "parameters": {
                "duration": "PT2M",
                "scope": "all"
            },
            "targets": {
                "Subnets": "Subnet"
            }
        },
        "Failover-RDS": {
            "actionId": "aws:rds:failover-db-cluster",
            "parameters": {},
            "targets": {
                "Clusters": "RDS-Cluster"
            }
        },
        "Pause-ElastiCache": {
            "actionId": "aws:elasticache: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."
}
```

Cross-Region: Connectivity

您可以使用 Cross-Region: Connectivity案例來封鎖從實驗區域到目的地區域的應用程式網路流量，並暫停 Amazon S3 和 Amazon DynamoDB 的跨區域複寫。跨區域：連線會影響來自您執行實驗之區域的傳出應用程式流量 (實驗區域)。來自您想要隔離實驗區域 (目的地區域) 之區域的無狀態傳入流量可能不會遭到封鎖。來自 AWS 受管服務的流量可能無法封鎖。

當無法從實驗區域存取目的地區域中的資源時，此案例可用來示範多區域應用程式可如預期運作。它包括透過鎖定傳輸閘道和路由表，封鎖從實驗區域到目的地區域的網路流量。它也會暫停 S3 和 DynamoDB 的跨區域複寫。預設會略過找不到目標的動作。

動作

下列動作會共同封鎖包含 AWS 服務的跨區域連線。動作會平行執行。根據預設，此案例會封鎖流量 3 小時，最多可增加 12 小時的持續時間。

中斷傳輸閘道連線

Cross Region: Connectivity 包含 aws：network：transit-gateway-disrupt-cross-region-connectivity，以封鎖從實驗區域中 VPCs 到傳輸閘道所連接目的地區域中 VPCs 的跨區域網路流量。這不會影響對實驗區域內 VPC 端點的存取，但會封鎖來自目的地區域中 VPC 端點目的地之實驗區域的流量。

此動作以連接實驗區域和目的地區域的傳輸閘道為目標。根據預設，它會以具有標籤名為 `DisruptTransitGateway`且值為 的傳輸閘道為目標`Allowed`。您可以在實驗範本中將此標籤新增至傳輸閘道，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的傳輸閘道，則會略過此動作。

中斷子網路連線

Cross Region: Connectivity 包含 aws：network：route-table-disrupt-cross-region-connectivity，以封鎖從實驗區域中 VPCs 到目的地區域中公有 AWS IP 區塊的跨區域網路流量。這些公有 IP 區塊包括目的地區域中的 AWS 服務端點，例如 S3 區域端點，以及受管服務的 AWS IP 區塊，例如用於負載平衡器和 Amazon API Gateway 的 IP 地址。此動作也會封鎖從實驗區域到目的地區域的跨區域 VPC 對等連線網路連線。它不會影響對實驗區域中 VPC 端點的存取，但會封鎖來自目的地區域中 VPC 端點目的地之實驗區域的流量。

此動作以實驗區域中的子網路為目標。根據預設，它會以標籤名為 且值`DisruptSubnet`為 的子網路為目標`Allowed`。您可以在實驗範本中將此標籤新增至子網路，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的子網路，則會略過此動作。

暫停 S3 複寫

Cross Region: Connectivity 包含 aws：s3：bucket-pause-replication，以暫停從實驗區域到目標儲存貯體目的地區域的 S3 複寫。從目的地區域到實驗區域的複寫將不受影響。案例結束後，儲存貯體複寫會從暫停的時間點繼續。請注意，複寫保持所有物件同步所需的時間，會根據實驗持續時間以及物件上傳至儲存貯體的速率而有所不同。

此動作將啟用跨區域複寫 (CRR) 的實驗區域中的 S3 儲存貯體目標設為目的地區域中的 S3 儲存貯體。根據預設，它會以標籤名為 且值`DisruptS3`為 的儲存貯體為目標`Allowed`。您可以在實驗範本中將此標籤新增至儲存貯體，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的儲存貯體，則會略過此動作。

暫停 DynamoDB 複寫

Cross-Region: Connectivity 包含 aws：dynamodb：global-table-pause-replication，以暫停實驗區域與所有其他區域之間的複寫，包括目的地區域。這可防止複寫到實驗區域內外，但不會影響其他區域之

間的複寫。案例結束後，資料表複寫會從暫停的時間點繼續。請注意，複寫保持所有資料同步所需的時間會根據實驗持續時間和資料表的變更率而有所不同。

此動作以實驗區域中的 DynamoDB 全域資料表為目標。根據預設，它會以標籤名為 `DisruptDynamoDb`且值為 的資料表為目標`Allowed`。您可以在實驗範本中將此標籤新增至資料表，或以您自己的標籤取代預設標籤。根據預設，如果找不到有效的全域資料表，則會略過此動作。

限制

- 此案例不包含停止條件。應用程式正確的停止條件應新增至實驗範本。

要求

- 將必要的許可新增至 AWS FIS 實驗角色。
- 資源標籤必須套用至實驗目標的資源。這些可以使用您自己的標記慣例或案例中定義的預設標籤。

許可

下列政策授予 AWS FIS 必要的許可，以執行具有 Cross-Region: Connectivity案例的實驗。此政策必須連接到實驗角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RouteTableDisruptConnectivity1",
            "Effect": "Allow",
            "Action": "ec2:CreateRouteTable",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity2",
            "Effect": "Allow",
            "Action": "ec2:CreateRouteTable",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        },
```

```
        {
            "Sid": "RouteTableDisruptConnectivity21",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateRouteTable",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity3",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity4",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateManagedPrefixList",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity5",
            "Effect": "Allow",
            "Action": "ec2:DeleteRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*",
                "arn:aws:ec2:*:*:vpc/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity6",
            "Effect": "Allow",
            "Action": "ec2:CreateRoute",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity7",
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity8",
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity9",
            "Effect": "Allow",
            "Action": "ec2:DeleteNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
```

```
                "ec2:ResourceTag/managedByFIS": "true"
            }
        }
    },
    {
        "Sid": "RouteTableDisruptConnectivity10",
        "Effect": "Allow",
        "Action": "ec2:CreateManagedPrefixList",
        "Resource": "arn:aws:ec2:*:*:prefix-list/*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/managedByFIS": "true"
            }
        }
    },
    {
        "Sid": "RouteTableDisruptConnectivity11",
        "Effect": "Allow",
        "Action": "ec2:DeleteManagedPrefixList",
        "Resource": "arn:aws:ec2:*:*:prefix-list/*",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/managedByFIS": "true"
            }
        }
    },
    {
        "Sid": "RouteTableDisruptConnectivity12",
        "Effect": "Allow",
        "Action": "ec2:ModifyManagedPrefixList",
        "Resource": "arn:aws:ec2:*:*:prefix-list/*",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/managedByFIS": "true"
            }
        }
    },
    {
        "Sid": "RouteTableDisruptConnectivity13",
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeVpcs",
            "ec2:DescribeVpcPeeringConnections",
```

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

```
        {
            "Sid": "RouteTableDisruptConnectivity18",
            "Effect": "Allow",
            "Action": "ec2:DisassociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity19",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity20",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity1",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateTransitGatewayRouteTable",
                "ec2:AssociateTransitGatewayRouteTable"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:transit-gateway-route-table/*",
                "arn:aws:ec2:*:*:transit-gateway-attachment/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity2",
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayAttachments",
                "ec2:DescribeTransitGateways"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion1",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion2",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion3",
            "Effect": "Allow",
            "Action": [
                "s3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "StringLike": {
                    "s3:DestinationRegion": "*"
                }
            }
        },
        {
            "Sid": "S3CrossRegion4",
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:PutReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::*",
```

```
            "Condition": {
                "BoolIfExists": {
                    "s3:isReplicationPauseRequest": "true"
                }
            }
        },
        {
            "Sid": "DdbCrossRegion1",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DdbCrossRegion",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:PutResourcePolicy",
                "dynamodb:GetResourcePolicy",
                "dynamodb:DeleteResourcePolicy"
            ],
            "Resource": [
                "arn:aws:dynamodb:*:*:table/*",
            ]
        }
    ]
}
```

案例內容

下列內容定義了案例。此 JSON 可用來儲存，並使用 AWS 命令列界面 (AWS CLI) 中的 create-experiment-template 命令來建立實驗範本。如需最新版本的案例，請造訪 FIS 主控台中的案例程式庫。

```
{
        "targets": {
                "Transit-Gateway": {
                        "resourceType": "aws:ec2:transit-gateway",
                        "resourceTags": {
                                "TgwTag": "TgwValue"
```

```
                        },
                        "selectionMode": "ALL"
                },
                "Subnet": {
                        "resourceType": "aws:ec2:subnet",
                        "resourceTags": {
                                "SubnetKey": "SubnetValue"
                        },
                        "selectionMode": "ALL",
                        "parameters": {}
                },
                "S3-Bucket": {
                        "resourceType": "aws:s3:bucket",
                        "resourceTags": {
                                "S3Impact": "Allowed"
                        },
                        "selectionMode": "ALL"
                },
                "DynamoDB-Global-Table": {
                        "resourceType": "aws:dynamodb:global-table",
                        "resourceTags": {
                                "DisruptDynamoDb": "Allowed"
                        },
                        "selectionMode": "ALL"
                }
        },
        "actions": {
                "Disrupt-Transit-Gateway-Connectivity": {
                        "actionId": "aws:network:transit-gateway-disrupt-cross-region-
connectivity",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "TransitGateways": "Transit-Gateway"
                        }
                },
                "Disrupt-Subnet-Connectivity": {
                        "actionId": "aws:network:route-table-disrupt-cross-region-
connectivity",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
```

```
                        },
                        "targets": {
                                "Subnets": "Subnet"
                        }
                },
                "Pause-S3-Replication": {
                        "actionId": "aws:s3:bucket-pause-replication",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "Buckets": "S3-Bucket"
                        }
                },
                "Pause-DynamoDB-Replication": {
                        "actionId": "aws:dynamodb:global-table-pause-replication",
                        "parameters": {
                                "duration": "PT3H"
                        },
                        "targets": {
                                "Tables": "DynamoDB-Global-Table"
                        }
                }
        },
        "stopConditions": [
                {
                        "source": "none"
                }
        ],
        "roleArn": "",
        "logConfiguration": {
                "logSchemaVersion": 2
        },
        "tags": {
                "Name": "Cross-Region: Connectivity"
        },
        "experimentOptions": {
                "accountTargeting": "single-account",
                "emptyTargetResolutionMode": "skip"
        },
        "description": "Block application network traffic from experiment Region to
 target Region and pause cross-Region replication"
}
```

使用 的多帳戶實驗 AWS FIS

您可以使用 AWS FIS 主控台或命令列來建立和管理多帳戶實驗範本。您可以透過將以實驗為目標的帳戶選項指定為 來建立多帳戶實驗`"multi-account"`，並新增目標帳戶組態。建立多帳戶實驗範本之後，您可以使用它來執行實驗。

透過多帳戶實驗，您可以在跨越區域內多個 AWS 帳戶的應用程式中設定和執行實際故障案例。您可以從影響多個目標帳戶中資源的協調器帳戶執行多帳戶實驗。

當您執行多帳戶實驗時，具有受影響資源的目標帳戶將透過其 AWS Health 儀表板收到通知，為目標帳戶中的使用者提供意識。透過多帳戶實驗，您可以：

- 在應用程式上執行真實的故障案例，這些應用程式跨越多個帳戶，並具有 AWS FIS 提供的中央控制項和護欄。
- 使用具有精細許可和標籤的 IAM 角色來控制多帳戶實驗的效果，以定義每個目標的範圍。
- 從 AWS Management Console 和 透過 AWS FIS 日誌集中檢視每個帳戶中 AWS FIS 採取的動作。
- 使用 AWS CloudTrail 監控和稽核每個帳戶中 AWS FIS 的 API 呼叫。

