aws

사용 설명서

AWS Fault Injection 서비스

AWS Fault Injection 서비스: 사용 설명서

Table of Contents

AWS Fault Injection Service란 무엇입니까?

AWS Fault Injection Service(AWS FIS)는 AWS 워크로드에서 오류 주입 실험을 수행할 수 있는 관리형 서비스입니다. 오류 주입은 카오스 엔지니어링의 원칙을 기반으로 합니다. 이러한 실험은 애플리케이션이 어떻게 반응하는지 관찰할 수 있도록 방해 이벤트를 생성하여 애플리케이션에 스트레스를 줍니다. 그런 다음 이 정보를 사용하여 애플리케이션이 예상대로 작동하도록 애플리케이션의 성능과 복원력을 개선할 수 있습니다.

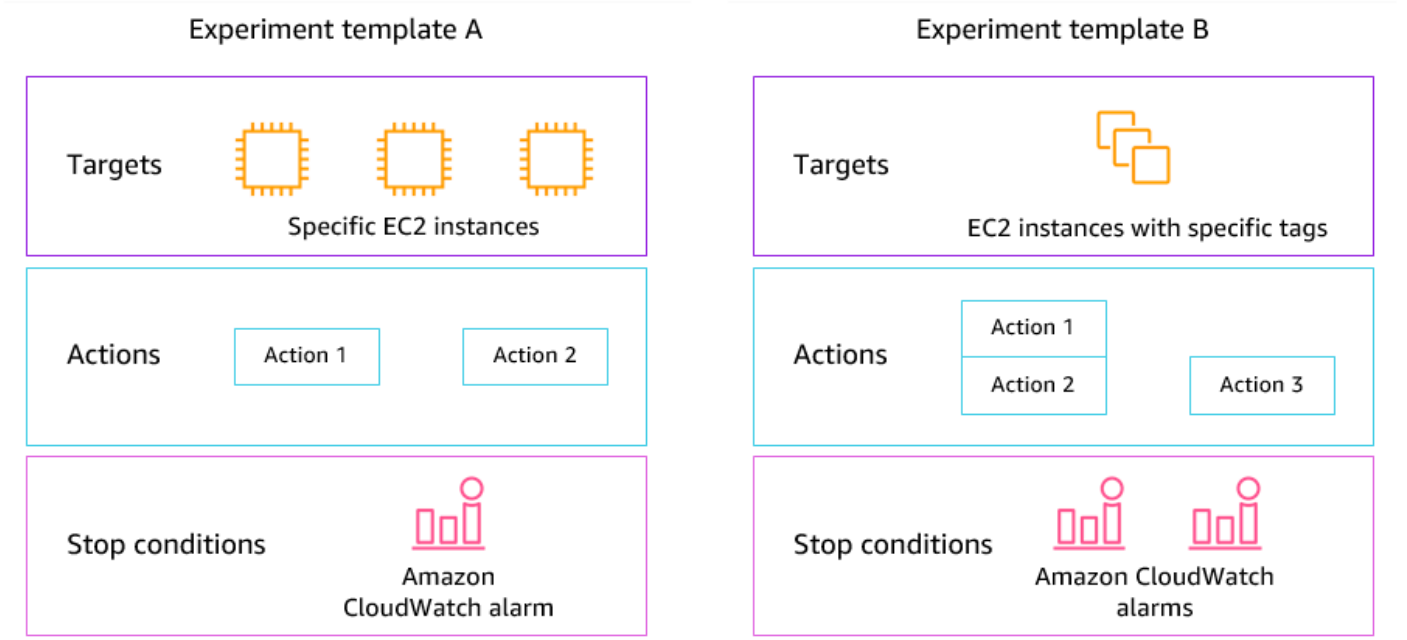
AWS FIS를 사용하려면 달리 찾기 어려울 수 있는 애플리케이션 문제를 발견하는 데 필요한 실제 조건을 생성하는 데 도움이 되는 실험을 설정하고 실행합니다. AWS FIS는 중단을 생성하는 템플릿과 특정 조건이 충족되는 경우 실험을 자동으로 롤백하거나 중지하는 등 프로덕션 환경에서 실험을 실행하는 데 필요한 제어 및 가드레일을 제공합니다.

> **Important**
>
> AWS FIS는 시스템의 실제 AWS 리소스에 대해 실제 작업을 수행합니다. 따라서 AWS FIS를 사용하여 프로덕션 환경에서 실험을 실행하기 전에 계획 단계를 완료하고 사전 프로덕션 환경에서 실험을 실행하는 것이 좋습니다.

실험 계획에 대한 자세한 내용은 테스트 신뢰성 및 AWS FIS 실험 계획을 참조하세요. AWS FIS에 대한 자세한 내용은 AWS Fault Injection Service를 참조하세요.

AWS FIS 개념

AWS FIS를 사용하려면 AWS 리소스에서 실험을 실행하여 장애 조건에서 애플리케이션 또는 시스템이 어떻게 작동하는지에 대한 이론을 테스트합니다. 실험을 실행하려면 먼저 실험 템플릿을 만들어야 합니다. 실험 템플릿은 실험의 청사진입니다. 여기에는 실험에서 수행할 작업, 목표, 중지 조건이 포함됩니다. 실험 템플릿을 만든 후 이를 사용하여 실험을 실행할 수 있습니다. 실험이 실행되는 동안 진행 상황을 추적하고 상태를 볼 수 있습니다. 실험의 모든 작업이 실행되면 실험이 완료된 것입니다.



작업

작업은 AWS FIS가 실험 중에 리소스에 대해 AWS 수행하는 활동입니다. AWS FIS는 AWS 리소스 유형에 따라 미리 구성된 작업 세트를 제공합니다. 각 작업은 실험 중 지정된 기간 동안 또는 실험을 중지할 때까지 실행됩니다. 작업은 순차적으로 또는 동시에 (병렬) 실행할 수 있습니다.

대상

대상은 AWS FIS가 실험 중에 작업을 수행하는 하나 이상의 AWS 리소스입니다. 특정 리소스를 선택하거나 태그 또는 상태와 같은 특정 기준에 따라 리소스 그룹을 선택할 수 있습니다.

중지 조건

AWS FIS는 AWS 워크로드에서 실험을 안전하게 실행하는 데 필요한 제어 및 가드레일을 제공합니다. 중지 조건은 Amazon CloudWatch 경보로 정의한 임계값에 도달할 경우 실험을 중지하는 메커니즘입니다. 실험이 실행되는 동안 중지 조건이 트리거되면 AWS FLS는 실험을 중지합니다.

지원됨 AWS 서비스

AWS FIS는 서비스 전반에 AWS 걸쳐 특정 유형의 대상에 대해 사전 구성된 작업을 제공합니다. 지원되는 서비스 및 해당 작업 목록은 AWS FIS 작업 참조를 참조하세요.

단일 계정 실험의 경우 대상 리소스는 실험 AWS 계정 과 동일한에 있어야 합니다. AWS FIS 다중 계정 실험을 사용하여 다른 AWS 계정 계정의 리소스를 대상으로 하는 AWS FIS 실험을 실행할 수 있습니다.

자세한 내용은 AWS FIS에 대한 작업 단원을 참조하십시오.

AWS FIS 액세스

다음 방법 중 하나로 AWS FIS를 사용할 수 있습니다.

- AWS Management Console - AWS FIS에 액세스하는 데 사용할 수 있는 웹 인터페이스를 제공합니다. 자세한 내용은 AWS Management Console작업 단원을 참조하세요.
- AWS Command Line Interface (AWS CLI) - AWS FIS를 비롯한 다양한 AWS 서비스에 대한 명령을 제공하며 Windows, macOS 및 Linux에서 지원됩니다. 자세한 내용은 AWS Command Line Interface 단원을 참조하십시오. AWS FIS 명령에 대한 자세한 내용은 명령 참조의 fis를 AWS CLI 참조하세요.
- AWS CloudFormation - AWS 리소스를 설명하는 템플릿을 생성합니다. 템플릿을 사용하여 이러한 리소스를 하나의 단위로 프로비저닝하고 관리할 수 있습니다. 자세한 내용은 AWS Fault Injection Service 리소스 유형 참조를 참조하세요.
- AWS SDKs- 언어별 APIs 제공하고 서명 계산, 요청 재시도 처리, 오류 처리와 같은 많은 연결 세부 정보를 처리합니다. 자세한 내용은 AWS SDK를 참조하십시오.
- HTTPS API - HTTPS 요청을 사용하여 직접 호출할 수 있는 하위 수준의 API 작업을 제공합니다. 자세한 내용은 AWS Fault Injection Service API 참조를 참조하세요.

AWS FIS 요금

실험의 대상 계정 수에 따라 작업이 시작부터 끝까지 실행되는 분당 요금이 부과됩니다. 자세한 내용은 AWS FIS 요금을 참조하세요.

AWS FIS 실험 계획

오류 주입은 서버 중단이나 API 제한과 같은 운영 중단 이벤트를 발생시켜 테스트 또는 프로덕션 환경에서 애플리케이션에 스트레스를 주는 프로세스입니다. 시스템이 어떻게 반응하는지 관찰한 다음 개선을 구현할 수 있습니다. 시스템에서 실험을 실행하면 시스템에 의존하는 고객에게 영향을 미치기 전에 통제된 방식으로 시스템상의 약점을 식별하는 데 도움이 될 수 있습니다. 그러면 문제를 사전에 해결하여 예측할 수 없는 결과를 예방할 수 있습니다.

AWS FIS를 사용하여 오류 주입 실험 실행을 시작하기 전에 다음 원칙과 지침을 숙지하는 것이 좋습니다.

> **Important**
>
> AWS FIS는 시스템의 실제 AWS 리소스에 대해 실제 작업을 수행합니다. 따라서 AWS FIS를 사용하여 실험을 실행하기 전에 먼저 사전 프로덕션 또는 테스트 환경에서 계획 단계와 테스트를 완료하는 것이 좋습니다.

내용

기본 원칙 및 지침

AWS FIS로 실험을 시작하기 전에 다음 단계를 수행합니다.

1. 실험 대상 배포 식별 - 먼저 대상 배포를 식별하세요. 실험이 처음인 경우 사전 프로덕션 또는 테스트 환경에서 시작하는 것이 좋습니다.
2. 애플리케이션 아키텍처 검토 - 각 구성 요소의 애플리케이션 구성 요소, 종속성 및 복구 절차를 모두 식별했는지 확인해야 합니다. 먼저 애플리케이션 아키텍처를 검토하세요. 애플리케이션에 따라 AWS Well-Architected Framework를 참조하세요.
3. 정상 상태 동작 정의 - 지연 시간, CPU 부하, 분당 로그인 실패, 재시도 횟수, 페이지 로드 속도 등 중요한 기술 및 비즈니스 지표를 기준으로 시스템의 정상 상태 동작을 정의합니다.
4. 가설 세우기 - 실험 중에 시스템 동작이 어떻게 변할 것으로 예상하는지에 대한 가설을 세우세요. 가설 정의는 다음과 같은 형식을 따릅니다.

##이 수행되는 경우 #### ## ## ### ### ##이 #을 초과해서는 안 됩니다.

예를 들어 인증 서비스에 대한 가설은 다음과 같을 수 있습니다. “네트워크 지연 시간이 10% 증가할 경우 로그인 실패 증가는 1% 미만입니다.” 실험이 완료된 후 애플리케이션 복원력이 비즈니스 및 기술 기대치에 부합하는지 평가합니다.

또한 AWS FIS로 작업할 때는 다음 지침을 따르는 것이 좋습니다.

- 항상 테스트 환경에서 AWS FIS 실험을 시작합니다. 프로덕션 환경에서는 절대 시작하지 마세요. 오류 주입 실험이 진행됨에 따라 테스트 환경 이외의 다른 통제된 환경에서도 실험할 수 있습니다.
- 하나의 대상에서 aws:ec2:stop-instances 작업을 실행하는 것과 같이 작고 간단한 실험부터 시작하여 애플리케이션 복원력에 대한 팀의 신뢰를 구축하세요.
- 오류 주입으로 인해 실제 문제가 발생할 수 있습니다. 주의를 기울여 진행하고 고객이 영향을 받지 않도록 테스트 인스턴스에 첫 번째 오류 주입이 이루어지도록 하세요.
- 테스트를 거듭하고 몇 가지 더 테스트해 보세요. 오류 주입은 잘 계획된 실험이 포함된 통제된 환경에서 구현되어야 합니다. 이를 통해 난류 조건을 견딜 수 있는 애플리케이션 및 도구의 성능에 대한 확신을 구축할 수 있습니다.
- 시작하기 전에 우수한 모니터링 및 경고 프로그램을 마련해 두는 것이 좋습니다. 이 기능이 없으면 실험의 영향을 이해하거나 측정할 수 없는데, 이는 지속 가능한 오류 주입 관행에 매우 중요한 요소입니다.

실험 계획 지침

AWS FIS를 사용하면 AWS 리소스에서 실험을 실행하여 오류 조건에서 애플리케이션 또는 시스템이 어떻게 작동하는지에 대한 이론을 테스트할 수 있습니다.

다음은 AWS FIS 실험을 계획하기 위한 권장 지침입니다.

- 정전 기록 검토 - 시스템의 과거 정전 및 이벤트를 검토하세요. 이를 통해 시스템의 전반적인 상태와 복원력을 파악할 수 있습니다. 시스템에서 실험을 시작하기 전에 시스템의 알려진 문제와 약점을 해결해야 합니다.
- 가장 큰 영향을 미치는 서비스 식별 - 서비스를 검토하고 서비스가 중단되거나 제대로 작동하지 않을 경우 최종 사용자나 고객에게 가장 큰 영향을 미치는 서비스를 식별하세요.

- 대상 시스템 식별 - 대상 시스템은 실험을 실행할 시스템입니다. AWS FIS를 처음 사용하거나 이전에 오류 주입 실험을 실행한 적이 없는 경우 사전 프로덕션 또는 테스트 시스템에서 실험을 실행하는 것으로 시작하는 것이 좋습니다.
- 팀원들과 논의 - 어떤 걱정을 하고 있는지 물어보세요. 고객의 우려를 입증하거나 반증하기 위한 가설을 세울 수 있습니다. 팀원들이 걱정하지 않는 것은 무엇인지 물어볼 수도 있습니다. 이 질문을 통해 두 가지 일반적인 오류, 즉 매몰 비용 오류와 확증 편향 오류가 드러날 수 있습니다. 팀원들의 답변을 기반으로 가설을 세우면 시스템 상태의 현실에 대한 자세한 정보를 제공하는 데 도움이 될 수 있습니다.
- 애플리케이션 아키텍처 검토 - 시스템 또는 애플리케이션을 검토하고 애플리케이션의 모든 구성 요소, 종속성 및 각 구성 요소의 복구 절차를 모두 식별했는지 확인하세요.

  AWS Well-Architected 프레임워크를 검토하는 것이 좋습니다. 이 프레임워크는 애플리케이션과 워크로드를 위한 안전하고 성능 및 복원력이 뛰어나며 효율적인 인프라를 구축할 수 있도록 지원합니다. 자세한 내용은 AWS Well-Architected를 참조하세요.
- 해당 지표 식별 - Amazon CloudWatch 지표를 사용하여 실험이 AWS 리소스에 미치는 영향을 모니터링할 수 있습니다. 이러한 지표를 사용하여 애플리케이션이 최적의 성능을 발휘할 때 기준선 또는 '안정 상태'를 판단할 수 있습니다. 그런 다음 실험 중 또는 실험 후에 이러한 지표를 모니터링하여 영향을 확인할 수 있습니다. 자세한 내용은 Amazon CloudWatch를 사용하여 AWS FIS 사용량 지표 모니터링 단원을 참조하십시오.
- 시스템에 적합한 성능 임곗값 정의 - 시스템에 허용 가능한 안정 상태를 나타내는 지표를 식별하세요. 이 지표를 사용하여 실험 중지 조건을 나타내는 CloudWatch 경보를 하나 이상 생성합니다. 경보가 트리거되면 실험이 자동으로 중지됩니다. 자세한 내용은 AWS FIS에 대한 중지 조건 단원을 참조하십시오.

AWS FIS 실험 템플릿 구성 요소

다음 구성 요소를 사용하여 실험 템플릿을 만들 수 있습니다.

작업

실행하려는 AWS FIS 작업. 지정한 순서대로 작업을 실행하거나 동시에 실행할 수 있습니다. 자세한 내용은 작업 단원을 참조하십시오.

대상

특정 작업이 수행되는 AWS 리소스입니다. 자세한 내용은 대상 단원을 참조하십시오.

중지 조건

허용되지 않는 애플리케이션 성능 임계값을 정의하는 CloudWatch 경보입니다. 실험이 실행되는 동안 중지 조건이 트리거되면 AWS FLS는 실험을 중지합니다. 자세한 내용은 중지 조건 단원을 참조하십시오.

실험 역할

사용자를 대신하여 실험을 실행할 수 있도록 AWS FIS에 필요한 권한을 부여하는 IAM 역할입니다. 자세한 내용은 실험 역할 단원을 참조하십시오.

실험 보고서 구성

실험 보고서를 활성화하는 구성입니다. 자세한 내용은 AWS FIS에 대한 실험 보고서 구성 단원을 참조하십시오.

실험 옵션

실험 템플릿의 옵션입니다. 자세한 내용은 에 대한 실험 옵션 AWS FIS 단원을 참조하십시오.

계정에 AWS FIS와 관련된 할당량이 있습니다. 예를 들어 실험 템플릿당 작업 수에는 할당량이 있습니다. 자세한 내용은 제한 사항 및 할당량 단원을 참조하십시오.

템플릿 구문

다음은 실험 템플릿의 구문입니다.

```
{
            "description": "string",
            "targets": {},
            "actions": {},
```

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

예시는 템플릿 예제 섹션을 참조하세요.

시작하기

를 사용하여 실험 템플릿을 생성하려면 섹션을 AWS Management Console참조하세요실험 템플릿 만들기.

를 사용하여 실험 템플릿을 생성하려면 섹션을 AWS CLI참조하세요AWS FIS 실험 템플릿 예제.

AWS FIS에 대한 작업

실험 템플릿을 만들려면 작업을 하나 이상 정의해야 합니다. AWS FIS에서 제공하는 사전 정의된 작업 목록은 섹션을 참조하세요작업 참조.

실험 중에는 작업을 한 번만 실행할 수 있습니다. 동일한 실험에서 동일한 AWS FIS 작업을 두 번 이상 실행하려면 다른 이름을 사용하여 템플릿에 여러 번 추가합니다.

내용

- 작업 구문
- 작업 식별자
- 작업 파라미터
- 작업 대상
- 작업 기간
- 작업 예

작업 구문

다음은 작업에 대한 구문입니다.

```
{
    "actions": {
```

```
        "action_name": {
            "actionId": "aws:service:action-type",
            "description": "string",
            "parameters": {
                "name": "value"
             },
            "startAfter": ["action_name", ...],
            "targets": {
                "ResourceType": "target_name"
            }
        }
    }
}
```

작업을 정의할 때 다음을 제공합니다.

action_name

작업의 이름입니다.

actionId

작업 식별자.

description

설명(선택 사항)입니다.

parameters

작업 파라미터.

startAfter

이 작업을 시작하기 전에 완료해야 하는 모든 작업입니다. 그렇지 않으면 실험이 시작될 때 이 작업이 실행됩니다.

targets

작업 대상.

예시는 the section called “작업 예” 섹션을 참조하세요.

작업 식별자

각 AWS FIS 작업에는 다음 형식의 식별자가 있습니다.

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

예를 들어 다음 작업은 대상 Amazon EC2 인스턴스를 중지합니다.

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

전체 작업 목록은 AWS FIS 작업 참조 섹션을 참조하세요.

작업 파라미터

일부 AWS FIS 작업에는 작업과 관련된 추가 파라미터가 있습니다. 이러한 파라미터는 작업이 실행될 때 AWS FIS에 정보를 전달하는 데 사용됩니다.

AWS FIS는 SSM 에이전트와 SSM 명령 문서를 사용하여 대상 인스턴스에 결함 조건을 생성하는 `aws:ssm:send-command` 작업을 사용하여 사용자 지정 결함 유형을 지원합니다. `aws:ssm:send-command` 작업에는 SSM 문서의 Amazon 리소스 이름(ARN)을 값으로 취하는 `documentArn` 파라미터가 포함됩니다. 실험 템플릿에 작업을 추가할 때 파라미터 값을 지정합니다.

`aws:ssm:send-command` 작업 파라미터 지정에 대한 자세한 내용은 aws:ssm:send-command 작업을 사용을 참조하세요.

가능한 경우 작업 파라미터 내에 롤백 구성(사후 조치라고도 함)을 입력할 수 있습니다. 사후 작업은 대상을 작업이 실행되기 전의 상태로 되돌립니다. 사후 작업은 작업 기간에 지정된 시간 이후에 실행됩니다. 모든 작업이 사후 작업을 지원할 수 있는 것은 아닙니다. 예를 들어, 작업으로 인해 Amazon EC2 인스턴스가 종료되는 경우, 종료된 후에는 인스턴스를 복구할 수 없습니다.

작업 대상

작업은 지정한 대상 리소스에서 실행됩니다. 대상을 정의한 후 작업을 정의할 때 이름을 지정할 수 있습니다.

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

AWS FIS 작업은 작업 대상에 대해 다음과 같은 리소스 유형을 지원합니다.

- AutoScalingGroups – Amazon EC2 Auto Scaling 그룹
- 버킷 - Amazon S3 버킷
- 클러스터 - Amazon EKS 클러스터

- 클러스터 - Amazon ECS, Aurora DSQL 또는 Amazon Aurora DB 클러스터
- DB 인스턴스 – Amazon RDS DB 인스턴스
- 함수 - AWS Lambda 함수
- 인스턴스 – Amazon EC2 인스턴스
- KinesisStreams – Kinesis 데이터 스트림
- ManagedResources - ARC 영역 전환에 대해 활성화된 Amazon EKS 클러스터, Amazon EC2 Application and Network Load Balancer 및 Amazon EC2 Auto Scaling 그룹입니다.
- MultiRegionClusters – Amazon MemoryDB 다중 리전 클러스터
- 노드 그룹 – Amazon EKS 노드 그룹
- 포드 – Amazon EKS의 Kubernetes 포드
- ReplicationGroups – ElastiCache 복제 그룹
- 역할 - IAM 역할
- 스팟 인스턴스 - Amazon EC2 스팟 인스턴스
- 서브넷 - VPC 서브넷
- 테이블 - Amazon DynamoDB 다중 리전 강력하며 최종적으로 일관된 글로벌 테이블
- 작업 - Amazon ECS 작업
- TransitGateways – 전송 게이트웨이
- VirtualInterfaces - Direct Connect 가상 인터페이스
- 볼륨 – Amazon EBS 볼륨
- VPCEndpoints – Amazon VPC 엔드포인트

예시는 the section called “작업 예” 섹션을 참조하세요.

작업 기간

작업에 작업 기간을 지정하는 데 사용할 수 있는 파라미터가 포함된 경우 기본적으로 지정된 기간이 경과한 후에만 작업이 완료된 것으로 간주됩니다. `emptyTargetResolutionMode` 실험 옵션을 `skip`로 설정한 경우, 확인된 대상이 없는 경우 작업이 'skipped' 상태로 즉시 완료됩니다. 예를 들어 기간을 5분으로 지정하면 AWS FIS는 5분 후에 작업이 완료된 것으로 간주합니다. 그런 다음 모든 작업이 완료되어야 다음 작업을 시작합니다.

기간은 작업 조건이 유지되는 기간 또는 지표가 모니터링되는 기간일 수 있습니다. 예를 들어, 지정된 기간 동안 지연 시간이 주입됩니다. 인스턴스 종료와 같이 거의 즉각적인 작업 유형의 경우, 지정된 기간 동안 중지 조건이 모니터링됩니다.

작업 파라미터 내에 사후 작업이 포함된 작업의 경우, 사후 작업은 작업이 완료된 후에 실행됩니다. 사후 작업을 완료하는 데 걸리는 시간으로 인해 지정된 작업 기간과 다음 작업의 시작(또는 다른 모든 작업이 완료된 경우 실험 종료) 시점 사이에 지연이 발생할 수 있습니다.

작업 예

작업의 예를 들면 다음과 같습니다.

예제

- EC2 인스턴스 중지
- 스팟 인스턴스 중단
- 네트워크 트래픽 방해
- EKS 작업자 종료
- ARC 영역 자동 전환 시작

예: EC2 인스턴스 중지

다음 작업은 *targetInstances*라는 대상을 사용하여 식별된 EC2 인스턴스를 중지합니다. 2분 후에 대상 인스턴스를 다시 시작합니다.

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

예: 스팟 인스턴스 중단

다음 작업은 *targetSpotInstances*라는 대상을 사용하여 식별된 스팟 인스턴스를 중지합니다. 스팟 인스턴스가 중단될 때까지 2분 정도 기다립니다.

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

예: 네트워크 트래픽 방해

다음 작업은 대상 서브넷과 다른 가용 영역에 있는 서브넷 간의 트래픽을 거부합니다.

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

예: EKS 작업자 종료

다음 작업은 *targetNodeGroups*라는 대상을 사용하여 식별된 EKS 클러스터의 EC2 인스턴스 중 50%를 종료합니다.

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

```
            "Nodegroups": "targetNodeGroups"
        }
    }
}
```

예: ARC 영역 자동 전환 시작

다음 작업은 파라미터 내 *duration-in-parameteres* 동안 관리형 리소스를 *az-in-parameters*에서 다른 곳으로 이동하는 ARC 영역 자동 전환을 시작합니다. 리소스 유형은 AWS FIS 실험 템플릿에서 대상 이름의 키로 ManagedResources 사용됩니다.

```
{
    "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 FIS의 대상

대상은 실험 중에 AWS Fault Injection Service(AWS FIS)에서 작업을 수행하는 하나 이상의 AWS 리소스입니다. 대상은 실험과 동일한 AWS 계정에 있을 수도 있고, 다중 계정 실험을 사용하여 다른 계정에 있을 수도 있습니다. 다른 계정의 리소스를 타겟팅하는 방법에 대해 자세히 알아보려면 다중 계정 실험 작업 단원을 참조하세요.

실험 템플릿을 생성할 때 대상을 정의합니다. 실험 템플릿에서 동일한 대상을 여러 작업에 사용할 수 있습니다.

AWS FIS는 작업 세트에서 작업을 시작하기 전에 실험 시작 시 모든 대상을 식별합니다. AWS FIS는 전체 실험에 대해 선택한 대상 리소스를 사용합니다. 대상이 발견되지 않으면 실험이 실패합니다.

목차

대상 구문

다음은 대상에 대한 구문입니다.

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

대상을 정의할 때 다음을 제공합니다.

target_name

대상의 이름입니다.

resourceType

리소스 유형.

resourceArns

특정 리소스의 Amazon 리소스 이름(ARN)입니다.

resourceTags

특정 리소스에 적용된 태그.

parameters

특정 속성을 사용하여 대상을 식별하는 파라미터.

filters

리소스 필터는 특정 속성을 사용하여 식별된 대상 리소스의 범위를 지정합니다.

selectionMode

식별된 리소스의 선택 모드.

예시는 the section called “예시 대상” 섹션을 참조하세요.

조건 키

각 AWS FIS 작업은 특정 AWS 리소스 유형에서 수행됩니다. 대상을 정의할 때 리소스 유형을 하나만 지정해야 합니다. 작업의 대상을 지정하는 경우, 대상은 작업에서 지원하는 리소스 유형이어야 합니다.

AWS FIS에서 지원하는 리소스 유형은 다음과 같습니다.

- aws:arc:zonal-shift-managed-resource – ARC 영역 전환에 등록된 AWS 리소스
- aws:directconnect:virtual-interface – Direct Connect 가상 인터페이스
- aws:dsql:cluster – Amazon Aurora DSQL 클러스터
- aws:dynamodb:global-table – Amazon DynamoDB 다중 리전 글로벌 테이블
- aws:ec2:autoscaling-group – Amazon EC2 Auto Scaling 그룹
- aws:ec2:ebs-volume – Amazon EBS 볼륨
- aws:ec2:instance – Amazon EC2 인스턴스
- aws:ec2:spot-instance – Amazon EC2 스팟 인스턴스
- aws:ec2:subnet – Amazon VPC 서브넷
- aws:ec2:transit-gateway – 전송 게이트웨이
- aws:ec2:vpc-endpoint – Amazon VPC 엔드포인트
- aws:ecs:cluster – Amazon ECS 클러스터
- aws:ecs:task – Amazon ECS 태스크
- aws:eks:cluster – Amazon EKS 클러스터
- aws:eks:nodegroup – Amazon EKS 노드 그룹
- aws:eks:pod – Kubernetes 포드
- aws:elasticache:replicationgroup – ElastiCache 복제 그룹
- aws:iam:role – IAM 역할
- aws:kinesis:stream – Amazon Kinesis 데이터 스트림
- aws:lambda:function – AWS Lambda 함수
- aws:memorydb:multi-region-cluster – Amazon MemoryDB 다중 리전 클러스터
- aws:rds:cluster – Amazon Aurora DB 클러스터
- aws:rds:db – Amazon RDS DB 인스턴스
- aws:s3:bucket – Amazon S3 버킷

대상 리소스 식별

AWS FIS 콘솔에서 대상을 정의할 때 대상으로 지정할 특정 AWS 리소스(특정 리소스 유형)를 선택할 수 있습니다. 또는 사용자가 제공한 기준에 따라 AWS FIS가 리소스 그룹을 식별하도록 할 수 있습니다.

대상 리소스를 식별하기 위해 다음을 지정할 수 있습니다.

- 리소스 IDs- 특정 리소스의 AWS 리소스 IDs. 모든 리소스 ID는 동일한 유형의 리소스를 나타내야 합니다.
- 리소스 태그 - 특정 AWS 리소스에 적용되는 태그입니다.
- 리소스 필터 - 특정 속성을 가진 리소스를 나타내는 경로와 값입니다. 자세한 내용은 리소스 필터 단원을 참조하십시오.
- 리소스 파라미터 - 특정 기준을 충족하는 리소스를 나타내는 파라미터입니다. 자세한 내용은 리소스 파라미터 단원을 참조하십시오.

고려 사항

- 동일한 대상에 리소스 ID와 리소스 태그를 모두 지정할 수는 없습니다.
- 동일한 대상에 리소스 ID와 리소스 태그를 모두 지정할 수는 없습니다.
- 빈 태그 값으로 리소스 태그를 지정하는 경우 와일드카드와 동일하지 않습니다. 지정된 태그 키와 빈 태그 값이 있는 태그가 있는 리소스를 매칭합니다.
- 둘 이상의 태그를 지정하는 경우 대상 리소스에 지정된 모든 태그가 있어야 태그를 선택할 수 있습니다(AND).

리소스 필터

리소스 필터는 특정 속성에 따라 대상 리소스를 식별하는 쿼리입니다. AWS FIS는 지정한 리소스 유형에 따라 AWS 리소스의 정식 설명이 포함된 API 작업의 출력에 쿼리를 적용합니다. 쿼리와 일치하는 속성을 가진 리소스는 대상 정의에 포함됩니다.

각 필터는 속성 경로 및 가능한 값으로 표현됩니다. 경로는 마침표로 구분된 일련의 요소로, 리소스에 대한 설명 작업 출력의 속성에 도달하기 위한 경로를 설명합니다. 각 기간은 요소의 확장을 나타냅니다. 리소스에 대한 Describe 작업의 출력이 카멜 표기법인 경우에도 각 요소는 파스칼 표기법으로 표현해야 합니다. 예를 들어, `availablityZone`이 아니라 `AvailabilityZone`을 속성 요소로 사용해야 합니다.

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

다음 로직은 모든 리소스 필터에 적용됩니다.

- 경로가 동일한 필터를 포함하여 여러 필터가 제공되는 경우 리소스를 선택하려면 모든 필터를 일치해야 합니다. AND
- 단일 필터에 여러 값이 제공되는 경우 리소스를 선택하려면 하나의 값과 일치해야 합니다. OR
- API 설명 호출의 경로 위치에 여러 값이 있는 경우 리소스를 선택하려면 하나의 값과 일치해야 합니다. OR
- 태그 키/값 페어에서 일치시키려면 태그별로 대상 리소스를 대신 선택해야 합니다(위 참조).

다음 표에는 각 리소스 유형에 대한 정식 설명을 가져오는 데 사용할 수 있는 API 작업과 AWS CLI 명령이 나와 있습니다. AWS FIS는 사용자를 대신하여 이러한 작업을 실행하여 지정한 필터를 적용합니다. 해당 설명서에는 기본적으로 결과에 포함되는 리소스가 설명되어 있습니다. 예를 들어 DescribeInstances 설명서에는 최근에 종료된 인스턴스가 결과에 표시될 수 있다고 나와 있습니다.

리소스 유형	API 작업	AWS CLI 명령
aws:arc:zonal-shift-managed-resource	ListManagedResources	list-managed-resources
aws:directconnect:virtual-interface	DescribeVirtualInterfaces	describe-virtual-interfaces
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes
aws:ec2:instance	DescribeInstances	describe-instances
aws:ec2:subnet	DescribeSubnets	describe-subnets

리소스 유형	API 작업	AWS CLI 명령
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ec2:vpc-endpoint	DescribeVpcEndpoints	describe-vpc-endpoints
aws:ecs:cluster	DescribeClusters	describe-clusters
aws:ecs:task	DescribeTasks	describe-tasks
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	describe-nodegroup
aws:elasticache:replication group	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	list-roles
aws:kinesis:stream	DescribeStreamSummary	describe-stream-summary
aws:lambda:function	ListFunctions	list-functions
aws:memorydb:multi-region-clustern	DescribeMultiRegionClusters	describe-multi-region-clusters
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
aws:dsql:cluster	GetCluster	get-cluster

예시는 the section called “필터 예제” 섹션을 참조하세요.

리소스 파라미터

리소스 파라미터는 특정 기준에 따라 대상 리소스를 식별합니다.

다음 리소스 유형은 파라미터를 지원합니다.

aws:ec2:ebs-volume

- `availabilityZoneIdentifier` - 대상 볼륨이 포함된 가용 영역의 코드(예: us-east-1a) 입니다.

aws:ec2:subnet

- `availabilityZoneIdentifier` - 대상 서브넷이 포함된 가용 영역의 코드(예: us-east-1a) 또는 AZ ID(예: use1-az1) 입니다.
- `vpc` - 대상 서브넷이 포함된 VPC입니다. 계정당 하나 이상의 VPC를 지원하지 않습니다.

aws:ecs:task

- `cluster` - 대상 작업이 포함된 클러스터입니다.
- `service` - 대상 작업이 포함된 서비스입니다.

aws:eks:pod

- `availabilityZoneIdentifier` – 선택 사항입니다. 대상 포드가 포함된 가용 영역입니다. 예를 들어 `us-east-1d`입니다. 포드의 호스트 IP와 클러스터 서브넷의 CIDR을 비교하여 포드의 가용 영역을 결정합니다.
- `clusterIdentifier` - 필수입니다. 대상 EKS 클러스터의 이름 또는 ARN.
- `namespace` - 필수입니다. 대상 포드의 Kubernetes 네임스페이스.
- `selectorType` - 필수입니다. 선택기 유형. 가능한 값은 `labelSelector`, `deploymentName`, `podName`입니다.
- `selectorValue` - 필수입니다. 선택기 값입니다. 이 값은 `selectorType`의 값에 따라 달라집니다.
- `targetContainerName` – 선택 사항입니다. 포드 사양에 정의된 대상 컨테이너의 이름입니다. 기본값은 각 대상 포드 사양에 정의된 첫 번째 컨테이너입니다.

aws:lambda:function

- `functionQualifier` – 선택 사항입니다. 대상으로 지정할 함수의 버전 또는 별칭입니다. 한정자를 지정하지 않으면 모든 호출이 대상으로 고려됩니다. 버전이 여러 개인 별칭을 지정하면 별칭이 포함된 ARN을 사용하여 호출되는 한 별칭에 포함된 모든 버전이 대상으로 간주됩니다. 특수 별칭`$LATEST`을 사용하는 경우 기본 함수 ARN에 대한 호출과 ARN`$LATEST`에를 포함한 호출이 결함 주입에 고려됩니다. Lambda 버전에 대한 자세한 내용은 AWS Lambda 사용 설명서의 Lambda 함수 버전 관리를 참조하세요.

aws:rds:cluster

- `writerAvailabilityZoneIdentifiers` – 선택 사항입니다. DB 클러스터 라이터의 가용 영역입니다. 가능한 값은 쉼표로 구분된 가용 영역 식별자 목록, `all`입니다.

aws:rds:db

- `availabilityZoneIdentifiers` – 선택 사항입니다. 영향을 받을 DB 인스턴스의 가용 영역입니다. 가능한 값은 쉼표로 구분된 가용 영역 식별자 목록, `all`입니다.

aws:elasticache:replicationgroup

- `availabilityZoneIdentifier` - 필수입니다. 대상 노드가 포함된 가용성 영역의 코드(예: us-east-1a) 또는 AZ ID(예: use1-az1)입니다.

선택 모드

선택 모드를 지정하여 식별된 리소스의 범위를 지정합니다. AWS FIS는 다음 선택 모드를 지원합니다.

- `ALL` - 모든 대상에 대해 작업을 실행합니다.
- `COUNT(n)` - 식별된 대상에서 무작위로 선택한 지정된 수의 대상에 대해 작업을 실행합니다. 예를 들어, COUNT(1)은 식별된 대상 중 하나를 선택합니다.
- `PERCENT(n)` - 식별된 대상에서 무작위로 선택한 지정된 비율의 대상에 대해 작업을 실행합니다. 예를 들어, PERCENT(25)는 식별된 대상의 25%를 선택합니다.

리소스 수가 홀수이고 50%를 지정하면 AWS FIS가 반올림됩니다. 예를 들어 5개의 Amazon EC2 인스턴스를 대상으로 추가하고 범위를 50%로 설정하면 AWS FIS는 2개의 인스턴스로 내림합니다. 리소스 1개보다 작은 비율을 지정할 수 없습니다. 예를 들어 4개의 Amazon EC2 인스턴스를 추가하고 범위를 5%로 설정하면 AWS FIS는 인스턴스를 선택할 수 없습니다.

동일한 대상 리소스 유형을 사용하여 여러 대상을 정의하는 경우 AWS FIS는 동일한 리소스를 여러 번 선택할 수 있습니다.

어떤 선택 모드를 사용하든, 지정한 범위에 리소스가 없는 것으로 확인되면 실험은 실패합니다.

예시 대상

예를 들면 다음과 같습니다.

예제

- 지정된 태그가 있는 지정된 VPC의 인스턴스

- 지정된 파라미터가 있는 작업

예: 지정된 태그가 있는 지정된 VPC의 인스턴스

이 예제의 가능한 대상은 env=prod 태그가 있는 지정된 VPC의 Amazon EC2 인스턴스입니다. 선택 모드는 AWS FIS가 이러한 대상 중 하나를 무작위로 선택하도록 지정합니다.

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

예: 지정된 파라미터가 있는 작업

이 예제의 가능한 대상은 지정된 클러스터 및 서비스를 사용하는 Amazon ECS 작업입니다. 선택 모드는 AWS FIS가 이러한 대상 중 하나를 무작위로 선택하도록 지정합니다.

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