本節可協助您開始使用多帳戶實驗。

主題

- 多帳戶實驗的概念
- 多帳戶實驗的最佳實務
- 多帳戶實驗的先決條件
- 建立多帳戶實驗範本
- 更新目標帳戶組態
- 刪除目標帳戶組態

多帳戶實驗的概念

以下是多帳戶實驗的主要概念：

- Orchestrator 帳戶 - Orchestrator 帳戶充當中央帳戶，以在 AWS FIS 主控台中設定和管理實驗，以及集中記錄。協調程式帳戶擁有 AWS FIS 實驗範本和實驗。

- 目標帳戶 - 目標帳戶是具有資源的個別 AWS 帳戶，這些資源可能會受到 AWS FIS 多帳戶實驗的影響。
- 目標帳戶組態 - 您可以透過將目標帳戶組態新增至實驗範本來定義屬於實驗一部分的目標帳戶。目標帳戶組態是多帳戶實驗所需的實驗範本元素。您可以設定帳戶 ID、IAM 角色和選用描述，為每個目標 AWS 帳戶定義一個。

多帳戶實驗的最佳實務

以下是使用多帳戶實驗的最佳實務：

- 當您設定多帳戶實驗的目標時，我們建議在所有目標帳戶中使用一致的資源標籤來鎖定目標。 AWS FIS 實驗會解析每個目標帳戶中具有一致標籤的資源。動作必須解決任何目標帳戶中至少一個目標資源，否則 將會失敗，但將 `emptyTargetResolutionMode` 設為 的實驗除外`skip`。每個帳戶都適用動作配額。如果您想要依資源 ARNs 鎖定資源，則每個動作套用相同的單一帳戶限制。
- 當您使用參數或篩選條件鎖定一或多個可用區域中的資源時，您應該指定 AZ ID，而不是 AZ 名稱。AZ ID 是跨帳戶可用區域的唯一且一致的識別符。若要了解如何尋找帳戶中可用區域的可用區域 ID，請參閱 AWS 資源的可用區域 IDs。

多帳戶實驗的先決條件

若要使用多帳戶實驗的停止條件，您必須先設定跨帳戶警示。當您建立多帳戶實驗範本時，會定義 IAM 角色。您可以在建立範本之前建立必要的 IAM 角色。

內容

- 多帳戶實驗的許可
- 多帳戶實驗的停止條件（選用）
- 多帳戶實驗的安全控制桿（選用）

多帳戶實驗的許可

多帳戶實驗使用 IAM 角色鏈結，將許可授予 AWS FIS ，以對目標帳戶中的資源採取動作。對於多帳戶實驗，您可以在每個目標帳戶和協調器帳戶中設定 IAM 角色。這些 IAM 角色需要目標帳戶與協調者帳戶之間，以及協調者帳戶與 之間的信任關係 AWS FIS。

目標帳戶的 IAM 角色包含對資源採取動作所需的許可，並透過新增目標帳戶組態為實驗範本建立。您將為協調器帳戶建立 IAM 角色，並具有擔任目標帳戶角色的許可，並與之建立信任關係 AWS FIS。此 IAM 角色用作實驗範本`roleArn`的 。

若要進一步了解角色鏈結，請參閱 IAM 使用者指南中的角色術語和概念。

在下列範例中，您將設定協調器帳戶 A 的許可，以執行目標帳戶 B `aws:ebs:pause-volume-io`中的 實驗。

1. 在帳戶 B 中，建立具有執行動作所需許可的 IAM 角色。如需每個動作所需的許可，請參閱 動作參考。下列範例顯示目標帳戶授予執行 EBS 暫停磁碟區 IO 動作 的許可the section called “aws:ebs:pause-volume-io”。

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

2. 接著，在帳戶 B 中新增信任政策，以建立與帳戶 A 的信任關係。選擇帳戶 A 的 IAM 角色名稱，您將在步驟 3 中建立該角色。

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

3. 在帳戶 A 中，建立 IAM 角色。此角色名稱必須符合您在步驟 2 信任政策中指定的角色。若要鎖定多個帳戶，您可以授予協調者擔任每個角色的許可。下列範例顯示帳戶 A 擔任帳戶 B 的許可。如果您有其他目標帳戶，請將其他角色 ARNs 新增至此政策。每個目標帳戶只能有一個角色 ARN。

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

4. 帳戶 A 的此 IAM 角色會用作實驗範本`roleArn`的 。下列範例顯示 IAM 角色中所需的信任政策，授予 AWS FIS 許可以擔任帳戶 A，即協調器帳戶。

```
{
```

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

您也可以使用 Stacksets 一次佈建多個 IAM 角色。若要使用 CloudFormation StackSets，您需要在 AWS 帳戶中設定必要的 StackSet 許可。若要進一步了解，請參閱使用 AWS CloudFormation StackSets。

多帳戶實驗的停止條件（選用）

停止條件是一種機制，可在實驗達到您定義為警示的閾值時停止實驗。若要設定多帳戶實驗的停止條件，您可以使用跨帳戶警示。您必須啟用每個目標帳戶中的共用，才能使用唯讀許可讓協調器帳戶可以使用警示。共用後，您可以使用指標數學來結合來自不同目標帳戶的指標。然後，您可以新增此警示做為實驗的停止條件。

若要進一步了解跨帳戶儀表板，請參閱在 CloudWatch 中啟用跨帳戶功能。

多帳戶實驗的安全控制桿（選用）

安全控制桿可用來停止所有執行中的實驗，並防止新的實驗開始。您可能想要使用安全控制桿，在特定期間或回應應用程式運作狀態警示時防止 FIS 實驗。每個 AWS 帳戶都有一個安全控制桿 AWS 區域。使用安全控制桿時，它會影響在與安全控制桿相同的帳戶和區域中執行的所有實驗。若要停止和防止多帳戶實驗，必須在執行實驗的相同帳戶和區域中使用安全控制桿。

建立多帳戶實驗範本

了解如何透過 建立實驗範本 AWS Management Console

請參閱 建立實驗範本。

使用 CLI 建立實驗範本

1. 開啟 AWS Command Line Interface
2. 若要從儲存的 JSON 檔案建立實驗，並將以實驗為目標的帳戶選項設為 `"multi-account"`（例如，`my-template.json`），請以您自己的值取代*##*的預留位置值，然後執行下列 create-experiment-template 命令。

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

   這會在回應中傳回實驗範本。`id` 從回應複製 ，這是實驗範本的 ID。

3. 執行 create-target-account-configuration 命令，將目標帳戶組態新增至實驗範本。使用步驟 2 `id`中的 作為 `--experiment-template-id` 參數的值，將*###*取代預留位置值，然後執行下列動作。`--description` 為選用參數。為每個目標帳戶重複此步驟。

```
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. 執行 get-target-account-configuration 命令，以擷取特定目標帳戶組態的詳細資訊。

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

5. 新增所有目標帳戶組態後，您可以執行 list-target-account-configurations 命令，查看已建立您的目標帳戶組態。

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

   您也可以執行 get-experiment-template 命令來驗證您已新增目標帳戶組態。範本將傳回唯讀欄位`targetAccountConfigurationsCount`，該欄位是實驗範本上所有目標帳戶組態的計數。

6. 當您準備好時，您可以使用 start-experiment 命令執行實驗範本。

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

更新目標帳戶組態

如果您想要變更帳戶的角色 ARN 或描述，您可以更新現有的目標帳戶組態。當您更新目標帳戶組態時，變更不會影響任何使用 範本的執行中實驗。

使用 更新目標帳戶組態 AWS Management Console

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本
3. 選取實驗範本，然後選擇動作、更新實驗範本。
4. 在側邊面板中，選擇步驟 3，設定服務存取。
5. 修改目標帳戶組態，然後選擇更新實驗範本。
6. 選取步驟 5，檢閱並建立。

使用 CLI 更新目標帳戶組態

執行 update-target-account-configuration 命令以執行命令，以您自己的值取代*##*的預留位置值。`--role-arn` 和 `--description` 參數是選用的，如果未包含，則不會更新。

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

刪除目標帳戶組態

如果您不再需要目標帳戶組態，則可以將其刪除。當您刪除目標帳戶組態時，任何使用範本的執行中實驗都不會受到影響。實驗會繼續執行，直到完成或停止為止。

使用 刪除目標帳戶組態 AWS Management Console

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、更新。
4. 在側邊面板中，選擇步驟 3，設定服務存取。
5. 在目標帳戶組態下，針對您要刪除的目標帳戶角色 ARN 選取移除。
6. 選取步驟 5，檢閱並建立。

7. 檢閱範本，然後選擇更新實驗範本。出現確認提示時，輸入update並選擇更新實驗範本。

使用 CLI 刪除目標帳戶組態

執行 delete-target-account-configuration 命令，以您自己的值取代##的預留位置值。

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

排程實驗

使用 AWS Fault Injection Service (FIS)，您可以對 AWS 工作負載執行故障注入實驗。這些實驗會在範本上執行，其中包含一或多個要在指定目標上執行的動作。您現在可以從 FIS 主控台將實驗排程為一次性任務或重複性任務。除了排程規則之外，FIS 現在還提供新的排程功能。FIS 現在與 EventBridge Scheduler 整合，並代表您建立規則。EventBridge 排程器是無伺服器排程器，可讓您從單一受管的中央服務建立、執行及管理任務。

> **Important**
>
> 具有 的實驗排程器 AWS Fault Injection Service 不適用於 AWS GovCloud （美國東部） 和 AWS GovCloud （美國西部）。

主題

- 建立排程器角色
- 建立實驗排程
- 更新實驗排程
- 停用或刪除實驗排程

建立排程器角色

執行角色是 AWS FIS 擔任的 IAM 角色，以便與 EventBridge 排程器互動，以及讓 Event Bridge 排程器啟動 FIS 實驗。您可以將許可政策連接至此角色，以授予 EventBridge Scheduler 調用 FIS 實驗的存取權。下列步驟說明如何建立新的執行角色和政策，以允許 EventBridge 開始實驗。

使用 AWS CLI 建立排程器角色

這是 Event Bridge 能夠代表客戶排程實驗所需的 IAM 角色。

1. 複製下列擔任角色的 JSON 政策，並將其儲存為 `fis-execution-role.json`。此信任政策允許 EventBridge Scheduler 代表您擔任該角色。

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

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

2. 從 AWS Command Line Interface (AWS CLI) 輸入下列命令來建立新的角色。`FisSchedulerExecutionRole` 將 取代為您要提供此角色的名稱。

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

如果成功，您會看到下列輸出：

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

3. 若要建立新的政策，以允許 EventBridge Scheduler 叫用實驗，請複製下列 JSON，並將其儲存為 `fis-start-experiment-permissions.json`。下列政策允許 EventBridge Scheduler 呼叫您帳戶中所有實驗範本`fis:StartExperiment`的動作。如果您想要將角色限制為單一實驗範本，請將 `*` 結尾的 取代`"arn:aws:fis:*:*:experiment-template/*"`為實驗範本的 ID。

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

4. 執行下列命令來建立新的許可政策。 `FisSchedulerPolicy` 將 取代為您要提供此政策的名稱。

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

如果成功，您會看到下列輸出。請注意政策 ARN。您可以在下一個步驟中使用此 ARN，將政策連接至我們的執行角色。

```
{
    "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. 執行下列命令，將政策連接至您的執行角色。`your-policy-arn` 將 取代為您在上一個步驟中建立的政策 ARN。`FisSchedulerExecutionRole` 將 取代為執行角色的名稱。

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

`attach-role-policy` 操作不會傳回命令列上的回應。

6. 您可以限制排程器只執行具有特定標籤值的 AWS FIS 實驗範本。例如，下列政策會授予所有 AWS FIS 實驗的 `fis:StartExperiment`許可，但限制排程器只執行標記為 的實驗範本 `Purpose=Schedule`。

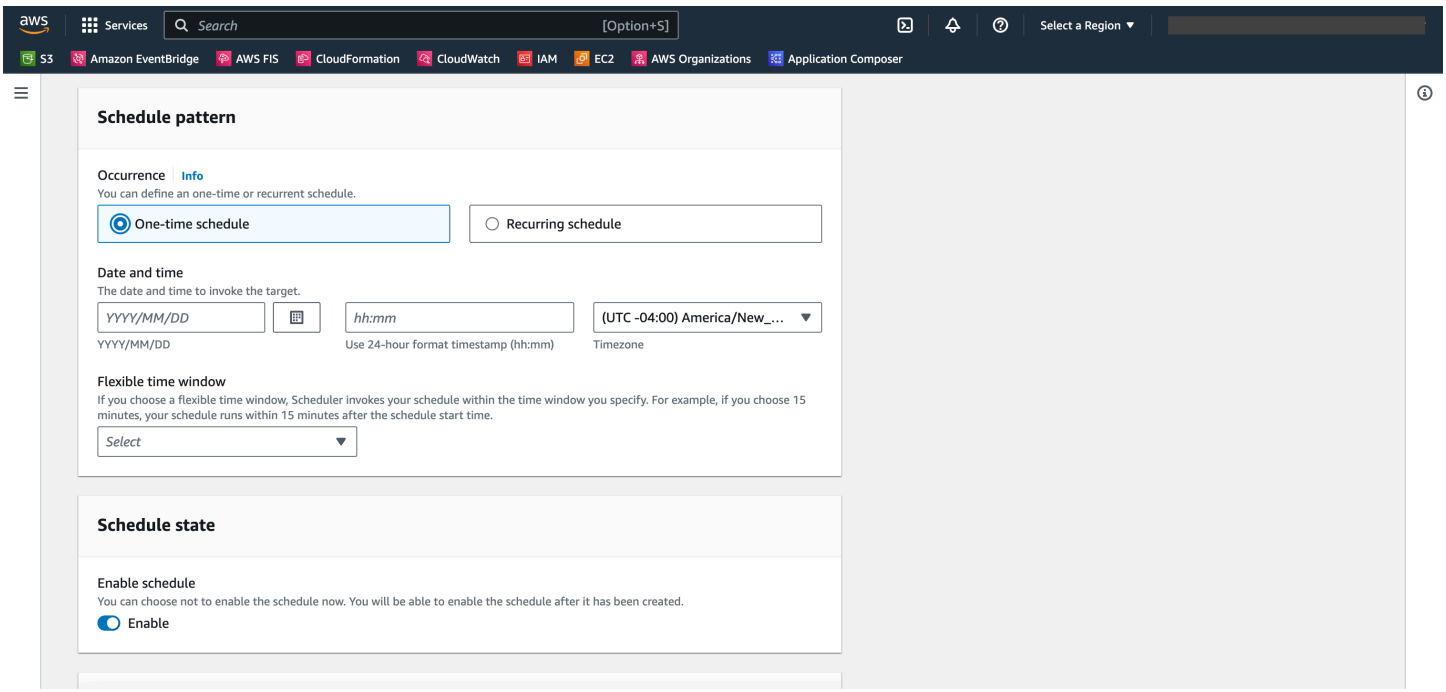
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": "arn:aws:fis:*:*:experiment/*"
        },
        {
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
        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

建立實驗排程

在排定實驗之前，您需要一或多個 ，實驗範本元件才能叫用排程。您可以使用現有的 AWS 資源，或建立新的資源。

實驗範本建立後，按一下動作，然後選取排程實驗。系統會將您重新導向至排程實驗頁面。將會為您填寫排程的名稱。

遵循排程模式區段，然後選擇一次性排程或重複排程。填寫必要的輸入欄位並導覽至許可。



排程狀態預設為啟用。注意：如果您停用排程狀態，即使您建立排程，也不會排程實驗。

AWS FIS 實驗排程器建置在 EventBridge 排程器之上。您可以參考支援的各種排程類型的文件。

使用主控台更新排程

1. 開啟 AWS FIS 主控台。
2. 在左側導覽窗格中，選擇實驗範本 。
3. 選擇您要為其建立排程的實驗範本。
4. 按一下動作，然後從下拉式清單中選取排程實驗。
   a. 在排程名稱下，會自動填入名稱。
   b. 在排程模式下，選取週期性排程。
   c. 在排程類型下，您可以選取以速率為基礎的排程，請參閱排程類型 。
   d. 在速率表達式下，選擇比實驗執行時間慢的速率，例如 5 分鐘。
   e. 在時間範圍下，選取您的時區 。
   f. 在開始日期和時間下，指定開始日期和時間。
   g. 在結束日期和時間下，指定結束日期和時間

   h. 在排程狀態下，切換啟用排程選項。
   i. 在許可下，選取使用現有角色，然後搜尋 `FisSchedulerExecutionRole`。
   j. 選擇 Next (下一步)。
5. 選取檢閱和建立排程、檢閱排程器詳細資訊，然後選擇建立排程。

更新實驗排程

您可以更新實驗排程，以便在適合您的特定日期和時間進行。

使用主控台更新實驗執行

1. 開啟 Amazon FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選擇資源類型：已建立排程的實驗範本。
4. 按一下範本的實驗 ID。然後導覽至排程索引標籤。
5. 檢查是否存在與實驗相關聯的現有排程。選取相關聯的排程，然後按一下按鈕更新排程。

停用或刪除實驗排程

若要停止實驗在排程中執行或執行，您可以刪除或停用規則。下列步驟將逐步引導您如何使用 AWS 主控台刪除或停用實驗執行。

刪除或停用規則

1. 開啟 Amazon FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選擇資源類型：已建立排程的實驗範本。
4. 按一下範本的實驗 ID。然後導覽至排程索引標籤。
5. 檢查是否存在與實驗相關聯的現有排程。選取相關聯的排程，然後按一下按鈕更新排程。
6. 執行以下任意一項：
   a. 若要刪除排程，請選取規則刪除排程旁的按鈕。輸入 `delete`，然後按一下刪除排程按鈕。
   b. 若要停用排程，請選取停用排程規則旁的按鈕。輸入`disable`並按一下停用排程按鈕。

的安全葉子 AWS FIS

安全控制桿用於停止所有執行中的實驗，並防止新的實驗啟動。您可能想要使用安全控制桿來防止在特定期間或回應應用程式運作狀態警示的 FIS 實驗。每個 AWS 帳戶都有一個安全控制桿 AWS 區域。

對於由安全控制桿停止的進行中實驗，您只需支付實驗停止之前執行的動作持續時間。無法啟動的實驗不會產生任何費用。下列各節提供如何開始使用安全控制桿的資訊。

主題

- 安全控制桿的概念
- 使用安全控制桿

安全控制桿的概念

安全控制桿可以接合或取消接合。

- 如果取消連接，則允許 FIS 實驗。根據預設，安全控制桿會分離。
- 如果參與，進行中實驗會停止，且不允許啟動新實驗。

受安全控制桿影響的實驗會以下列其中一種狀態結束：

- 已停止，如果實驗在安全控制桿參與時正在執行。
- 如果實驗是在安全控制桿已使用時啟動，則已取消。

您無法繼續或重新執行已停止或取消的實驗。不過，分離安全桿後，您可以使用相同的實驗範本啟動新的實驗。

Saftey 控制桿資源

安全控制桿是由 Amazon Resource Name (ARN) 定義的資源。安全控制桿包含下列參數：

- 狀態，其已參與或取消參與。
- 原因，這是使用者用來記錄為何變更鬆散控制桿狀態的字串輸入。

使用安全控制桿

本節將詳細說明如何使用 AWS FIS 主控台或命令列來檢視、參與和解除安全控制桿。

檢視安全控制桿

您可以依照下列步驟，檢視您帳戶和區域的安全控制桿狀態。

使用主控台檢視安全控制桿

1. 開啟 AWS FIS 主控台
2. 在導覽窗格中，選擇實驗。
3. 如果已使用安全控制桿，您會在頁面頂端看到提醒橫幅。如果沒有警示橫幅，則會解除安全控制桿。

使用 CLI 檢視安全控制桿

- 使用下列命令：

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

安全控制桿可以處於下列其中一種狀態：

- 停用 - 安全控制桿不會影響任何實驗。實驗可以自由執行。預設情況下，安全控制桿會取消連接。
- 參與 - 安全控制桿正在從分離變更為參與。可能仍有尚未停止的實驗。處於此狀態時，無法變更安全控制桿。
- Engaged - 安全控制桿處於作用中狀態，且沒有實驗正在執行。任何嘗試在安全控制桿接合時啟動的新實驗都會取消。

使用安全控制桿

使用 主控台與安全控制桿互動

1. 開啟 AWS FIS 主控台
2. 在導覽窗格中，選擇實驗。
3. 選擇停止所有實驗按鈕。

4. 輸入與安全控制桿互動的原因。
5. 選擇確認。

使用 CLI 設定安全控制桿

- 使用下列 命令。使用您自己的回應填入原因欄位。

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

停用安全控制桿

使用主控台解除安全控制桿

1. 開啟 AWS FIS 主控台
2. 在導覽窗格中，選擇實驗。
3. 選擇解除安全控制桿按鈕。
4. 輸入解除安全控制桿的原因。
5. 選擇確認。

使用 CLI 解除安全控制桿

- 使用下列命令：

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

監控 AWS FIS 實驗

您可以使用下列工具來監控 Fault Injection Service (AWS FIS) AWS 實驗的進度和影響。

AWS FIS 主控台和 AWS CLI

使用 AWS FIS 主控台或 AWS CLI 來監控執行中實驗的進度。您可以檢視實驗中每個動作的狀態，以及每個動作的結果。如需詳細資訊，請參閱the section called “檢視您的實驗”。

CloudWatch 用量指標和警示

使用 CloudWatch 用量指標來提供您的帳戶對 資源用量的可見性。 AWS FIS 用量指標對應至 AWS 服務配額。您可以設定警示，在您的用量接近服務配額時發出警示。如需詳細資訊，請參閱使用 CloudWatch 監控 。

您也可以建立 CloudWatch 警示來為 AWS FIS 實驗建立停止條件，該警示定義實驗何時超出界限。觸發警示時，實驗會停止。如需詳細資訊，請參閱停止條件。如需建立 CloudWatch 警示的詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的根據靜態閾值建立 CloudWatch 警示和根據異常偵測建立 CloudWatch CloudWatch 警示。 Amazon CloudWatch

AWS FIS 實驗記錄

啟用實驗日誌記錄，以在實驗執行時擷取實驗的詳細資訊。如需詳細資訊，請參閱 實驗記錄。

實驗狀態變更事件

Amazon EventBridge 可讓您自動回應系統事件或資源變更。 AWS FIS 會在實驗狀態變更時發出通知。您可以為您感興趣的事件建立規則，指定事件符合規則時要採取的自動動作。例如，傳送通知至 Amazon SNS 主題或叫用 Lambda 函數。如需詳細資訊，請參閱使用 EventBridge 監控。

CloudTrail 日誌

使用 AWS CloudTrail 來擷取對 AWS FIS API 發出的呼叫的詳細資訊，並將其儲存為 Amazon S3 中的日誌檔案。CloudTrail 也會為您執行實驗的資源記錄對服務 APIs 發出的呼叫。您可以使用這些 CloudTrail 日誌來判斷提出了哪些呼叫、提出呼叫的來源 IP 地址、提出呼叫的人員及時間等。

AWS 運作狀態儀表板通知

AWS Health 可讓您持續了解資源效能，以及 AWS 服務和帳戶的可用性。當您開始實驗時， AWS FIS 會向 AWS 運作狀態儀表板發出通知。通知會在每個帳戶中的實驗持續時間內出現，其中包含實驗中目標為的資源，包括多帳戶實驗。僅包含 `aws:ssm:start-automation-execution`和