```
        }
    }
}
```

필터 예제

예를 들면 다음과 같습니다.

예제

- EC2 인스턴스
- DB 클러스터

예: EC2 인스턴스

aws:ec2:instance 리소스 유형을 지원하는 작업에 대한 필터를 지정하면 AWS FIS는 Amazon EC2 describe-instances 명령을 사용하고 필터를 적용하여 대상을 식별합니다.

describe-instances 명령은 각 인스턴스가 `Instances` 아래 구조인 JSON 출력을 반환합니다. 다음은 `####`로 표시된 필드를 포함하는 부분 출력입니다. 이러한 필드를 사용하여 JSON 출력 구조에서 속성 경로를 지정하는 예제를 제공합니다.

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

리소스 필터를 사용하여 특정 가용 영역의 인스턴스를 선택하려면 AvailabilityZone의 속성 경로와 가용 영역의 코드를 값으로 지정하세요. 예제:

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

리소스 필터를 사용하여 특정 서브넷의 인스턴스를 선택하려면 SubnetId의 속성 경로와 서브넷의 ID를 값으로 지정하세요. 예제:

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

특정 인스턴스 상태에 있는 인스턴스를 선택하려면 Name의 속성 경로와 다음 상태 이름 중 하나를 값으로 지정하세요. pending | running | shutting-down | terminated | stopping | stopped. 예제:

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

여러 보안 그룹이 연결된 인스턴스를 선택하려면 GroupId 및 여러 보안 그룹 IDs의 속성 경로가 있는 단일 필터를 지정합니다. 예제:

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

```
            ]
    }
],
```

여러 보안 그룹이 모두 연결된 인스턴스를 선택하려면에 대한 속성 경로GroupId와 각 필터에 대한 단일 보안 그룹 ID를 사용하여 여러 필터를 지정합니다. 예제:

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

예: Amazon RDS 클러스터(DB 클러스터)

aws:rds:cluster 리소스 유형을 지원하는 작업에 대한 필터를 지정하면 AWS FIS는 Amazon RDS describe-db-clusters 명령을 실행하고 필터를 적용하여 대상을 식별합니다.

describe-db-clusters 명령은 각 DB 클러스터에 대해 다음과 유사한 JSON 출력을 반환합니다. 다음은 ####로 표시된 필드를 포함하는 부분 출력입니다. 이러한 필드를 사용하여 JSON 출력 구조에서 속성 경로를 지정하는 예제를 제공합니다.

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

특정 DB 엔진을 사용하는 DB 클러스터만 반환하는 리소스 필터를 적용하려면 다음 예와 같이 속성 경로를 Engine로 지정하고 값을 aurora-postgresql로 지정하세요.

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

특정 가용 영역의 DB 클러스터만 반환하는 리소스 필터를 적용하려면 다음 예와 같이 속성 경로와 값을 지정하세요.

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

AWS FIS에 대한 중지 조건

AWS Fault Injection Service(AWS FIS)는 AWS 워크로드에서 실험을 안전하게 실행할 수 있는 제어 및 가드레일을 제공합니다. 중지 조건은 Amazon CloudWatch 경보로 정의한 임계값에 도달할 경우 실험을 중지하는 메커니즘입니다. 실험 중에 중지 조건이 트리거되면 AWS FFS는 실험을 중지합니다. 중지된 실험은 재개할 수 없습니다.

중지 조건을 만들려면 먼저 애플리케이션 또는 서비스의 안정 상태를 정의하세요. 안정 상태는 애플리케이션이 최적의 성능을 발휘하는 시점으로, 비즈니스 또는 기술 지표로 정의됩니다. 지연 시간, CPU 부하, 재시도 횟수 등을 예로 들 수 있습니다. 안정 상태를 사용하면 애플리케이션 또는 서비스 성능이 허용되지 않는 상태에 도달할 경우 실험을 중지하는 데 사용할 수 있는 CloudWatch 경보를 생성할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon CloudWatch 경보 사용을 참조하세요.

사용자 계정에는 실험 템플릿에 지정할 수 있는 중지 조건 수에 대한 할당량이 있습니다. 자세한 내용은 AWS Fault Injection Service의 할당량 및 제한 사항 단원을 참조하십시오.

중지 조건 구문

실험 템플릿을 생성할 때 생성한 CloudWatch 경보를 지정하여 하나 이상의 중지 조건을 지정합니다.

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

다음 예는 실험 템플릿에 중지 조건이 지정되어 있지 않음을 나타냅니다.

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

자세히 알아보기

CloudWatch 경보를 생성하고 실험 템플릿에 중지 조건을 추가하는 방법을 보여주는 자습서는 인스턴스에서 CPU 스트레스 실행를 참조하세요.

AWS FIS에서 지원하는 리소스 유형에 사용할 수 있는 CloudWatch 지표에 대한 자세한 내용은 다음을 참조하세요.

- CloudWatch를 사용하여 인스턴스 모니터링
- Amazon ECS CloudWatch 지표
- CloudWatch를 사용한 Amazon RDS 지표 모니터링
- CloudWatch를 사용하여 명령 실행 지표 모니터링

AWS FIS 실험을 위한 IAM 역할

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. AWS FIS를 사용하려면 AWS FIS가 사용자를 대신하여 실험을 실행할 수 있도록 필요한 권한을 AWS FIS에 부여하는 IAM 역할을 생성해야 합니다. 실험 템플릿을 생성할 때 이 실험 역할을 지정합니다. 단일 계정 실험의 경우 실험 역할에 대한 IAM 정책은 실험 템플릿에서 대상으로 지정한 리소스를 수정할 수 있는 권한을 부여해야 합니다. 다중 계정 실험의 경우 실험 역할은 오케스트레이터 역할에 각 대상 계정에 대한 IAM 역할을 맡을 수 있는 권한을 부여해야 합니다. 자세한 내용은 다중 계정 실험에 대한 권한 단원을 참조하십시오.

최소 권한 부여 표준 보안 관행을 따르는 것이 좋습니다. 정책에 특정 리소스 ARN 또는 태그를 지정하여 이 작업을 수행할 수 있습니다.

AWS FIS를 빠르게 시작할 수 있도록 실험 역할을 생성할 때 지정할 수 있는 AWS 관리형 정책을 제공합니다. 또는 자체 인라인 정책 문서를 만들 때 이러한 정책을 모델로 사용할 수도 있습니다.

내용

- 사전 조건
- 옵션 1: 실험 역할 생성 및 AWS 관리형 정책 연결
- 옵션 2: 실험 역할 생성 및 인라인 정책 문서 추가

사전 조건

시작하기 전에 AWS CLI 를 설치하고 필요한 신뢰 정책을 생성합니다.

설치 AWS CLI

> 시작하기 전에 AWS CLI을(를) 설치하고 구성합니다. 를 구성할 때 자격 AWS 증명을 입력 AWS CLI하라는 메시지가 표시됩니다. 이 절차의 예제에서는 기본 리전도 구성했다고 가정합니다. 그렇지 않을 경우 각 명령에 `--region` 옵션을 적용합니다. 자세한 내용은 AWS CLI설치 또는 업데이트 및 AWS CLI구성을 참조하세요.

신뢰 관계 정책 만들기

실험 역할에는 AWS FIS 서비스가 역할을 수임할 수 있도록 허용하는 신뢰 관계가 있어야 합니다. `fis-role-trust-policy.json`라는 텍스트 파일을 만들어 다음 신뢰 관계 정책을 추가합니다.

JSON

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

혼동된 대리자 문제로부터 자신을 보호하기 위하여 `aws:SourceAccount` 및 `aws:SourceArn` 조건 키를 사용할 것을 권장합니다. 소스 계정은 실험의 소유자이고 소스 ARN은 실험의 ARN입니다. 예를 들어 신뢰 정책에 다음 조건 블록을 추가해야 합니다.

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

대상 계정 역할을 맡을 수 있는 권한 추가(다중 계정 실험만 해당)

다중 계정 실험의 경우 오케스트레이터 계정이 대상 계정 역할을 맡을 수 있는 권한이 필요합니다. 다음 예제를 수정하여 인라인 정책 문서로 추가하여 대상 계정 역할을 가정할 수 있습니다.

```
{
```

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

옵션 1: 실험 역할 생성 및 AWS 관리형 정책 연결

AWS FIS의 AWS 관리형 정책 중 하나를 사용하여 빠르게 시작할 수 있습니다.

실험 역할을 생성하고 AWS 관리형 정책을 연결하려면

1. 실험의 AWS FIS 작업에 대한 관리형 정책이 있는지 확인합니다. 그렇지 않으면 자체 인라인 정책 문서를 만들어야 합니다. 자세한 내용은 the section called “AWS 관리형 정책” 단원을 참조하십시오.
2. 다음 create-role 명령을 사용하여 역할을 만들고 사전 조건에 따라 만든 신뢰 정책을 추가합니다.

   ```
   aws iam create-role --role-name my-fis-role --assume-role-policy-document
    file://fis-role-trust-policy.json
   ```
3. 다음 attach-role-policy 명령을 사용하여 AWS 관리형 정책을 연결합니다.

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

   여기서 *fis-policy-arn*은 다음 중 하나입니다.

   - 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

옵션 2: 실험 역할 생성 및 인라인 정책 문서 추가

관리형 정책이 없는 작업이나 특정 실험에 필요한 권한만 포함하는 작업에 이 옵션을 사용하세요.

실험을 만들고 인라인 정책 문서를 추가하려면

1. 다음 create-role 명령을 사용하여 역할을 만들고 사전 조건에 따라 만든 신뢰 정책을 추가합니다.

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

2. `fis-role-permissions-policy.json`라는 텍스트 파일을 만들고 권한 정책을 추가합니다. 시작점으로 사용할 수 있는 예제를 보려면 다음을 참조하세요.

   - 오류 주입 작업 - 다음 정책에서 시작합니다.

     JSON

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

   - Amazon EBS 작업 - 다음 정책에서 시작합니다.

     JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:PauseVolumeIO"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*"
        }
    ]
}
```

- Amazon EC2 작업 - AWSFaultInjectionSimulatorEC2Access 정책에서 시작합니다.
- Amazon ECS 작업 - AWSFaultInjectionSimulatorECSAccess 정책에서 시작합니다.
- Amazon EKS 작업 – AWSFaultInjectionSimulatorEKSAccess 정책에서 시작합니다.
- 네트워크 작업 – AWSFaultInjectionSimulatorNetworkAccess 정책에서 시작합니다.
- Amazon RDS 작업 – AWSFaultInjectionSimulatorRDSAccess 정책에서 시작합니다.
- Systems Manager 작업 – AWSFaultInjectionSimulatorSSMAccess 정책에서 시작합니다.

3. 다음 put-role-policy 명령을 사용하여 이전 단계에서 생성한 권한 정책을 추가합니다.

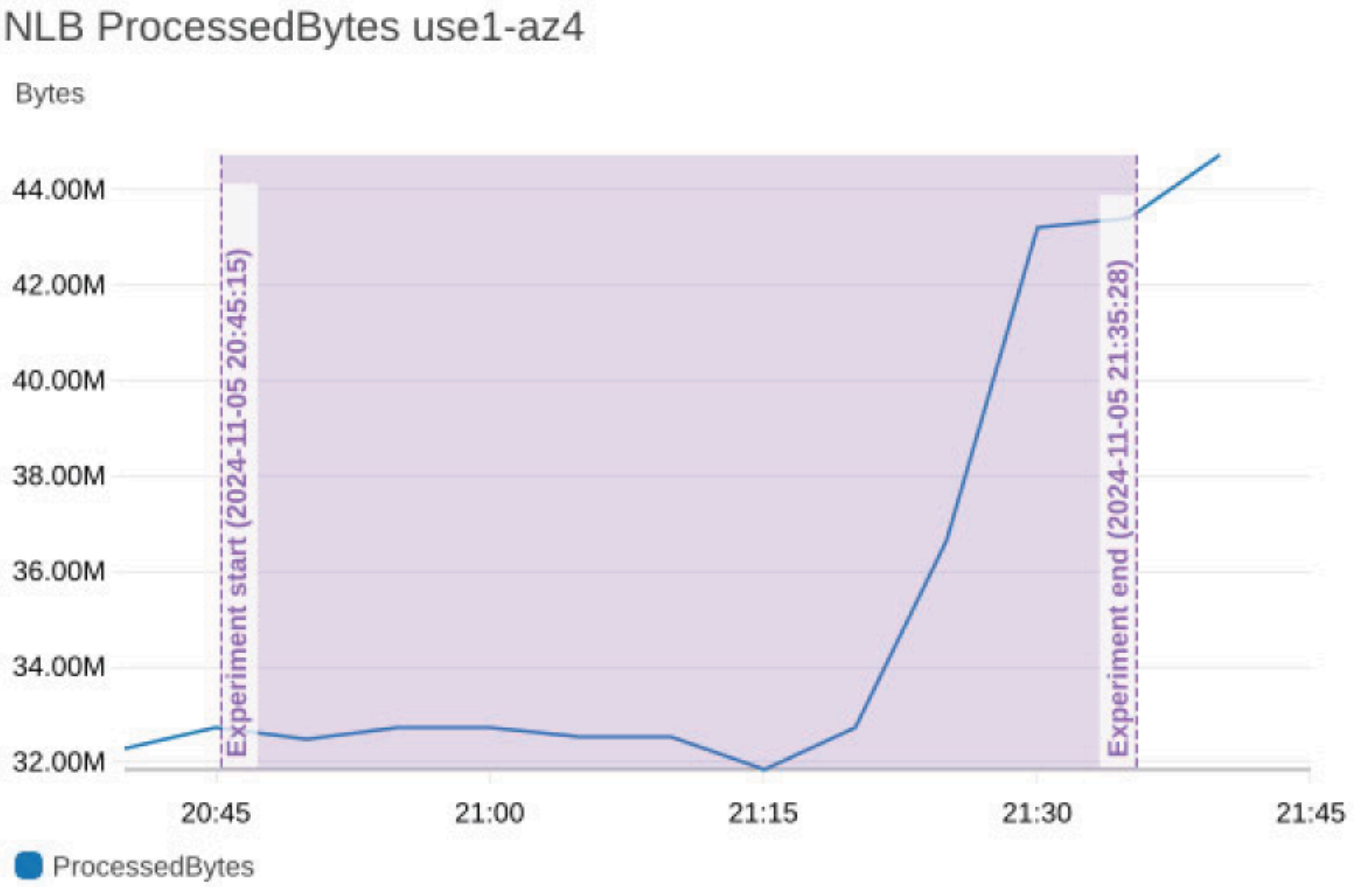
```
aws iam put-role-policy --role-name my-fis-role --policy-name my-fis-policy --
policy-document file://fis-role-permissions-policy.json
```

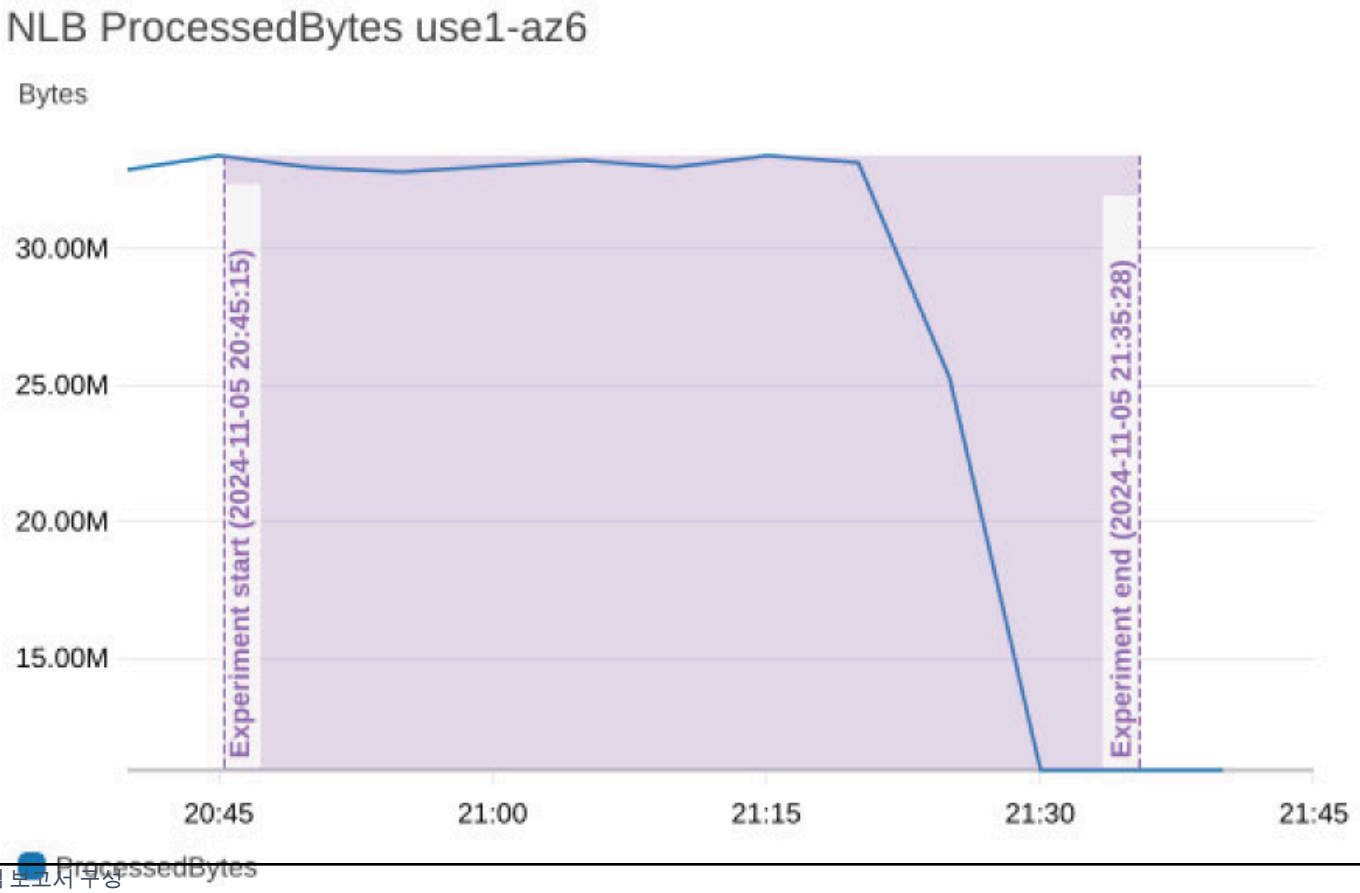
AWS FIS에 대한 실험 보고서 구성

AWS Fault Injection Service(FIS)를 활성화하여 실험에 대한 보고서를 생성할 수 있으므로 복원력 테스트의 증거를 더 쉽게 생성할 수 있습니다. 실험 보고서는 실험 작업을 요약하고 선택적으로 지정한 CloudWatch 대시보드에서 애플리케이션 응답을 캡처하는 PDF 문서입니다. 실험 보고서 예제를 보려면 여기에서 zip 파일을 다운로드하세요.

실험에 대해 생성된 보고서의 내용을 활성화하고 구성하려면 실험 템플릿에 대한 실험 보고서 구성을 정의합니다. CloudWatch 대시보드를 지정하면 아래 예와 같이 지정된 기간 동안 실험 시작 및 종료 시간에 주석이 달린 지정된 대시보드의 모든 위젯에 대한 스냅샷 그래프가 AWS 포함됩니다.

이 예제는 가용 영역(AZ)에서 패킷 손실 실험이 미치는 영향을 보여줍니다. AZ use1-az6에서 패킷 손실이 발생하면 트래픽이 use1-az6에서 use1-az4로 전환되므로 해당 AZ에서 로드 밸런서가 처리하는 바이트 수가 감소합니다.


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

실험이 종료되면 AWS FIS 콘솔에서 보고서를 다운로드할 수 있으며 Amazon S3 버킷에도 저장됩니다. 보고서 구성에 CloudWatch 대시보드를 포함하면 각 위젯의 이미지도 전송됩니다. 대상 미리 보기의 일부로 실행cancelled되거나 인 실험에는 보고서가 생성되지 않습니다(actionsMode가 로 설정된 경우skip-all). 실험이 실험 데이터 보존 한도를 초과하면 보고서는 Amazon S3 버킷에서만 사용할 수 있습니다. 내부 오류로 실패한 보고서를 제외하고 전송된 각 보고서에는 AWS FIS 요금이 적용됩니다. 자세한 내용은 AWS Fault Injection Service 요금 및 섹션을 참조하세요AWS Fault Injection Service의 할당량 및 제한 사항. GetMetricWidgetImage 및 GetDashboard 요청에 대한 Amazon S3 및 CloudWatch API 요금의 수집 및 스토리지 요금이 적용될 수 있습니다. CloudWatch GetMetricWidgetImage 자세한 내용은 Amazon S3 요금 및 CloudWatch 요금을 참조하세요.

내용

실험 보고서 구성 구문

다음은 실험 템플릿의 선택적 섹션인 실험 보고서 구성의 구문입니다.

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

를 사용하면 실험 보고서에 포함할 데이터의 출력 대상, 입력 데이터 및 기간을 사용자 지정할 experimentReportConfiguration수 있으므로 AWS FIS 실험의 영향과 결과를 더 잘 이해하는 데 도움이 될 수 있습니다. 실험 보고서 구성을 정의할 때 다음을 제공합니다.

출력

실험 보고서가 전달될 위치를 experimentReportConfiguration 지정하는 섹션입니다. 에서 다음을 s3Configuration 제공하여를 outputs지정합니다.

- bucketName - 보고서가 저장될 Amazon S3 버킷의 이름입니다. 버킷은 실험과 동일한 리전에 있어야 합니다.
- prefix (선택 사항) - 보고서가 저장될 Amazon S3 버킷 내의 접두사입니다. 접두사에 대한 액세스만 제한할 수 있도록이 필드를 사용하는 것이 좋습니다.

dataSources

실험 보고서에 포함될 추가 데이터 소스를 experimentReportConfiguration 지정하는의 선택적 섹션입니다.

- cloudWatchDashboards - 보고서에 포함될 CloudWatch 대시보드의 배열입니다. CloudWatch 대시보드 1개로 제한됩니다.
- dashboardIdentifier- CloudWatch 대시보드의 ARN입니다. 교차 리전 지표를 제외하고이 대시보드metric에 유형이 있는 모든 위젯의 스냅샷 그래프가 보고서에 포함됩니다.

preExperimentDuration

보고서에 포함할 CloudWatch 대시보드 지표의 실험 전 기간을 experimentReportConfiguration 정의하는의 선택적 섹션으로, 최대 30분입니다. 이 기간은 애플리케이션 안정 상태를 나타내는 기간이어야 합니다. 예를 들어 실험 전 기간이 5분이면 스냅샷 그래프에 실험이 시작되기 5분 전에 지표가 포함됩니다. 지속 시간의 형식은 ISO 8601이고 기본값은 20분입니다.

postExperimentDuration

보고서에 포함할 CloudWatch 대시보드 지표의 실험 후 기간을 experimentReportConfiguration 정의하는의 선택적 섹션으로, 최대 2시간입니다. 애플리케이션 안정 상태 또는 복구 기간을 나타내는 기간이어야 합니다. 예를 들어 실험 후 기간을 5분으로 지정하면 실험 종료 후 5분까지 스냅샷 그래프에 지표가 포함됩니다. 지속 시간의 형식은 ISO 8601이고 기본값은 20분입니다.

실험 보고서 권한

AWS FIS가 실험 보고서를 생성하고 저장할 수 있도록 하려면 AWS FIS 실험 IAM 역할에서 다음 작업을 허용해야 합니다.

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

AWS 보안 모범 사례를 따르고 실험 역할을 버킷 및 접두사로 제한하는 것이 좋습니다. 다음은 실험 역할 액세스를 제한하는 정책 설명의 예입니다.

JSON

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

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

고객 관리형 키(CMK)로 암호화된 Amazon S3 버킷에 전달된 보고서에 대한 추가 권한

에서 지정하는 Amazon S3 버킷`S3Configuration`이 CMK로 암호화된 경우 KMS 키 정책의 FIS 실험 역할에 다음과 같은 추가 권한을 부여해야 합니다.

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

다음은 FIS 실험 역할이 암호화된 버킷에 보고서를 작성할 수 있도록 허용하는 KMS 키 정책 설명의 예입니다.

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

실험 보고서 모범 사례

다음은 AWS FIS 실험 보고서 구성을 사용하는 모범 사례입니다.

- 실험을 시작하기 전에 대상 미리 보기를 생성하여 실험 템플릿이 예상대로 구성되어 있는지 확인합니다. 대상 미리 보기는 실험의 예상 대상에 대한 정보를 제공합니다. 자세한 내용은 실험 템플릿에서 대상 미리 보기 생성를 참조하세요.

- 실패한 실험 문제를 해결하는 데 보고서를 사용해서는 안 됩니다. 대신 실험 로그를 사용하여 실험 오류를 해결합니다. 이전에 실행하고 성공적으로 완료한 실험에만 보고서에 의존하는 것이 좋습니다.
- 실험 IAM 역할 입력을 제한하고 S3 대상 버킷 및 접두사에 대한 객체 액세스 권한을 얻습니다. 버킷/접두사는 AWS FIS 실험 보고서에만 사용하고 다른 AWS 서비스에이 버킷 및 접두사에 대한 액세스 권한을 부여하지 않는 것이 좋습니다.
- Amazon S3 객체 잠금을 사용하여 고정된 시간 동안 또는 무기한으로 보고서가 삭제되거나 덮어쓰이지 않도록 합니다. 자세한 내용은 객체 잠금을 사용하여 객체 잠금을 참조하세요.
- CloudWatch 대시보드가 동일한 리전 내의 별도 계정에 있는 경우 CloudWatch 교차 계정 관찰성을 사용하여 AWS FIS 오케스트레이터 계정을 모니터링 계정으로 활성화하고, AWS CLI 및 API의 CloudWatch 콘솔 또는 Observability Access Manager 명령에서 별도의 계정을 소스 계정으로 활성화할 수 있습니다. 자세한 내용은 CloudWatch 교차 계정 관찰성을 참조하세요.

에 대한 실험 옵션 AWS FIS

실험 옵션은 실험에 대한 선택적 설정입니다. 실험 템플릿에서 특정 실험 옵션을 정의할 수 있습니다. 실험을 시작할 때 추가 실험 옵션이 설정됩니다.

다음은 실험 템플릿에 정의하는 실험 옵션의 구문입니다.

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

실험 템플릿을 만들 때 실험 옵션을 지정하지 않으면 각 옵션의 기본값이 사용됩니다.

다음은 실험을 시작할 때 설정하는 실험 옵션의 구문입니다.

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

실험을 시작할 때 실험 옵션을 지정하지 않으면 기본값 `run-all`이 사용됩니다.

내용

계정 타겟팅

실험에서 대상으로 지정할 리소스가 있는 AWS 계정이 여러 개인 경우 계정 대상 지정 실험 옵션을 사용하여 다중 계정 실험을 정의할 수 있습니다. 여러 대상 계정의 리소스에 영향을 주는 오케스트레이터 계정에서 다중 계정 실험을 실행합니다. 오케스트레이터 계정은 AWS FIS 실험 템플릿과 실험을 소유합니다. 대상 계정은 AWS FIS 실험의 영향을 받을 수 있는 리소스가 있는 개별 AWS 계정입니다. 자세한 내용은 에 대한 다중 계정 실험 작업 AWS FIS 단원을 참조하십시오.

계정 타겟팅을 사용하여 대상 리소스의 위치를 표시합니다. 계정 타겟팅에 두 가지 값을 제공할 수 있습니다:

- 단일 계정 - 기본값입니다. 실험은 AWS FIS 실험이 실행되는 AWS 계정의 리소스만 대상으로 합니다.
- 다중 계정 - 실험이 여러 AWS 계정의 리소스를 대상으로 할 수 있습니다.

대상 계정 구성

다중 계정 실험을 실행하려면 하나 이상의 대상 계정 구성을 정의해야 합니다. 대상 계정 구성은 실험의 대상이 되는 리소스가 있는 각 계정에 대한 accountId, roleArn 및 설명을 지정합니다. 실험 템플릿에 대한 대상 계정 구성의 계정 ID는 고유해야 합니다.

다중 계정 실험 템플릿을 만들면 실험 템플릿은 실험 템플릿에 대한 모든 대상 계정 구성의 개수인 읽기 전용 필드인 `targetAccountConfigurationsCount`를 반환합니다.

다음은 대상 계정 구성에 대한 구문입니다.

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

대상 계정 구성을 만들 때 다음을 입력합니다.

accountId

대상 계정의 12자리 AWS 계정 ID입니다.

roleArn

대상 계정에서 작업을 수행할 수 있는 AWS FIS 권한을 부여하는 IAM 역할입니다.

description

설명(선택 사항)입니다.

대상 계정 구성으로 작업하는 방법에 대해 자세히 알아보려면 에 대한 다중 계정 실험 작업 AWS FIS 단원을 참조하세요.

빈 대상 확인 모드

이 모드에서는 대상 리소스가 확인되지 않은 경우에도 실험을 완료할 수 있는 옵션을 제공합니다.

- fail - 기본값입니다. 대상에 확인된 리소스가 없는 경우 실험이 `failed` 상태로 즉시 종료됩니다.
- skip – 대상에 확인된 리소스가 없는 경우 실험이 계속되고 확인된 대상이 없는 모든 작업이 건너뛰어집니다. ARN과 같은 고유 식별자를 사용하여 정의된 대상이 있는 작업은 건너뛸 수 없습니다. 고유 식별자를 사용하여 정의된 대상을 찾을 수 없는 경우 실험은 `failed` 상태로 즉시 종료됩니다.

작업 모드

작업 모드는 실험을 시작할 때 지정할 수 있는 선택적 파라미터입니다. 작업 모드를 `skip-all`로 설정하여 대상 리소스에 결함을 주입하기 전에 대상 미리 보기를 생성할 수 있습니다. 대상 미리 보기를 사용하면 다음을 확인할 수 있습니다.

- 예상 리소스를 대상으로 실험 템플릿을 구성했는지 여부. 이 실험을 시작할 때 대상으로 지정되는 실제 리소스는 미리 보기의 리소스와 다를 수 있습니다. 리소스가 임의로 제거, 업데이트 또는 샘플링될 수 있기 때문입니다.
- 로깅 구성이 올바르게 설정되었는지 여부.
- 다중 계정 실험의 경우 각 대상 계정 구성에 대해 IAM 역할을 올바르게 설정했는지 여부.

> **Note**
>
> `skip-all` 모드에서는 AWS FIS 실험을 실행하고 리소스에 대해 작업을 수행하는 데 필요한 권한이 있는지 확인할 수 없습니다.

작업 모드 파라미터는 다음 값 중 하나를 받습니다.

- `run-all` - (기본값) 실험이 대상 리소스에 대한 작업을 수행합니다.
- `skip-all` - 실험이 대상 리소스에 대한 모든 작업을 건너뜁니다.

실험을 시작할 때 작업 모드 파라미터를 설정하는 방법에 대한 자세한 내용은 실험 템플릿에서 대상 미리 보기 생성 섹션을 참조하세요.

AWS FIS 작업 참조

작업은 AWS Fault Injection Service ()를 사용하여 대상에서 실행하는 결함 주입 활동입니다AWS FIS. 는 AWS 서비스 전반에 걸쳐 특정 유형의 대상에 대해 미리 구성된 작업을 AWS FIS 제공합니다. 실험 템플릿에 작업을 추가한 다음, 실험을 실행하는 데 사용합니다.

이 참조에서는 작업 파라미터 및 필요한 IAM 권한에 대한 정보를 AWS FIS포함하여의 일반적인 작업에 대해 설명합니다. AWS FIS 콘솔 또는 AWS Command Line Interface ()의 list-actions 명령을 사용하여 지원되는 AWS FIS 작업을 나열할 수도 있습니다AWS CLI. 특정 작업의 이름을 알고 나면 get-action 명령을 사용하여 작업에 대한 세부 정보를 볼 수 있습니다. 에서 AWS FIS 명령을 사용하는 방법에 대한 자세한 AWS CLI내용은 명령 참조의 AWS Command Line Interface 사용 설명서 및 fis를 AWS CLI 참조하세요.

AWS FIS 작업 작동 방식에 대한 자세한 내용은 AWS FIS에 대한 작업 및 단원을 참조하십시오AWS Fault Injection Service가 IAM과 작동하는 방식.

작업

- 오류 주입 작업
- 복구 작업
- 대기 작업
- Amazon CloudWatch 작업
- Amazon DynamoDB 작업
- Amazon Aurora DSQL 작업
- Amazon EBS 작업
- Amazon EC2 작업
- Amazon ECS 작업
- Amazon EKS 작업
- Amazon ElastiCache 작업
- Amazon Kinesis Data Streams 작업
- AWS Lambda 작업
- Amazon MemoryDB 작업
- 네트워크 작업

- Amazon RDS 작업
- Amazon S3 작업
- Systems Manager 작업
- AWS Direct Connect 작업
- AWS FIS에서 Systems Manager SSM 문서 사용
- AWS FIS aws:ecs:task 작업 사용
- AWS FIS aws:eks:pod 작업 사용
- AWS FIS aws:lambda:function 작업 사용

오류 주입 작업

AWS FIS 는 다음과 같은 오류 주입 작업을 지원합니다.

작업

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

aws:fis:inject-api-internal-error

대상 IAM 역할이 만든 요청에 내부 오류를 삽입합니다. 구체적인 응답은 각 서비스 및 API에 따라 달라집니다. 자세한 내용은 해당 서비스의 SDK 및 API 설명서를 참조하세요.

리소스 유형

- aws:iam:role

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- service - 대상 AWS API 네임스페이스입니다. 지원되는 값은 `ec2` 및 입니다`kinesis`.
- percentage - 오류를 주입하기 위한 호출의 백분율(1~100).

- operations - 오류를 주입하는 작업으로, 쉼표로 구분합니다. ec2 네임스페이스에 대한 API 작업 목록은 Amazon EC2 API 참조 및 Amazon Kinesis Data Streams API 참조를 참조하세요.

권한

- fis:InjectApiInternalError

aws:fis:inject-api-throttle-error

대상 IAM 역할이 만든 요청에 제한 오류를 삽입합니다. 구체적인 응답은 각 서비스 및 API에 따라 달라집니다. 자세한 내용은 해당 서비스의 SDK 및 API 설명서를 참조하세요.

리소스 유형

- aws:iam:role

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- service - 대상 AWS API 네임스페이스입니다. 지원되는 값은 ec2 및 입니다kinesis.
- percentage - 오류를 주입하기 위한 호출의 백분율(1~100).
- operations - 오류를 주입하는 작업으로, 쉼표로 구분합니다. ec2 네임스페이스에 대한 API 작업 목록은 Amazon EC2 API 참조 및 Amazon Kinesis Data Streams API 참조를 참조하세요.

권한

- fis:InjectApiThrottleError

aws:fis:inject-api-unavailable-error

대상 IAM 역할이 만든 요청에 사용할 수 없음 오류를 삽입합니다. 구체적인 응답은 각 서비스 및 API에 따라 달라집니다. 자세한 내용은 해당 서비스의 SDK 및 API 설명서를 참조하세요.

리소스 유형

- aws:iam:role

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- service - 대상 AWS API 네임스페이스입니다. 지원되는 값은 `ec2` 및 입니다`kinesis`.
- percentage - 오류를 주입하기 위한 호출의 백분율(1~100).
- operations - 오류를 주입하는 작업으로, 쉼표로 구분합니다. `ec2` 네임스페이스에 대한 API 작업 목록은 Amazon EC2 API 참조 및 Amazon Kinesis Data Streams API 참조를 참조하세요.

권한

- `fis:InjectApiUnavailableError`

복구 작업

복구 작업은 장애 발생 후 위험을 완화하거나 애플리케이션을 보호하기 위해 수행됩니다.

AWS FIS 는 다음과 같은 복구 작업을 지원합니다.

aws:arc:start-zonal-autoshift

지원되는 리소스의 트래픽을 잠재적으로 손상된 가용 영역(AZ)에서 자동으로 이동하고 동일한 AWS 리전의 정상 AZs로 다시 라우팅합니다. 이렇게 하면 FIS를 통해 영역 자동 전환이 발생할 수 있습니다. 영역 자동 전환은 Amazon Application Recovery Controller(ARC)의 기능으로,가 AZ의 고객에게 잠재적으로 영향을 미칠 수 있는 장애가 있다고 AWS 판단할 때 사용자를 대신하여 리소스의 트래픽을 AZ에서 다른 곳으로 AWS 이동할 수 있습니다.

`aws:arc:start-zonal-autoshift` 작업을 실행하면는 이러한 요청에 대한 `expiresIn` 필드가 1분으로 설정된 StartZonalShift, UpdateZonalShift 및 CancelZonalShift APIs를 안전 메커니즘으로 사용하여 영역 전환을 AWS FIS 관리합니다. 이를 통해 AWS FIS 는 네트워크 중단 또는 시스템 문제와 같은 예상치 못한 이벤트가 발생할 경우 영역 전환을 신속하게 롤백할 수 있습니다. ARC 콘솔에서 만료 시간 필드는 AWS FIS관리형으로 표시되며 실제 예상 만료는 영역 전환 작업에 지정된 기간에 따라 결정됩니다.

리소스 유형

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

영역 전환 관리형 리소스는 Amazon EKS 클러스터, Amazon EC2 Application and Network Load Balancer, ARC 영역 자동 전환을 활성화할 수 있는 Amazon EC2 Auto Scaling 그룹을 포함한 리소스 유형입니다. 자세한 내용은 ARC 개발자 안내서의 지원되는 리소스 및 영역 자동 전환 리소스 활성화를 참조하세요.

파라미터

- duration - 트래픽이 이동되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- availabilityZoneIdentifier - 트래픽이이 AZ에서 멀어집니다. AZ 이름(us-east-1a) 또는 AZ ID(use1-az1)일 수 있습니다.
- managedResourceTypes - 트래픽이 이동할 리소스 유형으로, 쉼표로 구분됩니다. 가능한 옵션은 `ASG` (Auto Scaling 그룹), `ALB` (Application Load Balancer), `NLB` (Network Load Balancer) 및 `EKS` (Amazon EKS)입니다.
- zonalAutoshiftStatus - 대상으로 지정할 리소스의 `zonalAutoshiftStatus` 상태입니다. 가능한 옵션은 `ENABLED` `DISABLED`, 및 입니다`ANY`. 기본값은 `ENABLED`입니다.

권한

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

대기 작업

AWS FIS 는 다음 대기 작업을 지원합니다.

aws:fis:wait

AWS FIS 대기 작업을 실행합니다.

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

- 없음

Amazon CloudWatch 작업

AWS FIS 는 다음과 같은 Amazon CloudWatch 작업을 지원합니다.

aws:cloudwatch:assert-alarm-state

지정된 경보가 지정된 경보 상태 중 하나에 해당하는지 확인합니다.

리소스 유형

- 없음

파라미터

- alarmArns - 경보의 ARN으로, 쉼표로 구분됩니다. 최대 5개의 경보를 지정할 수 있습니다.
- alarmStates - 경보 상태로, 쉼표로 구분됩니다. 가능한 경보 상태는 `OK`, `ALARM` 및 `INSUFFICIENT_DATA`입니다.

권한

- `cloudwatch:DescribeAlarms`

Amazon DynamoDB 작업

AWS FIS 는 다음 Amazon DynamoDB 작업을 지원합니다.

aws:dynamodb:global-table-pause-replication

Amazon DynamoDB 다중 리전 전역 테이블 복제를 모든 복제본 테이블로 일시 중지합니다. 작업이 시작된 후 최대 5분 동안 테이블이 계속 복제될 수 있습니다.

다중 리전 강력히 일관된(MRSC) 글로벌 테이블

다음 문은 대상 DynamoDB MRSC 글로벌 테이블의 정책에 동적으로 추가됩니다.

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

```
         "Condition": {
            "DateLessThan": {
              "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
            },
            "ArnEquals": {
                "aws:PrincipalArn": "arn:aws:iam::123456789012:role/
aws-service-role/dynamodb.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable"
            }
         }
      }
   ]
}
```

대상 테이블에 연결된 리소스 정책이 없는 경우 실험 기간 동안 리소스 정책이 생성되고 실험이 종료되면 자동으로 삭제됩니다. 정책이 있는 경우 기존 정책 문에 대한 추가 수정 없이 결함 문이 기존 정책에 삽입됩니다. 그런 다음 실험이 끝날 때 정책에서 결함 문이 제거됩니다.

대상 Amazon DynamoDB MRSC 글로벌 테이블에는 추가 할당량이 적용됩니다. 이 할당량은 7일 롤링 윈도우에서 단일 테이블에 5,040분 이상의 장애가 발생할 수 없도록 강제합니다.

다중 리전 최종 일관성(MREC) 글로벌 테이블

다음 문은 대상 DynamoDB MREC 글로벌 테이블의 정책에 동적으로 추가됩니다.

```
{
   "Statement":[
      {
         "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxx",
         "Effect":"Deny",
         "Principal":{
            "AWS": "*"
         },
         "Action":[
            "dynamodb:GetItem",
            "dynamodb:PutItem",
            "dynamodb:UpdateItem",
            "dynamodb:DeleteItem",
            "dynamodb:DescribeTable",
            "dynamodb:UpdateTable",
            "dynamodb:Scan",
            "dynamodb:DescribeTimeToLive",
            "dynamodb:UpdateTimeToLive"
```

```
         ],
         "Resource":"arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable",
         "Condition": {
            "DateLessThan": {
              "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
            },
            "ArnEquals": {
                "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
            }
         }
      }
   ]
}
```

다음 문은 대상 DynamoDB MREC 글로벌 테이블의 스트림 정책에 동적으로 추가됩니다.

```
{
   "Statement":[
      {
         "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxx",
         "Effect":"Deny",
         "Principal":{
            "AWS": "*"
         },
         "Action":[
            "dynamodb:GetRecords",
            "dynamodb:DescribeStream",
            "dynamodb:GetShardIterator"
         ],
         "Resource":"arn:aws:dynamodb:us-east-1:123456789012:table/ExampleGlobalTable/
stream/2023-08-31T09:50:24.025",
         "Condition": {
            "DateLessThan": {
              "aws:CurrentTime": "2024-04-10T09:51:41.511Z"
            },
            "ArnEquals": {
                "aws:PrincipalArn": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
            }
         }
      }
   ]
```

```
}
```

대상 테이블 또는 스트림에 연결된 리소스 정책이 없는 경우 실험 기간 동안 리소스 정책이 생성되고 실험이 종료되면 자동으로 삭제됩니다. 정책이 있는 경우 기존 정책 문에 대한 추가 수정 없이 결함 문이 기존 정책에 삽입됩니다. 그런 다음 실험이 끝날 때 정책에서 결함 문이 제거됩니다.

리소스 유형

- aws:dynamodb:global-table

파라미터

- duration - AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

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

* 권한은 MRSC 글로벌 테이블을 대상으로 하는 경우에만 필요합니다.

Amazon Aurora DSQL 작업

AWS FIS 는 다음과 같은 Amazon Aurora DSQL 작업을 지원합니다.

aws:dsql:cluster-connection-failure

지정된 기간 동안 Aurora DSQL 클러스터에서 제어된 연결 실패를 생성하여 애플리케이션 복원력을 테스트합니다.

리소스 유형

- aws:dsql:cluster

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- percentage - 오류를 주입하기 위한 호출의 백분율(1~100).

권한

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

Aurora DSQL로 실험을 시작하려면 Aurora DSQL 사용 설명서의 결함 주입 테스트를 참조하세요.

Amazon EBS 작업

AWS FIS 는 다음 Amazon EBS 작업을 지원합니다.

작업

- aws:ebs:pause-volume-io
- aws:ebs:volume-io-latency

aws:ebs:pause-volume-io

대상 EBS 볼륨의 I/O 작업을 일시 중지합니다. 대상 볼륨은 동일한 가용 영역에 있어야 하며 Nitro System 기반 인스턴스에 연결되어 있어야 합니다. Outpost의 인스턴스에는 볼륨을 연결할 수 없습니다.

Amazon EC2 콘솔을 사용하여 실험을 시작하려면 Amazon EC2 사용 설명서의 Amazon EBS에서의 오류 테스트를 참조하세요.

리소스 유형

- aws:ec2:ebs-volume

파라미터

- duration - 소요 시간은 1초에서 12시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어 PT1M은 1분, PT5S는 5초, PT6H는 6시간을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다. 지속 시간이 짧으면(예: PT5S) 지정된 기간 동안 I/O가 일시 중지되지만 실험을 초기화하는 데 걸리는 시간 때문에 실험이 완료되는 데 시간이 더 오래 걸릴 수 있습니다.

권한

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

aws:ebs:volume-io-latency

대상 EBS 볼륨의 I/O 작업에 지연 시간을 주입합니다. 대상 볼륨은 동일한 가용 영역에 있어야 하며 Nitro 기반 인스턴스에 연결되어야 합니다. 볼륨은 Outpost의 인스턴스에 연결할 수 없습니다.

Amazon EC2 콘솔을 사용하여 실험을 시작하려면 Amazon EBS 사용 설명서의 Amazon EBS에서 결함 테스트를 참조하세요.

리소스 유형

- aws:ec2:ebs-volume

파라미터

- readIOPercentage - 지연 시간이 주입되는 읽기 I/O 작업의 백분율로, 0.1%~100%입니다. 이는 실험 중에 영향을 받는 볼륨에 대한 모든 읽기 I/O 작업의 백분율입니다. 기본값은 100입니다.
- readIOLatencyMilliseconds - 1ms(io2 볼륨) 또는 10ms(비io2 볼륨)에서 60초까지 밀리초 단위로 읽기 I/O 작업에 주입되는 지연 시간입니다. 이는 실험 중에 읽기 I/O의 지정된 백분율에서 관찰될 지연 시간 값입니다. 기본값은 100입니다.
- writeIOPercentage - 지연 시간이 주입되는 쓰기 I/O 작업의 백분율로, 0.1%에서 100.% 사이입니다. 이는 실험 중에 영향을 받는 볼륨에 대한 모든 쓰기 I/O 작업의 백분율입니다. 기본값은 100입니다.
- writeIOLatencyMilliseconds - 쓰기 I/O 작업에 밀리초 단위로 주입되는 지연 시간으로, 1ms(io2 볼륨) 또는 10ms(비io2 볼륨)에서 60초입니다. 이는 실험 중에 읽기 I/O의 특정 백분율에서 관찰되는 지연 시간 값입니다. 기본값은 100입니다.

- duration - 지연 시간이 1초에서 12시간까지 주입되는 기간입니다.

권한

- `ec2:DescribeVolumes`
- `ec2:InjectVolumeIOLatency`
- `tag:GetResources`

Amazon EC2 작업

AWS FIS 는 다음과 같은 Amazon EC2 작업을 지원합니다.

작업

- 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 는 AWS Systems Manager SSM 에이전트를 통한 오류 주입 작업도 지원합니다. Systems Manager는 EC2 인스턴스에서 실행하는 작업을 정의하는 SSM 문서를 사용합니다. 자체 문서를 사용하여 사용자 지정 오류를 주입하거나 사전 구성된 SSM 문서를 사용할 수 있습니다. 자세한 내용은 the section called “SSM 문서 작업” 단원을 참조하십시오.

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

대상 IAM 역할의 요청에 대해 `InsufficientInstanceCapacity` 오류 응답을 삽입합니다. 지원되는 작업은 RunInstances, CreateCapacityReservation, StartInstances, CreateFleet 호출입니다. 여러 가용성 영역에서 용량 요청을 포함하는 요청은 지원되지 않습니다. 이 작업은 리소스 태그, 필터 또는 파라미터를 사용하여 대상을 정의하는 것을 지원하지 않습니다.

Auto Scaling LaunchInstances 작업의 경우 응답의 `errors` 필드에 InsufficientInstanceCapacity 오류가 반환되지만 Auto Scaling 그룹의 원하는 용량은 계속 업데이트되므로 비동기 조정 프로세스가 잠재적으로 인스턴스를 시작할 수 있습니다. LaunchInstances를 사용하여 용량 부족 처리를 광범위하게

테스트하려면와 함께이 작업을 사용하는 것이 좋습니다the section called “aws:ec2:asg-insufficient-instance-capacity-error”.

리소스 유형

- aws:iam:role

파라미터

- duration - AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- availabilityZoneIdentifiers - 쉼표로 구분된 사용 가능 영역 목록. 영역 ID(예: `"use1-az1, use1-az2"`) 및 영역 이름(예: `"us-east-1a"`)을 지원합니다.
- percentage - 오류를 주입하기 위한 호출의 백분율(1~100).

권한

- 조건 키 `ec2:FisActionId` 값이 `aws:ec2:api-insufficient-instance-capacity-error`로 설정된 `ec2:InjectApiError` 조건 키와 타깃 IAM 역할로 설정된 `ec2:FisTargetArns` 조건 키가 있습니다.

정책 예제는 예: `ec2:InjectApiError`의 조건 키 사용을 참조하세요.

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

대상 Auto Scaling 그룹의 요청에 대해 `InsufficientInstanceCapacity` 오류 응답을 삽입합니다. 이 작업은 시작 템플릿을 사용하는 Auto Scaling 그룹만 지원합니다. 인스턴스 용량 부족 오류에 대해 자세히 알아보려면 Amazon EC2 사용 설명서를 참조하세요.

리소스 유형

- aws:ec2:autoscaling-group

파라미터

- duration - AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

- availabilityZoneIdentifiers - 쉼표로 구분된 사용 가능 영역 목록. 영역 ID(예: "use1-az1, use1-az2") 및 영역 이름(예: "us-east-1a")을 지원합니다.
- percentage – 선택 사항입니다. 대상 Auto Scaling 그룹의 시작 요청 중 오류를 주입할 비율(1~100)입니다. 기본값은 100입니다.

권한

- 조건 키 ec2:FisActionId 값이 aws:ec2:asg-insufficient-instance-capacity-error로 설정된 ec2:InjectApiError와 조건 키가 Auto Scaling 그룹을 대상으로 설정된 ec2:FisTargetArns입니다.
- autoscaling:DescribeAutoScalingGroups

정책 예제는 예: ec2:InjectApiError의 조건 키 사용을 참조하세요.

aws:ec2:reboot-instances

대상 EC2 인스턴스에서 Amazon EC2 API 작업 RebootInstances를 실행합니다.

리소스 유형

- aws:ec2:instance

파라미터

- 없음

권한

- ec2:RebootInstances
- ec2:DescribeInstances

AWS 관리형 정책

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:send-spot-instance-interruptions

대상 스팟 인스턴스를 중단합니다. 대상 스팟 인스턴스를 중단하기 2분 전에 스팟 인스턴스 중단 공지를 대상 스팟 인스턴스에 보냅니다. 중단 시간은 지정된 durationBeforeInterruption 파라미터에 의해 결정됩니다. 중단 시점으로부터 2분 후에 스팟 인스턴스는 중단 동작에 따라 종료되거나 중지됩니다. AWS FIS 에 의해 중지된 스팟 인스턴스는 다시 시작할 때까지 중지된 상태로 유지됩니다.

작업이 시작된 직후 대상 인스턴스는 EC2 인스턴스 재조정 권장 사항을 수신합니다. durationBeforeInterruption을 지정한 경우 재조정 권장 사항과 중단 공지 사이에 지연이 있을 수 있습니다.

자세한 내용은 the section called “스팟 인스턴스 중단 테스트” 단원을 참조하십시오. 또는 Amazon EC2 콘솔을 사용하여 실험을 시작하려면 Amazon EC2 사용 설명서의 스팟 인스턴스 중단 시작을 참조하세요.

리소스 유형

- aws:ec2:spot-instance

파라미터

- durationBeforeInterruption - 인스턴스 중단까지 걸리는 대기 시간(2~15분). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT2M은 2분을 나타냅니다. AWS FIS 콘솔에서 분 수를 입력합니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:stop-instances

대상 EC2 인스턴스에서 Amazon EC2 API 작업 StopInstances를 실행합니다.

리소스 유형

- aws:ec2:instance

파라미터

- startInstancesAfterDuration – 선택 사항입니다. 인스턴스를 시작하기 전에 대기하는 데 걸리는 시간은 1분에서 12시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 단위 숫자를 입력합니다. 인스턴스에 암호화된 EBS 볼륨이 있는 경우 볼륨을 암호화하는 데 사용되는 KMS 키에 AWS FIS 권한을 부여하거나 KMS 키 정책에 실험 역할을 추가해야 합니다.
- completeIfInstancesTerminated – 선택 사항입니다. true인 경우 그리고 `startInstancesAfterDuration`도 true인 경우, 대상 EC2 인스턴스가 FIS 외부의 별도 요청에 의해 종료되어 다시 시작할 수 없는 경우에도 이 작업은 실패하지 않습니다. 예를 들어, Auto Scaling 그룹은 이 작업이 완료되기 전에 해당 그룹이 제어하는 중지된 EC2 인스턴스를 종료할 수 있습니다. 기본값은 false입니다.

권한

- `ec2:StopInstances`
- `ec2:StartInstances`
- `ec2:DescribeInstances` – 선택 사항입니다. 작업 종료 시 인스턴스 상태의 유효성을 검사하기 위해 completeIfInstancesTerminated와 함께 필요합니다.
- `kms:CreateGrant` – 선택 사항입니다. 암호화된 볼륨으로 인스턴스를 다시 시작하려면 startInstancesAfterDuration에 필요합니다.

AWS 관리형 정책

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:terminate-instances

대상 EC2 인스턴스에서 Amazon EC2 API 작업 TerminateInstances를 실행합니다.

리소스 유형

- aws:ec2:instance

파라미터

- 없음

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEC2Access

Amazon ECS 작업

AWS FIS 는 다음과 같은 Amazon ECS 작업을 지원합니다.

작업

- 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

Amazon ECS API 작업 UpdateContainerInstancesState를 실행하여 대상 클러스터에 있는 기본 Amazon EC2 인스턴스의 지정된 비율을 소모합니다.

리소스 유형

- aws:ecs:cluster

파라미터

- drainagePercentage - 백분율(1~100).
- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:stop-task

Amazon ECS API 작업 StopTask를 실행하여 대상 작업을 중지합니다.

리소스 유형

- aws:ecs:task

파라미터

- 없음

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:task-cpu-stress

대상 작업에 CPU 스트레스를 실행합니다. AWSFIS-Run-CPU-Stress SSM 문서를 사용합니다. 작업은에서 관리해야 합니다 AWS Systems Manager. 자세한 내용은 ECS 태스크 작업 단원을 참조하십시오.

리소스 유형

- aws:ecs:task

파라미터

- duration - ISO 8601 형식의 스트레스 테스트 기간입니다.
- percent – 선택 사항입니다. 0(무부하)에서 100(완전 부하)까지의 목표 부하 백분율. 기본값은 100입니다.
- workers – 선택 사항입니다. 사용할 스트레스 요인의 수입니다. 기본값은 0이며, 모든 스트레스 요인을 사용합니다.
- installDependencies – 선택 사항입니다. 이 값이 `True`이면 Systems Manager가 SSM 에이전트의 사이드카 컨테이너에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 stress-ng입니다.

권한

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

aws:ecs:task-io-stress

대상 작업에 I/O 스트레스를 실행합니다. AWSFIS-Run-IO-Stress SSM 문서를 사용합니다. 작업은에서 관리해야 합니다 AWS Systems Manager. 자세한 내용은 ECS 태스크 작업 단원을 참조하십시오.

리소스 유형

- aws:ecs:task

파라미터

- duration - ISO 8601 형식의 스트레스 테스트 기간입니다.
- percent – 선택 사항입니다. 스트레스 테스트 중에 사용할 파일 시스템의 여유 공간 비율입니다. 기본값은 80%입니다.
- workers – 선택 사항입니다. 작업자 수. 작업자는 순차, 임의 및 메모리 매핑된 읽기/쓰기 작업, 강제 동기화 및 캐시 삭제를 혼합하여 수행합니다. 여러 하위 프로세스가 동일한 파일에서 다양한 I/O 작업을 수행합니다. 기본값은 1입니다.
- installDependencies – 선택 사항입니다. 이 값이 `True`이면 Systems Manager가 SSM 에이전트의 사이드카 컨테이너에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 stress-ng입니다.

권한

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

aws:ecs:task-kill-process

killall 명령을 사용하여 작업에서 지정된 프로세스를 중지합니다. AWSFIS-Run-Kill-Process SSM 문서를 사용합니다. 작업 정의의 `pidMode`가 `task`로 설정되어야 합니다. 작업은에서 관리해야 합니다 AWS Systems Manager. 자세한 내용은 ECS 태스크 작업 단원을 참조하십시오.

리소스 유형

- aws:ecs:task

파라미터

- processName - 중지할 프로세스의 이름.

- signal – 선택 사항입니다. 명령과 함께 전송할 신호입니다. 가능한 값은 SIGTERM(수신자가 무시하도록 선택할 수 있는 값)과 SIGKILL(무시할 수 없는 값)입니다. 기본값은 SIGTERM입니다.
- installDependencies – 선택 사항입니다. 이 값이 True이면 Systems Manager가 SSM 에이전트의 사이드카 컨테이너에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 True입니다. 종속성은 killall입니다.

권한

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

aws:ecs:task-network-blackhole-port

Amazon ECS Fault Injection 엔드포인트를 사용하여 지정된 프로토콜 및 포트에 대한 인바운드 또는 아웃바운드 트래픽을 삭제합니다. AWSFIS-Run-Network-Blackhole-Port-ECS SSM 문서를 사용합니다. 작업 정의의 pidMode가 task로 설정되어야 합니다. 작업은에서 관리해야 합니다 AWS Systems Manager. 작업 정의에서 networkMode를 bridge로 설정할 수 없습니다. 자세한 내용은 ECS 태스크 작업 단원을 참조하십시오.

useEcsFaultInjectionEndpoints를 로 설정하면 false결함이 iptables 도구를 사용하고 AWSFIS-Run-Network-Blackhole-Port SSM 문서를 사용합니다.

리소스 유형

- aws:ecs:task

파라미터

- duration - ISO 8601 형식의 테스트 기간입니다.
- port - 포트 번호입니다.
- trafficType - 트래픽 유형입니다. 가능한 값은 ingress와 egress입니다.
- protocol – 선택 사항입니다. 프로토콜. 가능한 값은 tcp와 udp입니다. 기본값은 tcp입니다.

- installDependencies – 선택 사항입니다. 이 값이 `True`이면 Systems Manager가 SSM 에이전트의 사이드카 컨테이너에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 atd, curl-minimal dig 및 입니다jq.
- useEcsFaultInjectionEndpoints – 선택 사항입니다. true로 설정하면 Amazon ECS Fault Injection APIs 사용됩니다. 기본값은 false입니다.

권한

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

aws:ecs:task-network-latency

Amazon ECS Fault Injection 엔드포인트를 사용하여 특정 소스로 송신 트래픽에 대한 지연 시간과 지터를 네트워크 인터페이스에 추가합니다. AWSFIS-Run-Network-Latency-ECS SSM 문서를 사용합니다. 작업 정의의 `pidMode`가 `task`로 설정되어야 합니다. 작업은에서 관리해야 합니다 AWS Systems Manager. 작업 정의에서 `networkMode`를 `bridge`로 설정할 수 없습니다. 자세한 내용은 ECS 태스크 작업 단원을 참조하십시오.

이 로 `useEcsFaultInjectionEndpoints` 설정되면 장애`false`는 `tc` 도구를 사용하고 AWSFIS-Run-Network-Latency-Sources SSM 문서를 사용합니다.

`flowsPercent` 파라미터를 사용하여 연결의 백분율에 지연 시간을 추가합니다. `flowsPercent` 파라미터를 사용하려면 ECS 에이전트 버전이 `1.100.0` 이상이어야 합니다.

`sources` 파라미터에서 AZ 이름 또는 AZ IDs를 사용하려면 작업의 모든 대상이 동일한 VPC에 있어야 합니다.

리소스 유형

- aws:ecs:task

파라미터

- duration - ISO 8601 형식의 테스트 기간입니다.

- delayMilliseconds – 선택 사항입니다. 밀리 초 단위의 지연 시간입니다. 기본값은 200입니다.
- jitterMilliseconds – 선택 사항입니다. 밀리 초 단위의 지터입니다. 기본값은 10입니다.
- flowsPercent – 선택 사항입니다. 작업의 영향을 받는 네트워크 흐름의 백분율입니다. 의 기본값은 100%입니다.
- sources – 선택 사항입니다. 공백 없이 쉼표로 구분된 소스입니다. 가능한 값은 IPv4 주소, IPv4 CIDR 블록, 도메인 이름, AZ 이름(us-east-1a), AZ ID(use1-az1), ALL, 및 DYNAMODB입니다S3. DYNAMODB 또는 S3를 지정하는 경우 이는 현재 리전의 리전 엔드포인트에만 적용됩니다. 기본값은 모든 IPv4 트래픽과 일치하는 ALL입니다.
- installDependencies – 선택 사항입니다. 이 값이 True이면 Systems Manager가 SSM 에이전트의 사이드카 컨테이너에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 True입니다. 종속성은 atd, curl-minimaldig, jq 및 입니다lsof.
- useEcsFaultInjectionEndpoints – 선택 사항입니다. true로 설정하면 Amazon ECS Fault Injection APIs 사용됩니다. 기본값은 false입니다.

권한

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

aws:ecs:task-network-packet-loss

Amazon ECS Fault Injection 엔드포인트를 사용하여 특정 소스로 송신 트래픽에 대한 패킷 손실을 네트워크 인터페이스에 추가합니다. AWSFIS-Run-Network-Packet-Loss-ECS SSM 문서를 사용합니다. 작업 정의의 pidMode가 task로 설정되어야 합니다. 작업은에서 관리해야 합니다 AWS Systems Manager. 작업 정의에서 networkMode를 bridge로 설정할 수 없습니다. 자세한 내용은 ECS 태스크 작업 단원을 참조하십시오.

이 로 useEcsFaultInjectionEndpoints 설정되면 장애false는 tc 도구를 사용하고 AWSFIS-Run-Network-Packet-Loss-Sources SSM 문서를 사용합니다.

flowsPercent 파라미터를 사용하여 연결의 백분율에 패킷 손실을 주입합니다. flowsPercent 파라미터를 사용하려면 ECS 에이전트 버전이 1.100.0 이상이어야 합니다.

sources 파라미터에서 AZ 이름 또는 AZ IDs를 사용하려면 작업의 모든 대상이 동일한 VPC에 있어야 합니다.

리소스 유형

- aws:ecs:task

파라미터

- duration - ISO 8601 형식의 테스트 기간입니다.
- lossPercent – 선택 사항입니다. 패킷 손실의 백분율. 기본값은 7%입니다.
- flowsPercent – 선택 사항입니다. 작업의 영향을 받는 네트워크 흐름의 백분율입니다. 의 기본값은 100%입니다.
- sources – 선택 사항입니다. 공백 없이 쉼표로 구분된 소스입니다. 가능한 값은 IPv4 주소, IPv4 CIDR 블록, 도메인 이름, AZ 이름(us-east-1a), AZ ID(use1-az1), ALL, 및 DYNAMODB입니다S3. DYNAMODB 또는 S3를 지정하는 경우 이는 현재 리전의 리전 엔드포인트에만 적용됩니다. 기본값은 모든 IPv4 트래픽과 일치하는 ALL입니다.
- installDependencies – 선택 사항입니다. 이 값이 True이면 Systems Manager가 SSM 에이전트의 사이드카 컨테이너에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 True입니다. 종속성은 atd, curl-minimaldig, jq 및 입니다lsof.
- useEcsFaultInjectionEndpoints – 선택 사항입니다. true로 설정하면 Amazon ECS Fault Injection APIs 사용됩니다. 기본값은 false입니다.

권한

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

Amazon EKS 작업

AWS FIS 는 다음과 같은 Amazon EKS 작업을 지원합니다.

작업

- 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

단일 대상 클러스터에서 ChaosMesh 또는 Litmus 실험을 실행합니다. 대상 클러스터에 ChaosMesh 또는 Litmus를 설치해야 합니다.

실험 템플릿을 생성하고 `aws:eks:cluster` 대상 유형을 정의할 때는 이 작업을 단일 Amazon 리소스 이름(ARN)으로 타겟팅해야 합니다. 이 작업은 리소스 태그, 필터 또는 파라미터를 사용하여 대상을 정의하는 것을 지원하지 않습니다.

ChaosMesh를 설치할 때 적절한 컨테이너 런타임을 지정해야 합니다. Amazon EKS 버전 1.23부터 기본 런타임이 Docker에서 containerd로 변경되었습니다. 버전 1.24부터 Docker가 제거되었습니다.

리소스 유형

- aws:eks:cluster

파라미터

- kubernetesApiVersion – Kubernetes 사용자 지정 리소스의 API 버저전. 가능한 값: `chaos-mesh.org/v1alpha1` | `litmuschaos.io/v1alpha1`.

- kubernetesKind - Kubernetes 사용자 지정 리소스 종류. 값은 API 버전에 따라 달라집니다.
  - `chaos-mesh.org/v1alpha1` – 가능한 값: `AWSChaos` | `DNSChaos` | `GCPChaos` | `HTTPChaos` | `IOChaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodIOChaos` | `PodNetworkChaos` | `Schedule` | `StressChaos` | `TimeChaos` |
  - `litmuschaos.io/v1alpha1` – 가능한 값: `ChaosEngine`.
- kubernetesNamespace – Kubernetes 네임스페이스.
- kubernetesSpec – JSON 형식으로 된 Kubernetes 사용자 지정 리소스의 `spec` 섹션.
- maxDuration - 자동화 실행이 완료되는 데 허용되는 최대 시간은 1분에서 12시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

이 작업에는 AWS Identity and Access Management(IAM) 권한이 필요하지 않습니다. 이 작업을 사용하는 데 필요한 권한은 RBAC 인증을 사용하여 Kubernetes에서 제어합니다. 자세한 내용은 공식 Kubernetes 문서의 RBAC 승인 사용을 참조하세요. Chaos Mesh에 대한 자세한 내용은 공식 Chaos Mesh 문서를 참조하세요. Litmus에 대한 자세한 내용은 공식 Litmus 문서를 참조하세요.

aws:eks:pod-cpu-stress

대상 포드에 CPU 스트레스를 실행합니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

리소스 유형

- aws:eks:pod

파라미터

- duration - ISO 8601 형식의 스트레스 테스트 기간입니다.
- percent – 선택 사항입니다. 0(무부하)에서 100(완전 부하)까지의 목표 부하 백분율. 기본값은 100입니다.
- workers – 선택 사항입니다. 사용할 스트레스 요인의 수입니다. 기본값은 0이며, 모든 스트레스 요인을 사용합니다.
- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.

- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.
- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.
- fisPodSecurityPolicy – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드와 임시 컨테이너에 사용할 Kubernetes 보안 표준 정책입니다. 가능한 값은 `privileged`, `baseline`, `restricted`입니다. 이 작업은 모든 정책 수준과 호환됩니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-delete

대상 포드를 삭제합니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

리소스 유형

- aws:eks:pod

파라미터

- gracePeriodSeconds – 선택 사항입니다. 포드가 정상적으로 종료될 때까지 기다리는 시간(초)입니다. 값이 0이면 작업을 즉시 수행합니다. 값이 nil인 경우 포드의 기본 유예 기간을 사용합니다.

- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.
- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.
- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.
- fisPodSecurityPolicy – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드와 임시 컨테이너에 사용할 Kubernetes 보안 표준 정책입니다. 가능한 값은 `privileged`, `baseline`, `restricted`입니다. 이 작업은 모든 정책 수준과 호환됩니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-io-stress

대상 포드에서 I/O 스트레스를 실행합니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

리소스 유형

- aws:eks:pod

파라미터

- duration - ISO 8601 형식의 스트레스 테스트 기간입니다.

- workers – 선택 사항입니다. 작업자 수. 작업자는 순차, 임의 및 메모리 매핑된 읽기/쓰기 작업, 강제 동기화 및 캐시 삭제를 혼합하여 수행합니다. 여러 하위 프로세스가 동일한 파일에서 다양한 I/O 작업을 수행합니다. 기본값은 1입니다.
- percent – 선택 사항입니다. 스트레스 테스트 중에 사용할 파일 시스템의 여유 공간 비율입니다. 기본값은 80%입니다.
- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.
- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.
- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.
- fisPodSecurityPolicy – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드와 임시 컨테이너에 사용할 Kubernetes 보안 표준 정책입니다. 가능한 값은 `privileged`, `baseline`, `restricted`입니다. 이 작업은 모든 정책 수준과 호환됩니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-memory-stress

대상 포드에 메모리 스트레스를 실행합니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

리소스 유형

- aws:eks:pod

파라미터

- duration - ISO 8601 형식의 스트레스 테스트 기간입니다.
- workers – 선택 사항입니다. 사용할 스트레스 요인의 수입니다. 기본값은 1입니다.
- percent – 선택 사항입니다. 스트레스 테스트 중에 사용할 가상 메모리의 비율입니다. 기본값은 80% 입니다.
- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.
- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.
- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.
- fisPodSecurityPolicy – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드와 임시 컨테이너에 사용할 Kubernetes 보안 표준 정책입니다. 가능한 값은 `privileged`, `baseline`, `restricted`입니다. 이 작업은 모든 정책 수준과 호환됩니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-blackhole-port

지정된 프로토콜 및 포트에 대한 인바운드 또는 아웃바운드 트래픽을 삭제합니다. Kubernetes 보안 표준 `privileged`정책과만 호환됩니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

리소스 유형

- aws:eks:pod

파라미터

- duration - ISO 8601 형식의 테스트 기간입니다.
- protocol – 프로토콜입니다. 가능한 값은 `tcp`와 `udp`입니다.
- trafficType - 트래픽 유형입니다. 가능한 값은 `ingress`와 `egress`입니다.
- port - 포트 번호입니다.
- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.
- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.
- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-latency

특정 소스에서 오가는 트래픽을 처리하는 tc 도구를 사용하여 네트워크 인터페이스에 지연 시간과 지터를 추가합니다. Kubernetes 보안 표준 `privileged`정책과만 호환됩니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

`flowsPercent` 파라미터를 사용하여 연결의 백분율에 지연 시간을 추가합니다.