等目標之動作的多帳戶實驗`aws:fis:wait`不會發出通知。用於允許實驗的角色的相關資訊會列在受影響的資源下。若要進一步了解 AWS Health Dashboard，請參閱《AWS Health 使用者指南》中的 AWS Health Dashboard。

> **Note**
>
> AWS Health 會盡力交付事件。

使用 Amazon CloudWatch 監控 AWS FIS 用量指標

您可以使用 Amazon CloudWatch 來監控 AWS FIS 實驗對目標的影響。您也可以監控 AWS FIS 用量。

如需檢視實驗狀態的詳細資訊，請參閱 檢視您的實驗。

監控 AWS FIS 實驗

當您規劃 AWS FIS 實驗時，請識別 CloudWatch 指標，可用來識別實驗目標資源類型的基準或「穩定狀態」。開始實驗後，您可以監控透過實驗範本選取之目標的 CloudWatch 指標。

如需 AWS FIS 支援之目標資源類型的可用 CloudWatch 指標的詳細資訊，請參閱下列內容：

- 使用 CloudWatch 監控您的執行個體
- Amazon ECS CloudWatch 指標
- 使用 CloudWatch 監控 Amazon RDS 指標
- 使用 CloudWatch 監控執行命令指標

AWS FIS 用量指標

您可以使用 CloudWatch 用量指標來提供您帳戶的資源用量可見度。使用這些指標，以 CloudWatch 圖表和儀表板視覺化目前的服務使用狀況。

AWS FIS 用量指標對應至 AWS 服務配額。您可以設定警示，在您的用量接近服務配額時發出警示。如需 CloudWatch 警示的詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

AWS FIS 會在 AWS/Usage 命名空間中發佈下列指標。

指標	描述
ResourceCount	您的帳戶中正在執行的特定資源總數。資源由與指標相關聯的維度定義。

下列維度用於精簡 AWS FIS 發佈的使用指標。

維度	描述
Service	包含 資源 AWS 的服務名稱。對於 AWS FIS 用量指標，此維度的值為 FIS。
Type	正在報告的實體類型。目前， AWS FIS 用量指標的唯一有效值為 Resource。
Resource	正在執行的資源類型。可能的值ExperimentTemplates 適用於實驗範本，以及ActiveExperiments 主動實驗。
Class	此維度會保留供日後使用。

使用 Amazon EventBridge 監控 AWS FIS 實驗

當實驗狀態變更時， AWS FIS 會發出通知。這些通知會透過 Amazon EventBridge （先前稱為 CloudWatch Events) 做為事件提供。 AWS FIS 會盡最大努力發出這些事件。事件會以接近即時的方式遞送到 EventBridge。

使用 EventBridge，您可以建立觸發程式設計動作以回應事件的規則。例如，您可以設定呼叫 SNS 主題以傳送電子郵件通知的規則，或呼叫 Lambda 函數以採取一些動作。

如需有關 EventBridge 的詳細資訊，請參閱《Amazon Eventbridge 使用者指南》中的 Amazon EventBridge 入門。

以下是實驗狀態變更事件的語法：

```
{
    "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`

狀態已變更的實驗 ID。

`experiment-template-id`

實驗使用的實驗範本 ID。

`new_value`

實驗的新狀態。可能值如下：

- `completed`
- `failed`
- `initiating`
- `running`
- `stopped`
- `stopping`

`old_value`

實驗的先前狀態。可能值如下：

- `initiating`
- `pending`
- `running`
- `stopping`

AWS FIS 的實驗記錄

您可以使用實驗記錄來擷取實驗執行時的詳細資訊。

您需要根據與每個日誌目的地類型相關聯的成本支付實驗日誌記錄的費用。如需詳細資訊，請參閱 Amazon CloudWatch 定價 (在付費方案、日誌、已終止日誌下） 和 Amazon S3 定價。

許可

您必須授予 AWS FIS 許可，將日誌傳送至您設定的每個日誌目的地。如需詳細資訊，請參閱《Amazon CloudWatch Logs 使用者指南》中的以下內容：

- 傳送至 CloudWatch Logs 的日誌
- 傳送至 Amazon S3 的日誌

日誌結構描述

以下是用於實驗記錄的結構描述。目前的結構描述版本為 2。的欄位`details`取決於 的值`log_type`。的欄位`resolved_targets`取決於 的值`target_type`。如需詳細資訊，請參閱the section called “日誌記錄範例”。

```
{
    "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

    }
}
```

版本備註

- 第 2 版推出：
  - `target_type` 欄位 和 會將`resolved_targets`欄位從 ARNs 清單變更為物件清單。`resolved_targets` 物件的有效欄位取決於 的值`target_type`，這是目標的資源類型。
  - 新增`account_id`欄位的 `action-error`和 `target-resolution-detail`事件類型。
- 第 1 版是初始版本。

日誌目的地

AWS FIS 支援將日誌交付至下列目的地：

- Amazon S3 儲存貯體
- Amazon CloudWatch Logs 日誌群組

S3 日誌交付

日誌會傳送到下列位置。

```
bucket-and-optional-prefix/AWSLogs/account-id/fis/region/experiment-
id/YYYY/MM/DD/account-id_awsfislogs_region_experiment-id_YYYYMMDDHHMMZ_hash.log
```

可能需要幾分鐘的時間，日誌才會交付至儲存貯體。

CloudWatch Logs 日誌交付

日誌會交付至名為 /aws/fis/*experiment-id* 的日誌串流。

日誌會在不到一分鐘內交付至日誌群組。

日誌記錄範例

以下是在隨機選取的 EC2 執行個體上執行 aws:ec2:reboot-instances動作之實驗的範例日誌記錄。

記錄

- experiment-start
- target-resolution-start
- target-resolution-detail
- target-resolution-end
- action-start
- action-end
- action-error
- 實驗結束

experiment-start

以下是experiment-start事件的範例記錄。

```
{
    "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事件的範例記錄。

```
{
    "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事件的範例記錄。如果目標解析失敗，記錄也會包含error_information 欄位。

```
{
    "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

如果目標解析失敗，記錄也會包含 `error_information` 欄位。如果 `total_pages` 大於 1，則已解析的目標數量超過一筆記錄的大小限制。還有其他包含其餘已解析目標`target-resolution-end`的記錄。

以下是 EC2 動作`target-resolution-end`事件的範例記錄。

```
{
    "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 動作`target-resolution-end`事件的範例記錄。

```
{
    "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`事件的範例記錄。如果實驗範本指定 動作的參數，則記錄也會包含 `parameters` 欄位。

```
{
    "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`事件的範例記錄。只有在動作失敗時，才會傳回此事件。會針對動作失敗的每個帳戶傳回。

```
{
```

```
    "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`事件的範例記錄。

```
{
    "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."
        }
    }
}
```

實驗結束

以下是`experiment-end`事件的範例記錄。

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

啟用實驗記錄

實驗記錄預設為停用。若要接收實驗的實驗日誌，您必須從已啟用記錄的實驗範本建立實驗。您第一次執行設定為使用先前未曾用於記錄的目的地的實驗時，我們會延遲實驗來設定傳送至此目的地的日誌交付，這大約需要 15 秒。

使用主控台啟用實驗記錄

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。
3. 選取實驗範本，然後選擇動作、更新實驗範本。
4. 對於日誌，設定目的地選項。若要將日誌傳送至 S3 儲存貯體，請選擇傳送至 Amazon S3 儲存貯體，然後輸入儲存貯體名稱和字首。若要將日誌傳送至 CloudWatch Logs，請選擇傳送至 CloudWatch Logs，然後輸入日誌群組。
5. 選擇更新實驗範本。

使用 啟用實驗記錄 AWS CLI

使用 update-experiment-template 命令並指定日誌組態。

停用實驗記錄

如果您不想再收到實驗的日誌，您可以停用實驗日誌記錄。

使用主控台停用實驗記錄

1. 在 https：//https://console.aws.amazon.com/fis/ 開啟 AWS FIS 主控台。
2. 在導覽窗格中，選擇實驗範本。

3. 選取實驗範本，然後選擇動作、更新實驗範本。
4. 對於日誌，清除傳送至 Amazon S3 儲存貯體和傳送至 CloudWatch Logs。
5. 選擇更新實驗範本。

使用 停用實驗記錄 AWS CLI

使用 update-experiment-template 命令並指定空白日誌組態。

使用 記錄 API 呼叫 AWS CloudTrail

AWS Fault Injection Service (AWS FIS) 已與 整合 AWS CloudTrail，此服務提供使用者、角色或 AWS FIS 中 AWS 服務所採取動作的記錄。CloudTrail 會將 AWS FIS 的所有 API 呼叫擷取為事件。擷取的呼叫包括 FIS AWS 主控台的呼叫，以及對 AWS FIS API 操作的程式碼呼叫。如果您建立線索，則可以啟用 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 AWS FIS 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。使用 CloudTrail 收集的資訊，您可以判斷對 AWS FIS 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

使用 CloudTrail

建立帳戶 AWS 帳戶 時，您的 上會啟用 CloudTrail。當活動在 AWS FIS 中發生時，該活動會記錄在 CloudTrail 事件中，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 中檢視、搜尋和下載最近的事件 AWS 帳戶。如需詳細資訊，請參閱《使用 CloudTrail 事件歷史記錄檢視事件》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/view-cloudtrail-events.html。

若要持續記錄 中的事件 AWS 帳戶，包括 AWS FIS 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台中建立追蹤時，追蹤會套用至所有 AWS 區域。追蹤會記錄 AWS 分割區中所有 區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析 CloudTrail 日誌中收集的事件資料並對其採取行動。如需詳細資訊，請參閱下列內容：

- 為 AWS 您的帳戶建立追蹤
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案

CloudTrail 會記錄所有 AWS FIS 動作，並記錄在 AWS Fault Injection Service API 參考中。對於在目標資源上執行的實驗動作，請檢視擁有資源之服務的 API 參考文件。例如，如需在 Amazon EC2 執行個體上執行的動作，請參閱 Amazon EC2 API 參考。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 AWS FIS 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下是呼叫 AWS FIS `StopExperiment`動作的 CloudTrail 日誌項目範例。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
    "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/example",
        "accountId": "111122223333",
        "userName": "example"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2020-12-03T09:40:42Z",
        "mfaAuthenticated": "false"
```

```
      }
    }
  },
  "eventTime": "2020-12-03T09:44:20Z",
  "eventSource": "fis.amazonaws.com",
  "eventName": "StopExperiment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.51.100.25",
  "userAgent": "Boto3/1.22.9 Python/3.8.13 Linux/5.4.186-113.361.amzn2int.x86_64
 Botocore/1.25.9",
  "requestParameters": {
    "clientToken": "1234abc5-6def-789g-012h-ijklm34no56p",
    "experimentTemplateId": "ABCDE1fgHIJkLmNop",
    "tags": {}
  },
  "responseElements": {
    "experiment": {
      "actions": {
        "exampleAction1": {
          "actionId": "aws:ec2:stop-instances",
          "duration": "PT10M",
          "state": {
            "reason": "Initial state",
            "status": "pending"
          },
          "targets": {
            "Instances": "exampleTag1"
          }
        },
        "exampleAction2": {
          "actionId": "aws:ec2:stop-instances",
          "duration": "PT10M",
          "state": {
            "reason": "Initial state",
            "status": "pending"
          },
          "targets": {
            "Instances": "exampleTag2"
          }
        }
      },
      "creationTime": 1605788649.95,
      "endTime": 1606988660.846,
      "experimentTemplateId": "ABCDE1fgHIJkLmNop",
```