리소스 유형

- aws:eks:pod

파라미터

- duration - ISO 8601 형식의 테스트 기간입니다.
- interface – 선택 사항입니다. 쉼표로 구분된 네트워크 인터페이스입니다. ALL 및 DEFAULT 값이 지원됩니다. 기본값은 운영 체제의 기본 네트워크 인터페이스를 대상으로 `DEFAULT`하는 입니다.
- delayMilliseconds – 선택 사항입니다. 밀리 초 단위의 지연 시간입니다. 기본값은 200입니다.
- jitterMilliseconds – 선택 사항입니다. 밀리 초 단위의 지터입니다. 기본값은 10입니다.
- flowsPercent – 선택 사항입니다. 작업의 영향을 받는 네트워크 흐름의 백분율입니다. 의 기본값은 100%입니다.
- sources – 선택 사항입니다. 공백 없이 쉼표로 구분된 소스입니다. 가능한 값은 IPv4 주소, IPv4 CIDR 블록, 도메인 이름, AZ 이름(us-east-1a), AZ ID(use1-az1), ALL, 및 `DYNAMODB`입니다`S3`. `DYNAMODB` 또는 `S3`를 지정하는 경우 이는 현재 리전의 리전 엔드포인트에만 적용됩니다. 도메인 이름의 경우 IP 주소를 수집하기 위해 10회의 DNS 확인 시도가 이루어집니다. DNS 로드 밸런싱 및 교체로 인해이 작업은 도메인이 해결할 수 있는 가능한 모든 IP 주소를 손상시키지 않을 수 있습니다. 기본값은 모든 IPv4 트래픽과 일치하는 ALL입니다.
- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.
- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.

- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-packet-loss

tc 도구를 사용하여 네트워크 인터페이스에 패킷 손실을 추가합니다. Kubernetes 보안 표준 `privileged`정책과만 호환됩니다. 자세한 내용은 EKS 포드 작업 단원을 참조하십시오.

`flowsPercent` 파라미터를 사용하여 연결의 백분율에 패킷 손실을 주입합니다.

리소스 유형

- aws:eks:pod

파라미터

- duration - ISO 8601 형식의 테스트 기간입니다.
- interface – 선택 사항입니다. 쉼표로 구분된 네트워크 인터페이스입니다. ALL 및 DEFAULT 값이 지원됩니다. 기본값은 운영 체제의 기본 네트워크 인터페이스를 대상으로 `DEFAULT`하는 입니다.
- lossPercent – 선택 사항입니다. 패킷 손실의 백분율. 기본값은 7%입니다.
- flowsPercent – 선택 사항입니다. 작업의 영향을 받는 네트워크 흐름의 백분율입니다. 의 기본값은 100%입니다.

- sources – 선택 사항입니다. 공백 없이 쉼표로 구분된 소스입니다. 가능한 값은 IPv4 주소, IPv4 CIDR 블록, 도메인 이름, AZ 이름(us-east-1a), AZ ID(use1-az1), ALL, 및 DYNAMODB입니다S3. DYNAMODB 또는 S3를 지정하는 경우 이는 현재 리전의 리전 엔드포인트에만 적용됩니다. 도메인 이름의 경우 IP 주소를 수집하기 위해 10회의 DNS 확인 시도가 이루어집니다. DNS 로드 밸런싱 및 교체로 인해이 작업은 도메인이 해결할 수 있는 가능한 모든 IP 주소를 손상시키지 않을 수 있습니다. 기본값은 모든 IPv4 트래픽과 일치하는 ALL입니다.
- kubernetesServiceAccount - Kubernetes 서비스 계정. 필요한 권한에 대한 자세한 내용은 the section called “Kubernetes 서비스 계정 구성” 단원을 참조하세요.
- fisPodContainerImage – 선택 사항입니다. 오류 인젝터 포드를 만드는 데 사용된 컨테이너 이미지. 기본값은에서 제공하는 이미지를 사용하는 것입니다 AWS FIS. 자세한 내용은 the section called “포드 컨테이너 이미지” 단원을 참조하십시오.
- maxErrorsPercent – 선택 사항입니다. 오류 주입이 실패하기 전에 실패할 수 있는 대상의 비율입니다. 기본값은 0입니다.
- fisPodLabels – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 레이블입니다.
- fisPodAnnotations – 선택 사항입니다. FIS에서 생성한 결함 오케스트레이션 포드에 연결된 Kubernetes 주석입니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:terminate-nodegroup-instances

대상 노드 그룹에서 Amazon EC2 API 작업 TerminateInstances를 실행합니다. Amazon EKS 관리형 노드 그룹과만 호환됩니다. 자체 관리형 노드 그룹은 지원되지 않습니다. 자세한 내용은 EKS 컴퓨팅 관리를 참조하세요.

리소스 유형

- aws:eks:nodegroup

파라미터

- instanceTerminationPercentage - 종료할 인스턴스의 백분율(1~100).

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEKSAccess

Amazon ElastiCache 작업

AWS FIS 는 다음 ElastiCache 작업을 지원합니다.

aws:elasticache:replicationgroup-interrupt-az-power

다중 AZ가 활성화된 대상 ElastiCache 복제 그룹에 대해 지정된 가용 영역의 노드에 대한 전원을 중단합니다. 복제 그룹당 한 번에 하나의 가용 영역만 영향을 받을 수 있습니다. 기본 노드가 타깃이 되면 복제 지연이 가장 적은 해당 읽기 복제본이 프라이머리 노드로 승격됩니다. 지정된 가용성 영역의 읽기 복제본 교체는 이 작업이 진행되는 동안 차단되므로 대상 복제 그룹이 줄어든 용량으로 작동합니다. 이 작업의 대상은 Redis 엔진과 Valkey 엔진을 모두 지원합니다. 작업은 "서버리스" 배포 옵션을 지원하지 않습니다.

리소스 유형

- aws:elasticache:replicationgroup

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

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

> **Note**
>
> ElastiCache 인터럽트 AZ 전원 작업은 이제 Valkey 및 Redis를 포함한 모든 복제 그룹 유형을 지원합니다. 이 기능을 더 잘 나타내기 위해 작업의 이름이 변경되었습니다. 현재를 사용 중인 경우 최신 기능을 활용`aws:elasticache:replicationgroup-interrupt-az-power`하려면 새 작업으로 마이그레이션하는 `aws:elasticache:interrupt-cluster-az-power`것이 좋습니다.

Amazon Kinesis Data Streams 작업

Amazon Kinesis Data Streams는 다음과 같은 Kinesis 작업을 지원합니다.

작업

- aws:kinesis:stream-provisioned-throughput-exception
- aws:kinesis:stream-expired-iterator-exception

aws:kinesis:stream-provisioned-throughput-exception

대상 Kinesis Data Streams에 대한 요청에 `ProvisionedThroughputExceededException` 오류 응답을 삽입합니다. 지원되는 작업에는 `GetRecords`, `GetShardIterator`, 및 `PutRecord`가 포함됩니다`PutRecords`.

리소스 유형

- aws:kinesis:stream

파라미터

- 기간 - 1분에서 12시간까지의 기간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- 백분율 - 오류를 주입할 호출의 백분율(1~100)입니다.

권한

- `kinesis:InjectApiError`

aws:kinesis:stream-expired-iterator-exception

지정된 Kinesis Data Streams를 대상으로 하는 `GetRecords` 호출에 대한 `ExpiredIteratorException` 오류 응답을 삽입합니다.

리소스 유형

- aws:kinesis:stream

파라미터

- 기간 - 1분에서 12시간까지의 기간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- 백분율 - 오류를 주입할 호출의 백분율(1~100)입니다.

권한

- `kinesis:InjectApiError`

AWS Lambda 작업

AWS Lambda 는 다음 Lambda 작업을 지원합니다.

작업

- aws:lambda:invocation-add-delay
- aws:lambda:invocation-error

- aws:lambda:invocation-http-integration-response

aws:lambda:invocation-add-delay

지정한 수 밀리초 동안 함수 시작을 지연합니다. 이 작업의 효과는 Lambda 콜드 스타트와 비슷하지만 추가 시간은 청구된 기간의 일부로 소요되며 새 실행 환경에만 영향을 주지 않고 모든 실행 환경에 적용됩니다. 즉, Lambda 콜드 스타트와이 지연이 모두 발생할 수 있습니다. 지연 시간 값을 Lambda 함수에 구성된 제한 시간보다 높게 설정하면이 작업은 충실도가 높은 제한 시간 이벤트에 대한 액세스도 제공합니다.

리소스 유형

- aws:lambda:function

파라미터

- duration - 작업이 지속되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- invocationPercentage – 선택 사항입니다. 오류를 주입할 함수 호출의 백분율(1~100)입니다. 기본값은 100입니다.
- startupDelayMilliseconds – 선택 사항입니다. 함수 코드 호출과 실행 사이에 밀리초(0~900,000) 단위로 대기하는 시간입니다. 기본값은 1000입니다.

권한

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

aws:lambda:invocation-error

Lambda 함수 호출을 실패로 표시합니다. 이 작업은 경보 및 재시도 구성과 같은 오류 처리 메커니즘을 테스트하는 데 유용합니다. 이 작업을 사용하는 동안 오류를 반환하기 전에 함수 코드를 실행할지 여부를 선택합니다.

리소스 유형

- aws:lambda:function

파라미터

- duration - 작업이 지속되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- invocationPercentage – 선택 사항입니다. 오류를 주입할 함수 호출의 백분율(1~100)입니다. 기본값은 100입니다.
- preventExecution - 값이 true인 경우 작업은 함수를 실행하지 않고 오류를 반환합니다.

권한

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

aws:lambda:invocation-http-integration-response

함수의 동작을 수정합니다. 콘텐츠 유형 및 HTTP 응답 코드를 선택하여 ALB, API-GW 및 VPC Lattice와의 통합을 지원합니다. 업스트림 또는 다운스트림 통합에 선택적으로 영향을 미치려면 수정된 응답을 직접 반환할지 또는 함수가 실행을 완료한 후 함수를 실행하고 응답을 교체할지 여부를 선택할 수 있습니다.

리소스 유형

- aws:lambda:function

파라미터

- contentTypeHeader - Lambda 함수에서 반환할 HTTP 콘텐츠 유형 헤더의 문자열 값입니다.
- duration - 작업이 지속되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

- invocationPercentage – 선택 사항입니다. 오류를 주입할 함수 호출의 백분율(1~100)입니다. 기본값은 100입니다.
- preventExecution - 값이 true인 경우 작업은 함수를 실행하지 않고 응답을 반환합니다.
- statusCode - Lambda 함수에서 반환할 HTTP 상태 코드(000~999)의 값입니다.

권한

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

Amazon MemoryDB 작업

AWS FIS 는 다음 MemoryDB 작업을 지원합니다.

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

한 리전 클러스터와 다중 리전 클러스터 내의 다른 모든 리전 클러스터 간의 복제를 일시 중지합니다. 대상 리전 클러스터는 FIS 실험이 실행 중인 리전의 클러스터입니다. 복제가 일시 중지된 동안에는 다중 리전 클러스터를 업데이트할 수 없습니다. 작업이 완료되면 다중 리전 클러스터가 사용 가능한 상태로 돌아가는 데 몇 분 정도 걸릴 수 있습니다. Amazon MemoryDB 다중 리전에 대한 자세한 내용은 Amazon MemoryDB 다중 리전 개발자 안내서를 참조하세요. 리전 가용성은 MemoryDB 다중 리전 사전 조건 및 제한 사항을 참조하세요.

리소스 유형

- aws:memorydb:multi-region-cluster

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

- memorydb:DescribeMultiRegionClusters
- memorydb:PauseMultiRegionClusterReplication
- tag:GetResources

네트워크 작업

AWS FIS 는 다음과 같은 네트워크 작업을 지원합니다.

작업

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

aws:network:disrupt-connectivity

대상 서브넷과 연결된 원래 네트워크 액세스 제어 목록(네트워크 ACL)을 일시적으로 복제하여 대상 서브넷에 대한 지정된 트래픽을 거부합니다. FIS는 태그 managedbyFIS=true가 있는 복제된 네트워크 ACL에 거부 규칙을 추가하고 작업 기간 동안 서브넷과 연결합니다. 작업 완료 시 FIS는 복제된 네트워크 ACL을 삭제하고 원래 네트워크 ACL 연결을 복원합니다.

리소스 유형

- aws:ec2:subnet

파라미터

- scope - 거부할 트래픽의 유형입니다. 범위가 all이 아닌 경우 네트워크 ACL의 최대 항목 수는 20개입니다. 가능한 값은 다음과 같습니다.
  - all - 서브넷으로 들어오고 나가는 모든 트래픽을 거부합니다. 이 옵션은 서브넷의 네트워크 인터페이스로 들어오고 나가는 트래픽을 포함하여 서브넷 내 트래픽을 허용한다는 점에 유의하세요.
  - availability-zone - 다른 가용 영역에 있는 서브넷으로 들어오고 나가는 VPC 내부 트래픽을 거부합니다. VPC에서 대상으로 지정할 수 있는 최대 서브넷 수는 30개입니다.

  - dynamodb - 현재 리전의 DynamoDB 리전 엔드포인트로 들어오고 나가는 트래픽을 거부합니다.
  - prefix-list - 지정된 접두사 목록으로 들어오고 나가는 트래픽을 거부합니다.
  - s3 - 현재 리전의 Amazon S3 리전 엔드포인트로 들어오고 나가는 트래픽을 거부합니다.
  - s3express - 대상 서브넷의 AZ에서 Amazon S3 Express One Zone의 영역 엔드포인트와 주고받는 트래픽을 거부합니다. 대상 서브넷은 현재 S3 Express One Zone을 사용할 수 있는 AZs에 있어야 합니다. 자세한 내용은 S3 Express One Zone 가용 영역 및 리전을 참조하세요.
  - vpc - VPC로 들어오고 나가는 트래픽을 거부합니다.
- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- prefixListIdentifier - 범위가 prefix-list인 경우 고객이 관리하는 접두사 목록의 식별자입니다. 이름, ID 또는 ARN을 지정할 수 있습니다. 접두사 목록에는 최대 10개의 항목이 있을 수 있습니다.

권한

- ec2:CreateNetworkAcl - managedByFIS=true 태그를 사용하여 네트워크 ACL을 생성합니다.
- ec2:CreateNetworkAclEntry - 네트워크 ACL에 managedByFIS=true 태그가 있어야 합니다.
- ec2:CreateTags
- ec2:DeleteNetworkAcl - 네트워크 ACL에 managedByFIS=true 태그가 있어야 합니다.
- ec2:DescribeManagedPrefixLists
- ec2:DescribeNetworkAcls
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- ec2:GetManagedPrefixListEntries
- ec2:ReplaceNetworkAclAssociation

AWS 관리형 정책

- AWSFaultInjectionSimulatorNetworkAccess

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

대상 서브넷에서 시작하여 지정된 리전으로 향하는 트래픽을 차단합니다. 격리할 리전의 모든 경로를 포함하는 라우팅 테이블을 생성합니다. FIS가 이러한 라우팅 테이블을 생성할 routes per route

table 수 있도록 하려면에 대한 Amazon VPC 할당량을 250(또는 region 파라미터가 us-east-1인 경우 350)에 기존 라우팅 테이블의 경로 수를 더한 값으로 늘립니다.

리소스 유형

- aws:ec2:subnet

파라미터

- region - 격리할 리전의 코드(예: eu-west-1)입니다.
- duration - 작업이 지속되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

- 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

† managedByFIS=true 태그를 사용하여 범위를 지정합니다. 실험 중에이 tag. AWS FIS adds 및 removes 태그를 관리할 필요가 없습니다.

AWS 관리형 정책

- AWSFaultInjectionSimulatorNetworkAccess

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

지정된 리전으로 향하는 대상 전송 게이트웨이 피어링 첨부 파일의 트래픽을 차단합니다.

리소스 유형

- aws:ec2:transit-gateway

파라미터

- region - 격리할 리전의 코드(예: eu-west-1)입니다.
- duration - 작업이 지속되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorNetworkAccess

aws:network:disrupt-vpc-endpoint

대상 인터페이스 VPC 엔드포인트의 인바운드 및 아웃바운드 트래픽을 차단합니다. FIS는 빈 규칙으로 관리형 보안 그룹을 생성하고 대상 VPC 엔드포인트의 보안 그룹을이 관리형 보안 그룹으로 임시 대체합니다. 작업 실행 중에 대상 리소스를 수정하면 작업이 실패하고 리소스가 실험 전 상태로 복원되지 않습니다. 또한 작업 실행 중에 FIS 관리형 보안 그룹이 수정되면 FIS에서 삭제하지 않습니다.

리소스 유형

- aws:ec2:vpc-endpoint

파라미터

- `duration` - 작업이 지속되는 시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyVpcEndpoint`
- `ec2:CreateSecurityGroup`
- `ec2:DeleteSecurityGroup`
- `ec2:RevokeSecurityGroupEgress`
- `ec2:CreateTags`
- `vpce:AllowMultiRegion` *

* 권한은 교차 리전 VPC 엔드포인트를 대상으로 하는 경우에만 필요합니다.

Amazon RDS 작업

AWS FIS 는 다음과 같은 Amazon RDS 작업을 지원합니다.

작업

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

aws:rds:failover-db-cluster

대상 Aurora DB 클러스터에서 Amazon RDS API 작업 FailoverDBCluster를 실행합니다. RDS 클러스터 및 DocumentDB 클러스터가 지원됩니다.

리소스 유형

- aws:rds:cluster

파라미터

- 없음

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorRDSAccess

aws:rds:reboot-db-instances

대상 DB 인스턴스에서 Amazon RDS API 작업 RebootDBInstance를 실행합니다. RDS 클러스터 및 DocumentDB 클러스터가 지원됩니다.

리소스 유형

- aws:rds:db

파라미터

- forceFailover – 선택 사항입니다. 값이 true이고 인스턴스가 다중 AZ인 경우, 한 가용 영역에서 다른 가용 영역으로 강제 장애 조치합니다. 기본값은 false입니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorRDSAccess

Amazon S3 작업

AWS FIS 는 다음 Amazon S3 작업을 지원합니다.

작업

- aws:s3:bucket-pause-replication

aws:s3:bucket-pause-replication

대상 소스 버킷에서 대상 버킷으로의 복제를 일시 중지합니다. 대상 버킷은 다른 AWS 리전에 있을 수도 있고 원본 버킷과 동일한 리전 내에 있을 수도 있습니다. 기존 객체는 작업이 시작된 후 최대 1시간 동안 계속 복제될 수 있습니다. 이 작업은 태그별 타겟팅만 지원합니다. Amazon S3 복제에 대해 자세히 알아보려면 Amazon S3 사용 설명서를 참조하세요.

리소스 유형

- aws:s3:bucket

파라미터

- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.
- region - 대상 버킷이 위치한 AWS 리전입니다.
- destinationBuckets – 선택 사항입니다. 쉼표로 구분된 대상 S3 버킷 목록입니다.
- prefixes – 선택 사항입니다. 복제 규칙 필터에서 쉼표로 구분된 S3 객체 키 접두사 목록입니다. 접두사를 기반으로 하는 필터가 있는 대상 버킷의 복제 규칙이 일시 중지됩니다.

권한

- 조건 키 `S3:IsReplicationPauseRequest`가 `True`로 설정된 `S3:PutReplicationConfiguration`
- 조건 키 `S3:IsReplicationPauseRequest`가 `True`로 설정된 `S3:GetReplicationConfiguration`
- `S3:PauseReplication`
- `S3:ListAllMyBuckets`
- `tag:GetResources`

정책 예제는 예: `aws:s3:bucket-pause-replication`의 조건 키 사용을 참조하세요.

Systems Manager 작업

AWS FIS 는 다음과 같은 Systems Manager 작업을 지원합니다.

작업

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

aws:ssm:send-command

대상 EC2 인스턴스에서 Systems Manager API 작업 SendCommand를 실행합니다. Systems Manager 문서(SSM 문서)는 Systems Manager가 인스턴스에서 수행하는 작업을 정의합니다. 자세한 내용은 aws:ssm:send-command 작업을 사용 단원을 참조하십시오.

리소스 유형

- aws:ec2:instance

파라미터

- documentArn - 문서의 Amazon 리소스 이름(ARN)입니다. 콘솔에서 사전 구성된 AWS FIS SSM 문서 중 하나에 해당하는 작업 유형에서 값을 선택하면이 파라미터가 완료됩니다.
- documentVersion – 선택 사항입니다. 문서의 버전입니다. 비어 있으면 기본 버전이 실행됩니다.

- documentParameters - 조건부. 문서에 허용되는 필수 및 선택 파라미터. 형식은 키가 문자열이고 값이 문자열 또는 문자열 배열인 JSON 객체입니다.
- duration - 소요 시간(1분에서 12시간). AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

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

AWS 관리형 정책

- AWSFaultInjectionSimulatorEC2Access

aws:ssm:start-automation-execution

Systems Manager API 작업 StartAutomationExecution을 실행합니다.

리소스 유형

- 없음

파라미터

- documentArn - 자동화 문서의 Amazon 리소스 이름(ARN)입니다.
- documentVersion – 선택 사항입니다. 문서의 버전입니다. 비어 있으면 기본 버전이 실행됩니다.
- documentParameters - 조건부. 문서에 허용되는 필수 및 선택 파라미터. 형식은 키가 문자열이고 값이 문자열 또는 문자열 배열인 JSON 객체입니다.
- maxDuration - 자동화 실행이 완료되는 데 허용되는 최대 시간은 1분에서 12시간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어, PT1M은 1분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간 수를 입력합니다.

권한

- `ssm:GetAutomationExecution`

- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `iam:PassRole` – 선택 사항입니다. 자동화 문서가 역할을 맡는 경우 필수입니다.

AWS 관리형 정책

- AWSFaultInjectionSimulatorSSMAccess

AWS Direct Connect 작업

AWS FIS 는 다음 AWS Direct Connect 작업을 지원합니다.

작업

- aws:directconnect:virtual-interface-disconnect

aws:directconnect:virtual-interface-disconnect

온프레미스 네트워크와 대상 가상 인터페이스(VIF)와 연결된 피어 간의 BGP(Border Gateway Protocol) 세션을 일시적으로 중단하여 AWS Direct Connect 연결 복원력을 테스트합니다.VIFs 실험을 시작하기 전에 FIS는 실험의 대상인 모든 VIFs가 '사용 가능' 상태이고 각 VIF에 '사용 가능' 상태 및 '위쪽' BGP 상태의 모든 BGP 피어가 있는지 확인합니다. 실험 중에 대상 가상 인터페이스에 대한 BGP 피어링 세션은 다운 상태로 전환됩니다. Direct Connect 장애 조치 테스트에 대한 자세한 내용은 AWS Direct Connect 설명서를 참조하세요.

리소스 유형

- aws:directconnect:virtual-interface

파라미터

- `duration` - 10분에서 12시간까지의 기간입니다. AWS FIS API에서 값은 ISO 8601 형식의 문자열입니다. 예를 들어 PT10M은 10분을 나타냅니다. AWS FIS 콘솔에서 초, 분 또는 시간을 입력합니다.

권한

- `directconnect:DescribeVirtualInterfaces`

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

AWS FIS에서 Systems Manager SSM 문서 사용

AWS FIS는 AWS Systems Manager SSM 에이전트 및 AWS FIS 작업를 통해 사용자 지정 장애 유형을 지원합니다aws:ssm:send-command. 일반적인 결함 주입 작업을 생성하는 데 사용할 수 있는 사전 구성된 Systems Manager SSM 문서(SSM 문서)는 AWSFIS 접두사로 시작하는 퍼블릭 AWS 문서로 사용할 수 있습니다.

SSM 에이전트는 Amazon EC2 인스턴스, 온프레미스 서버 또는 가상 머신(VM)에 설치 및 구성할 수 있는 Amazon 소프트웨어입니다. 이를 통해 Systems Manager가 이러한 리소스를 관리할 수 있습니다. 에이전트는 Systems Manager의 요청을 처리하고 요청에 지정된 대로 실행합니다. 자체 SSM 문서를 포함하여 사용자 지정 오류를 삽입하거나 Amazon 소유의 공개 문서 중 하나를 참조할 수 있습니다.

요구 사항

SSM 에이전트가 대상에서 작업을 실행해야 하는 작업의 경우 다음을 확인해야 합니다.

- 에이전트가 대상에 설치되어 있습니다. SSM 에이전트는 일부 Amazon Machine Images(AMI)에 기본적으로 설치됩니다. 그렇지 않으면 인스턴스에 SSM 에이전트를 설치할 수 있습니다. 자세한 정보는 AWS Systems Manager 사용 설명서의 EC2 인스턴스에 수동으로 SSM 에이전트 설치를 참조하세요.
- Systems Manager에는 인스턴스에서 작업을 수행할 수 있는 권한이 있습니다. IAM 인스턴스 프로파일을 사용하여 액세스 권한을 부여합니다. 자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager용 IAM 인스턴스 프로파일 생성 및 EC2 인스턴스에 IAM 인스턴스 프로파일 연결을 참조하세요.

aws:ssm:send-command 작업을 사용

SSM 문서는 Systems Manager가 관리형 인스턴스에서 실행하는 작업을 정의합니다. Systems Manager에는 사전 구성된 여러 문서가 포함되어 있으며 사용자가 직접 만들 수도 있습니다. SSM 문서를 직접 만드는 방법에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 문서 생성을 참조하세요. SSM 문서 전반에 대한 자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager 문서를 참조하세요.

AWS FIS는 사전 구성된 SSM 문서를 제공합니다. AWS Systems Manager 콘솔의 문서에서 사전 구성된 SSM 문서를 볼 수 있습니다. https://console.aws.amazon.com/systems-manager/documents. AWS FIS 콘솔에서 사전 구성된 문서 중에서 선택할 수도 있습니다. 자세한 내용은 사전 구성된 AWS FIS SSM 문서 단원을 참조하십시오.

AWS FIS 실험에서 SSM 문서를 사용하려면 aws:ssm:send-command 작업을 사용할 수 있습니다. 이 작업은 대상 인스턴스에서 지정된 SSM 문서를 가져와 실행합니다.

실험 템플릿에서 `aws:ssm:send-command` 작업을 사용할 때는 다음을 포함하여 작업에 대한 추가 파라미터를 지정해야 합니다.

- documentArn - 필수입니다. SSM 문서의 Amazon 리소스 이름(ARN)입니다.
- documentParameters - 조건부. SSM 문서에 허용되는 필수 및 선택 파라미터. 형식은 키가 문자열이고 값이 문자열 또는 문자열 배열인 JSON 객체입니다.
- documentVersion – 선택 사항입니다. 실행할 SSM 문서의 버전입니다.

Systems Manager 콘솔 또는 명령줄을 사용하여 SSM 문서에 대한 정보(문서 파라미터 포함)를 볼 수 있습니다.

콘솔을 사용하여 SSM 문서에 대한 정보를 확인하려면

1. https://console.aws.amazon.com/systems-manager/ AWS Systems Manager 콘솔을 엽니다.
2. 탐색 창에서 Documents를 선택합니다.
3. 문서를 선택하고 세부 정보 탭을 선택합니다.

명령줄을 사용하여 SSM 문서에 대한 정보를 확인하려면

SSM describe-document 명령을 사용합니다.

작업 상태에 대해 자세히 알아보기

SSM 작업 상태는 SSM 명령 상태에 따라 결정됩니다.

사전 구성된 AWS FIS SSM 문서

실험 템플릿의 `aws:ssm:send-command` 작업과 함께 사전 구성된 AWS FIS SSM 문서를 사용할 수 있습니다.

요구 사항

- AWS FIS에서 제공하는 사전 구성된 SSM 문서는 다음 운영 체제에서만 지원됩니다.
  - Amazon Linux 2023, Amazon Linux 2
  - Ubuntu
  - RHEL 8, 9
  - CentOS 9
- AWS FIS에서 제공하는 사전 구성된 SSM 문서는 EC2 인스턴스에서만 지원됩니다. 온프레미스 서버 등, 다른 유형의 관리형 노드에서는 지원되지 않습니다.

이러한 SSM 문서를 ECS 작업 실험에 사용하려면 해당 the section called “Amazon ECS 작업”를 사용하세요. 예를 들어, aws:ecs:task-cpu-stress 작업에는 AWSFIS-Run-CPU-Stress 문서가 사용됩니다.

문서

- 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 문서의 작업 기간과 DurationSeconds 간의 차이

일부 SSM 문서는 자체 실행 시간을 제한합니다. 예를 들어 DurationSeconds 파라미터는 사전 구성된 일부 AWS FIS SSM 문서에서 사용됩니다. 따라서 AWS FIS 작업 정의에 두 개의 독립 기간을 지정해야 합니다.

- Action duration: 단일 작업이 포함된 실험의 경우 작업 기간은 실험 기간과 동일합니다. 여러 작업의 경우 실험 기간은 개별 작업 기간과 실행 순서에 따라 달라집니다. AWS FIS는 작업 기간이 경과할 때까지 각 작업을 모니터링합니다.

- 문서 파라미터 DurationSeconds: SSM 문서가 실행될 기간(초 단위로 지정)입니다.

두 가지 유형의 기간에 대해 서로 다른 값을 선택할 수 있습니다.

- Action duration exceeds DurationSeconds: 작업이 완료되기 전에 SSM 문서 실행이 완료됩니다. AWS FIS는 후속 작업이 시작되기 전에 작업 기간이 경과할 때까지 기다립니다.
- Action duration is shorter than DurationSeconds: SSM 문서는 작업이 완료된 후에도 실행을 계속합니다. SSM 문서 실행이 아직 진행 중이고 작업 기간이 경과한 경우 작업 상태는 완료됨으로 설정됩니다. AWS FIS는 작업 기간이 경과할 때까지만 실행을 모니터링합니다.

일부 SSM 문서에는 가변 기간이 있습니다. 예를 들어 AWS FIS SSM 문서에는 사전 조건을 설치할 수 있는 옵션이 있어 전체 실행 기간을 지정된 DurationSeconds 파라미터 이상으로 연장할 수 있습니다. 따라서 작업 기간과 DurationSeconds를 동일한 값으로 설정하면 SSM 스크립트가 작업 기간보다 오래 실행될 수 있습니다.

AWSFIS-Run-CPU-Stress

stress-ng 도구를 사용하여 인스턴스에서 CPU 스트레스를 실행합니다. AWSFIS-Run-CPU-Stress SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- DurationSeconds - 필수입니다. CPU 스트레스 테스트의 지속 시간(초)입니다.
- CPU – 선택 사항입니다. 사용할 CPU 스트레스 요인의 수입니다. 기본값은 0이며, 모든 CPU 스트레스 요인을 사용합니다.
- LoadPercent – 선택 사항입니다. 0(무부하)에서 100(완전 부하)까지의 목표 CPU 부하 백분율. 기본값은 100입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 stress-ng입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Disk-Fill

인스턴스의 루트 볼륨에 디스크 공간을 할당하여 디스크 전체 오류를 시뮬레이션합니다. AWSFIS-Run-Disk-Fill SSM 문서를 사용합니다.

이 결함을 주입하는 실험이 수동으로 또는 중지 조건을 통해 중지되는 경우 AWS FIS는 실행 중인 SSM 문서를 취소하여 롤백을 시도합니다. 그러나 오류 또는 오류와 애플리케이션 작업으로 인해 디스크가 100% 가득 차면 Systems Manager에서 취소 작업을 완료하지 못할 수 있습니다. 따라서 실험을 중단해야 하는 경우 디스크가 100% 가득 차지 않도록 하세요.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- DurationSeconds - 필수입니다. 디스크 채우기 테스트의 지속 시간(초)입니다.
- Percent – 선택 사항입니다. 디스크 채우기 테스트 중에 할당할 디스크의 비율입니다. 기본값은 95%입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 atd, kmod 및 입니다fallocate.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-IO-Stress

stress-ng 도구를 사용하여 인스턴스에서 IO 스트레스를 실행합니다. AWSFIS-Run-IO-Stress SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- DurationSeconds - 필수입니다. IO 스트레스 테스트의 지속 시간(초)입니다.
- Workers – 선택 사항입니다. 순차, 임의 및 메모리 매핑된 읽기/쓰기 작업, 강제 동기화 및 캐시 삭제를 혼합하여 수행하는 작업자 수. 여러 하위 프로세스가 동일한 파일에서 다양한 I/O 작업을 수행합니다. 기본값은 1입니다.
- Percent – 선택 사항입니다. IO 스트레스 테스트 중에 사용할 파일 시스템의 여유 공간 비율입니다. 기본값은 80%입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 stress-ng입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Kill-Process

killall 명령을 사용하여 인스턴스에서 지정된 프로세스를 중지합니다. AWSFIS-Run-Kill-Process SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- ProcessName - 필수입니다. 중지할 프로세스의 이름입니다.

- Signal – 선택 사항입니다. 명령과 함께 전송할 신호입니다. 가능한 값은 SIGTERM(수신자가 무시하도록 선택할 수 있는 값)과 SIGKILL(무시할 수 없는 값)입니다. 기본값은 SIGTERM입니다.
- InstallDependencies – 선택 사항입니다. 값이 True이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 True입니다. 종속성은 killall입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Memory-Stress

stress-ng 도구를 사용하여 인스턴스에 메모리 스트레스를 실행합니다. AWSFIS-Run-Memory-Stress SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- DurationSeconds - 필수입니다. 메모리 스트레스 테스트의 지속 시간(초)입니다.
- Workers – 선택 사항입니다. 가상 메모리 스트레스 요인의 수. 기본값은 1입니다.
- Percent - 필수입니다. 메모리 스트레스 테스트 중에 사용할 가상 메모리의 비율입니다.
- InstallDependencies – 선택 사항입니다. 값이 True이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 True입니다. 종속성은 stress-ng입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Network-Blackhole-Port

iptables 도구를 사용하여 프로토콜 및 포트에 대한 인바운드 또는 아웃바운드 트래픽을 삭제합니다. AWSFIS-Run-Network-Blackhole-Port SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- Protocol - 필수입니다. 프로토콜. 가능한 값은 `tcp`와 `udp`입니다.
- Port - 필수입니다. 포트 번호입니다.
- TrafficType – 선택 사항입니다. 트래픽 유형입니다. 가능한 값은 `ingress`와 `egress`입니다. 기본값은 `ingress`입니다.
- DurationSeconds - 필수입니다. 네트워크 블랙홀 테스트 시간(초)입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 atd, dig, lsof 및 iptables입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Network-Latency

tc 도구를 사용하여 네트워크 인터페이스에 지연 시간을 추가합니다. AWSFIS-Run-Network-Latency SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- Interface – 선택 사항입니다. 네트워크 인터페이스입니다. 기본값은 `eth0`입니다.
- DelayMilliseconds – 선택 사항입니다. 밀리 초 단위의 지연 시간입니다. 기본값은 200입니다.

- DurationSeconds - 필수입니다. 네트워크 지연 시간 테스트 시간(초)입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 atd, dig 및 tc입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Network-Latency-Sources

특정 소스에서 오가는 트래픽을 처리하는 tc 도구를 사용하여 네트워크 인터페이스에 지연 시간과 지터를 추가합니다. AWSFIS-Run-Network-Latency-Sources SSM 문서를 사용합니다.

`FlowsPercent` 파라미터를 사용하여 연결의 백분율에 지연 시간을 추가합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- Interface – 선택 사항입니다. 쉼표로 구분된 네트워크 인터페이스입니다. ALL 및 DEFAULT 값이 지원됩니다. 기본값은 운영 체제의 기본 네트워크 인터페이스를 대상으로 `DEFAULT`하는 입니다.
- DelayMilliseconds – 선택 사항입니다. 밀리 초 단위의 지연 시간입니다. 기본값은 200입니다.
- JitterMilliseconds – 선택 사항입니다. 밀리 초 단위의 지터입니다. 기본값은 10입니다.
- FlowsPercent – 선택 사항입니다. 작업의 영향을 받는 네트워크 흐름의 백분율입니다. 의 기본값은 100%입니다.
- Sources - 필수입니다. 쉼표로 구분되고 공백이 없는 소스입니다. 가능한 값은 IPv4 주소, IPv4 CIDR 블록, 도메인 이름, AZ 이름(us-east-1a), AZ ID(use1-az1), ALL, 및 `DYNAMODB`입니다`S3`. `DYNAMODB` 또는 `S3`를 지정하는 경우 이는 현재 리전의 리전 엔드포인트에만 적용됩니다.
- TrafficType – 선택 사항입니다. 트래픽 유형입니다. 가능한 값은 `ingress`와 `egress`입니다. 기본값은 `ingress`입니다.