```
      "id": "ABCDE1fgHIJkLmNop",
      "roleArn": "arn:aws:iam::111122223333:role/AllowFISActions",
      "startTime": 1605788650.109,
      "state": {
        "reason": "Experiment stopped",
        "status": "stopping"
      },
      "stopConditions": [
        {
          "source": "aws:cloudwatch:alarm",
          "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:example"
        }
      ],
      "tags": {},
      "targets": {
        "ExampleTag1": {
          "resourceTags": {
            "Example": "tag1"
          },
          "resourceType": "aws:ec2:instance",
          "selectionMode": "RANDOM(1)"
        },
        "ExampleTag2": {
          "resourceTags": {
            "Example": "tag2"
          },
          "resourceType": "aws:ec2:instance",
          "selectionMode": "RANDOM(1)"
        }
      }
    }
  },
  "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
  "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"

}
```

以下是 API 動作的範例 CloudTrail 日誌項目，做為包含 AWS FIS `aws:ssm:send-command` AWS 動作之實驗的一部分而調用。`userIdentity` 元素會反映透過擔任角色取得的臨時登入資料所提出的請求。擔任的角色名稱會出現在 中`userName`。實驗的 ID EXP21nT17WMzA6dnUgz 會出現在中，`principalId`並做為擔任角色的 ARN 的一部分。

```
{
    "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"
}
```

故障診斷 AWS FIS

若要對錯誤進行故障診斷， 會從 `GetExperiment` API 和 FIS 實驗日誌 AWS FIS 傳回詳細錯誤。當實驗狀態失敗時，錯誤會傳回為實驗狀態的一部分。當多個動作失敗時，第一個失敗的動作會傳回為實驗錯誤。您可以檢閱您的 FIS 實驗日誌是否有任何其他錯誤。若要了解如何記錄和監控 AWS FIS 實驗，請參閱 監控 AWS FIS 實驗。

根據失敗的類型，您可能會收到下列其中一個錯誤：

- 原因：特定失敗的詳細描述。原因值不應用於自動化，因為它們可能會變更。
- 程式碼：失敗的類型。除非下表另有指定，否則程式碼值不應用於自動化，因為這些值可能會有所變更。
- 位置：失敗實驗範本 區段的內容，例如 動作或目標。
- 帳戶 ID：失敗發生 AWS 的帳戶。

錯誤代碼

錯誤代碼	程式碼描述
`ConfigurationFailure`	動作、目標、實驗或日誌未正確設定。檢查錯誤，`location`並確保參數和組態正確。
`DependentServiceFailure`	另一個 AWS 服務失敗。再次嘗試執行實驗。
`InternalFailure`	執行實驗時發生內部錯誤。您可以根據此錯誤碼自動化 。
`InvalidTarget`	在目標解析期間或動作開始時，無法解析目標。這可能是下列其中一個原因所造成： • 目標不存在，例如已刪除或 ARN 不正確。 • 您的目標有一個標籤，不會解析任何資源。 • 有一個動作未連結至目標。 若要進行故障診斷，請檢閱您的日誌，以識別哪些目標尚未解決。檢查所有動作是否連結至目

錯誤代碼	程式碼描述
	標，以及您的資源 ID 或標籤是否存在，並且沒有拼字錯誤。
`AuthorizationFailure`	因許可錯誤導致實驗失敗的主要原因有兩個： • 您鎖定的 IAM 角色沒有適當的許可來解析目標或對資源採取動作。若要修正此錯誤，請在 FIS 動作參考中檢閱動作所需的許可，並將其新增至實驗 IAM 角色。 • FIS AWS 的服務連結角色 (SLR) 建立遭到組織中的服務控制政策 (SCP) 拒絕。FIS 使用 SLR 來管理實驗的監控和資源選擇。如需詳細資訊，請參閱AWS FIS 的服務連結角色許可。
`QuotaExceededFailure`	已超過資源類型的配額。若要判斷配額是否可以增加，請參閱 Fault Injection Service AWS 的配額和限制。

Fault Injection Service AWS 的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。作為AWS 合規計畫的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 Fault Injection Service AWS 的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 AWS FIS 時套用共同責任模型。下列主題說明如何設定 AWS FIS 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS FIS 資源。

目錄

- Fault Injection Service AWS 中的資料保護
- Fault Injection Service AWS 的身分和存取管理
- Fault Injection Service AWS 中的基礎設施安全性
- 使用介面 VPC 端點存取 AWS FIS (AWS PrivateLink)

Fault Injection Service AWS 中的資料保護

AWS 共同責任模型適用於 Fault Injection Service AWS 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。

- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 AWS FIS 或使用主控台、API AWS CLI或 AWS SDKs的其他 AWS 服務 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

靜態加密

AWS FIS 一律會加密靜態資料。 AWS FIS 中的資料會使用透明的伺服器端加密進行靜態加密。這可協助降低保護敏感資料所涉及的操作負擔和複雜性。您可以透過靜態加密，建立符合加密合規和法規要求，而且對安全性要求甚高的應用程式。

傳輸中加密

AWS FIS 會加密服務與其他整合 AWS 服務之間傳輸中的資料。在 AWS FIS 和整合服務之間傳遞的所有資料都會使用 Transport Layer Security (TLS) 加密。如需其他整合 AWS 服務的詳細資訊，請參閱支援的 AWS 服務。

Fault Injection Service AWS 的身分和存取管理

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （ 登入 ） 和授權 （ 具有許可 ） 來使用 AWS FIS 資源。IAM 是您可以免費使用 AWS 服務 的 。

目錄

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- Fault Injection Service AWS 如何與 IAM 搭配使用
- AWS Fault Injection Service 政策範例

- 使用 Fault Injection Service AWS 的服務連結角色
- AWS Fault Injection Service AWS 的 受管政策

目標對象

使用方式 AWS Identity and Access Management (IAM) 會有所不同，取決於您在 AWS FIS 中執行的工作。

服務使用者 – 如果您使用 AWS FIS 服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS FIS 功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。

服務管理員 – 如果您在公司負責 AWS FIS 資源，您可能擁有 AWS FIS 的完整存取權。您的任務是判斷服務使用者應存取的 AWS FIS 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。

IAM 管理員 – 如果您是 IAM 管理員，建議您了解如何撰寫政策以管理 AWS FIS 存取的詳細資訊。

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的適用於 API 請求的AWS Signature 第 4 版。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱《AWS IAM Identity Center 使用者指南》中的多重要素驗證和《IAM 使用者指南》中的 IAM 中的AWS 多重要素驗證。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取帳戶中的所有 AWS 服務 和 資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的需要根使用者憑證的任務。

聯合身分

最佳實務是要求人類使用者，包括需要管理員存取權的使用者，使用 聯合身分提供者 AWS 服務 來使用臨時憑證來存取 。

聯合身分是您企業使用者目錄、Web 身分提供者、 AWS Directory Service、Identity Center 目錄，或 AWS 服務 是透過身分來源提供的登入資料存取的任何使用者。當聯合身分存取時 AWS 帳戶，它們會擔任 角色，而角色會提供臨時登入資料。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連接並同步到您自己的身分來源中的一組使用者 AWS 帳戶 和群組，以便在所有 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者是 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供臨時憑證。如需更多資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。若要暫時在 中擔任 IAM 角色 AWS Management Console，您可以從使用者切換至 IAM 角色（主控台）。

您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

使用臨時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱《IAM 使用者指南》中的為第三方身分提供者 (聯合) 建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以將政策直接連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。
  - 服務連結角色 – 服務連結角色是一種連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的

程式取得臨時憑證。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM 角色來授予許可權給 Amazon EC2 執行個體上執行的應用程式。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件， AWS 當與身分或資源建立關聯時， 會定義其許可。當委託人（使用者、根使用者或角色工作階段）發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策之間選擇，請參閱《IAM 使用者指南》中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 的多個 的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) - RCP 是 JSON 政策，可用來設定您帳戶中資源的可用許可上限，採取這種方式就不需要更新附加至您所擁有的每個資源的 IAM 政策。RCP 會限制成員帳戶中資源的許可，並可能影響身分的有效許可，包括 AWS 帳戶根使用者，無論它們是否屬於您的組織。如需 Organizations 和 RCPs的詳細資訊，包括 AWS 服務 支援 RCPs 清單，請參閱AWS Organizations《 使用者指南》中的資源控制政策 (RCPs)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

Fault Injection Service AWS 如何與 IAM 搭配使用

在您使用 IAM 管理對 AWS FIS 的存取之前，請先了解哪些 IAM 功能可與 AWS FIS 搭配使用。

可與 Fault Injection Service 搭配使用的 IAM AWS 功能

IAM 功能	AWS FIS 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵 (服務特定)	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
主體許可	是
服務角色	是
服務連結角色	是

若要全面了解 AWS FIS 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱《AWS IAM 使用者指南》中的與 IAM 搭配使用的 服務。

AWS FIS 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

AWS FIS 的身分型政策範例

若要檢視 AWS FIS 身分型政策的範例，請參閱 AWS Fault Injection Service 政策範例。

AWS FIS 中的資源型政策

支援資源型政策：否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，做為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當委託人和資源位於不同位置時 AWS 帳戶，信任帳戶中的 IAM 管理員也必須授予委託人實體（使用者或角色）存取資源的許可。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

AWS FIS 的政策動作

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼条件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS FIS 動作清單，請參閱《服務授權參考》中的 Fault Injection Service AWS 定義的動作。

AWS FIS 中的政策動作在動作之前使用以下字首：

```
fis
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

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

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 `List` 文字的所有動作，請包含以下動作：

```
"Action": "fis:List*"
```

AWS FIS 的政策資源

支援政策資源：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。陳述式必須包含 `Resource` 或 `NotResource` 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

有些 AWS FIS API 動作支援多個資源。若要在單一陳述式中指定多項資源，請使用逗號分隔 ARN。

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

```
]
```

若要查看 AWS FIS 資源類型及其 ARNs的清單，請參閱《服務授權參考》中的 Fault Injection Service AWS 定義的資源類型。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 Fault Injection Service AWS 定義的動作。

AWS FIS 的政策條件索引鍵

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素 (或 `Condition` 區塊) 可讓您指定使陳述式生效的條件。`Condition` 元素是選用項目。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 `Condition` 元素，或是在單一 `Condition` 元素中指定多個索引鍵， AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值， 會使用邏輯OR操作 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的 IAM 政策元素：變數和標籤。

AWS 支援全域條件金鑰和服務特定的條件金鑰。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AWS FIS 條件金鑰清單，請參閱《服務授權參考》中的 Fault Injection Service AWS 的條件金鑰。若要了解您可以使用條件金鑰的動作和資源，請參閱 Fault Injection Service AWS 定義的動作。

若要檢視 AWS FIS 身分型政策的範例，請參閱 AWS Fault Injection Service 政策範例。

AWS FIS ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 與 AWS FIS

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在 中 AWS，這些屬性稱為標籤。您可以將標籤連接到 IAM 實體 （使用者或角色） 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的使用屬性型存取控制 (ABAC)。

若要檢視以身分為基礎的政策範例，以根據資源的標籤限制對資源的存取，請參閱 範例：使用標籤控制資源用量。

搭配 AWS FIS 使用臨時登入資料

支援臨時憑證：是

當您使用臨時登入資料登入時，有些 AWS 服務 無法運作。如需詳細資訊，包括哪些 AWS 服務 使用臨時登入資料，請參閱《AWS 服務 IAM 使用者指南》中的 使用 IAM。

如果您 AWS Management Console 使用使用者名稱和密碼以外的任何方法登入 ，則會使用臨時登入資料。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取 時，該程序會自動建立臨時登入資料。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的從使用者切換至 IAM 角色 (主控台)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱 IAM 中的暫時性安全憑證。

AWS FIS 的跨服務主體許可

支援轉寄存取工作階段 (FAS)：是

當您使用 IAM 使用者或角色在 中執行動作時 AWS，您會被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合

AWS 服務 請求向下游服務提出請求。只有在服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱轉發存取工作階段。