- DurationSeconds - 필수입니다. 네트워크 지연 시간 테스트 시간(초)입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`이면 Systems Manager가 대상 인스턴스에 필요한 종속성을 설치합니다(아직 설치되지 않은 경우). 기본값은 `True`입니다. 종속성은 atd, dig, jq, 및 lsof입니다tc.

이 문서를 사용할 때 실험 역할에는 다음 권한이 필요합니다.

- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

```
{"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

tc 도구를 사용하여 네트워크 인터페이스에 패킷 손실을 추가합니다. AWSFIS-Run-Network-Packet-Loss SSM 문서를 사용합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- Interface – 선택 사항입니다. 네트워크 인터페이스입니다. 기본값은 `eth0`입니다.
- LossPercent – 선택 사항입니다. 패킷 손실의 백분율. 기본값은 7%입니다.
- DurationSeconds - 필수입니다. 네트워크 패킷 손실 테스트 시간(초)입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`인 경우 Systems Manager는 대상 인스턴스에 필요한 종속성을 설치합니다. 기본값은 `True`입니다. 종속성은 atd, lsof, dig 및 tc입니다.

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

AWSFIS-Run-Network-Packet-Loss-Sources

특정 소스에서 오가는 트래픽을 처리하는 tc 도구를 사용하여 네트워크 인터페이스에 패킷 손실을 추가합니다. AWSFIS-Run-Network-Packet-Loss-Sources SSM 문서를 사용합니다.

`FlowsPercent` 파라미터를 사용하여 연결의 백분율에 패킷 손실을 주입합니다.

작업 유형(콘솔 전용)

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

ARN

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

문서 파라미터

- Interface – 선택 사항입니다. 쉼표로 구분된 네트워크 인터페이스입니다. ALL 및 DEFAULT 값이 지원됩니다. 기본값은 운영 체제의 기본 네트워크 인터페이스를 대상으로 `DEFAULT`하는 입니다.
- LossPercent – 선택 사항입니다. 패킷 손실의 백분율. 기본값은 7%입니다.
- FlowsPercent – 선택 사항입니다. 작업의 영향을 받는 네트워크 흐름의 백분율입니다. 의 기본값은 100%입니다.
- Sources - 필수입니다. 쉼표로 구분되고 공백이 없는 소스입니다. 가능한 값은 IPv4 주소, IPv4 CIDR 블록, 도메인 이름, AZ 이름(us-east-1a), AZ ID(use1-az1), ALL, 및 `DYNAMODB`입니다`S3`. `DYNAMODB` 또는 `S3`를 지정하는 경우 이는 현재 리전의 리전 엔드포인트에만 적용됩니다.
- TrafficType – 선택 사항입니다. 트래픽 유형입니다. 가능한 값은 `ingress`와 `egress`입니다. 기본값은 `ingress`입니다.
- DurationSeconds - 필수입니다. 네트워크 패킷 손실 테스트 시간(초)입니다.
- InstallDependencies – 선택 사항입니다. 값이 `True`인 경우 Systems Manager는 대상 인스턴스에 필요한 종속성을 설치합니다. 기본값은 `True`입니다. 종속성은 atd, dig, jq, 및 lsof입니다tc.

이 문서를 사용할 때 실험 역할에는 다음 권한이 필요합니다.

- `ec2:DescribeInstances`
- `ec2:DescribeSubnets`

다음은 콘솔에 입력할 수 있는 문자열의 예입니다.

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

예제

실험 템플릿 예시를 보려면 the section called “사전 구성된 AWS FIS SSM 문서 실행” 단원을 참조하세요.

자습서 예시는 인스턴스에서 CPU 스트레스 실행 단원을 참조하세요.

제한 사항

- 다음 문서는 병렬로 실행할 수 없습니다.
  - 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

롤백 스크립트

AWS FIS SSM 문서는 결함 주입 실험 후 시스템 상태를 복원하는 안전 메커니즘으로 롤백 스크립트를 자동으로 생성합니다. 이러한 스크립트는 작업이 실패하거나 예기치 않게 종료되더라도 주입된 결함이 제거되도록 합니다.

롤백 스크립트 생성

롤백 스크립트는 오류 주입 실험이 시작될 때 자동으로 생성됩니다.

생성 세부 정보

- 위치 - `/var/lib/amazon/ssm/` 디렉터리에 스크립트가 생성됩니다.
- 이름 지정 패턴 - *`FAULT_NAME`*`-`*`FAULT_IDENTIFIER`*`-Rollback.sh` 여기서 *`FAULT_IDENTIFIER`*는 무작위로 생성된 32자 문자열입니다.
- 타이밍 - 결함 주입이 시작되기 전에 각 결함 주입 실험이 시작될 때 생성됩니다.
- 콘텐츠 - 특정 오류를 되돌리는 데 필요한 모든 환경 변수와 명령을 포함합니다.

예를 들어 네트워크 지연 시간 실험은에서 롤백 스크립트를 생성할 수 있습니다`/var/lib/amazon/ssm/NetworkLatency-abc123-Rollback.sh`.

롤백 로깅

롤백 스크립트는 이중 로깅을 구현하여 문제 해결 및 감사 목적으로 모든 롤백 활동을 캡처합니다.

로그 파일 위치

롤백 스크립트가 실행되면 다음 두 위치에 로그가 생성됩니다.

- 임시 파일 - `/tmp/aws-fis-rollback-`*`TIMESTAMP`*`-`*`PID`*`.log`
- 시스템 로그 - 시설과 함께 syslog로 전송됨 `local0.info`

로그 파일 이름 지정

임시 로그 파일은 다음 명명 규칙을 사용합니다.

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

여기서 *`YYYY-MM-DDTHH:MM:SSZ`*는 UTC 타임스탬프이고 *`PID`*는 롤백 스크립트의 프로세스 ID입니다.

Syslog 구성

롤백 로그는 다음 구성으로 syslog로 전송됩니다.

- 태그 - `aws-fis-rollback`
- 우선순위 – `local0.info`
- 형식 - `[YYYY-MM-DDTHH:MM:SSZ]` *`log_message`*

롤백 로그를 보려면

다음 명령을 사용하여 시스템 저널의 모든 롤백 로그를 봅니다.

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

문제 해결

다음 절차를 사용하여 문제를 해결하세요.

SSM 문서 관련 문제를 해결하려면

1. https://console.aws.amazon.com/systems-manager/ AWS Systems Manager 콘솔을 엽니다.
2. 탐색 창에서 노드 관리, 명령 실행을 선택합니다.
3. 명령 기록 탭에서 필터를 사용하여 문서 실행 위치를 찾습니다.
4. 명령의 ID를 선택하여 세부 정보 페이지를 엽니다.
5. 인스턴스의 ID를 선택합니다. 각 단계의 출력 및 오류를 검토하세요.

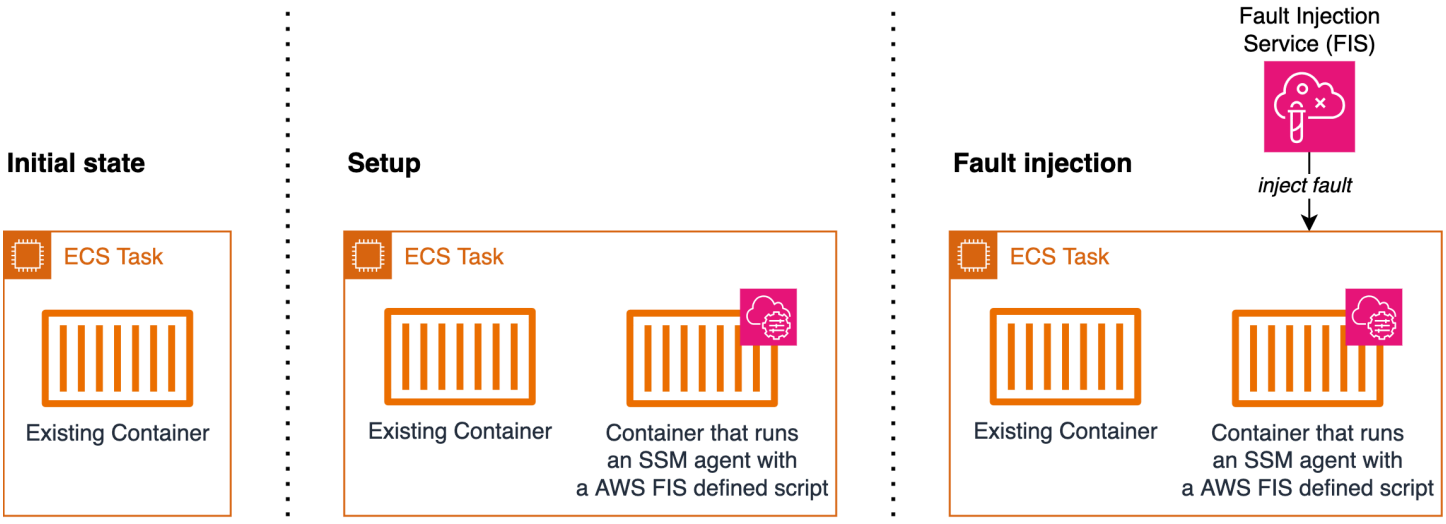
AWS FIS aws:ecs:task 작업 사용

aws:ecs:task 작업을 사용하여 Amazon ECS 태스크에 오류를 주입할 수 있습니다. Amazon EC2 및 Fargate 용량 유형이 지원됩니다.

이러한 작업은 AWS Systems Manager(SSM) 문서를 사용하여 결함을 주입합니다. `aws:ecs:task` 작업을 사용하려면 Amazon Elastic Container Service(Amazon ECS) 태스크 정의에 SSM 에이전트가 있는 컨테이너를 추가해야 합니다. 컨테이너는 Amazon ECS 태스크를 SSM 서비스에서 관리형 인스턴스로 등록하는 AWS FIS 정의 스크립트를 실행합니다. 또한 이 스크립트는 태스크 메타데이터를 검색하여 관리형 인스턴스에 태그를 추가합니다. 설정을 통해 AWS FIS는 대상 작업을 확인할 수 있습니다. 이 단락의 내용은 아래 다이어그램의 설정에 해당합니다.

를 대상으로 AWS FIS 실험을 실행하면 `aws:ecs:task` AWS AWS FIS 실험 템플릿에서 지정한 대상 Amazon ECS 작업을 리소스 태그를 사용하여 SSM 관리형 인스턴스 세트에 매핑합니다`ECS_TASK_ARN`. 이 태그 값은 SSM 문서가 실행되어야 하는 연결된 Amazon ECS 태스크의 ARN입니다. 이 단락의 내용은 아래 다이어그램의 결함 주입에 해당합니다.

다음 다이어그램은 하나의 기존 컨테이너가 있는 태스크에 대한 설정 및 결함 주입을 보여줍니다.



작업

- 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”

제한 사항

- 다음 작업은 병렬로 실행할 수 없습니다.
  - aws:ecs:task-network-blackhole-port
  - aws:ecs:task-network-latency
  - aws:ecs:task-network-packet-loss
- Amazon ECS Exec을 활성화한 경우 이러한 작업을 사용하려면 먼저 비활성화해야 합니다.
- SSM 문서 실행은 실험에 완료됨 상태가 있더라도 취소됨 상태일 수 있습니다. Amazon ECS 작업을 실행할 때 고객이 제공한 기간은 실험의 작업 기간과 Amazon EC2 Systems Manager(SSM) 문서 기간 모두에 사용됩니다. 작업이 시작된 후 SSM 문서가 실행되기까지 약간의 시간이 걸립니다. 따라서 지정된 작업 기간에 도달할 때까지 SSM 문서의 실행을 완료하는 데 몇 초가 남을 수 있습니다. 실험 작업 기간에 도달하면 작업이 중지되고 SSM 문서 실행이 취소됩니다. 결함 주입에 성공했습니다.

요구 사항

- AWS FIS 실험 역할에 다음 권한을 추가합니다.
  - `ecs:DescribeTasks`
  - `ssm:SendCommand`
  - `ssm:ListCommands`
  - `ssm:CancelCommand`
- 다음 권한을 Amazon ECS 작업 IAM 역할에 추가합니다.
  - `ssm:CreateActivation`

  - `ssm:AddTagsToResource`
  - `iam:PassRole`

  참고로 관리형 인스턴스 역할의 ARN을 `iam:PassRole`의 리소스로 지정할 수 있습니다.
- Amazon ECS 작업 실행 IAM 역할을 생성하고 AmazonECSTaskExecutionRolePolicy 관리형 정책을 추가합니다.
- 태스크 정의에서 환경 변수 `MANAGED_INSTANCE_ROLE_NAME`을 관리형 인스턴스 역할 이름으로 설정합니다. 이 역할은 SSM에서 관리형 인스턴스로 등록된 태스크에 연결됩니다.
- 관리형 인스턴스 역할에 다음 권한을 추가합니다.
  - `ssm:DeleteActivation`
  - `ssm:DeregisterManagedInstance`
- 관리형 인스턴스 역할에 AmazonSSMManagedInstanceCore 관리형 정책을 추가합니다.
- Amazon ECS 태스크 정의에 SSM 에이전트 컨테이너를 추가합니다. 명령 스크립트는 Amazon ECS 작업을 관리형 인스턴스로 등록합니다.

```
{
    "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": []
}
```

더 읽기 쉬운 스크립트 버전은 the section called “스크립트의 참조 버전”을 참조하세요.

- Amazon ECS 태스크 정의에서 enableFaultInjection 필드를 설정하여 Amazon ECS Fault Injection APIs를 활성화합니다.

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

- Fargate 작업에서 aws:ecs:task-network-blackhole-portaws:ecs:task-network-latency, 또는 aws:ecs:task-network-packet-loss 작업을 사용하는 경우 작업에 useEcsFaultInjectionEndpoints 파라미터가 로 설정되어 있어야 합니다true.
- aws:ecs:task-kill-process, aws:ecs:task-network-latency, 또는 aws:ecs:task-network-packet-loss 작업을 사용하는 경우 Amazon ECS 태스크 정의는 aws:ecs:task-network-blackhole-port를 로 pidMode 설정해야 합니다task.
- EC2 시작 유형의 태스크에서 aws:ecs:task-network-blackhole-portaws:ecs:task-network-latency, 또는 aws:ecs:task-network-packet-loss 작업을 사용하는 경우 태스크 정의의 네트워킹 옵션을 awsvpc 또는 로 설정해야 합니다host.

스크립트의 참조 버전

다음은 참조용으로 요구 사항 섹션에 있는 더 읽기 쉬운 버전의 스크립트입니다.

```
#!/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
```

실험 템플릿 예시

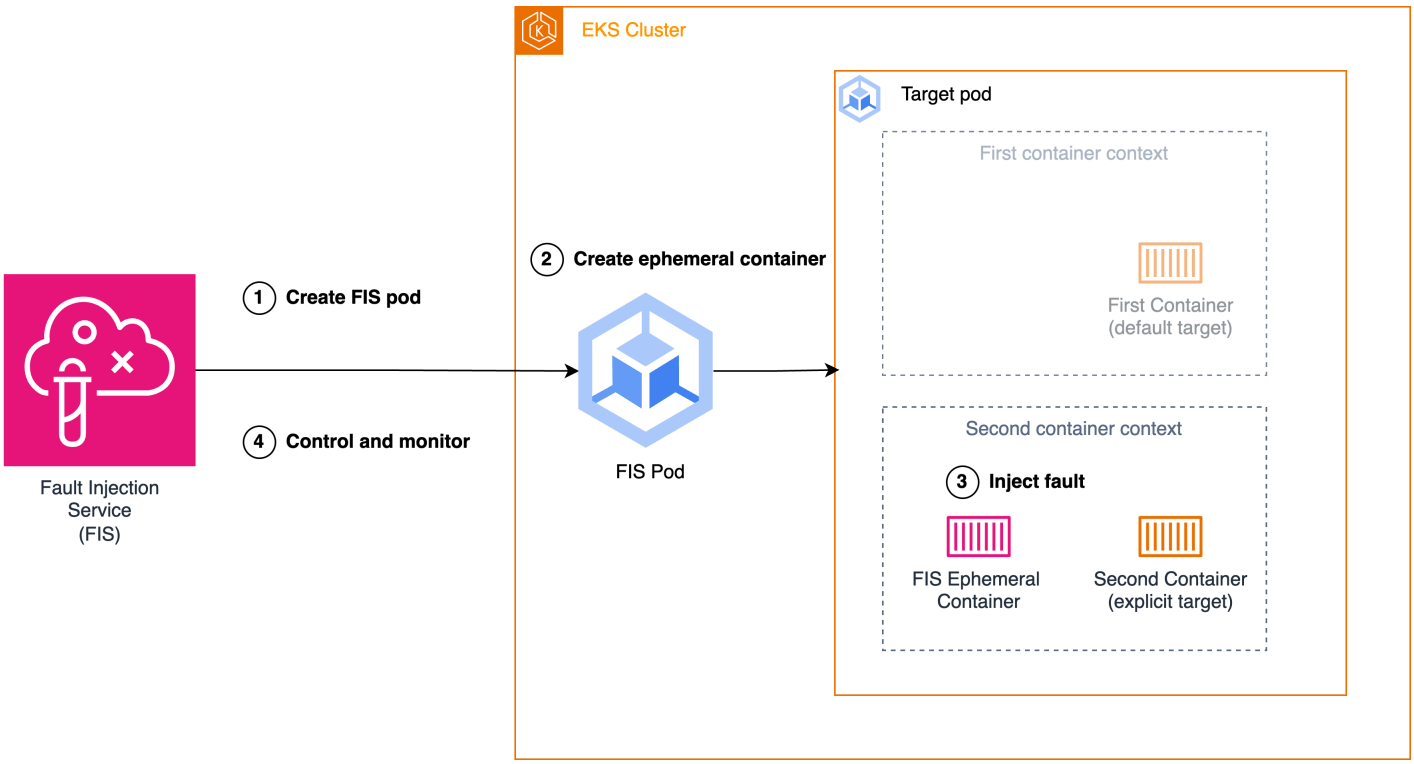
다음은 the section called “aws:ecs:task-cpu-stress” 작업에 대한 예제 실험 템플릿입니다.

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

AWS FIS aws:eks:pod 작업 사용

aws:eks:pod 작업을 사용하여 EKS 클러스터에서 실행되는 Kubernetes 포드에 오류를 주입할 수 있습니다.

작업이 시작되면 FIS는 FIS 포드 컨테이너 이미지를 검색합니다. 그런 다음이 이미지를 사용하여 대상 EKS 클러스터에 포드를 생성합니다. 새로 생성된 포드는 오류를 주입, 제어 및 모니터링하는 역할을 합니다. aws:eks:pod-delete을 제외한 모든 FIS EKS 작업의 경우 기존 포드 내에 임시 컨테이너를 생성할 수 있는 Kubernetes 기능인 임시 컨테이너를 사용하여 결함 주입을 수행합니다. 임시 컨테이너는 대상 컨테이너와 동일한 네임스페이스에서 시작되며 원하는 결함 주입 작업을 실행합니다. 대상 컨테이너가 지정되지 않은 경우 포드 사양의 첫 번째 컨테이너가 대상으로 선택됩니다.



1. FIS는 실험 템플릿에 지정된 대상 클러스터에 FIS 포드를 생성합니다.
2. FIS 포드는 대상 컨테이너와 동일한 네임스페이스의 대상 포드에 임시 컨테이너를 생성합니다.
3. 임시 컨테이너는 대상 컨테이너의 네임스페이스에 결함을 주입합니다.
4. FIS 포드는 임시 컨테이너의 결함 주입을 제어 및 모니터링하며, FIS는 FIS 포드를 제어 및 모니터링합니다.

실험이 완료되거나 오류가 발생하면 임시 컨테이너와 FIS 포드가 제거됩니다.

작업

- 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”

제한 사항

- 다음 작업은에서 작동하지 않습니다. AWS Fargate
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- 다음 작업은 `bridge` 네트워크 모드를 지원하지 않습니다.
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- 다음 작업에는 임시 컨테이너 내의 루트 권한이 필요합니다.
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss

  임시 컨테이너는 대상 포드의 보안 컨텍스트에서 권한을 상속합니다. 루트가 아닌 사용자로 포드의 컨테이너를 실행해야 하는 경우 대상 포드의 컨테이너에 대해 별도의 보안 컨텍스트를 설정할 수 있습니다.
- 실험 템플릿에서는 리소스 ARN 또는 리소스 태그를 사용하여 aws:eks:pod 유형의 대상을 식별할 수 없습니다. 필수 리소스 파라미터를 사용하여 대상을 식별해야 합니다.
- aws:eks:pod-network-latency 및 aws:eks:pod-network-packet-loss 작업은 병렬로 실행해서는 안 되며 동일한 포드를 대상으로 해야 합니다. 지정한 `maxErrors` 파라미터의 값에 따라 작업이 완료 또는 실패 상태로 종료될 수 있습니다.

  - `maxErrorsPercent`가 0(기본값)이면 작업이 failed 상태로 종료됩니다.
  - 그렇지 않으면 실패가 `maxErrorsPercent` 예산에 합산됩니다. 실패한 주입 횟수가 제공된 `maxErrors`에 도달하지 않으면 작업이 완료된 상태로 종료됩니다.
  - 대상 포드에 삽입된 임시 컨테이너의 로그에서 이러한 실패를 식별할 수 있습니다. `Exit Code: 16`와 함께 실패합니다.
- 작업 aws:eks:eks:pod-network-blackhole-port는 동일한 포드를 대상으로 하고 동일한를 사용하는 다른 작업과 병렬로 실행해서는 안 됩니다`trafficType`. 서로 다른 트래픽 유형을 사용하는 병렬 작업이 지원됩니다.
- FIS는 대상 포드의가 로 설정된 경우에만 결함 주입 `securityContext` 상태를 모니터링할 수 있습니다`readOnlyRootFilesystem: false`. 이 구성이 없으면 모든 EKS 포드 작업이 실패합니다.

요구 사항

- 컴퓨터에 AWS CLI 를 설치합니다. 이는 AWS CLI 를 사용하여 IAM 역할을 생성하는 경우에만 필요합니다. 자세한 내용은 AWS CLI설치 또는 업데이트를 참조하세요.
- 컴퓨터에 kubectl를 설치합니다. 이는 EKS 클러스터와 상호 작용하여 대상 애플리케이션을 구성하거나 모니터링하는 경우에만 필요합니다. 자세한 내용은 https://kubernetes.io/docs/tasks/tools/를 참조하세요.
- 현재 지원되는 최소 EKS 버전은 1.23입니다.

실험 역할 생성

실험을 실행하려면 실험에 대한 IAM 역할을 구성해야 합니다. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오. 이 역할에 필요한 권한은 사용 중인 작업에 따라 달라집니다. 작업에 필요한 권한을 찾으려면 `aws:eks:pod`를 대상으로 하는AWS FIS 작업을 참조하세요.

Kubernetes 서비스 계정 구성

지정된 Kubernetes 네임스페이스의 대상으로 실험을 실행하도록 Kubernetes 서비스 계정을 구성합니다. 다음 예제에서 서비스 계정은 *`myserviceaccount`*이고 네임스페이스는 *`###`*입니다. 참고로 default는 표준 Kubernetes 네임스페이스 중 하나입니다.

Kubernetes 서비스 계정을 구성하려면

1. 이름이 `rbac.yaml`인 파일을 만들고 다음을 추가합니다.

```
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. 다음 명령을 실행합니다.

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

IAM 사용자 및 역할에 Kubernetes API에 대한 액세스 권한 부여

EKS 설명서의 IAM 자격 증명을 Kubernetes 권한과 연결에 나온 단계를 따르세요.

옵션 1: 액세스 항목 생성

Access Entries를 사용하는 것이 좋습니다. 다음 명령을 사용하여 IAM 역할을 Kubernetes 사용자 *fis-experiment*와 연결하는 액세스 항목을 생성할 수 있습니다. 자세한 내용은 IAM 사용자에게 EKS 액세스 항목을 사용하여 Kubernetes에 대한 액세스 권한 부여를 참조하세요.

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

> **Important**
>
> 액세스 항목을 활용하려면 EKS 클러스터의 인증 모드를 API_AND_CONFIG_MAP 또는 API 모드로 구성해야 합니다.

옵션 2: aws-auth ConfigMap에 항목 추가

다음 명령을 사용하여 자격 증명 매핑을 생성할 수 있습니다. 자세한 내용은 eksctl 설명서의 IAM 사용자 및 역할 관리를 참조하세요.

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

> **Important**
>
> eksctl 툴킷을 활용하여 자격 증명 매핑을 구성하면 `aws-auth` ConfigMap 내에 항목이 생성됩니다. 이렇게 생성된 항목은 경로 구성 요소의 포함을 지원하지 않는다는 점에 유의하세요. 따라서 입력으로 제공된 ARN에는 경로 세그먼트(예: `arn:aws:iam::123456789012:role/service-role/fis-experiment-role`)가 포함되어서는 안 됩니다.

포드 컨테이너 이미지

AWS FIS에서 제공하는 포드 컨테이너 이미지는 Amazon ECR에서 호스팅됩니다. Amazon ECR의 이미지를 참조할 때는 전체 이미지 URI를 사용해야 합니다.

포드 컨테이너 이미지는 AWS ECR 퍼블릭 갤러리에서도 사용할 수 있습니다.

AWS 리전	이미지 URI
미국 동부(오하이오)	`051821878176.dkr.ecr.us-east-2.amazonaws.com/aws-fis-pod:0.1`
미국 동부(버지니아 북부)	`731367659002.dkr.ecr.us-east-1.amazonaws.com/aws-fis-pod:0.1`
미국 서부(캘리포니아 북부)	`080694859247.dkr.ecr.us-west-1.amazonaws.com/aws-fis-pod:0.1`
미국 서부(오리건)	`864386544765.dkr.ecr.us-west-2.amazonaws.com/aws-fis-pod:0.1`
아프리카(케이프타운)	`056821267933.dkr.ecr.af-south-1.amazonaws.com/aws-fis-pod:0.1`
아시아 태평양(홍콩)	`246405402639.dkr.ecr.ap-east-1.amazonaws.com/aws-fis-pod:0.1`
아시아 태평양(뭄바이)	`524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1`

AWS 리전	이미지 URI
아시아 태평양(오사카)	148336246925.dkr.ecr.ap-northeast-3.amazonaws.com/aws-fis-pod:0.1
아시아 태평양(서울)	526524659354.dkr.ecr.ap-northeast-2.amazonaws.com/aws-fis-pod:0.1
아시아 태평양(싱가포르)	316401638346.dkr.ecr.ap-southeast-1.amazonaws.com/aws-fis-pod:0.1
아시아 태평양(시드니)	488104106298.dkr.ecr.ap-southeast-2.amazonaws.com/aws-fis-pod:0.1
아시아 태평양(도쿄)	635234321696.dkr.ecr.ap-northeast-1.amazonaws.com/aws-fis-pod:0.1
캐나다(중부)	490658072207.dkr.ecr.ca-central-1.amazonaws.com/aws-fis-pod:0.1
유럽(프랑크푸르트)	713827034473.dkr.ecr.eu-central-1.amazonaws.com/aws-fis-pod:0.1
유럽(아일랜드)	205866052826.dkr.ecr.eu-west-1.amazonaws.com/aws-fis-pod:0.1
유럽(런던)	327424803546.dkr.ecr.eu-west-2.amazonaws.com/aws-fis-pod:0.1
유럽(밀라노)	478809367036.dkr.ecr.eu-south-1.amazonaws.com/aws-fis-pod:0.1
유럽(파리)	154605889247.dkr.ecr.eu-west-3.amazonaws.com/aws-fis-pod:0.1
유럽(스페인)	395402409451.dkr.ecr.eu-south-2.amazonaws.com/aws-fis-pod:0.1
유럽(스톡홀름)	263175118295.dkr.ecr.eu-north-1.amazonaws.com/aws-fis-pod:0.1

AWS 리전	이미지 URI
유럽(취리히)	604225987275.dkr.ecr.eu-central-2.amazonaws.com/aws-fis-pod:0.1
Middle East (Bahrain)	065825543785.dkr.ecr.me-south-1.amazonaws.com/aws-fis-pod:0.1
중동(UAE)	438374459301.dkr.ecr.me-central-1.amazonaws.com/aws-fis-pod:0.1
남아메리카(상파울루)	767113787785.dkr.ecr.sa-east-1.amazonaws.com/aws-fis-pod:0.1
AWS GovCloud(미국 동부)	246533647532.dkr.ecr.us-gov-east-1.amazonaws.com/aws-fis-pod:0.1
AWS GovCloud(미국 서부)	246529956514.dkr.ecr.us-gov-west-1.amazonaws.com/aws-fis-pod:0.1

실험 템플릿 예시

다음은 the section called “aws:eks:pod-network-latency” 작업에 대한 예제 실험 템플릿입니다.

```
{
    "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": {
```

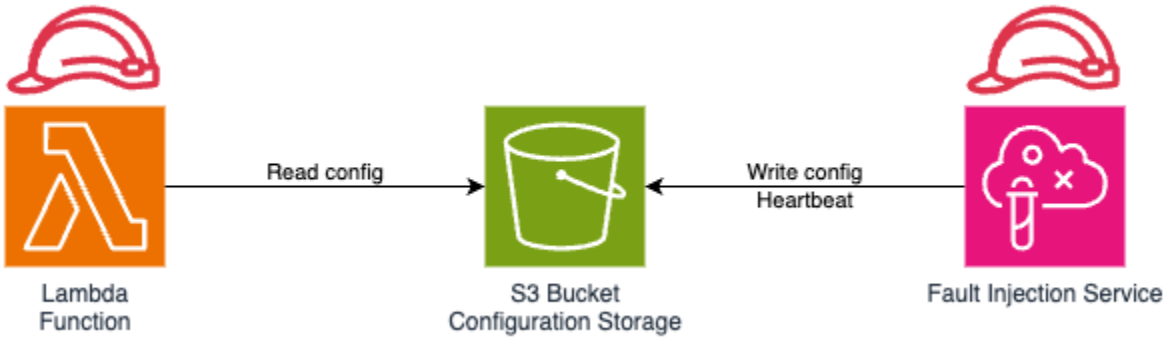
```
        "EksPod-latency": {
            "actionId": "aws:eks:pod-network-latency",
            "description": "Add latency",
            "parameters": {
                "kubernetesServiceAccount": "myserviceaccount",
                "duration": "PT5M",
                "delayMilliseconds": "200",
                "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"
    }
}
```

AWS FIS aws:lambda:function 작업 사용

aws:lambda:function 작업을 사용하여 AWS Lambda 함수 호출에 오류를 주입할 수 있습니다.

이러한 작업은 AWS FIS 관리형 확장을 사용하여 오류를 주입합니다. aws:lambda:function 작업을 사용하려면 확장을 Lambda 함수에 계층으로 연결하고와 확장 간에 통신하도록 Amazon S3 버킷 AWS FIS 을 구성해야 합니다.

aws:lambda:function을 대상으로 AWS FIS 실험을 실행하면 아래 다이어그램과 같이 Lambda 함수에서 Amazon S3 구성을 AWS FIS 읽고 지정된 Amazon S3 위치에 오류 삽입 정보를 씁니다.



작업

- the section called “aws:lambda:invocation-add-delay”
- the section called “aws:lambda:invocation-error”
- the section called “aws:lambda:invocation-http-integration-response”

제한 사항

- AWS FIS Lambda 확장은 응답 스트리밍을 사용하는 함수와 함께 사용할 수 없습니다. 결함이 적용되지 않더라도 AWS FIS Lambda 확장은 스트리밍 구성을 억제합니다. 자세한 내용은 AWS Lambda 사용 설명서의 Lambda 함수에 대한 응답 스트리밍을 참조하세요.

사전 조건

AWS FIS Lambda 작업을 사용하기 전에 다음 일회성 작업을 완료했는지 확인합니다.

- 에서 실험을 시작하려는 리전에 Amazon S3 버킷을 생성합니다 - 여러 실험에 단일 Amazon S3 버킷을 사용하고 여러 AWS 계정 간에 버킷을 공유할 수 있습니다. 그러나 각각에 대해 별도의 버킷이 있어야 합니다 AWS 리전.
- Amazon S3 버킷에 대한 Lambda 확장에 대한 읽기 액세스 권한을 부여하는 IAM 정책 생성 - 다음 템플릿에서 `my-config-distribution-bucket`를 위에서 생성한 Amazon S3 버킷의 이름으로 바꾸고를 사용하려는 Amazon S3 버킷의 폴더 `FisConfigs` 이름으로 바꿉니다.

  JSON

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

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

- AWS FIS 실험에 대한 쓰기 액세스 권한을 Amazon S3 버킷에 부여하는 IAM 정책 생성 - 다음 템플릿에서 `my-config-distribution-bucket`를 위에서 생성한 Amazon S3 버킷의 이름으로 바꾸고를 사용하려는 Amazon S3 버킷의 폴더 `FisConfigs` 이름으로 바꿉니다.

  JSON

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

```
            "Action": [
                "lambda:GetFunction"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowFisToDoTagLookups",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
}
```

Lambda 함수 구성

영향을 미치려는 모든 Lambda 함수에 대해 아래 단계를 따릅니다.

1. 위에서 생성한 Amazon S3 읽기 액세스 정책을 Lambda 함수에 연결합니다.
2. AWS FIS 확장을 함수에 계층으로 연결합니다. 계층 ARNsLambda용 AWS FIS 확장의 사용 가능한 버전.
3. `AWS_FIS_CONFIGURATION_LOCATION` 예를 들어 변수를 Amazon S3 구성 폴더의 ARN으로 설정합니다`arn:aws:s3:::my-config-distribution-bucket/FisConfigs/`.
4. `AWS_LAMBDA_EXEC_WRAPPER` 변수를 `/opt/aws-fis/bootstrap`로 설정합니다.

AWS FIS 실험 구성

실험을 실행하기 전에 사전 조건에서 생성한 Amazon S3 쓰기 액세스 정책을 AWS FIS Lambda 작업을 사용할 실험 역할에 연결했는지 확인합니다. AWS FIS 실험을 설정하는 방법에 대한 자세한 내용은 섹션을 참조하세요AWS FIS 실험 템플릿 관리.

로깅

AWS FIS Lambda 확장은 콘솔 및 CloudWatch 로그에 로그를 씁니다. `AWS_FIS_LOG_LEVEL` 변수를 사용하여 로깅을 구성할 수 있습니다. 지원되는 값은 `INFO`, `WARN` 및 `ERROR`입니다. 로그는 Lambda 함수에 대해 구성된 로그 형식으로 작성됩니다.

다음은 텍스트 형식의 로그 예제입니다.

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

다음은 JSON 형식의 로그 예제입니다.