AWS FIS 的服務角色

支援服務角色：是

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可權給 AWS 服務。

AWS FIS 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理 AWS FIS 服務連結角色的詳細資訊，請參閱 使用 Fault Injection Service AWS 的服務連結角色。

AWS Fault Injection Service 政策範例

根據預設，使用者和角色沒有建立或修改 AWS FIS 資源的許可。他們也無法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 來執行任務。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 AWS FIS 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的 Fault Injection Service AWS 的動作、資源和條件索引鍵。

目錄

- 政策最佳實務
- 範例：使用 AWS FIS 主控台
- 範例：列出可用的 AWS FIS 動作

- 範例：為特定動作建立實驗範本
- 範例：開始實驗
- 範例：使用標籤控制資源用量
- 範例：刪除具有特定標籤的實驗範本
- 範例：允許使用者檢視他們自己的許可
- 範例：使用 的條件索引鍵 ec2:InjectApiError
- 範例：使用 的條件索引鍵 aws:s3:bucket-pause-replication

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AWS FIS 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_configure-api-require.html中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

範例：使用 AWS FIS 主控台

若要存取 AWS Fault Injection Service 主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AWS FIS 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

下列範例政策授予許可，以使用 AWS FIS 主控台列出和檢視所有 AWS FIS 資源，但不能建立、更新或刪除這些資源。它還授予許可，以檢視您可以在實驗範本中指定的所有 AWS FIS 動作所使用的可用資源。

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

範例：列出可用的 AWS FIS 動作

下列政策授予許可，以列出可用的 AWS FIS 動作。

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

範例：為特定動作建立實驗範本

下列政策授予許可，以建立 動作 的實驗範本`aws:ec2:stop-instances`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "fis:CreateExperimentTemplate"
```

```
      ],
      "Resource": [
        "arn:aws:fis:*:*:action/aws:ec2:stop-instances",
        "arn:aws:fis:*:*:experiment-template/*"
      ]
    },
    {
      "Sid": "PolicyPassRoleExample",
      "Effect": "Allow",
      "Action": [
          "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::account-id:role/role-name"
      ]
    }
  ]
}
```

範例：開始實驗

下列政策授予使用指定 IAM 角色和實驗範本啟動實驗的許可。它還允許 AWS FIS 代表使用者建立服務連結角色。如需詳細資訊，請參閱使用 Fault Injection Service AWS 的服務連結角色。

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

```
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": "fis.amazonaws.com"
            }
        }
    }
  ]
}
```

範例：使用標籤控制資源用量

下列政策授予從具有標籤 的實驗範本執行實驗的許可Purpose=Test。它不會授予建立或修改實驗範本的許可，或使用沒有指定標籤的範本執行實驗。

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

範例：刪除具有特定標籤的實驗範本

下列政策授予許可，以刪除標籤為 的實驗範本Purpose=Test。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fis:DeleteExperimentTemplate"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

範例：允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

範例：使用 的條件索引鍵 `ec2:InjectApiError`

下列範例政策使用 `ec2:FisTargetArns` 條件索引鍵來限制目標資源的範圍。此政策允許 AWS FIS 動作 `aws:ec2:api-insufficient-instance-capacity-error`和 `aws:ec2:asg-insufficient-instance-capacity-error`。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:api-insufficient-instance-capacity-error",
                    ],
                    "ec2:FisTargetArns": [
                        "arn:aws:iam:*:*:role:role-name"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:asg-insufficient-instance-capacity-error"
                    ],
                    "ec2:FisTargetArns": [

 "arn:aws:autoscaling:*:*:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
                    ]
                }
```

```
            }
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:DescribeAutoScalingGroups",
            "Resource": "*"
        }
    ]
}
```

範例：使用 的條件索引鍵 aws:s3:bucket-pause-replication

下列範例政策使用 S3:IsReplicationPauseRequest條件索引鍵，GetReplicationConfiguration只在 AWS FIS 動作 的內容中使用 AWS 時允許 PutReplicationConfiguration和 aws:s3:bucket-pause-replication。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "S3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "StringEquals": {
                    "s3:DestinationRegion": "region"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "S3:PutReplicationConfiguration",
                "S3:GetReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "BoolIfExists": {
                    "s3:IsReplicationPauseRequest": "true"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "S3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
    }
```

使用 Fault Injection Service AWS 的服務連結角色

AWS Fault Injection Service 使用 AWS Identity and Access Management (IAM) 服務連結角色。服務連結角色是直接連結至 AWS FIS 的唯一 IAM 角色類型。服務連結角色由 AWS FIS 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定 AWS FIS，因為您不需要手動新增必要的許可來管理實驗的監控和資源選擇。 AWS FIS 會定義其服務連結角色的許可，除非另有定義，否則只有 AWS FIS 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

除了服務連結角色之外，您還必須指定 IAM 角色，授予許可來修改您在實驗範本中指定為目標的資源。如需詳細資訊，請參閱AWS FIS 實驗的 IAM 角色。

您必須先刪除相關的 資源，才能刪除服務連結角色。這可保護您的 AWS FIS 資源，因為您不會不小心移除存取資源的許可。

AWS FIS 的服務連結角色許可

AWS FIS 使用名為 AWSServiceRoleForFIS 的服務連結角色，讓它能夠管理實驗的監控和資源選擇。

AWSServiceRoleForFIS 服務連結角色信任下列服務擔任該角色：

- `fis.amazonaws.com`

AWSServiceRoleForFIS 服務連結角色使用 受管政策 AmazonFISServiceRolePolicy。此政策可讓 AWS FIS 管理實驗的監控和資源選擇。如需詳細資訊，請參閱《 AWS 受管政策參考》中的 AmazonFISServiceRolePolicy。

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。若要成功建立 AWSServiceRoleForFIS 服務連結角色，搭配 使用 AWS FIS 的 IAM 身分必須具有必要的許可。若要授予必要許可，請將下列政策連接至 IAM 身分。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "fis.amazonaws.com"
                }
            }
        }
    ]
}
```

如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

建立 AWS FIS 的服務連結角色

您不需要手動建立一個服務連結角色。當您在 AWS Management Console AWS CLI、 或 AWS API 中啟動 AWS FIS 實驗時， AWS FIS 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您啟動 AWS FIS 實驗時， AWS FIS 會再次為您建立服務連結角色。

編輯 AWS FIS 的服務連結角色

AWS FIS 不允許您編輯 AWSServiceRoleForFIS 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱「IAM 使用者指南」的編輯服務連結角色。

刪除 AWS FIS 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，在手動刪除服務連結角色之前，您必須先清除資源。

> **Note**
>
> 如果 AWS FIS 服務在您嘗試清除資源時使用角色，則清除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

清除 AWSServiceRoleForFIS 使用的 AWS FIS 資源

請確定目前沒有執行任何實驗。如有必要，請停止您的實驗。如需詳細資訊，請參閱停止實驗。

使用 IAM 手動刪除服務連結角色

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForFIS 服務連結角色。如需詳細資訊，請參閱「IAM 使用者指南」中的刪除服務連結角色。

AWS FIS 服務連結角色支援的區域

AWS FIS 支援在所有提供服務的區域中使用服務連結角色。如需詳細資訊，請參閱 AWS Fault Injection Service 端點和配額。

AWS Fault Injection Service AWS 的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受 AWS 管政策中定義的許可，則更新會影響政策連接的所有主體身分（使用者、群組和角色）。 AWS 服務 當新的 啟動或新的 API 操作可用於現有服務時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 受管政策：AmazonFISServiceRolePolicy

此政策會連接到名為 AWSServiceRoleForFIS 的服務連結角色，以允許 AWS FIS 管理實驗的監控和資源選擇。如需詳細資訊，請參閱使用 Fault Injection Service AWS 的服務連結角色。

AWS 受管政策：AWSFaultInjectionSimulatorEC2Access

在實驗角色中使用此政策授予 AWS FIS 許可，以執行使用 AWS Amazon EC2 的 FIS 動作的實驗。如需詳細資訊，請參閱the section called “實驗角色”。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSFaultInjectionSimulatorEC2Access。

AWS 受管政策：AWSFaultInjectionSimulatorECSAccess

在實驗角色中使用此政策授予 AWS FIS 許可，以執行使用 AWS Amazon ECS FIS 動作的實驗。如需詳細資訊，請參閱the section called “實驗角色”。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSFaultInjectionSimulatorECSAccess。

AWS 受管政策：AWSFaultInjectionSimulatorEKSAccess

在實驗角色中使用此政策授予 AWS FIS 許可，以執行使用 AWS Amazon EKS FIS 動作的實驗。如需詳細資訊，請參閱the section called “實驗角色”。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSFaultInjectionSimulatorEKSAccess。

AWS 受管政策：AWSFaultInjectionSimulatorNetworkAccess

在實驗角色中使用此政策來授予 AWS FIS 許可，以執行使用 AWS FIS 聯網動作的實驗。如需詳細資訊，請參閱the section called “實驗角色”。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSFaultInjectionSimulatorNetworkAccess。

AWS 受管政策：AWSFaultInjectionSimulatorRDSAccess

在實驗角色中使用此政策授予 AWS FIS 許可，以執行使用 AWS Amazon RDS FIS 動作的實驗。如需詳細資訊，請參閱the section called “實驗角色”。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSFaultInjectionSimulatorRDSAccess。

AWS 受管政策：AWSFaultInjectionSimulatorSSMAccess

在實驗角色中使用此政策授予 AWS FIS 許可，以執行使用 AWS Systems Manager FIS 動作的實驗。如需詳細資訊，請參閱the section called “實驗角色”。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSFaultInjectionSimulatorSSMAccess。

AWSAWS 受管政策的 FIS 更新

檢視自此服務開始追蹤 FIS AWS AWS 受管政策更新以來的詳細資訊。

變更	描述	日期
AWSFaultInjectionSimulatorECSAccess – 更新現有政策	新增允許 AWS FIS 解析 ECS 目標的許可。	2024 年 1 月 25 日
AWSFaultInjectionSimulatorNetworkAccess – 更新現有政策	新增允許 AWS FIS 使用 aws:network:route-table-disrupt-cross-region-connectivity和 aws:network:transit-gateway-disrupt-cross-region-connectivity動作執行實驗的許可。	2024 年 1 月 25 日
AWSFaultInjectionSimulatorEC2Access – 更新現有政策	新增允許 AWS FIS 解析 EC2 執行個體的許可。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorEKSAccess – 更新現有政策	新增允許 AWS FIS 解析 EKS 目標的許可。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorRDSAccess – 更新現有政策	新增允許 AWS FIS 解析 RDS 目標的許可。	2023 年 11 月 13 日
AWSFaultInjectionSimulatorEC2Access – 更新現有政策	新增許可，以允許 AWS FIS 在 EC2 執行個體上執行 SSM 文件，並終止 EC2 執行個體。	2023 年 6 月 2 日

變更	描述	日期
AWSFaultInjectionSimulatorSSMAccess – 更新現有政策	新增允許 AWS FIS 在 EC2 執行個體上執行 SSM 文件的許可。	2023 年 6 月 2 日
AWSFaultInjectionSimulatorECSAccess – 更新現有政策	新增允許 AWS FIS 使用新aws:ecs:task動作執行實驗的許可。	2023 年 6 月 1 日
AWSFaultInjectionSimulatorEKSAccess – 更新現有政策	新增允許 AWS FIS 使用新aws:eks:pod動作執行實驗的許可。	2023 年 6 月 1 日
AWSFaultInjectionSimulatorEC2Access – 新政策	新增政策，以允許 AWS FIS 執行使用 Amazon EC2 的 AWS FIS 動作的實驗。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorECSAccess – 新政策	新增政策，以允許 AWS FIS 執行針對 Amazon ECS 使用 AWS FIS 動作的實驗。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorEKSAccess – 新政策	新增政策，以允許 AWS FIS 執行針對 Amazon EKS 使用 AWS FIS 動作的實驗。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorNetworkAccess – 新政策	新增政策，以允許 AWS FIS 執行使用 AWS FIS 聯網動作的實驗。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorRDSAccess – 新政策	新增政策，以允許 AWS FIS 執行針對 Amazon RDS 使用 AWS FIS 動作的實驗。	2022 年 10 月 26 日
AWSFaultInjectionSimulatorSSMAccess – 新政策	新增政策，以允許 AWS FIS 執行使用 Systems Manager AWS FIS 動作的實驗。	2022 年 10 月 26 日
AmazonFISServiceRolePolicy – 更新至現有政策	新增允許 AWS FIS 描述子網路的許可。	2022 年 10 月 26 日
AmazonFISServiceRolePolicy – 更新至現有政策	新增允許 AWS FIS 描述 EKS 叢集的許可。	2022 年 7 月 7 日