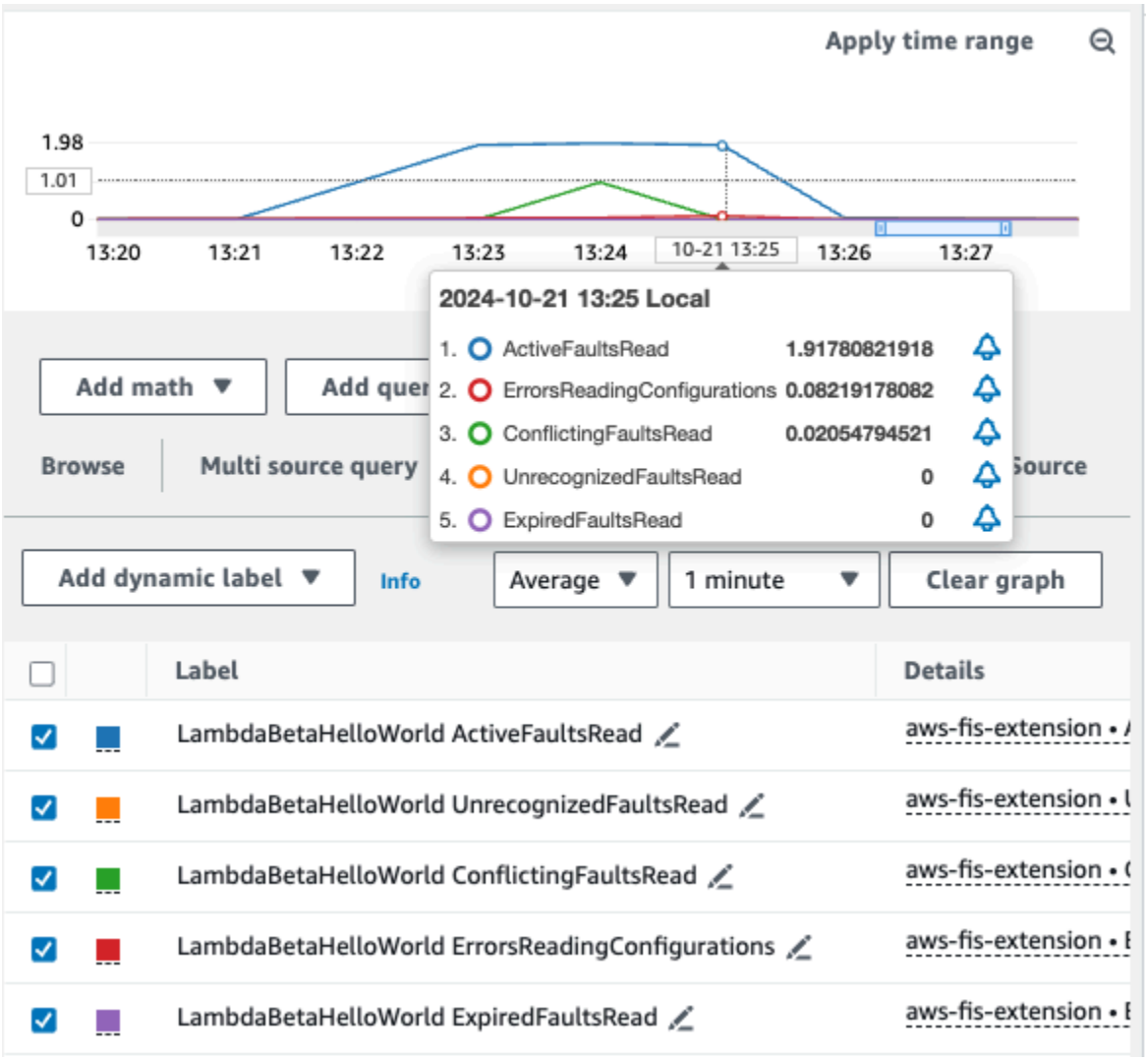
```
{
  "timestamp": "2024-10-08T17:15:36.953905Z",
  "level": "INFO",
  "fields": {
    "message": "AWS FIS EXTENSION - adding 5000 milliseconds of latency to function
 invocation",
    "requestId":"0608bf70-908f-4a17-bbfe-3782cd783d8b"
  }
}
```

내보낸 로그를 Amazon CloudWatch 지표 필터와 함께 사용하여 사용자 지정 지표를 생성할 수 있습니다. 지표 필터에 대한 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 필터를 사용하여 로그 이벤트에서 지표 생성을 참조하세요.

CloudWatch 임베디드 지표 형식(EMF) 사용

`AWS_FIS_EXTENSION_METRICS` 변수를 로 설정하여 EMF 로그를 내보내도록 AWS FIS Lambda 확장을 구성할 수 있습니다`all`. 기본적으로 확장은 EMF 로그를 내보내지 않으며 `AWS_FIS_EXTENSION_METRICS` 기본값은 입니다`none`. EMF 로그는 CloudWatch 콘솔의 `aws-fis-extension namespace`에 게시됩니다.

`aws-fis-extension` 네임스페이스 내에서 그래프에 표시할 특정 지표를 선택할 수 있습니다. 아래 예제는 `aws-fis-extension` 네임스페이스에서 사용 가능한 지표 중 일부를 보여줍니다.



고급 주제

이 섹션에서는가 Lambda 확장 및 특수 사용 사례와 함께 AWS FIS 작동하는 방법에 대한 추가 정보를 제공합니다.

주제

- 폴링 이해
- 동시성 이해
- 호출 백분율 이해
- SnapStart에 대한 특별 고려 사항
- 빈도가 빠른 함수에 대한 특별 고려 사항
- Lambda 런타임 API 프록시를 사용하여 여러 확장 구성

- 컨테이너 런타임과 AWS FIS 함께 사용
- AWS FIS Lambda 환경 변수

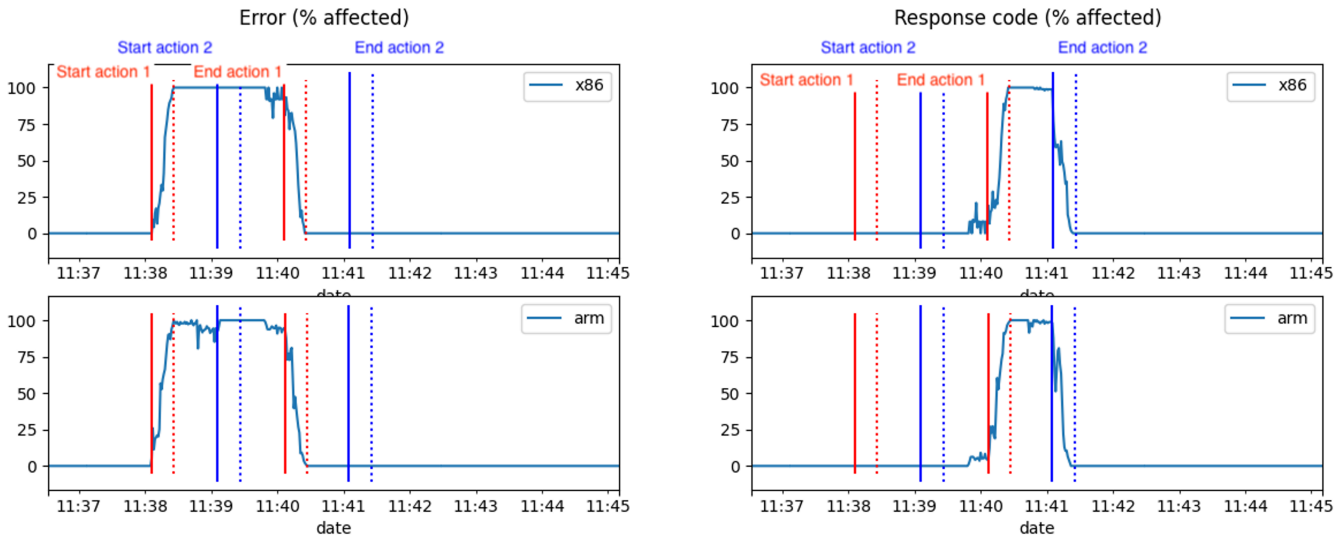
폴링 이해

장애가 모든 호출에 영향을 미치기 전에 최대 60초의 증가 기간을 확인할 수 있습니다. 이는 실험이 시작될 때까지 기다리는 동안 Lambda 확장이 구성 정보를 자주 폴링하지 않기 때문입니다. `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` 환경 변수(기본값 60초)를 설정하여 폴링 간격을 조정할 수 있습니다. 값이 낮을수록 폴링 빈도는 더 높아지지만 성능에 더 큰 영향과 비용이 발생합니다. 또한 오류가 주입된 후 최대 20초의 램프 다운 기간이 표시될 수 있습니다. 이는 실험이 실행되는 동안 확장이 더 자주 폴링되기 때문입니다.

동시성 이해

여러 작업을 동시에 사용하여 동일한 Lambda 함수를 대상으로 지정할 수 있습니다. 작업이 모두 서로 다른 경우 모든 작업이 적용됩니다. 예를 들어 오류를 반환하기 전에 초기 지연을 추가할 수 있습니다. 두 개의 동일하거나 충돌하는 작업이 동일한 함수에 적용되는 경우 시작 날짜가 가장 빠른 작업만 적용됩니다.

아래 그림은 aws:lambda:invocation-error와 aws:lambda:invocation-http-integration-response라는 두 가지 충돌하는 작업을 보여줍니다. 처음에는 aws:lambda:invocation-error가 11:38에 증가하여 2분 동안 실행됩니다. 그런 다음 aws:lambda:invocation-http-integration-response는 11:39에 시작하려고 시도하지만 첫 번째 작업이 완료된 후 11:40까지 적용되지 않습니다. 실험 타이밍을 유지하기 위해 aws:lambda:invocation-http-integration-response는 원래 의도한 시간인 11:41에 여전히 완료됩니다.



호출 백분율 이해

AWS Fault Injection Service Lambda 작업은 하나 이상의 AWS Lambda 함수 ARNs을 선택할 수 있는 aws:lambda:function 대상을 사용합니다. Lambda 작업은 이러한 ARNs을 사용하여 선택한 AWS Fault Injection Service Lambda 함수를 호출할 때마다 오류를 주입할 수 있습니다. 호출의 일부에만 오류를 주입할 수 있도록 각 작업을 통해 값이 0~100인 `invocationPercentage` 파라미터를 지정할 수 있습니다. `invocationPercentage` 파라미터를 사용하면 호출 비율이 100% 미만인 경우에도 작업이 동시에 이루어지도록 할 수 있습니다.

SnapStart에 대한 특별 고려 사항

AWS Lambda SnapStart가 활성화된 함수는 실험이 이미 실행 중인 경우에도 첫 번째 장애 구성을 선택`AWS_FIS_SLOW_POLL_INTERVAL_SECONDS`하기 전에의 전체 기간을 기다릴 가능성이 더 높습니다. 이는 Lambda SnapStart가 단일 스냅샷을 여러 실행 환경의 초기 상태로 사용하고 임시 스토리지를 유지하기 때문입니다. AWS Fault Injection Service Lambda 확장의 경우 폴링 빈도를 유지하고 실행 환경 초기화 시 초기 구성 검사를 건너뜁니다. Lambda SnapStart에 대한 자세한 내용은 사용 설명서의 Lambda SnapStart를 사용한 시작 성능 개선을 참조하세요. AWS Lambda

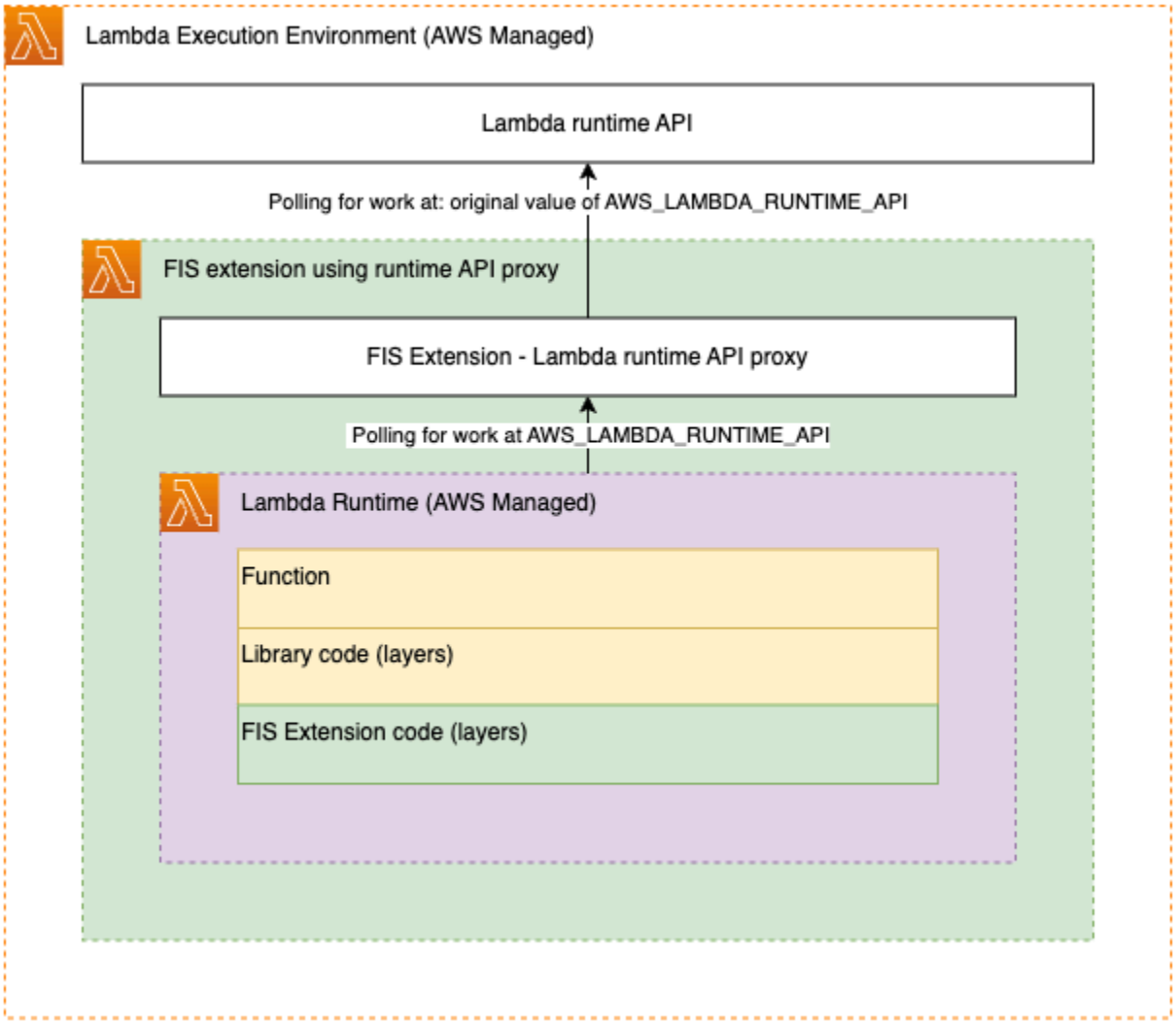
빈도가 빠른 함수에 대한 특별 고려 사항

Lambda 함수가 평균 폴링 기간인 70밀리초 미만으로 실행되는 경우 폴링 스레드는 장애 구성을 얻기 위해 여러 호출이 필요할 수 있습니다. 예를 들어 15분마다 한 번씩 함수가 자주 실행되지 않으면 폴링이 완료되지 않습니다. 폴링 스레드가 완료될 수 있도록 하려면 `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` 파라미터를 설정합니다. 확장은 함수를 시작하기 전에 진행 중인 폴링이 완료될 때까지 설정한 기간까지 기다립니다. 이렇게 하면 청구된 함수 기간이 늘어나고 일부 간접 호출이 추가로 지연됩니다.

Lambda 런타임 API 프록시를 사용하여 여러 확장 구성

Lambda 확장은 AWS Lambda 런타임 API 프록시를 사용하여 함수 호출이 런타임에 도달하기 전에 함수 호출을 가로챌 수 있습니다. 런타임 API의 프록시를 AWS Lambda 런타임에 노출하고 `AWS_LAMBDA_RUNTIME_API` 변수에서 해당 위치를 알림으로써이 작업을 수행합니다.

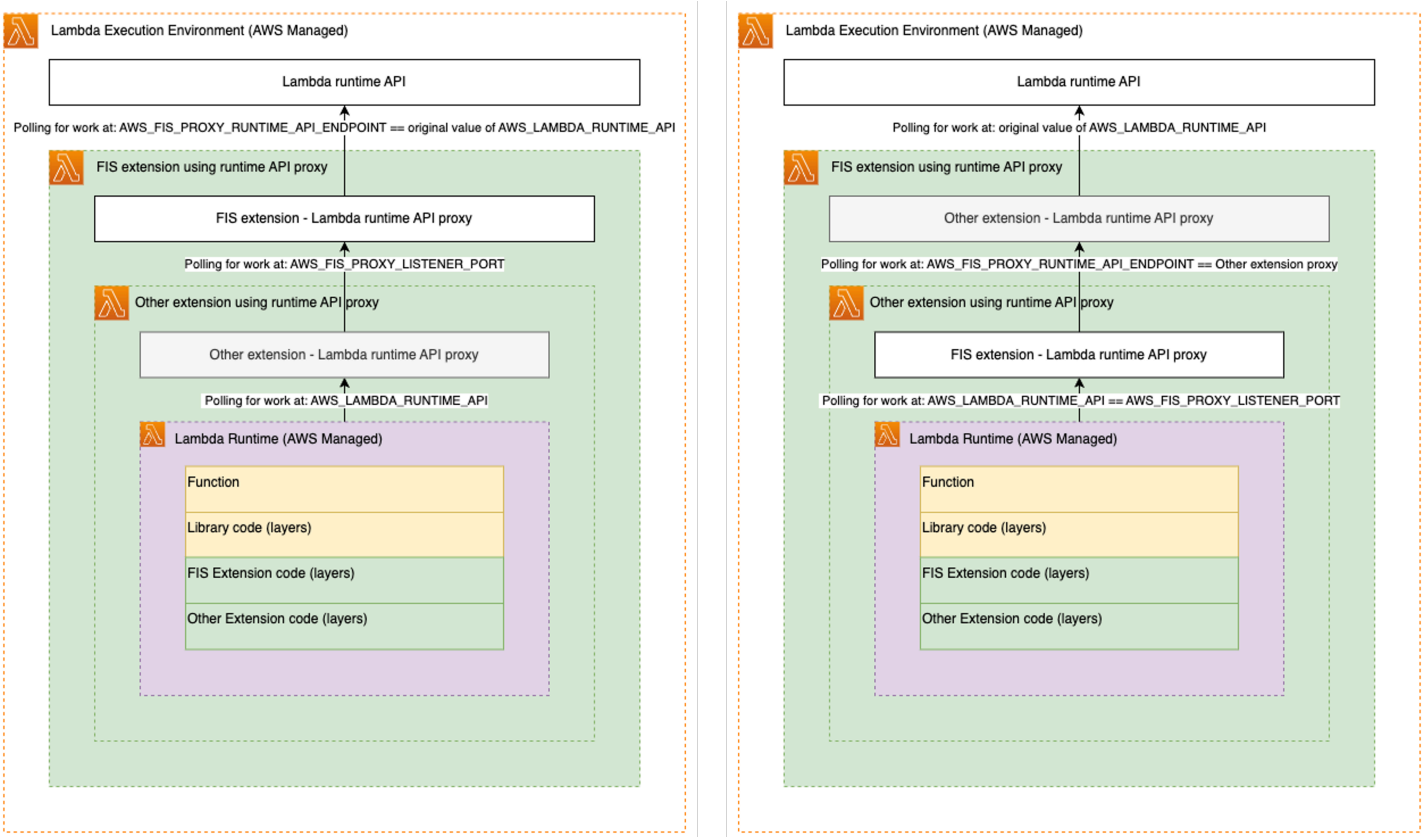
다음 다이어그램은 Lambda 런타임 API 프록시를 사용하는 단일 확장에 대한 구성을 보여줍니다.



AWS Lambda 런타임 API 프록시 패턴을 사용하여 AWS FIS Lambda 확장을 다른 확장과 함께 사용하려면 사용자 지정 부트스트랩 스크립트를 사용하여 프록시를 연결해야 합니다. AWS FIS Lambda 확장은 다음 환경 변수를 허용합니다.

- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT` - AWS Lambda 런타임 API의 로컬 IP 및 리스너 포트를 `127.0.0.1:9876` 나타내는 형식의 문자열을 가져옵니다. 이는의 원래 값 `AWS_LAMBDA_RUNTIME_API` 또는 다른 프록시의 위치일 수 있습니다.
- `AWS_FIS_PROXY_LISTENER_PORT` - 기본적으로 AWS FIS 확장이 자체 프록시를 시작해야 하는 포트 번호를 가져옵니다`9100`.

이러한 설정을 사용하면 Lambda 런타임 API 프록시를 사용하여 두 가지 순서로 AWS FIS 확장을 다른 확장과 연결할 수 있습니다.



AWS Lambda 런타임 API 프록시에 대한 자세한 내용은 사용 AWS Lambda 설명서의 AWS Lambda 런타임 API 프록시 확장을 사용하여 런타임 보안 및 거버넌스 강화 및 사용자 지정 런타임에 Lambda 런타임 API 사용을 참조하세요.

컨테이너 런타임과 AWS FIS 함께 사용

`AWS_LAMBDA_RUNTIME_API` 환경 변수를 허용하는 컨테이너 이미지를 사용하는 AWS Lambda 함수의 경우 아래 단계에 따라 AWS FIS Lambda 확장을 컨테이너 이미지에 패키징할 수 있습니다.

1. 확장을 추출할 계층의 ARN을 결정합니다. ARN을 찾는 방법에 대한 자세한 내용은 섹션을 참조하세요Lambda 함수 구성.
2. AWS Command Line Interface (CLI)를 사용하여 확장에 대한 세부 정보를 요청합니다`aws lambda get-layer-version-by-arn --arn fis-extension-arn`. 응답에는 FIS 확장을 ZIP 파일로 다운로드할 수 있는 미리 서명된 URL이 포함된 `Location` 필드가 포함됩니다.
3. 확장 프로그램의 내용을 Docker 파일 시스템의 `/opt`에 압축을 풉니다. 다음은 NodeJS Lambda 런타임을 기반으로 하는 Dockerfile의 예입니다.

```
# 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" ]
```

컨테이너 이미지에 대한 자세한 내용은 AWS Lambda 사용 설명서의 컨테이너 이미지를 사용하여 Lambda 함수 생성을 참조하세요.

AWS FIS Lambda 환경 변수

다음은 AWS FIS Lambda 확장의 환경 변수 목록입니다.

- `AWS_FIS_CONFIGURATION_LOCATION` - 필수 항목입니다. AWS FIS 가 활성 장애 구성을 작성하고 확장이 장애 구성을 읽는 위치입니다. 위치는 버킷 및 경로를 포함한 Amazon S3 ARN 형식이어야 합니다. 예를 들어 `arn:aws:s3:::my-fis-config-bucket/FisConfigs/`입니다.
- `AWS_LAMBDA_EXEC_WRAPPER` - 필수. AWS FIS Lambda 확장을 구성하는 데 사용되는 AWS Lambda 래퍼 스크립트의 위치입니다. 확장에 포함된 `/opt/aws-fis/bootstrap` 스크립트로 설정해야 합니다.
- `AWS_FIS_LOG_LEVEL` - 선택 사항입니다. AWS FIS Lambda 확장에서 내보낸 메시지의 로그 수준입니다. 지원되는 값은 `INFO`, `WARN` 및 `ERROR`입니다. 설정하지 않으면 AWS FIS 확장은 기본적으로 로 설정됩니다`INFO`.
- `AWS_FIS_EXTENSION_METRICS` - 선택 사항입니다. 가능한 값은 `all` 및 none입니다. 확장으로 설정하면 `all`에서 EMF 지표를 내보냅니다`aws-fis-extension namespace`.
- `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` - 선택 사항입니다. 이 옵션을 설정하면 확장이 오류를 주입하지 않고 구성 위치에 오류 구성이 추가될 때까지 기다리는 동안 폴링 간격(초)이 재정의됩니다. 기본값은 `60`입니다.
- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT` - 선택 사항입니다. 설정하면 값이 재정의`AWS_LAMBDA_RUNTIME_API`되어 AWS FIS 확장이 AWS Lambda 런타임 API와 상호 작용하

여 함수 호출을 제어하는 위치를 정의합니다. 와 같은 IP:PORT가 필요합니다`127.0.0.1:9000`. 에 대한 자세한 내용은 사용 설명서의 사용자 지정 런타임에 Lambda 런타임 API 사용을 `AWS_LAMBDA_RUNTIME_API`참조하세요. https://docs.aws.amazon.com/lambda/latest/dg/runtimes-api.html AWS Lambda

- `AWS_FIS_PROXY_LISTENER_PORT` - 선택 사항입니다. AWS FIS Lambda 확장이 다른 확장 또는 런타임에서 사용할 수 있는 AWS Lambda 런타임 API 프록시를 노출하는 포트를 정의합니다. 기본값은 `9100`입니다.
- `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` - 선택 사항입니다. `0`이 아닌 값으로 설정된 경우이 변수는 장애 구성을 평가하고 런타임 호출을 시작하기 전에 확장이 진행 중인 비동기 폴링이 완료될 때까지 대기하는 밀리초 수를 정의합니다. 기본값은 `0`입니다.

Lambda용 AWS FIS 확장의 사용 가능한 버전

이 섹션에는 AWS FIS Lambda 확장 버전에 대한 정보가 포함되어 있습니다. 확장은 x86-64 및 ARM64(Graviton2) 플랫폼용으로 개발된 Lambda 함수를 지원합니다. Lambda 함수는 현재 호스팅 AWS 리전 되는에 대해 특정 Amazon 리소스 이름(ARN)을 사용하도록 구성되어야 합니다. 아래에서 AWS 리전 및 ARN 세부 정보를 볼 수 있습니다.

주제

- AWS FIS Lambda 확장 릴리스 정보
- Lambda 확장 ARNs에 대한 액세스 가이드
- Lambda 확장 버전 번호 찾기

AWS FIS Lambda 확장 릴리스 정보

다음 표에서는 AWS FIS Lambda 확장의 최신 버전에 대한 변경 사항을 설명합니다.

버전	시작 날짜	참고
1.0.0	2024-10-29	초기 릴리스

Lambda 확장 ARNs에 대한 액세스 가이드

콘솔을 사용하여 퍼블릭 파라미터를 검색하려면 AWS 계정 및 AWS 리전 에 파라미터가 하나 이상 있어야 합니다. 퍼블릭 파라미터를 검색하려면 파라미터 스토어에서 퍼블릭 파라미터 검색을 참조하세요.

콘솔 액세스:

1. https://console.aws.amazon.com/systems-manager/ AWS Systems Manager 콘솔을 엽니다.
2. 탐색 창에서 파라미터 스토어를 선택합니다.
3. [퍼블릭 파라미터(Public parameters)] 탭을 선택합니다.
4. [서비스 선택(Select a service)] 드롭다운을 선택합니다. 드롭다운 옵션에서를 선택합니다`fis`.
5. (선택 사항) 검색 창에 자세한 정보를 입력하여 선택한 파라미터를 필터링합니다. arm64 아키텍처의 경우 "arm64"를 입력하여 파라미터를 필터링합니다. x86_64 아키텍처의 경우 "x86_64"를 입력하여 파라미터를 필터링합니다.
6. 사용할 퍼블릭 파라미터를 선택합니다.
7. 파라미터 세부 정보에서 ARN 값을 찾습니다. 대상 Lambda 함수에서 계층 확장을 구성하는 데 사용할 ARN을 복사합니다.

AWS CLI 액세스:

SSM 파라미터 이름

다양한 아키텍처에 사용할 수 있는 SSM 파라미터 이름은 다음과 같습니다.

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 명령 형식

확장 ARNs을 검색하려면 다음 AWS CLI 명령 형식을 사용합니다. 여기서 parameterName은 아키텍처의 이름이고 region은 대상입니다. AWS 리전

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

사용 예

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

응답 형식

명령은 다음과 같은 파라미터 세부 정보가 포함된 JSON 객체를 반환합니다. Lambda 계층의 ARN은 파라미터 객체의 값 필드에 포함됩니다. 대상 Lambda 함수에서 계층 확장을 구성하는 데 사용할 ARN을 복사합니다.

```
 {
     "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"
        }
 }
```

프로그래밍 방식 액세스:

코드형 인프라(IaC)를 사용하여 Lambda 함수를 빌드하거나 구성할 때 프로그래밍 방식으로 이러한 퍼블릭 파라미터를 검색합니다. 이 접근 방식은 AWS FIS 확장 계층 ARN이 하드코딩된 경우 필요한 수동 코드 업데이트 없이 최신 계층 버전 ARN으로 Lambda 함수를 유지하는 데 도움이 됩니다. 다음 리소스는 공통 IaC 플랫폼을 사용하여 퍼블릭 파라미터를 검색하는 방법을 보여줍니다.

- AWS SDK를 사용하여 퍼블릭 파라미터 가져오기
- AWS CDK를 사용하여 AWS Systems Manager Parameter Store에서 퍼블릭 파라미터 가져오기
- Terraform을 사용하여 퍼블릭 파라미터 가져오기

Lambda 확장 버전 번호 찾기

다음 절차에 따라 현재 구성된 AWS FIS Lambda 확장의 버전 번호를 찾습니다.

1. https://console.aws.amazon.com/lambda/ AWS Lambda 콘솔을 엽니다.
2. `AWS-FIS-Extension` 계층을 추가하려는 Lambda 함수를 선택합니다.
3. 계층 섹션에서 편집을 선택합니다.
4. 계층 편집 섹션에서 계층 추가를 선택합니다.
5. 계층 선택 섹션에서 ARN 지정을 선택합니다.
6. AWS 리전 및 아키텍처에 해당하는 AWS FIS 확장 계층의 ARN을 입력합니다. 콘솔 AWS CLI또는 이전 섹션에 설명된 프로그래밍 방식 액세스 방법을 사용하여 ARN을 찾을 수 있습니다.
7. 확인을 선택하여 계층 ARN이 유효한지 확인한 다음 추가를 선택합니다.
8. 테스트를 사용해 함수를 테스트합니다.
9. 테스트가 완료되면 로그 출력을 확인하십시오. 실행 세부 정보 섹션에서 AWS FIS Lambda 확장 버전을 찾습니다.

AWS FIS 실험 템플릿 관리

AWS FIS 콘솔 또는 명령줄을 사용하여 실험 템플릿을 생성하고 관리할 수 있습니다. 실험 템플릿에는 실험 중에 지정된 대상에서 실행할 수 있는 하나 이상의 작업이 포함되어 있습니다. 또한 실험이 범위를 벗어나는 것을 방지하는 중지 조건도 포함되어 있습니다. 실험 템플릿의 구성 요소에 대한 자세한 내용은 실험 템플릿 구성 요소 섹션을 참조하세요. 실험 템플릿을 만든 후 이를 사용하여 실험을 실행할 수 있습니다.

작업

- 실험 템플릿 만들기
- 실험 템플릿 보기
- 실험 템플릿에서 대상 미리 보기 생성
- 템플릿에서 실험 시작
- 실험 템플릿 업데이트
- 실험 템플릿에 태그 지정
- 실험 템플릿 만들기
- AWS FIS 실험 템플릿 예제

실험 템플릿 만들기

시작하기 전에 다음 작업을 완료하세요.

- 실험 계획 세우기.
- 사용자를 대신하여 작업을 수행할 수 있는 권한을 AWS FIS 서비스에 부여하는 IAM 역할을 생성합니다. 자세한 내용은 AWS FIS 실험을 위한 IAM 역할 단원을 참조하십시오.
- AWS FIS에 액세스할 수 있는지 확인합니다. 자세한 내용을 알아보려면 AWS FIS 정책 예제를 참조하세요.

콘솔을 사용하여 실험 템플릿을 생성하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿 생성을 선택합니다.

4. 1단계, 템플릿 세부 정보 지정에서 다음을 수행합니다.

   a. 설명 및 이름에와 같은 템플릿에 대한 설명을 입력합니다`Amazon S3 Network Disrupt Connectivity`.

   b. (선택 사항) 계정 타겟팅의 경우 여러 계정을 선택하여 다중 계정 실험 템플릿을 구성합니다.

   c. 다음을 선택하고 2단계, 작업 및 대상 지정으로 이동합니다.

5. 작업 경우 템플릿의 작업 세트를 지정합니다. 각 작업에 대해 작업 추가를 선택하고 다음을 완료합니다.

   - 이름에서 작업의 이름을 입력합니다.

     허용되는 문자는 영숫자, 하이픈(-), 밑줄(_)입니다. 이름은 문자로 시작해야 합니다. 공백은 사용할 수 없습니다. 각 작업 이름은 이 템플릿에서 고유해야 합니다.

   - (선택 사항) 설명에, 작업에 대한 설명을 입력합니다. 최대 길이는 512자입니다.
   - (선택 사항) 시작하기 전에서, 현재 작업이 시작되기 전에 완료되어야 하는 이 템플릿에 정의된 다른 작업을 선택합니다. 그렇지 않으면 실험이 시작될 때 이 작업이 실행됩니다.
   - 작업 유형에서 AWS FIS 작업을 선택합니다.
   - 대상의 경우 대상 섹션에서 정의한 대상을 선택합니다. 이 작업에 대한 대상을 아직 정의하지 않은 경우 AWS FIS가 새 대상을 생성합니다.
   - 작업 파라미터에는 작업의 파라미터를 지정합니다. 이 섹션은 AWS FIS 작업에 파라미터가 있는 경우에만 나타납니다.
   - 저장을 선택합니다.

6. 대상의 경우 작업을 수행할 대상 리소스를 정의합니다. 대상으로 하나 이상의 리소스 ID 또는 하나의 리소스 태그를 지정해야 합니다. 편집을 선택하여 AWS FIS가 이전 단계에서 생성한 대상을 편집하거나 대상 추가를 선택합니다. 각 대상에서 다음을 수행합니다.

   - 이름에 대상의 이름을 입력합니다.

     허용되는 문자는 영숫자, 하이픈(-), 밑줄(_)입니다. 이름은 문자로 시작해야 합니다. 공백은 사용할 수 없습니다. 각 대상 이름은 이 템플릿에서 고유해야 합니다.

   - 리소스 유형에서는 작업에 지원되는 리소스 유형을 선택합니다.
   - 대상 메서드에서 다음 중 하나를 수행합니다.
     - 리소스 ID를 선택한 다음 리소스 ID를 선택하거나 추가합니다.
     - 리소스 태그, 필터, 파라미터를 선택한 다음 필요한 태그와 필터를 추가합니다. 자세한 내용은 the section called “대상 리소스 식별” 단원을 참조하십시오.

- 선택 모드에서는 개수를 선택하여 지정된 수의 식별된 대상에 대해 작업을 실행하거나 백분율을 선택하여 지정된 비율의 식별된 대상에 대해 작업을 실행합니다. 기본적으로 작업은 식별된 모든 대상에서 실행됩니다.
- 저장을 선택합니다.

7. 생성한 대상으로 작업을 업데이트하려면 작업 아래에서 작업을 찾아 편집을 선택한 다음 대상을 업데이트하세요. 실험 템플릿에서 동일한 대상을 여러 작업에 사용할 수 있습니다.
8. (선택 사항) 실험 옵션에서 빈 대상 해상도 모드의 동작을 선택합니다.
9. 다음을 선택하여 3단계, 서비스 액세스 구성으로 이동합니다.
10. 서비스 액세스의 경우 기존 IAM 역할 사용을 선택한 다음 이 자습서의 사전 조건에 설명된 대로 생성한 IAM 역할을 선택합니다. 역할이 표시되지 않는 경우 해당 역할에 필요한 신뢰 관계가 있는지 확인하세요. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.
11. (다중 계정 실험만 해당) 대상 계정 구성의 경우 각 대상 계정에 대해 역할 ARN과 선택적 설명을 추가합니다. 대상 계정 역할 ARN을 CSV 파일로 업로드하려면 모든 대상 계정에 대한 역할 ARN 업로드를 선택한 다음 .CSV 파일 선택을 선택합니다.
12. 다음을 선택하여 4단계, 선택적 설정 구성으로 이동합니다.
13. (선택 사항) 중지 조건의 경우 중지 조건에 대한 Amazon CloudWatch 경보를 선택합니다. 자세한 내용은 AWS FIS에 대한 중지 조건 단원을 참조하십시오.
14. (선택 사항) 로그의 경우 대상 옵션을 구성합니다. S3 버킷으로 로그를 보내려면 Amazon S3 버킷으로 전송을 선택하고 버킷 이름과 접두사를 입력합니다. 로그를 CloudWatch Logs로 보내려면 CloudWatch Logs로 전송 선택하고 로그 그룹을 입력합니다.
15. (선택 사항) 태그의 경우 새 태그 추가를 선택하고 태그 키와 태그 값을 지정합니다. 추가한 태그는 템플릿을 사용하여 실행되는 실험이 아니라 실험 템플릿에 적용됩니다.
16. 다음을 선택하여 5단계, 검토 및 생성으로 이동합니다.
17. 템플릿을 검토하고 실험 템플릿 생성을 선택합니다. 확인 메시지가 표시되면를 입력한 `create`다음 실험 템플릿 생성을 선택합니다.

CLI를 사용하여 실험 템플릿을 만들려면

create-experiment-template 명령을 사용합니다.

JSON 파일에서 실험 템플릿을 로드할 수 있습니다.

`--cli-input-json` 파라미터를 사용합니다.