變更	描述	日期
AmazonFISServiceRolePolicy – 更新至現有政策	新增允許 AWS FIS 列出和描述叢集中任務的許可。	2022 年 2 月 7 日
AmazonFISServiceRolePolicy – 更新至現有政策	已移除 `events:DescribeRule` 動作`events:ManagedBy` 的條件。	2022 年 1 月 6 日
AmazonFISServiceRolePolicy – 更新至現有政策	新增許可，以允許 AWS FIS 擷取用於停止條件之 CloudWatch 警示的歷史記錄。	2021 年 6 月 30 日
AWS FIS 開始追蹤變更	AWS FIS 開始追蹤其 AWS 受管政策的變更	2021 年 3 月 1 日

Fault Injection Service AWS 中的基礎設施安全性

作為受管服務， AWS Fault Injection Service 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及 如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 已發佈的 API 呼叫，透過網路存取 AWS FIS。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

使用介面 VPC 端點存取 AWS FIS (AWS PrivateLink)

您可以建立介面 VPC 端點，在 VPC 與 AWS Fault Injection Service 之間建立私有連線。VPC 端點採用 AWS PrivateLink技術，可讓您在沒有網際網路閘道、NAT 裝置、VPN 連接或 AWS Direct Connect 連接的情況下私密存取 AWS FIS APIs。VPC 中的執行個體不需要公有 IP 地址，即可與 AWS FIS APIs通訊。

每個介面端點都由子網路中的一個或多個彈性網路介面表示。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的AWS 服務 透過 存取 AWS PrivateLink 。

AWS FIS VPC 端點的考量事項

在您設定 AWS FIS 的介面 VPC 端點之前，請檢閱《 AWS PrivateLink 指南》中的AWS 服務 使用介面 VPC 端點存取 。

AWS FIS 支援從您的 VPC 呼叫其所有 API 動作。

建立 AWS FIS 的介面 VPC 端點

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface () 為 AWS FIS 服務建立 VPC 端點 AWS CLI。如需詳細資訊，請參閱《AWS PrivateLink 指南》中的建立 VPC 端點。

使用下列服務名稱建立 AWS FIS 的 VPC 端點：`com.amazonaws.region.fis`。

如果您為端點啟用私有 DNS，您可以使用區域的預設 DNS 名稱向 AWS FIS 提出 API 請求，例如 `fis.us-east-1.amazonaws.com`。

建立 AWS FIS 的 VPC 端點政策

您可以將端點政策連接至控制 AWS FIS 存取的 VPC 端點。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的使用端點政策控制對 VPC 端點的存取。

範例：特定 AWS FIS 動作的 VPC 端點政策

下列 VPC 端點政策會將所有資源上列出的 AWS FIS 動作的存取權授予所有委託人。

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
```

```
            "fis:ListExperimentTemplates",
            "fis:StartExperiment",
            "fis:StopExperiment",
            "fis:GetExperiment"
         ],
         "Resource":"*",
         "Principal":"*"
      }
   ]
}
```

範例：拒絕從特定 存取的 VPC 端點政策 AWS 帳戶

下列 VPC 端點政策會拒絕指定 AWS 帳戶 存取所有動作和資源，但會授予所有其他 AWS 帳戶 存取所有動作和資源的權限。