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

자세한 내용은 AWS Command Line Interface 사용 설명서의 CLI 스켈레톤 템플릿 생성을 참조하세요. 템플릿 예시를 보려면 AWS FIS 실험 템플릿 예제 단원을 참조하세요.

실험 템플릿 보기

생성한 실험 템플릿을 볼 수 있습니다.

콘솔을 사용하여 실험 템플릿을 보려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 특정 템플릿에 대한 정보를 보려면 실험 템플릿 ID를 선택합니다.
4. 세부 정보 섹션에서 템플릿의 설명 및 중지 조건을 볼 수 있습니다.
5. 실험 템플릿의 작업을 보려면 작업을 선택합니다.
6. 실험 템플릿의 대상을 보려면 대상을 선택합니다.
7. 실험 템플릿의 태그를 보려면 태그를 선택합니다.

CLI를 사용하여 실험 템플릿을 보려면

list-experiment-templates 명령을 사용하여 실험 템플릿 목록을 가져오고 get-experiment-template 명령을 사용하여 특정 실험 템플릿에 대한 정보를 가져올 수 있습니다.

실험 템플릿에서 대상 미리 보기 생성

실험을 시작하기 전에 대상 미리 보기를 생성하여 실험 템플릿이 예상 리소스를 대상으로 구성되었는지 확인할 수 있습니다. 실제 실험을 시작할 때 대상으로 지정되는 리소스는 미리 보기의 리소스와 다를 수 있습니다. 리소스가 임의로 제거, 업데이트 또는 샘플링될 수 있기 때문입니다. 대상 미리보기를 생성하면 모든 작업을 건너뛰는 실험이 시작됩니다.

> **Note**
>
> 대상 미리 보기를 생성하는 경우 리소스에 대한 작업을 수행하는 데 필요한 권한이 있는지 확인할 수 없습니다.

콘솔을 사용하여 대상 미리 보기를 시작하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿의 대상을 보려면 대상을 선택합니다.
4. 실험 템플릿의 대상 리소스를 확인하려면 미리 보기 생성을 선택합니다. 실험을 실행하면 이 대상 미리 보기가 최근 실험의 대상으로 자동으로 업데이트됩니다.

CLI를 사용하여 대상 미리 보기를 시작하려면

- 다음 start-experiment 명령을 실행합니다. 이탤릭체로 표시된 값을 고유한 값으로 바꿉니다.

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

템플릿에서 실험 시작

실험 템플릿을 만든 후 해당 템플릿을 사용하여 실험을 시작할 수 있습니다.

실험을 시작하면 지정된 템플릿의 스냅샷을 만들고 이 스냅샷을 사용하여 실험을 실행합니다. 따라서 실험이 실행되는 동안 실험 템플릿이 업데이트되거나 삭제되더라도 해당 변경 사항은 진행 중인 실험에 영향을 주지 않습니다.

실험을 시작하면 AWS FIS가 사용자를 대신하여 서비스 연결 역할을 생성합니다. 자세한 내용은 AWS Fault Injection Service에 서비스 연결 역할 사용 단원을 참조하십시오.

실험을 시작한 후에는 언제든지 중지할 수 있습니다. 자세한 내용은 실험 중지 단원을 참조하십시오.

콘솔을 사용하여 실험을 시작하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 실험 시작을 선택합니다.
4. (선택 사항) 실험에 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
5. 실험 시작을 선택합니다. 확인 메시지가 나타나면 **start**을 입력하고 실험 시작을 선택합니다.

CLI를 사용하여 실험을 시작하려면

start-experiment 명령을 사용합니다.

실험 템플릿 업데이트

기존 실험 템플릿을 업데이트할 수 있습니다. 실험 템플릿을 업데이트해도 해당 템플릿을 사용하는 실행 중인 실험에는 변경 내용이 영향을 주지 않습니다.

콘솔을 사용하여 실험 템플릿을 업데이트하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 업데이트를 선택합니다.
4. 필요에 따라 템플릿 세부 정보를 수정하고 실험 템플릿 업데이트를 선택합니다.

CLI를 사용하여 실험 템플릿을 업데이트하려면

update-experiment-template 명령을 사용합니다.

실험 템플릿에 태그 지정

실험에 태그를 적용하여 실험 템플릿을 체계적으로 구성할 수 있습니다. 또한 태그 기반 IAM 정책을 구현하여 실험 템플릿에 대한 액세스를 제어할 수 있습니다.

콘솔을 사용하여 실험 템플릿에 태그를 지정하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 태그 관리를 선택합니다.
4. 새 태그를 추가하려면 새 태그 추가를 선택하고 키와 값을 지정합니다.

   태그를 제거하려면 해당 태그에서 제거를 선택합니다.
5. 저장을 선택합니다.

CLI를 사용하여 실험 템플릿에 태그를 지정하려면

tag-resource 명령을 사용합니다.

실험 템플릿 만들기

실험 템플릿이 더 이상 필요하지 않으면 삭제할 수 있습니다. 실험 템플릿을 삭제해도 해당 템플릿을 사용하는 실행 중인 실험은 변경 내용의 영향을 받지 않습니다. 실험은 완료되거나 중지될 때까지 계속 실행됩니다. 하지만 삭제된 실험 템플릿은 콘솔의 실험 페이지에서 볼 수 없습니다.

콘솔을 사용하여 실험 템플릿을 삭제하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **delete**를 입력하고 실험 템플릿 삭제를 선택합니다.

CLI를 사용하여 실험 템플릿을 삭제하려면

delete-experiment-template 명령을 사용합니다.

AWS FIS 실험 템플릿 예제

AWS FIS API 또는 명령줄 도구를 사용하여 실험 템플릿을 생성하는 경우 JavaScript Object Notation(JSON)으로 템플릿을 구성할 수 있습니다. 실험 템플릿의 구성 요소에 대한 자세한 내용은 AWS FIS 실험 템플릿 구성 요소 단원을 참조하세요.

예제 템플릿 중 하나를 사용하여 실험을 만들려면 JSON 파일(예: `my-template.json`)에 저장하고 `###`로 표시된 자리 표시자 값을 원하는 값으로 바꾼 후 다음 create-experiment-template 명령어를 실행합니다.

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

템플릿 예제

- 필터를 기반으로 EC2 인스턴스 중지
- 지정된 수의 EC2 인스턴스 중지
- 사전 구성된 AWS FIS SSM 문서 실행
- 사전 정의된 자동화 런북 실행
- 대상 IAM 역할을 사용하여 EC2 인스턴스에서의 API 작업을 제한하세요.
- Kubernetes 클러스터 내 포드의 CPU 스트레스 테스트

- 지정된 수의 Kinesis Data Streams에 대한 프로비저닝된 처리량 예외
- 실험 역할 권한 예제

필터를 기반으로 EC2 인스턴스 중지

다음 예제에서는 지정된 리전의 모든 실행 중인 VPC에서 지정된 태그를 가진 모든 Amazon EC2 인스턴스를 중지합니다. 2분 후에 인스턴스가 다시 시작됩니다.

```
{
    "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"
}
```

지정된 수의 EC2 인스턴스 중지

다음 예시에서는 지정된 태그가 있는 인스턴스 3개를 중지합니다. AWS FIS는 임의로 중지할 특정 인스턴스를 선택합니다. 2분 후에 이러한 인스턴스를 다시 시작합니다.

```
{
    "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"
}
```

사전 구성된 AWS FIS SSM 문서 실행

다음 예제에서는 미리 구성된 AWS FIS SSM 문서를 사용하여 지정된 EC2 인스턴스에서 60초 동안 CPU 오류 주입을 실행합니다. AWSFIS-Run-CPU-Stress. AWS FIS는 2분 동안 실험을 모니터링합니다.

```
{
    "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"
}
```

사전 정의된 자동화 런북 실행

다음 예제는 Systems Manager에서 제공하는 런북인 AWS-PublishSNSNotification을 사용하여 Amazon SNS에 알림을 게시합니다. 역할에는 지정된 SNS 주제에 알림을 게시할 권한이 있어야 합니다.

```
{
    "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"
}
```

대상 IAM 역할을 사용하여 EC2 인스턴스에서의 API 작업을 제한하세요.

다음 예제에서는 대상 정의에 지정된 IAM 역할에 의해 수행된 API 직접 호출에 대해 작업 정의에 지정된 API 호출의 100%를 스로틀링합니다.

> **Note**
>
> Auto Scaling 그룹의 멤버인 EC2 인스턴스를 대상으로 지정하려면 aws:ec2:asg-insufficient-instance-capacity-error 작업을 사용하고 Auto Scaling 그룹별로 대상을 지정합니다. 자세한 내용은 aws:ec2:asg-insufficient-instance-capacity-error 단원을 참조하십시오.

```
{
    "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 클러스터 내 포드의 CPU 스트레스 테스트

다음 예시에서는 Chaos Mesh를 사용하여 Amazon EKS Kubernetes 클러스터의 포드 CPU에 1분 동안 스트레스 테스트를 실시합니다.

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

다음 예시에서는 Litmus를 사용하여 Amazon EKS Kubernetes 클러스터의 포드 CPU에 1분 동안 스트레스 테스트를 실시합니다.

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

지정된 수의 Kinesis Data Streams에 대한 프로비저닝된 처리량 예외

다음 예제에서는 지정된 태그가 있는 최대 5개의 Kinesis Data Streams 요청의 100%에 대해 프로비저닝된 처리량 예외가 발생합니다. AWS FIS는 임의로 영향을 미칠 스트림을 선택합니다. 5분 후 오류가 제거됩니다.

```
{
    "description": "Kinesis stream experiment",
    "targets": {
        "KinesisStreams-Target-1": {
            "resourceType": "aws:kinesis:stream",
            "resourceTags": {
                   "tag-key": "tag-value"
            },
            "selectionMode": "COUNT(5)"
        }
    },
    "actions": {
         "kinesis": {
              "actionId": "aws:kinesis:stream-provisioned-throughput-exception",
              "description": "my-stream",
              "parameters": {
                   "duration": "PT5M",
                   "percentage": "100",
                   "service": "kinesis"
              },
              "targets": {
                    "KinesisStreams": "KinesisStreams-Target-1"
              }
         }
   },
   "stopConditions": [
         {
              "source": "none"
         }
   ],
```

```
    "roleArn": "arn:aws:iam::111122223333:role/role-name",
    "tags": {},
    "experimentOptions": {
        "accountTargeting": "single-account",
        "emptyTargetResolutionMode": "fail"
    }
}
```

실험 역할 권한 예제

다음 권한을 사용하면 요청의 50%에 영향을 미치는 특정 스트림에서 aws:kinesis:stream-provisioned-throughput-exception 및 aws:kinesis:stream-expired-iterator-exception 작업을 실행할 수 있습니다.

```
{
    "Version": "2012-10-17",
     "Statement": [
        {
            "Effect": "Allow",
            "Action": "kinesis:InjectApiError",
            "Resource": "*"
            "Condition": {
                "ForAllValues:StringEquals": {
                    "kinesis:FisActionId": [
                        "aws:kinesis:stream-provisioned-throughput-exception",
                        "aws:kinesis:stream-expired-iterator-exception"
                    ],
                    "kinesis:FisTargetArns": [
                        "arn:aws:kinesis:us-east-1:111122223333:stream/stream-name"
                    ],
                },
                "NumericEquals": {
                    "kinesis:FisInjectPercentage": "50"
                }
            }
        },
        {
             "Action": [
                  "kinesis:DescribeStreamSummary",
              ],
             "Resource": "*",
             "Effect": "Allow"
        }
```

```
    ]
}
```

AWS FIS 실험 관리

AWS FIS를 사용하면 AWS 워크로드에서 결함 주입 실험을 수행할 수 있습니다. 시작하려면 실험 템플릿을 만드세요. 실험 템플릿을 만든 후 이를 사용하여 실험을 시작할 수 있습니다.

다음 중 한 경우에 실험이 끝납니다.

- 템플릿의 모든 작업이 성공적으로 완료되었습니다.
- 중지 조건이 트리거됩니다.
- 오류로 인해 작업을 완료할 수 없습니다. 대상을 찾을 수 없는 경우를 예로 들 수 있습니다.
- 실험은 수동으로 중지됩니다.

중지되거나 실패한 실험은 재개할 수 없습니다. 또한 완료된 실험은 다시 실행할 수 없습니다. 하지만 동일한 실험 템플릿에서 새 실험을 시작할 수 있습니다. 새 실험에서 다시 지정하기 전에 실험 템플릿을 선택적으로 업데이트할 수도 있습니다.

작업

- 실험 시작
- 실험 보기
- 실험에 태그 지정
- 실험 중지
- 확인된 대상 나열

실험 시작

실험 템플릿에서 실험을 시작합니다. 자세한 내용은 템플릿에서 실험 시작 단원을 참조하십시오.

Amazon EventBridge를 사용하여 실험을 일회성 작업 또는 반복 작업으로 예약할 수 있습니다. 자세한 내용은 자습서: 반복 실험 예약 단원을 참조하십시오.

다음 기능 중 하나를 사용하여 실험을 모니터링할 수 있습니다.

- AWS FIS 콘솔에서 실험을 봅니다. 자세한 내용은 실험 보기 단원을 참조하십시오.
- 실험의 대상 리소스에 대한 Amazon CloudWatch 지표 또는 AWS FIS 사용 지표를 확인하세요. 자세한 내용은 CloudWatch를 사용한 모니터링 단원을 참조하십시오.

- 실험 로깅을 활성화하여 실행 중인 실험에 대한 세부 정보를 캡처할 수 있습니다. 자세한 내용은 실험 로깅을 참조하세요.

실험 보기

실행 중인 실험의 진행 상황을 볼 수 있으며 완료, 중지 또는 실패한 실험을 볼 수 있습니다.

중지, 완료, 실패한 실험은 120일 후 계정에서 자동으로 제거됩니다.

콘솔을 사용하여 실험을 보려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험을 선택합니다.
3. 실험의 실험 ID를 선택하여 세부 정보 페이지를 엽니다.
4. 다음 중 한 개 이상을 수행할 수 있습니다.
   - 세부 정보, 상태에서 실험 상태를 확인합니다.
   - 실험 작업에 대한 정보를 보려면 작업 탭을 선택합니다.
   - 실험 대상 대한 정보를 보려면 대상 탭을 선택합니다.
   - 타임라인 탭을 선택하면 시작 및 종료 시간을 기준으로 작업을 시각적으로 표현할 수 있습니다.

콘솔을 사용하여 실험을 보려면

list-experiments 명령을 사용하여 실험 목록을 가져오고 get-experiment 명령을 사용하여 특정 실험에 대한 정보를 가져올 수 있습니다.

실험 상태

실험은 다음 상태 중 하나일 수 있습니다.

- 보류 중 - 실험이 보류 중입니다.
- 시작 중 - 실험 시작 준비 중입니다.
- 실행 중 - 실험이 진행 중입니다.
- 완료 - 실험의 모든 작업이 성공적으로 완료되었습니다.
- 중지 중 - 중지 조건이 트리거되었거나 실험이 수동으로 중지되었습니다.
- 중지됨 - 실험에서 실행 중이거나 보류 중인 모든 작업이 중지됩니다.

- 실패 - 권한이 충분하지 않거나 구문이 잘못되는 등의 오류로 인해 실험이 실패했습니다.
- 취소됨 - 안전 레버가 동작하여 실험이 중지되었거나 시작되지 않았습니다.

작업 상태

작업은 다음 상태 중 하나일 수 있습니다.

- 보류 중 - 실험이 아직 시작되지 않았거나 실험 후반부에 작업을 시작해야 하기 때문에 작업이 보류 중입니다.
- 시작 중 - 작업 시작 준비 중입니다.
- 실행 중 - 작업이 실행 중입니다.
- 완료됨 - 작업이 성공적으로 완료되었습니다.
- 취소됨 - 작업이 시작되기 전에 실험이 중지되었습니다.
- 건너뜀 - 작업을 건너뛰었습니다.
- 중지 중 - 작업을 중지하는 중입니다.
- 중지됨 - 실험에서 실행 중이거나 보류 중인 모든 작업이 중지됩니다.
- 실패 - 권한이 충분하지 않거나 구문이 잘못되는 등의 클라이언트 오류로 인해 작업이 실패했습니다.

실험에 태그 지정

실험에 태그를 적용하여 실험을 구성하는 데 도움을 줄 수 있습니다. 또한 태그 기반 IAM 정책을 구현하여 실험에 대한 액세스를 제어할 수 있습니다.

콘솔을 사용하여 실험에 태그를 지정하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험을 선택합니다.
3. 실험을 선택한 다음 작업, 태그 관리를 선택합니다.
4. 새 태그를 추가하려면 새 태그 추가를 선택하고 키와 값을 지정합니다.

   태그를 제거하려면 해당 태그에서 제거를 선택합니다.
5. 저장을 선택합니다.

CLI를 사용하여 실험에 태그를 지정하려면

tag-resource 명령을 사용합니다.

실험 중지

실행 중인 실험을 언제든지 중지할 수 있습니다. 실험을 중지하면 해당 작업에 대해 완료되지 않은 모든 게시 작업은 실험이 중지되기 전에 완료됩니다. 중지된 실험은 재개할 수 없습니다.

콘솔을 사용하여 실험을 중지하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험을 선택합니다.
3. 실험을 선택하고 실험 중지를 선택합니다.
4. 확인 대화 상자에서 실험 중지를 선택합니다.

CLI를 사용하여 실험을 중지하려면

stop-experiment 명령을 사용합니다.

확인된 대상 나열

대상 확인이 종료된 후 실험의 확인된 대상에 대한 정보를 볼 수 있습니다.

콘솔을 사용하여 확인된 대상을 보려면 다음을 수행하세요.

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험을 선택합니다.
3. 실험을 선택하고 보고서를 선택합니다.
4. 리소스에서 확인된 대상 정보를 봅니다.

CLI를 사용하여 확인된 대상을 보려면 다음을 수행하세요.

list-experiment-resolved-targets 명령을 사용합니다.

AWS Fault Injection Service 자습서

다음 자습서에서는 AWS Fault Injection Service(AWS FIS)를 사용하여 실험을 생성하고 실행하는 방법을 보여줍니다.

자습서

- 자습서: AWS FIS를 사용하여 인스턴스 중지 및 시작 테스트
- 자습서: AWS FIS를 사용하여 인스턴스에서 CPU 스트레스 실행
- 자습서: AWS FIS를 사용하여 스팟 인스턴스 중단 테스트
- 자습서: 연결 이벤트 시뮬레이션
- 자습서: 반복 실험 예약

자습서: AWS FIS를 사용하여 인스턴스 중지 및 시작 테스트

AWS Fault Injection Service(AWS FIS)를 사용하여 애플리케이션이 인스턴스 중지 및 시작을 처리하는 방법을 테스트할 수 있습니다. 이 자습서를 사용하여 AWS FIS `aws:ec2:stop-instances` 작업을 사용하여 인스턴스 하나와 두 번째 인스턴스를 중지하는 실험 템플릿을 생성합니다.

사전 조건

이 자습서를 완료하려면 다음을 수행합니다.

- 계정에서 두 개의 테스트 EC2 인스턴스를 시작합니다. 인스턴스를 시작한 후 두 인스턴스의 ID를 기록합니다.
- AWS FIS 서비스가 사용자를 대신하여 `aws:ec2:stop-instances` 작업을 수행할 수 있도록 IAM 역할을 생성합니다. 자세한 내용은 AWS FIS 실험을 위한 IAM 역할 단원을 참조하십시오.
- AWS FIS에 액세스할 수 있는지 확인합니다. 자세한 내용을 알아보려면 AWS FIS 정책 예제를 참조하세요.

1단계: 실험 템플릿 만들기

AWS FIS 콘솔을 사용하여 실험 템플릿을 생성합니다. 템플릿에서 각각 3분 동안 순차적으로 실행되는 두 개의 작업을 지정합니다. 첫 번째 작업은 AWS FIS가 임의로 선택하는 테스트 인스턴스 중 하나를 중지합니다. 두 번째 작업은 두 테스트 인스턴스를 모두 중지합니다.

실험 템플릿 만들기

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿 생성을 선택합니다.
4. 1단계, 템플릿 세부 정보 지정에서 다음을 수행합니다.
   a. 설명 및 이름에와 같은 템플릿에 대한 설명을 입력합니다`Amazon S3 Network Disrupt Connectivity`.
   b. 다음을 선택하고 2단계, 작업 및 대상 지정으로 이동합니다.
5. 작업에서 다음을 수행합니다.
   a. 작업 추가를 선택합니다.
   b. 작업의 이름을 입력합니다. 예를 들면 **`stopOneInstance`**를(을) 입력합니다.
   c. 작업 유형으로는 aws:ec2:stop-instances를 선택합니다.
   d. 대상의 경우 AWS FIS가 생성하는 대상을 그대로 유지합니다.
   e. 작업 파라미터의 경우 기간 이후 인스턴스 시작은 3분(PT3M)으로 지정합니다.
   f. 저장(Save)을 선택합니다.
6. 대상에서 다음을 수행합니다.
   a. 이전 단계에서 AWS FIS가 자동으로 생성한 대상에 대해 편집을 선택합니다.
   b. 기본 이름을 좀 더 이해하기 쉬운 이름으로 바꾸세요. 예를 들면 **`oneRandomInstance`**를(을) 입력합니다.
   c. 리소스 유형이 aws:ec2:instance인지 확인하세요.
   d. 대상 메서드의 경우 리소스 ID를 선택한 다음, 두 테스트 인스턴스의 ID를 선택합니다.
   e. 선택 모드에서는 개수를 선택합니다. 리소스 수에 **`1`**를 입력합니다.
   f. 저장(Save)을 선택합니다.
7. 대상 추가를 선택하고 다음과 같이 합니다.
   a. 대상의 이름을 입력합니다. 예를 들면 **`bothInstances`**를(을) 입력합니다.
   b. 리소스 유형에서 aws:ec2:instance를 선택합니다.
   c. 대상 메서드의 경우 리소스 ID를 선택한 다음, 두 테스트 인스턴스의 ID를 선택합니다.
   d. 선택 모드에서는 모두를 선택합니다.
   e. 저장(Save)을 선택합니다.

8. 작업 섹션에서 작업 추가를 선택합니다. 다음을 수행합니다.

   a. 이름에서 작업의 이름을 입력합니다. 예를 들면 **stopBothInstances**를(을) 입력합니다.

   b. 작업 유형으로는 aws:ec2:stop-instances를 선택합니다.

   c. 다음 시간 후 시작에서 추가한 첫 번째 작업(**stopOneInstance**)을 선택합니다.

   d. 대상에서, 추가한 두 번째 대상(**bothInstances**)을 선택합니다.

   e. 작업 파라미터의 경우 기간 이후 인스턴스 시작은 3분(PT3M)으로 지정합니다.

   f. 저장(Save)을 선택합니다.

9. 다음을 선택하여 3단계, 서비스 액세스 구성으로 이동합니다.

10. 서비스 액세스의 경우 기존 IAM 역할 사용을 선택한 다음 이 자습서의 사전 조건에 설명된 대로 생성한 IAM 역할을 선택합니다. 역할이 표시되지 않는 경우 해당 역할에 필요한 신뢰 관계가 있는지 확인하세요. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

11. 다음을 선택하여 4단계, 선택적 설정 구성으로 이동합니다.

12. (선택 사항) 태그의 경우 새 태그 추가를 선택하고 태그 키와 태그 값을 지정합니다. 추가한 태그는 템플릿을 사용하여 실행되는 실험이 아니라 실험 템플릿에 적용됩니다.

13. 다음을 선택하여 5단계, 검토 및 생성으로 이동합니다.

14. 템플릿을 검토하고 실험 템플릿 생성을 선택합니다. 확인 메시지가 표시되면 `create`를 입력한 다음 실험 템플릿 생성을 선택합니다.

(선택사항) 실험 템플릿 JSON을 보려면

내보내기 탭을 선택합니다. 다음은 이전 콘솔 절차에서 생성한 JSON 예제입니다.

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

2단계: 실험 시작

실험 템플릿 생성을 완료하면 이를 사용하여 실험을 시작할 수 있습니다.

실험을 시작하려면

1. 방금 만든 실험 템플릿의 세부정보 페이지로 이동해야 합니다. 그렇지 않으면 실험 템플릿을 선택한 다음 실험 템플릿의 ID를 선택하여 세부 정보 페이지를 엽니다.
2. 실험 시작을 선택합니다.
3. (선택 사항) 실험에 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
4. 실험 시작을 선택합니다. 확인 메시지가 나타나면 **start**을 입력하고 실험 시작을 선택합니다.

3단계: 실험 진행 상황 추적하기

실험이 완료, 중지 또는 실패할 때까지 진행 중인 실험의 진행 상황을 추적할 수 있습니다.

실험 진행 상황 추적하기

1. 방금 시작한 실험의 세부정보 페이지로 이동해야 합니다. 그렇지 않으면 실험을 선택한 다음 실험의 ID를 선택하여 세부 정보 페이지를 엽니다.
2. 실험 상태를 보려면 세부 정보 창에서 상태를 확인하세요. 자세한 내용은 실험 상태를 참조하세요.
3. 실험 상태가 실행 중이면 다음 단계로 이동합니다.

4단계: 실험 결과 확인

예상대로 실험에 의해 인스턴스가 중지되고 시작되었는지 확인할 수 있습니다.

실험 결과를 확인하려면

1. 새 브라우저 탭 또는 창에 https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다. 이렇게 하면 Amazon EC2 콘솔에서 실험 결과를 보면서 AWS FIS 콘솔에서 실험 진행 상황을 계속 추적할 수 있습니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 첫 번째 작업의 상태가 보류 중에서 실행 중(AWS FIS 콘솔)으로 변경되면 대상 인스턴스 중 하나의 상태가 실행 중에서 중지됨(Amazon EC2 콘솔)으로 변경됩니다.
4. 3분 후 첫 번째 작업의 상태는 완료됨으로 변경되고, 두 번째 작업의 상태는 실행 중으로 변경되며, 다른 대상 인스턴스의 상태는 중지됨으로 변경됩니다.
5. 3분 후 두 번째 작업의 상태는 완료됨으로 변경되고, 대상 인스턴스의 상태는 실행 중으로 변경되며, 실험의 상태는 완료됨으로 변경됩니다.

5단계: 정리

이 실험용으로 생성한 테스트 EC2 인스턴스가 더 이상 필요하지 않으면 종료할 수 있습니다.

인스턴스를 종료하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 Instances(인스턴스)를 선택합니다.
3. 테스트 인스턴스를 모두 선택하고 Instance state(인스턴스 상태), Terminate instance(인스턴스 종료)를 선택합니다.
4. 확인 메시지가 나타나면 종료를 선택합니다.

실험 템플릿이 더 이상 필요하지 않으면 삭제할 수 있습니다.

AWS FIS 콘솔을 사용하여 실험 템플릿을 삭제하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **`delete`**를 입력한 다음 실험 템플릿 삭제를 선택합니다.

자습서: AWS FIS를 사용하여 인스턴스에서 CPU 스트레스 실행

AWS Fault Injection Service(AWS FIS)를 사용하여 애플리케이션이 CPU 스트레스를 처리하는 방법을 테스트할 수 있습니다. 이 자습서를 사용하여 AWS FIS를 사용하여 인스턴스에서 CPU 스트레스를 실행하는 사전 구성된 SSM 문서를 실행하는 실험 템플릿을 생성합니다. 이 자습서에서는 인스턴스의 CPU 사용률이 구성된 임계값을 초과할 경우 중지 조건을 사용하여 실험을 중단합니다.

자세한 내용은 the section called “사전 구성된 AWS FIS SSM 문서” 단원을 참조하십시오.

사전 조건

AWS FIS를 사용하여 CPU 스트레스를 실행하려면 먼저 다음 사전 조건을 완료하세요.

IAM 역할 생성

역할을 생성하고 AWS FIS가 사용자를 대신하여 `aws:ssm:send-command` 작업을 사용할 수 있도록 하는 정책을 연결합니다. 자세한 내용은 AWS FIS 실험을 위한 IAM 역할 단원을 참조하십시오.

AWS FIS에 대한 액세스 확인

AWS FIS에 액세스할 수 있는지 확인합니다. 자세한 내용을 알아보려면 AWS FIS 정책 예제를 참조하세요.

테스트 EC2 인스턴스 준비

- 사전 구성된 SSM 문서의 요구에 따라 Amazon Linux 2 또는 Ubuntu를 사용하여 EC2 인스턴스를 시작합니다.
- 인스턴스는 SSM으로 관리해야 합니다. 인스턴스가 SSM으로 관리되는지 확인하려면 플릿 관리자 콘솔을 여세요. 인스턴스가 SSM으로 관리되지 않는 경우, SSM 에이전트가 설치되어 있고 인스턴스에 AmazonSSMManagedInstanceCore 정책이 적용된 IAM 역할이 연결되어 있는지 확인하세요. 설치된 SSM 에이전트를 확인하려면 인스턴스에 연결하고 다음 명령을 실행합니다.

  Amazon Linux 2

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

  Ubuntu

  ```
  apt list amazon-ssm-agent
  ```
- 인스턴스에 대한 세부 모니터링을 활성화합니다. 추가 비용을 지불하면 데이터를 1분 기간으로 사용할 수 있습니다. 인스턴스를 선택하고 작업, 모니터링 및 문제 해결, 세부 모니터링 관리를 선택합니다.

1단계: 중지 조건에 대한 CloudWatch 경보 생성

CPU 사용률이 지정한 임계값을 초과하는 경우 실험을 중지할 수 있도록 CloudWatch 경보를 구성합니다. 다음 절차는 대상 인스턴스의 CPU 사용률 50%로 임계값을 설정합니다. 자세한 내용은 중지 조건 단원을 참조하십시오.

CPU 사용률이 임계값을 초과할 때 알려주는 경보를 만들려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 대상 인스턴스를 선택하고 작업, 모니터링 및 문제 해결, CloudWatch 경보 관리를 선택합니다.
4. 경보 알림에서 토글을 사용하여 Amazon SNS 알림을 끕니다.

5. 경보 임계값의 경우 다음 설정을 사용하세요.
   - 샘플 그룹화 기준: 최대값
   - 샘플링할 데이터 유형: CPU 사용률
   - 퍼센트: **50**
   - 기간: **1 Minute**
6. 경보 구성이 완료되면 생성을 선택합니다.

2단계: 실험 템플릿 만들기

AWS FIS 콘솔을 사용하여 실험 템플릿을 생성합니다. 템플릿에서 실행할 다음 작업을 지정합니다. aws:ssm:send-command/AWSFIS-Run-CPU-Stress.

실험 템플릿을 생성하는 방법

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿 생성을 선택합니다.
4. 1단계, 템플릿 세부 정보 지정에서 다음을 수행합니다.
   a. 설명 및 이름에 템플릿에 대한 설명을 입력합니다.
   b. 다음을 선택하고 2단계, 작업 및 대상 지정으로 이동합니다.
5. 작업에서 다음을 수행합니다.
   a. 작업 추가를 선택합니다.
   b. 작업의 이름을 입력합니다. 예를 들면 **runCpuStress**를 입력합니다.
   c. 작업 유형으로는 aws:ssm:send-command/AWSFIS-Run-CPU-Stress를 선택합니다. 그러면 SSM 문서의 ARN이 문서 ARN에 자동으로 추가됩니다.
   d. 대상의 경우 AWS FIS가 자동으로 생성하는 대상을 유지합니다.
   e. 작업 파라미터, 문서 파라미터에 다음을 입력합니다.

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

   f. 작업 파라미터 지속 시간에 5분(PT5M)을 지정합니다.
   g. 저장을 선택합니다.

6. 대상에서 다음을 수행합니다.

   a. 이전 단계에서 AWS FIS가 자동으로 생성한 대상에 대해 편집을 선택합니다.

   b. 기본 이름을 좀 더 이해하기 쉬운 이름으로 바꾸세요. 예를 들면 **testInstance**를 입력합니다.

   c. 리소스 유형이 aws:ec2:instance인지 확인하세요.

   d. 대상 메서드의 경우 리소스 ID를 선택한 다음 테스트 인스턴스의 ID를 선택합니다.

   e. 선택 모드에서는 모두를 선택합니다.

   f. 저장을 선택합니다.

7. 다음을 선택하여 3단계, 서비스 액세스 구성으로 이동합니다.

8. 서비스 액세스의 경우 기존 IAM 역할 사용을 선택한 다음 이 자습서의 사전 조건에 설명된 대로 생성한 IAM 역할을 선택합니다. 역할이 표시되지 않는 경우 해당 역할에 필요한 신뢰 관계가 있는지 확인하세요. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

9. 다음을 선택하여 4단계, 선택적 설정 구성으로 이동합니다.

10. 중지 조건의 경우, 1단계에서 생성한 CloudWatch 경보를 선택합니다.

11. (선택 사항) 태그의 경우 새 태그 추가를 선택하고 태그 키와 태그 값을 지정합니다. 추가한 태그는 템플릿을 사용하여 실행되는 실험이 아니라 실험 템플릿에 적용됩니다.

12. 다음을 선택하여 5단계, 검토 및 생성으로 이동합니다.

13. 템플릿을 검토하고 실험 템플릿 생성을 선택합니다. 확인 메시지가 표시되면를 입력한 `create`다음 실험 템플릿 생성을 선택합니다.

(선택사항) 실험 템플릿 JSON을 보려면

내보내기 탭을 선택합니다. 다음은 이전 콘솔 절차에서 생성한 JSON 예제입니다.

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

3단계: 실험 시작

실험 템플릿 생성을 완료하면 이를 사용하여 실험을 시작할 수 있습니다.

실험을 시작하려면

1. 방금 만든 실험 템플릿의 세부정보 페이지로 이동해야 합니다. 그렇지 않으면 실험 템플릿을 선택한 다음 실험 템플릿의 ID를 선택하여 세부 정보 페이지를 엽니다.
2. 실험 시작을 선택합니다.
3. (선택 사항) 실험에 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
4. 실험 시작을 선택합니다. 확인 메시지가 표시되면 **start**를 입력합니다. 실험 시작을 선택합니다.

4단계: 실험 진행 상황 추적하기

실험이 완료, 중지 또는 실패할 때까지 진행 중인 실험의 진행 상황을 추적할 수 있습니다.

실험 진행 상황 추적하기

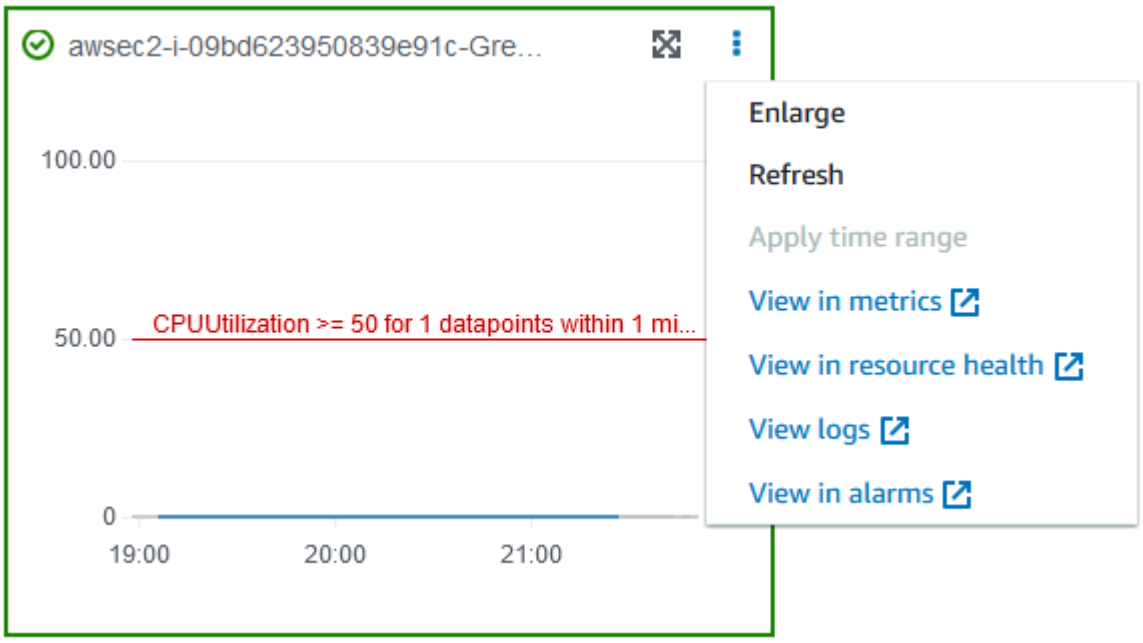
1. 방금 시작한 실험의 세부정보 페이지로 이동해야 합니다. 그렇지 않으면 실험을 선택한 다음 실험의 ID를 선택하여 실험의 세부 정보 페이지를 엽니다.
2. 실험 상태를 보려면 세부 정보 창에서 상태를 확인하세요. 자세한 내용은 실험 상태를 참조하세요.
3. 실험 상태가 실행 중으로 표시되면 다음 단계로 이동합니다.

5단계: 실험 결과 확인

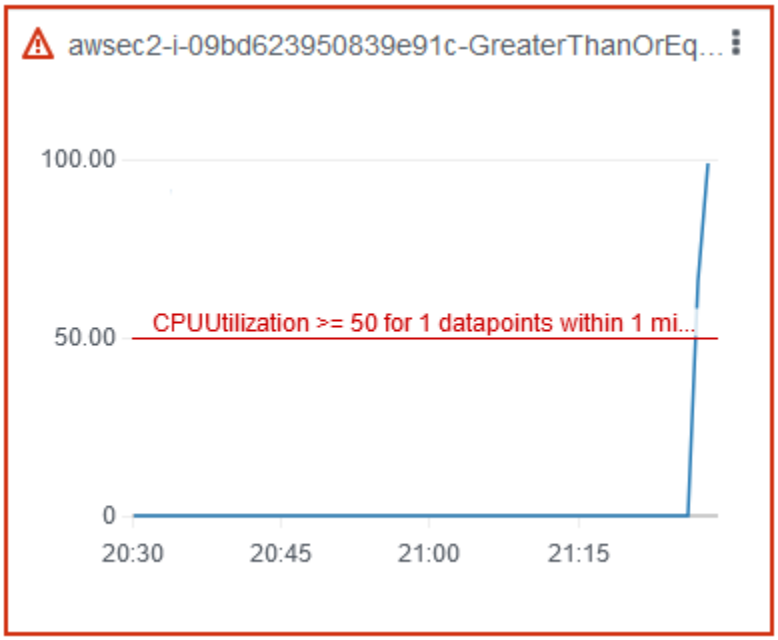
실험이 실행되는 동안 인스턴스의 CPU 사용률을 모니터링할 수 있습니다. CPU 사용률이 임계값에 도달하면 경보가 트리거되고 중지 조건에 따라 실험이 중단됩니다.

실험 결과를 확인하려면

1. 중지 조건 탭을 선택합니다. 녹색 테두리와 녹색 체크 표시 아이콘은 경보의 초기 상태가 OK임을 나타냅니다. 빨간색 선은 경보 임계값을 나타냅니다. 더 자세한 그래프를 원하면 위젯 메뉴에서 확대를 선택합니다.



2. CPU 사용률이 임계값을 초과하면 중지 조건 탭의 빨간색 테두리와 빨간색 느낌표 아이콘이 경보 상태가 ALARM로 변경되었음을 나타냅니다. 세부 정보 창에서는 실험 상태가 중지됨입니다. 상태를 선택하면 “중지 조건에 따라 실험이 중단됨”이라는 메시지가 표시됩니다.



3. CPU 사용률이 임계값 아래로 감소하면 녹색 테두리와 녹색 체크 표시 아이콘은 경보 상태가 OK로 변경되었음을 나타냅니다.
4. (선택 사항) 위젯 메뉴에서 경보에서 보기를 선택합니다. 그러면 CloudWatch 콘솔에서 경보 세부 정보 페이지가 열리고, 여기에서 경보에 대한 자세한 내용을 확인하거나 경보 설정을 편집할 수 있습니다.

6단계: 정리

이 실험용으로 생성한 테스트 EC2 인스턴스가 더 이상 필요하지 않으면 종료할 수 있습니다.

인스턴스를 해지하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 테스트 인스턴스를 선택하고 인스턴스 상태, 인스턴스 종료를 선택합니다.
4. 확인 메시지가 나타나면 종료를 선택합니다.

실험 템플릿이 더 이상 필요하지 않으면 삭제할 수 있습니다.

AWS FIS 콘솔을 사용하여 실험 템플릿을 삭제하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.

2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **delete**를 입력한 다음 실험 템플릿 삭제를 선택합니다.

자습서: AWS FIS를 사용하여 스팟 인스턴스 중단 테스트

스팟 인스턴스는 온디맨드 요금과 비교하여 최대 90% 할인된 가격으로 제공되는 예비 EC2 용량을 사용합니다. 그러나 Amazon EC2는 용량이 다시 필요할 때 스팟 인스턴스를 중단할 수 있습니다. 스팟 인스턴스를 사용할 때는 중단 가능성에 대비해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 스팟 인스턴스 중단을 참조하세요.

AWS Fault Injection Service(AWS FIS)를 사용하여 애플리케이션이 스팟 인스턴스 중단을 처리하는 방법을 테스트할 수 있습니다. 이 자습서를 사용하여 AWS FIS `aws:ec2:send-spot-instance-interruptions` 작업을 사용하여 스팟 인스턴스 중 하나를 중단하는 실험 템플릿을 생성합니다.

또는 Amazon EC2 콘솔을 사용하여 실험을 시작하려면 Amazon EC2 사용 설명서의 스팟 인스턴스 중단 시작을 참조하세요.

사전 조건

AWS FIS를 사용하여 스팟 인스턴스를 중단하려면 먼저 다음 사전 조건을 완료합니다.

1. IAM 역할 생성

역할을 생성하고 AWS FIS가 사용자를 대신하여 `aws:ec2:send-spot-instance-interruptions` 작업을 수행할 수 있도록 하는 정책을 연결합니다. 자세한 내용은 AWS FIS 실험을 위한 IAM 역할 단원을 참조하십시오.

2. AWS FIS에 대한 액세스 확인

AWS FIS에 액세스할 수 있는지 확인합니다. 자세한 내용을 알아보려면 AWS FIS 정책 예제를 참조하세요.

3. (선택 사항) 스팟 인스턴스 요청 생성

이 실험에 새 스팟 인스턴스를 사용하려면 run-instances 명령을 사용하여 스팟 인스턴스를 요청하세요. 기본값은 중단된 스팟 인스턴스를 종료하는 것입니다. 중단 동작을 `stop`로 설정하는 경우 유형도 `persistent`로 설정해야 합니다. 이 자습서에서는 최대 절전 모드 프로세스가 즉시 시작되므로 중단 동작을 `hibernate`로 설정하지 마세요.