```
{
   "Statement":[
      {
         "Effect": "Allow",
         "Action": "*",
         "Resource": "*",
         "Principal": "*"
      },
      {
         "Effect":"Deny",
         "Action": "*",
         "Resource": "*",
         "Principal": {
           "AWS": [ "123456789012" ]
         }
      }
   ]
}
```

標記您的 AWS FIS 資源

標籤是您或 AWS 指派給 AWS 資源的中繼資料標籤。每個標籤皆包含鍵與值。對於您指派的標籤，您可以定義索引鍵與值。例如，您可以將金鑰定義為 資源`test`的 `purpose`，並將值定義為 。

標籤可協助您執行以下操作：

- 識別和組織您的 AWS 資源。許多 AWS 服務支援標記，因此您可以將相同的標籤指派給來自不同 服務的資源，以指出資源相關。
- 控制對 AWS 資源的存取。如需詳細資訊，請參閱《IAM 使用者指南》中的使用標籤控制存取權限。

標記限制

下列基本限制適用於 AWS FIS 資源上的標籤：

- 您可以指派給資源的標籤數量上限：50
- 索引鍵長度上限：128 個 Unicode 字元
- 數值長度上限：256 個 Unicode 字元
- 索引鍵和值的有效字元：a-z、A-Z、0-9、空格和下列字元：_ . : / = + - 和 @
- 鍵和值會區分大小寫
- 您不能使用 `aws:`做為金鑰的字首，因為它已保留供 AWS 使用

使用標籤

下列 Fault Injection Service AWS (AWS FIS) 資源支援標記：

- 動作
- Experiments
- 實驗範本

您可以使用 主控台來處理實驗和實驗範本的標籤。如需詳細資訊，請參閱下列內容：

- 標記實驗
- 標記實驗範本

您可以使用下列 AWS CLI 命令來使用動作、實驗和實驗範本的標籤：

- tag-resource – 將標籤新增至資源。
- untag-resource – 從資源移除標籤。
- list-tags-for-resource – 列出特定資源的標籤。

Fault Injection Service AWS 的配額和限制

您的 AWS 帳戶 具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定的。您可以在下表中請求增加標記為可調整的配額。

若要檢視您帳戶中 AWS FIS 的配額，請開啟 Service Quotas 主控台。在導覽窗格中，選擇 AWS 服務，然後選取 AWS Fault Injection Service。高達 且包含自動核准配額的值會立即套用。自動核准的配額概述於下表的描述欄中。如果您需要超過自動核准限制的配額，請提交請求。超過自動核准限制的值會由客戶支援審查，並盡可能核准。

若要請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。

您的 AWS 帳戶 具有下列與 AWS FIS 相關的配額。

名稱	預設	可調整	描述
動作持續時間，以小時為單位	每個支援的區域：12	否	在目前區域中，允許在此帳戶中執行一個動作的最大時數。
每個實驗範本的動作	每個受支援的區域：20	否	您可以在目前區域中此帳戶中的實驗範本中建立的最大動作數量。
作用中實驗	每個受支援的區域：5	否	您可以在目前區域中的此帳戶中同時執行的作用中實驗數量上限。
以天為單位完成實驗資料保留	每個支援的區域：120	否	允許 AWS FIS 保留目前區域中此帳戶中已完成實驗資料的最大天數。
實驗持續時間，以小時為單位	每個支援的區域：12	否	在目前區域中，允許在此帳戶中執行一個實驗的最大時數。

名稱	預設	可調整	描述
實驗範本	每個受支援的區域：500	否	您可以在目前區域中在此帳戶中建立的實驗範本數目上限。
aws：network：route-table-disrupt-cross-region-connectivity 中的受管字首清單數目上限	每個受支援的區域：15	否	每個動作允許的最大受管字首清單數量 aws：network：route-table-disrupt-cross-region-connectivity。
aws：network：route-table-disrupt-cross-region-connectivity 中的路由表數目上限	每個受支援的區域：10	否	每個動作允許的最大路由表數量 aws：network：route-table-disrupt-cross-region-connectivity。
aws：network：route-table-disrupt-cross-region-connectivity 中的路由數量上限	每個受支援的區域：200	否	每個動作允許的最大路由數 aws：network：route-table-disrupt-cross-region-connectivity。
每個實驗的平行動作	每個受支援的區域：10	否	您可以在目前區域中此帳戶中的實驗中平行執行的最大動作數量。
每個實驗範本的停止條件	每個受支援的區域：5	否	您可以在目前區域中新增至此帳戶中實驗範本的停止條件數目上限。

名稱	預設	可調整	描述
aws：ec2：asg-insufficient-instance-capacity-error 的目標 Auto Scaling 群組	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：ec2：asg-insufficient-instance-capacity-error 可以鎖定的 Auto Scaling 群組數量上限。配額增加請求會自動核准最多 500 個值。
aws：s3：bucket-pause-replication 的目標儲存貯體	每個受支援的區域：20	是	每次實驗使用標籤識別目標時，aws：s3：bucket-pause-replication 可以鎖定的 S3 儲存貯體數量上限。配額增加請求會自動核准最多 25 個值。
aws：ecs：drain-container-instances 的目標叢集	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：ecs：drain-container-instances 可以鎖定的叢集數量上限。配額增加請求會自動核准最多 100 個值。
aws：rds：failover-db-cluster 的目標叢集	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：rds：failover-db-cluster 可以鎖定的叢集數量上限。配額增加請求會自動核准高達 160 的值。

名稱	預設	可調整	描述
aws：rds：reboot-db-instances 的目標 DBInstances reboot-db-instances	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：rds：reboot-db-instances 可以鎖定的目標 DBInstances 數目上限。配額增加請求會自動核准最多 130 個值。
aws：ec2：reboot-instances 的目標執行個體	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：ec2：reboot-instances 可以鎖定的執行個體數量上限。配額增加請求會自動核准高達 600 的值。
aws：ec2：stop-instances 的目標執行個體	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：ec2：stop-instances 可以鎖定的執行個體數量上限。配額增加請求會自動核准最多 400 個值。
aws：ec2：terminate-instances 的目標執行個體	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：ec2：terminate-instances 可以鎖定的執行個體數量上限。配額增加請求會自動核准最多 300 個值。

名稱	預設	可調整	描述
aws：ssm：send-command 的目標執行個體	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：ssm：send-mand 可以鎖定的執行個體數量上限。配額增加請求會自動核准最多 50 個值。
aws：eks：terminate-nodegroup-instances 的目標節點群組	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：eks：terminate-nodegroup-instances 可以鎖定的 Nodegroups 數量上限。配額增加請求會自動核准最多 100 個值。
aws：eks：pod-cpu-stress 的目標 Pod	每個受支援的區域：50	是	每個實驗使用參數識別目標時，aws：eks：pod-cpu-stress 可以鎖定的最大 Pod 數量。配額增加請求會自動核准高達 1000 的值。
aws：eks：pod-delete 的目標 Pod	每個受支援的區域：50	是	當您使用參數識別每個實驗的目標時，aws：eks：pod-delete 可以鎖定的最大 Pod 數量。配額增加請求會自動核准高達 1000 的值。

名稱	預設	可調整	描述
aws：eks：pod-io-stress 的目標 Pod	每個受支援的區域：50	是	每次實驗使用參數識別目標時，aws：eks：pod-io-stress 可以鎖定的最大 Pod 數量。配額增加請求會自動核准高達 1000 的值。
aws：eks：pod-memory-stress 的目標 Pod	每個受支援的區域：50	是	每次實驗使用參數識別目標時，aws：eks：pod-memory-stress 可以鎖定的最大 Pod 數量。配額增加請求會自動核准高達 1000 的值。
aws：eks：pod-network-blackhole-port 的目標 Pod	每個受支援的區域：50	是	每次實驗使用參數識別目標時，aws：eks：pod-network-blackhole-port 可以鎖定的 Pod 數量上限。配額增加請求會自動核准高達 1000 的值。
aws：eks：pod-network-latency 的目標 Pod	每個受支援的區域：50	是	每次實驗使用參數識別目標時，aws：eks：pod-network-latency 可以鎖定的 Pod 數量上限。配額增加請求會自動核准高達 1000 的值。

名稱	預設	可調整	描述
aws：eks：pod-network-packet-loss 的目標 Pod	每個受支援的區域：50	是	每次實驗使用參數識別目標時，aws：eks：pod-network-packet-loss 的目標 Pod 數量上限。配額增加請求會自動核准高達 1000 的值。
aws：elasticache：interrupt-cluster-az-power 的目標 ReplicationGroups - 已規劃棄用	每個受支援的區域：5	是	每個實驗使用標籤/參數識別目標時，aws：elasticache：interrupt-cluster-az-power 可以鎖定的 ReplicationGroups 數量上限。配額增加請求會自動核准最多 5 個值。
aws：elasticache：replicationgroup-interrupt-az-power 的目標 ReplicationGroups	每個受支援的區域：5	是	每個實驗可以鎖定的 ReplicationGroups 數量上限：elasticache：replicationgroup-interrupt-az-power。每日限制適用於以 ReplicationGroups 為目標。如需詳細資訊，請造訪：https：//https://docs.aws.amazon.com/fis/latest/userguide/fis-quotas.html。配額增加請求會自動核准最多 20 個值。

名稱	預設	可調整	描述
aws:ec2:send-spot-instance-interruptions 的目標 Spot 執行個體	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，可以鎖定的 SpotInstances 數量上限 aws：ec2：send-spot-instance-interruptions。配額增加請求會自動核准最多 100 個值。
aws：network：disrupt-connectivity 的目標子網路	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：network：disrupt-connectivity 可以鎖定的子網路數量上限。高於 5 的配額僅適用於參數範圍：全部。如果您需要提高其他範圍類型的配額，請聯絡客戶支援。配額增加請求會自動核准最多 100 個值。
aws：network：route-table-disrupt-cross-region-connectivity 的目標子網路	每個受支援的區域：6	是	每次實驗使用標籤識別目標時，可以鎖定的子網路數量上限 aws：network：route-table-disrupt-cross-region-connectivity。配額增加請求會自動核准最多 50 個值。
aws：ecs：stop-task 的目標任務	每個受支援的區域：5	是	每個實驗使用標籤識別目標時，aws：ecs：stop-task 可以鎖定的任務數量上限。配額增加請求會自動核准最多 500 個值。

名稱	預設	可調整	描述
aws：ecs：task-cpu-stress 的目標任務	每個受支援的區域：5	是	每次實驗使用標籤/參數識別目標時，aws：ecs：task-cpu-stress 可以鎖定的任務數量上限。配額增加請求會自動核准最多 50 個值。
aws：ecs：task-io-stress 的目標任務	每個受支援的區域：5	是	每個實驗使用標籤/參數識別目標時，aws：ecs：task-io-stress 可以鎖定的任務數量上限。配額增加請求會自動核准最多 50 個值。
aws：ecs：task-kill-process 的目標任務	每個受支援的區域：5	是	每次實驗使用標籤/參數識別目標時，aws：ecs：task-kill-process 可以鎖定的任務數量上限。配額增加請求會自動核准最多 50 個值。
aws：ecs：task-network-blackhole-port 的目標任務	每個受支援的區域：5	是	每次實驗使用標籤/參數識別目標時，aws：ecs：task-network-blackhole-port 可以鎖定的任務數量上限。配額增加請求會自動核准最多 50 個值。

名稱	預設	可調整	描述
aws：ecs：task-network-latency 的目標任務	每個受支援的區域：5	是	每次實驗使用標籤/參數識別目標時，aws：ecs：task-network-latency 可以鎖定的任務數量上限。配額增加請求會自動核准最多 50 個值。
aws：ecs：task-network-packet-loss 的目標任務	每個受支援的區域：5	是	每個實驗使用標籤/參數識別目標時，aws：ecs：task-network-packet-loss 的目標任務數量上限。配額增加請求會自動核准最多 50 個值。
Target TransitGateways for aws：network：transit-gateway-disrupt-cross-region-connectivity	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，可以鎖定的傳輸閘道數量上限：network：transit-gateway-disrupt-cross-region-connectivity。配額增加請求會自動核准最多 50 個值。
aws：ebs：pause-volume-io 的目標磁碟區	ca-central-1：160 eu-central-1：160 每個其他支援的區域：5	是	每個實驗使用標籤識別目標時，aws：ebs：pause-volume-io 可以鎖定的最大磁碟區數量。配額增加請求會自動核准高達 160 的值。

名稱	預設	可調整	描述
每個實驗範本的目標帳戶組態	每個受支援的區域：10	是	您可以在目前區域中為此帳戶中的實驗範本建立的目標帳戶組態數目上限。配額增加請求會自動核准最多 40 個值。
aws：lambda：invocation-add-delay 動作的目標函數。	每個受支援的區域：5	是	當您每個實驗使用標籤識別目標時，aws：lambda：invocation-add-delay 可以鎖定的 Lambda 函數數量上限。配額增加請求會自動核准最多 140 個值。
aws：lambda：invocation-error 動作的目標函數。	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：lambda：invocation-error 可以鎖定的 Lambda 函數數量上限。配額增加請求會自動核准最多 140 個值。
aws：lambda：invocation-http-integration-response 動作的目標函數。	每個受支援的區域：5	是	每次實驗使用標籤識別目標時，aws：lambda：invocation-http-integration-response 可以鎖定的 Lambda 函數數量上限。配額增加請求會自動核准最多 140 個值。

名稱	預設	可調整	描述
aws：dynamodb：global-table-pause-replication 動作的目標資料表	每個受支援的區域：5	是	每個實驗可以鎖定的全域資料表數量上限 aws：dynamodb：global-table-pause-replication。配額增加請求會自動核准最多 60 個值。

您的 AWS FIS 使用受下列其他限制的約束：

名稱	限制
aws：elasticache：replicationgroup-interrupt-az-power 的目標 ReplicationGroups 每日限制	限制為每個區域每個帳戶每天目標 20 個 ReplicationGroups。您可以在 AWS Support Center 主控台中建立支援案例來請求增加。

文件歷史紀錄

下表說明 AWS Fault Injection Service 使用者指南中的重要文件更新。

變更	描述	日期
AWS FIS 中的 ARC 支援	您可以使用 AWS FIS 來測試 ARC 區域自動轉移如何在 AZ 電源中斷期間自動復原您的應用程式。	2025 年 3 月 26 日
新的實驗報告組態	您現在可以讓 AWS FIS 產生實驗的報告，以摘要來自 CloudWatch 儀表板的實驗動作和回應。	2024 年 11 月 12 日
新的 Lambda 動作	您現在可以使用 aws：lambda：function 動作，將錯誤注入 Lambda 函數的調用中。	2024 年 10 月 31 日
新的安全控制桿功能	AWS FIS 現在支援安全控制桿，可讓您快速停止所有執行中的實驗，並防止新的實驗啟動。	2024 年 9 月 3 日
新的故障診斷章節	AWS FIS 新增了故障診斷指南，其中包含失敗實驗的錯誤代碼和內容。	2024 年 8 月 13 日
新動作	您現在可以使用 `aws:dynamodb:global-table-pause-replication` 動作來暫停目標全域資料表及其複本資料表之間的資料複寫。將不再支援 `aws:dynamodb:encrypted-glob`	2024 年 4 月 24 日

	al-table-pause-replication 動作。	
新動作模式實驗選項	您可以將動作模式設定為 skip-all，以在執行實驗之前產生目標預覽。	2024 年 3 月 13 日
AWS 受管政策更新	AWS FIS 已更新現有的 受管政策。	2024 年 1 月 25 日
新的案例和動作	您現在可以使用 AWS FIS 案例跨區域：連線和可用區域可用性：電力中斷。	2023 年 11 月 30 日
新動作	您現在可以使用 aws:ec2:asg-insufficient-instance-capacity-error動作。	2023 年 11 月 30 日
新動作	您現在可以使用 aws:ec2:api-insufficient-instance-capacity-error動作。	2023 年 11 月 30 日
新動作	您現在可以使用 aws:network:route-table-disrupt-cross-region-connectivity動作。	2023 年 11 月 30 日
新動作	您現在可以使用 aws:network:transit-gateway-disrupt-cross-region-connectivity動作。	2023 年 11 月 30 日
新動作	您現在可以使用 aws:dynamodb:encrypted-global-table-pause-replication動作。	2023 年 11 月 30 日
新動作	您現在可以使用 aws:s3:bucket-pause-replication動作。	2023 年 11 月 30 日

新動作	您現在可以使用 aws:elasticache:interrupt-cluster-az-power動作。	2023 年 11 月 30 日
新的實驗選項	您現在可以將 AWS FIS 實驗選項用於帳戶目標和空目標解析。	2023 年 11 月 27 日
AWS FIS 的名稱變更	將服務名稱更新為 AWS Fault Injection Service。	2023 年 11 月 15 日
AWS 受管政策更新	AWS FIS 已更新現有的 受管政策。	2023 年 11 月 13 日
新的案例程式庫	您現在可以使用 AWS FIS 案例程式庫功能。	2023 年 11 月 7 日
新的實驗排程器	您現在可以使用 AWS FIS 實驗排程器功能。	2023 年 11 月 7 日
AWS 受管政策更新	AWS FIS 已更新現有的 受管政策。	2023 年 6 月 2 日
新動作	您可以使用新的 aws:ecs:task和 aws:eks:pod動作。	2023 年 6 月 1 日
AWS 受管政策更新	AWS FIS 已更新現有的 受管政策。	2023 年 6 月 1 日
新的預先設定 SSM 文件	您可以使用下列預先設定的 SSM 文件：AWSFIS-Run-Disk-Fill。	2023 年 4 月 28 日
新動作	您可以使用 aws:ebs:pause-volume-io動作，在目標磁碟區與其連接的執行個體之間暫停 I/O。	2023 年 1 月 27 日

新動作	您可以使用 aws:network:disrupt-connectivity動作來拒絕目標子網路的特定流量類型。	2022 年 10 月 26 日
新動作	您可以使用 aws:eks:inject-kubernetes-custom-resource動作在單一目標叢集上執行 ChaosMesh 或 Litmus 實驗。	2022 年 7 月 7 日
實驗記錄	您可以設定實驗範本，將實驗活動日誌傳送至 CloudWatch Logs 或 S3 儲存貯體。	2022 年 2 月 28 日
新通知	當實驗狀態變更時， AWS FIS 會發出通知。這些通知可透過 Amazon EventBridge 做為事件提供。	2022 年 2 月 24 日
新動作	您可以使用 aws:ecs:stop-task動作來停止指定的任務。	2022 年 2 月 9 日
新動作	您可以使用 aws:cloudwatch:assert-alarm-state動作來驗證指定的警示是否處於其中一個指定的警示狀態。	2021 年 11 月 5 日
新的預先設定 SSM 文件	您可以使用下列預先設定的 SSM 文件：AWSFIS-Run-IO-Stress、AWSFIS-Run-Network-Blackhold-Port、AWSFIS-Run-Network-Latency-Sources、AWSFIS-Run-Network-Packet-Loss 和 AWSFIS-Run-Network-Packet-Loss-Sources。	2021 年 11 月 4 日

新動作	您可以使用 aws:ec2:send-spot-instance-interruptions動作將 Spot 執行個體中斷通知傳送至目標 Spot 執行個體，然後中斷目標 Spot 執行個體。	2021 年 10 月 20 日
新動作	您可以使用 aws:ssm:start-automation-execution動作來啟動 Automation Runbook 的執行。	2021 年 9 月 17 日
初始版本	Fault Injection Service AWS 使用者指南的初始版本。	2021 年 3 月 15 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。