```
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
```

다음은 spot-options.json 파일의 예입니다.

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

예제 명령의 --query 옵션을 사용하면 명령이 스팟 인스턴스의 인스턴스 ID만 반환하도록 할 수 있습니다. 출력의 예시는 다음과 같습니다.

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

4. AWS FIS가 대상 스팟 인스턴스를 식별할 수 있도록 태그 추가

대상 스팟 인스턴스에 Name=interruptMe 태그를 추가하려면 create-tags 명령을 사용합니다.

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

1단계: 실험 템플릿 만들기

AWS FIS 콘솔을 사용하여 실험 템플릿을 생성합니다. 템플릿에서 실행할 작업을 지정합니다. 작업은 지정된 태그가 있는 스팟 인스턴스를 중단합니다. 태그가 있는 스팟 인스턴스가 두 개 이상 있는 경우 AWS FIS는 그 중 하나를 임의로 선택합니다.

실험 템플릿을 생성하는 방법

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿 생성을 선택합니다.
4. 1단계, 템플릿 세부 정보 지정에서 다음을 수행합니다.
   a. 이름 및 설명에 템플릿에 대한 이름과 설명을 입력합니다.
   b. 다음을 선택하고 2단계, 작업 및 대상 지정으로 이동합니다.
5. 작업에서 다음을 수행합니다.
   a. 작업 추가를 선택합니다.
   b. 작업의 이름을 입력합니다. 예를 들면 **interruptSpotInstance**를(을) 입력합니다.
   c. 작업 유형의 경우 aws:ec2:send-spot-instance-interruptions를 선택합니다.
   d. 대상의 경우 AWS FIS가 생성하는 대상을 그대로 유지합니다.
   e. 작업 파라미터의 경우 중단 전 기간을 2분(PT2M)으로 지정합니다.
   f. 저장(Save)을 선택합니다.
6. 대상에서 다음을 수행합니다.
   a. AWS FIS가 이전 단계에서 자동으로 생성한 대상에 대해 편집을 선택합니다.
   b. 기본 이름을 좀 더 이해하기 쉬운 이름으로 바꾸세요. 예를 들면 **oneSpotInstance**를(을) 입력합니다.
   c. 리소스 유형이 aws:ec2:spot-instance인지 확인하세요.
   d. 대상 메서드의 경우 리소스 태그, 필터, 파라미터를 선택합니다.
   e. 리소스 태그에 대해 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다. 이 자습서의 사전 요구 사항에 설명된 대로 스팟 인스턴스에 추가한 태그를 사용하여 중단하세요.
   f. 리소스 필터의 경우 새 필터 추가를 선택하고 경로로 **State.Name**, 값으로 **running**를 입력합니다.
   g. 선택 모드에서는 개수를 선택합니다. 리소스 수에 **1**를 입력합니다.
   h. 저장(Save)을 선택합니다.
7. 다음을 선택하여 3단계, 서비스 액세스 구성으로 이동합니다.
8. 서비스 액세스의 경우 기존 IAM 역할 사용을 선택한 다음 이 자습서의 사전 조건에 설명된 대로 생성한 IAM 역할을 선택합니다. 역할이 표시되지 않는 경우 해당 역할에 필요한 신뢰 관계가 있는지 확인하세요. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

9. 다음을 선택하여 4단계, 선택적 설정 구성으로 이동합니다.
10. (선택 사항) 태그의 경우 새 태그 추가를 선택하고 태그 키와 태그 값을 지정합니다. 추가한 태그는 템플릿을 사용하여 실행되는 실험이 아니라 실험 템플릿에 적용됩니다.
11. 다음을 선택하여 5단계, 검토 및 생성으로 이동합니다.
12. 템플릿을 검토하고 실험 템플릿 생성을 선택합니다. 확인 메시지가 표시되면 `create`를 입력한 다음 실험 템플릿 생성을 선택합니다.

(선택사항) 실험 템플릿 JSON을 보려면

내보내기 탭을 선택합니다. 다음은 이전 콘솔 절차에서 생성한 JSON 예제입니다.

```
{
    "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"
    }
}
```

2단계: 실험 시작

실험 템플릿 생성을 완료하면 이를 사용하여 실험을 시작할 수 있습니다.

실험을 시작하려면

1. 방금 만든 실험 템플릿의 세부정보 페이지로 이동해야 합니다. 그렇지 않으면 실험 템플릿을 선택한 다음 실험 템플릿의 ID를 선택하여 세부 정보 페이지를 엽니다.
2. 실험 시작을 선택합니다.
3. (선택 사항) 실험에 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
4. 실험 시작을 선택합니다. 확인 메시지가 나타나면 **start**을 입력하고 실험 시작을 선택합니다.

3단계: 실험 진행 상황 추적하기

실험이 완료, 중지 또는 실패할 때까지 진행 중인 실험의 진행 상황을 추적할 수 있습니다.

실험 진행 상황 추적하기

1. 방금 시작한 실험의 세부정보 페이지로 이동해야 합니다. 그렇지 않으면 실험을 선택한 다음 실험의 ID를 선택하여 세부 정보 페이지를 엽니다.
2. 실험 상태를 보려면 세부 정보 창에서 상태를 확인하세요. 자세한 내용은 실험 상태를 참조하세요.
3. 실험 상태가 실행 중이면 다음 단계로 이동합니다.

4단계: 실험 결과 확인

이 실험에 대한 작업이 완료되면 다음과 같이 진행됩니다.

- 대상 스팟 인스턴스가 인스턴스 리밸런싱 권고를 수신합니다.

- 스팟 인스턴스 중단 공지는 Amazon EC2가 인스턴스를 종료 또는 중지하기 2분 전에 생성됩니다.
- 2분 후 스팟 인스턴스가 종료되거나 중지됩니다.
- AWS FIS에 의해 중지된 스팟 인스턴스는 다시 시작할 때까지 중지된 상태로 유지됩니다.

인스턴스가 실험에 의해 중단되었는지 확인

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 별도의 브라우저 탭 또는 창으로 Spot Requests(스팟 요청)와 Instances(인스턴스)를 엽니다.
3. Spot Requests(스팟 요청)에서 스팟 인스턴스 요청을 선택합니다. 초기 상태는 `fulfilled`입니다. 실험이 완료되면 상태가 다음과 같이 변경됩니다.
   - `terminate` - 상태가 `instance-terminated-by-experiment`로 변경됩니다.
   - `stop` - 상태가 `marked-for-stop-by-experiment`으로 변경되었다가 `instance-stopped-by-experiment`로 변경됩니다.
4. Instances(인스턴스)에서 스팟 인스턴스를 선택합니다. 초기 상태는 `Running`입니다. 스팟 인스턴스 중단 알림을 받고 2분 후 상태가 다음과 같이 변경됩니다.
   - `stop` - 상태가 `Stopping`으로 변경되었다가 `Stopped`로 변경됩니다.
   - `terminate` - 상태가 `Shutting-down`으로 변경되었다가 `Terminated`로 변경됩니다.

5단계: 정리

중단 동작이 `stop`인 이 실험에 대한 테스트 스팟 인스턴스를 생성했는데 더 이상 필요하지 않은 경우 스팟 인스턴스 요청을 취소하고 스팟 인스턴스를 종료할 수 있습니다.

요청을 취소하고를 사용하여 인스턴스를 종료하려면 AWS CLI

1. cancel-spot-instance-requests 명령을 사용하여 스팟 인스턴스 요청을 취소합니다.

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

2. 인스턴스를 종료하려면 terminate-instances 명령을 사용하세요.

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

실험 템플릿이 더 이상 필요하지 않으면 삭제할 수 있습니다.

AWS FIS 콘솔을 사용하여 실험 템플릿을 삭제하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **delete**를 입력한 다음 실험 템플릿 삭제를 선택합니다.

자습서: 연결 이벤트 시뮬레이션

AWS Fault Injection Service(AWS FIS)를 사용하여 다양한 연결 이벤트를 시뮬레이션할 수 있습니다. AWS FIS는 다음 방법 중 하나로 네트워크 연결을 차단하여 연결 이벤트를 시뮬레이션합니다.

- `all` - 서브넷으로 들어오고 나가는 모든 트래픽을 거부합니다. 이 옵션은 서브넷의 네트워크 인터페이스로 들어오고 나가는 트래픽을 포함하여 서브넷 내 트래픽을 허용한다는 점에 유의하세요.
- `availability-zone` - 다른 가용 영역에 있는 서브넷으로 들어오고 나가는 VPC 내부 트래픽을 거부합니다.
- `dynamodb` - 현재 리전의 DynamoDB 리전 엔드포인트로 들어오고 나가는 트래픽을 거부합니다.
- `prefix-list` - 지정된 접두사 목록으로 들어오고 나가는 트래픽을 거부합니다.
- `s3` - 현재 리전의 Amazon S3 리전 엔드포인트로 들어오고 나가는 트래픽을 거부합니다.
- `s3express` - 대상 서브넷의 AZ에서 Amazon S3 Express One Zone의 영역 엔드포인트와 주고받는 트래픽을 거부합니다. 대상 서브넷은 현재 S3 Express One Zone을 사용할 수 있는 AZs에 있어야 합니다. 자세한 내용은 S3 Express One Zone 가용 영역 및 리전을 참조하세요.
- `vpc` - VPC로 들어오고 나가는 트래픽을 거부합니다.

이 자습서를 사용하여 AWS FIS `aws:network:disrupt-connectivity` 작업을 사용하여 대상 서브넷에서 Amazon S3와의 연결 손실을 도입하는 실험 템플릿을 생성합니다.

주제

- 사전 조건
- 1단계: AWS FIS 실험 템플릿 생성
- 2단계: Amazon S3 엔드포인트에 대해 핑 전송
- 3단계: AWS FIS 실험 시작
- 4단계: AWS FIS 실험 진행 상황 추적

- 5단계: Amazon S3 네트워크 중단 확인
- 5단계: 정리

사전 조건

이 자습서를 시작하기 전에에 적절한 권한이 있는 역할과 테스트 Amazon EC2 인스턴스 AWS 계정가 필요합니다.

에서 권한이 있는 역할 AWS 계정

역할을 생성하고 AWS FIS가 사용자를 대신하여 `aws:network:disrupt-connectivity` 작업을 수행할 수 있도록 하는 정책을 연결합니다.

IAM 역할에는 다음의 정책이 필요합니다.

- AWSFaultInjectionSimulatorNetworkAccess – Amazon EC2 네트워킹 및 기타 필수 서비스에서 네트워크 인프라와 관련된 AWS FIS 작업을 수행하는 데 필요한 AWS FIS 서비스 권한을 부여합니다.

> **Note**
>
> 간소화를 위해이 자습서에서는 AWS 관리형 정책을 사용합니다. 프로덕션 사용 용도의 경우 사용 사례에 필요한 최소한의 권한만 부여하는 것이 좋습니다.
> IAM 역할을 생성하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS FIS 실험을 위한 IAM 역할(AWS CLI) 또는 IAM 역할 생성(콘솔)을 참조하세요.

테스트 Amazon EC2 인스턴스

테스트 Amazon EC2 인스턴스를 시작하고 연결합니다. 다음 자습서를 사용하여 Amazon EC2 인스턴스를 시작하고 연결할 수 있습니다. Amazon EC2 사용 설명서의 자습서: Amazon EC2 Linux 인스턴스 시작.

1단계: AWS FIS 실험 템플릿 생성

AWS FIS를 사용하여 실험 템플릿을 생성합니다 AWS Management Console. AWS FIS 템플릿은 작업, 대상, 중지 조건 및 실험 역할로 구성됩니다. 템플릿 작동 방식에 대한 자세한 내용은 AWS FIS용 실험 템플릿을 참조하세요.

시작하기 전에 다음 항목이 준비되었는지 확인합니다.

- 올바른 권한을 가진 IAM 역할.
- Amazon EC2 인스턴스.
- Amazon EC2 인스턴스의 서브넷 ID.

실험 템플릿을 생성하는 방법

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿 생성을 선택합니다.
4. 1단계, 템플릿 세부 정보 지정에서 다음을 수행합니다.
   a. 설명 및 이름에와 같은 템플릿에 대한 설명을 입력합니다`Amazon S3 Network Disrupt Connectivity`.
   b. 다음을 선택하고 2단계, 작업 및 대상 지정으로 이동합니다.
5. 작업에서 작업 추가를 선택합니다.
   a. 이름에 `disruptConnectivity`를 입력합니다.
   b. 작업 유형에서 aws:network:disrupt-connectivity를 선택합니다.
   c. 작업 파라미터에서 기간을 `2 minutes`로 설정합니다.
   d. 범위에서 s3를 선택합니다.
   e. 상단에서 저장을 선택합니다.
6. 대상 아래에서 자동으로 생성된 대상을 확인할 수 있습니다. 편집을 선택합니다.
   a. 리소스 유형이`aws:ec2:subnet`인지 확인하세요.
   b. 대상 메서드에서 리소스 ID를 선택한 다음, 사전 조건 단계에서 Amazon EC2 인스턴스를 생성할 때 사용한 서브넷을 선택합니다.
   c. 선택 모드가 모두인지 확인하세요.
   d. 저장을 선택합니다.
7. 다음을 선택하여 3단계, 서비스 액세스 구성으로 이동합니다.
8. 서비스 액세스에서 이 자습서의 사전 조건에 설명된 대로 생성한 IAM 역할을 선택합니다. 역할이 표시되지 않는 경우 해당 역할에 필요한 신뢰 관계가 있는지 확인하세요. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

9. 다음을 선택하여 4단계, 선택적 설정 구성으로 이동합니다.
10. (선택 사항) 중지 조건 아래에서 조건이 발생할 경우 실험을 중단할 CloudWatch 경보를 선택할 수 있습니다. 자세한 내용은 AWS FIS의 중지 조건을 참조하세요.
11. (선택 사항) 로그에서 Amazon S3 버킷을 선택하거나 CloudWatch로 실험에 사용할 로그를 보낼 수 있습니다.
12. 다음을 선택하여 5단계, 검토 및 생성으로 이동합니다.
13. 템플릿을 검토하고 실험 템플릿 생성을 선택합니다. 확인 메시지가 표시되면를 입력한 `create`다음 실험 템플릿 생성을 선택합니다.

2단계: Amazon S3 엔드포인트에 대해 핑 전송

Amazon EC2 인스턴스가 Amazon S3 엔드포인트에 도달할 수 있는지 확인하세요.

1. 사전 조건 단계에 만든 Amazon EC2 인스턴스에 연결합니다.

   문제 해결을 위해 Amazon EC2 사용 설명서의 인스턴스 연결 문제 해결을 참조하세요.
2. 인스턴스가 있는 AWS 리전 를 확인합니다. 이렇게 하려면 Amazon EC2 콘솔에 있거나 다음 명령을 실행합니다.

   ```
   hostname
   ```

   예를 들어 `us-west-2`에서 Amazon EC2 인스턴스를 시작한 경우 다음과 같은 출력이 표시됩니다.

   ```
   [ec2-user@ip-172.16.0.0 ~]$ hostname
   ip-172.16.0.0.us-west-2.compute.internal
   ```
3. 에서 Amazon S3 엔드포인트 ping AWS 리전. *AWS ##*을 해당 리전으로 바꿉니다.

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

   출력의 경우 다음 예제와 같이 핑에 성공하여 패킷 손실이 0% 인 것을 볼 수 있습니다.

   ```
   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
```

3단계: AWS FIS 실험 시작

방금 만든 실험 템플릿으로 실험을 시작합니다.

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 실험 템플릿을 선택합니다.
3. 만든 실험 템플릿의 ID를 선택하여 세부 정보 페이지를 엽니다.
4. 실험 시작을 선택합니다.
5. (선택 사항) 확인 페이지에서 실험에 사용할 태그를 추가합니다.
6. 확인 페이지에서 실험 시작을 선택합니다.

4단계: AWS FIS 실험 진행 상황 추적

실험이 완료, 중지 또는 실패할 때까지 진행 중인 실험의 진행 상황을 추적할 수 있습니다.

1. 방금 시작한 실험의 세부정보 페이지로 이동해야 합니다. 그렇지 않은 경우 실험을 선택한 다음 실험의 ID를 선택하여 해당 세부정보 페이지를 여세요.
2. 실험 상태를 보려면 세부 정보 창에서 상태를 확인하세요. 자세한 내용은 실험 상태를 참조하세요.
3. 실험 상태가 실행 중이면 다음 단계로 이동합니다.

5단계: Amazon S3 네트워크 중단 확인

Amazon S3 엔드포인트에 핑을 보내 실험 진행 상황을 확인할 수 있습니다.

- Amazon EC2 인스턴스에서 사용자 AWS 리전의 Amazon S3 엔드포인트에 핑을 보냅니다. *AWS ##*을 해당 리전으로 바꿉니다.

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

  출력의 경우 다음 예제와 같이 핑에 실패하여 패킷 손실이 100% 인 것을 볼 수 있습니다.

  ```
  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
```

5단계: 정리

이 실험을 위해 만든 Amazon EC2 인스턴스나 AWS FIS 템플릿이 더 이상 필요하지 않은 경우 이 인스턴스를 제거할 수 있습니다.

Amazon EC2 인스턴스를 제거하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 인스턴스를 선택합니다.
3. 테스트 인스턴스를 선택하고 인스턴스 상태, 인스턴스 종료를 차례로 선택합니다.
4. 확인 메시지가 나타나면 종료를 선택합니다.

AWS FIS 콘솔을 사용하여 실험 템플릿을 삭제하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택한 다음 작업, 실험 템플릿 삭제를 선택합니다.
4. 확인 메시지가 나타나면 `delete`를 입력한 다음 실험 템플릿 삭제를 선택합니다.

자습서: 반복 실험 예약

AWS Fault Injection Service(AWS FIS)를 사용하면 워크로드에서 오류 주입 실험을 수행할 수 있습니다 AWS . 이러한 실험은 지정된 대상에서 실행할 하나 이상의 작업이 포함된 템플릿에서 실행됩니다. 또한를 사용하면 실험을 일회성 작업 또는 반복 작업으로 예약할 Amazon EventBridge수 있습니다.

이 자습서를 사용하여 5분마다 AWS FIS 실험 템플릿을 실행하는 EventBridge 일정을 생성합니다.

작업

- 사전 조건
- 1단계: IAM 역할 및 정책 생성
- 2단계: Amazon EventBridge 스케줄러 생성

- 3단계: 실험 확인
- 4단계: 정리

사전 조건

이 자습서를 시작하기 전에 에는 일정에 따라 실행하려는 AWS FIS 실험 템플릿이 있어야 합니다. 이미 제대로 작동하는 실험 템플릿이 있다면 템플릿 ID 및 AWS 리전을 메모해 두세요. 그렇지 않으면 the section called “테스트 인스턴스 중지 및 시작”의 지침에 따라 템플릿을 만든 다음 이 자습서로 돌아갈 수 있습니다.

1단계: IAM 역할 및 정책 생성

IAM 역할 및 정책을 생성하려면

1. IAM 콘솔(https://console.aws.amazon.com/iam/)을 엽니다.
2. 왼쪽 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. 사용자 지정 신뢰 정책을 선택한 다음 다음 코드 조각을 삽입하여 Amazon EventBridge 스케줄러가 사용자를 대신하여 역할을 수임하도록 허용합니다.

   JSON

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

   다음을 선택합니다.
4. 권한 추가, 정책 생성을 선택합니다.

5. JSON을 선택하고 다음 정책을 삽입합니다. *your-experiment-template-id* 값을 사전 조건 단계에 있는 실험의 템플릿 ID로 바꾸세요.

   JSON

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

   특정 태그 값이 있는 AWS FIS 실험 템플릿만 실행하도록 스케줄러를 제한할 수 있습니다. 예를 들어 다음 정책은 모든 AWS FIS 실험에 대한 `StartExperiment` 권한을 부여하지만 스케줄러가 태그가 지정된 실험 템플릿만 실행하도록 제한합니다`Purpose=Schedule`.

   JSON

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

```
                    "aws:ResourceTag/Purpose": "Schedule"
                }
            }
        }
    ]
}
```

다음: 태그를 선택합니다.

6. 다음: 검토를 선택합니다.
7. 정책 검토에서 정책 이름을 FIS_RecurringExperiment로 지정한 다음 정책 생성을 선택합니다.
8. 권한 추가에서 새 FIS_RecurringExperiment 정책을 역할에 추가하고 다음을 선택합니다.
9. 이름 지정, 검토 및 생성 아래에서 역할 이름에 FIS_RecurringExperiment_role을 입력하고 역할 생성을 선택합니다.

2단계: Amazon EventBridge 스케줄러 생성

Amazon EventBridge 스케줄러를 생성하려면

1. Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)을 엽니다.
2. 왼쪽 탐색 창에서 일정을 선택합니다.
3. AWS FIS 실험 템플릿 AWS 리전 과 동일한에 있는지 확인합니다.
4. 일정 생성을 선택하고 다음 필드를 입력합니다.
   - 일정 이름 아래에 다음을 입력합니다. FIS_recurring_experiment_tutorial
   - 일정 패턴에서 반복 일정을 선택합니다.
   - 일정 유형에서 요금 기반 일정을 선택합니다.
   - rate 표현식에서 5분을 선택합니다.
   - 유연한 기간 아래에서 끄기를 선택합니다.
   - (선택 사항) 기간 아래에서 시간대를 선택합니다.
   - 다음을 선택합니다.
5. 대상 선택에서 모든 API를 선택한 다음 AWS FIS를 검색합니다.
6. AWS FIS를 선택한 다음 StartExperiment를 선택합니다.

7. 입력에 다음 JSON 페이로드를 삽입합니다. *your-experiment-template-id* 값을 실험의 템플릿 ID로 바꾸세요. ClientToken는 스케줄러의 고유 식별자입니다. 이 자습서에서는 Amazon EventBridge 스케줄러에서 허용하는 컨텍스트 키워드를 사용합니다. 자세한 내용은 Amazon EventBridge 사용 설명서의 컨텍스트 속성 추가를 참조하세요.

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

   다음을 선택합니다.
8. (선택 사항) 설정에서 재시도 정책, DLQ(Dead Letter Queue) 및 암호화 설정을 지정할 수 있습니다. 또는 기본값 을 그대로 유지합니다.
9. 권한 아래에서 기존 역할 사용을 선택한 다음 FIS_RecurringExperiment_role을 검색합니다.
10. 다음을 선택합니다.
11. 일정 검토 및 생성에서 스케줄러 세부 정보를 검토한 다음 일정 생성을 선택합니다.

3단계: 실험 확인

AWS FIS 실험이 일정에 따라 실행되었는지 확인하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 실험을 선택합니다.
3. 일정을 만든 후 5분이 지나면 실험이 진행 중인 것을 확인할 수 있습니다.

4단계: 정리

Amazon EventBridge 스케줄러를 비활성화하려면

1. Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)을 엽니다.
2. 왼쪽 탐색 창에서 일정을 선택합니다.
3. 새로 만든 스케줄러를 선택한 다음 비활성화를 선택합니다.

AWS FIS 시나리오 라이브러리 작업

시나리오는 애플리케이션이 실행되는 컴퓨팅 리소스의 중단과 같이 고객이 애플리케이션의 복원력을 테스트하기 위해 적용할 수 있는 이벤트 또는 조건을 정의합니다. 시나리오는 AWS에서 생성하고 소유하며, 일반적인 애플리케이션 장애에 대해 사전 정의된 대상 그룹과 장애 조치(예: 자동 크기 조정 그룹 내 인스턴스 30% 중지)를 제공하여 차별화되지 않은 작업의 부담을 최소화합니다.

시나리오는 콘솔 전용 시나리오 라이브러리를 통해 제공되며 AWS FIS 실험 템플릿을 사용하여 실행합니다. 시나리오를 사용하여 실험을 실행하려면 라이브러리에서 시나리오를 선택하고 워크로드 세부 정보와 일치하는 파라미터를 지정한 다음 계정에 실험 템플릿으로 저장합니다.

주제

시나리오 보기

콘솔을 사용하여 시나리오 보기

1. 에서 AWS FIS 콘솔을 엽니다https://console.aws.amazon.com/fis/.
2. 탐색 창에서 시나리오 라이브러리를 선택합니다.
3. 특정 시나리오에 대한 정보를 보려면 시나리오 카드를 선택하여 분할 패널을 불러옵니다.
   - 페이지 하단의 분할 패널에 있는 설명 탭에서 시나리오에 대한 간략한 설명을 볼 수 있습니다. 또한 필요한 대상 리소스의 요약과 시나리오에 사용할 리소스를 준비하기 위해 취해야 하는 조치가 포함된 사전 요구 사항에 대한 간략한 요약을 찾을 수 있습니다. 마지막으로 시나리오의 대상 및 작업에 대한 추가 정보뿐만 아니라 기본 설정으로 실험이 성공적으로 실행될 때의 예상 기간도 확인할 수 있습니다.
   - 페이지 하단의 분할 패널에 있는 콘텐츠 탭에서 시나리오에서 생성될 실험 템플릿의 일부가 채워진 버전을 미리 볼 수 있습니다.
   - 페이지 하단의 분할 패널에 있는 세부 정보 탭에서 시나리오 구현 방법에 대한 자세한 설명을 찾을 수 있습니다. 여기에는 시나리오의 개별 측면을 근사화하는 방법에 대한 자세한 정보가 포함될 수 있습니다. 해당하는 경우, 어떤 지표를 중지 조건으로 사용하고 관찰성을 제공하여 실험에서

배울 수 있는지에 대해서도 읽어볼 수 있습니다. 마지막으로 결과 실험 템플릿을 확장하는 방법에 대한 권장 사항을 찾을 수 있습니다.

시나리오 사용

콘솔을 사용하여 시나리오 사용:

1. 에서 AWS FIS 콘솔을 엽니다https://console.aws.amazon.com/fis/.
2. 탐색 창에서 시나리오 라이브러리를 선택합니다.
3. 특정 시나리오에 대한 정보를 보려면 시나리오 카드를 선택하여 분할 패널을 불러옵니다.
4. 시나리오를 사용하려면 시나리오 카드를 선택하고 시나리오로 템플릿 생성을 선택합니다.
5. 실험 템플릿 생성 보기에서 누락된 항목을 모두 채웁니다.
   a. 일부 시나리오에서는 여러 작업 또는 대상에서 공유되는 파라미터를 편집할 수 있습니다. 이 기능은 공유 파라미터 편집에 의한 변경을 포함하여 시나리오를 변경하면 비활성화됩니다. 이 기능을 사용하려면 공유 파라미터 편집 버튼을 선택합니다. 모달에서 파라미터를 편집하고 저장 버튼을 선택합니다.
   b. 일부 실험 템플릿에는 작업 또는 대상 파라미터가 누락되어 있을 수 있으며, 각 작업 및 대상 카드에 강조 표시되어 있습니다. 각 카드의 편집 버튼을 선택하고 누락된 정보를 추가한 다음 카드에서 저장 버튼을 선택합니다.
   c. 모든 템플릿에는 서비스 액세스 실행 역할이 필요합니다. 이 실험 템플릿에 대해 기존 역할을 선택하거나 새로운 역할을 생성할 수 있습니다.
   d. 기존 AWS CloudWatch 경보를 선택하여 하나 이상의 선택적 중지 조건을 정의하는 것이 좋습니다. AWS FIS에 대한 중지 조건에 대해 자세히 알아보세요. 경보가 아직 구성되지 않은 경우 Amazon CloudWatch 경보 사용의 지침에 따라 나중에 실험 템플릿을 업데이트할 수 있습니다.
   e. Amazon CloudWatch Logs 또는 Amazon S3 버킷에 대한 선택적 실험 로그를 활성화하는 것이 좋습니다. AWS FIS에 대한 실험 로깅에 대해 자세히 알아봅니다. 적절한 리소스가 아직 구성되지 않은 경우 나중에 실험 템플릿을 업데이트할 수 있습니다.
6. 실험 템플릿 생성에서 실험 템플릿 생성을 선택합니다.
7. AWS FIS 콘솔의 실험 템플릿 보기에서 실험 시작을 선택합니다. AWS FIS 실험 템플릿 관리에 대해 자세히 알아보세요.

시나리오 내보내기

시나리오는 콘솔 전용 환경입니다. 실험 템플릿과 비슷하지만 시나리오는 완전한 실험 템플릿이 아니므로 AWS FIS로 직접 가져올 수 없습니다. 시나리오를 자체 자동화의 일부로 사용하려는 경우 다음 두 가지 경로 중 하나를 사용할 수 있습니다.

1. 의 단계에 따라 유효한 AWS FIS 실험 템플릿을 시나리오 사용 생성하고 해당 템플릿을 내보냅니다.
2. 시나리오 보기의 절차를 따르고, 3단계에 콘텐츠 탭에서 시나리오 콘텐츠를 복사 및 저장한 다음 누락된 파라미터를 수동으로 추가하여 유효한 실험 템플릿을 생성합니다.

시나리오 참조

시나리오 라이브러리에 포함된 시나리오는 가능한 경우 태그를 사용하도록 설계되었으며, 각 시나리오는 시나리오 설명의 사전 조건 및 작동 방식 섹션에서 필수 태그를 설명합니다. 미리 정의된 태그로 리소스에 태그를 지정하거나 공유 파라미터 편집 환경을 사용하여 자체 태그를 설정할 수 있습니다(참조시나리오 사용).

이 참조에서는 AWS FIS 시나리오 라이브러리의 일반적인 시나리오에 대해 설명합니다. AWS FIS 콘솔을 사용하여 지원되는 시나리오를 나열할 수도 있습니다.

자세한 내용은 AWS FIS 시나리오 라이브러리 작업 단원을 참조하십시오.

AWS FIS는 다음과 같은 Amazon EC2 시나리오를 지원합니다. 이러한 시나리오는 태그를 사용하는 인스턴스를 대상으로 합니다. 자체 태그를 사용하거나 시나리오에 포함된 기본 태그를 사용할 수 있습니다. 이러한 시나리오 중 일부는 SSM 문서를 사용합니다.

- EC2 스트레스: 인스턴스 오류 - 하나 이상의 EC2 인스턴스를 중지하여 인스턴스 오류가 미치는 영향을 살펴봅니다.

  현재 리전에서 특정 태그가 부착된 인스턴스를 대상으로 합니다. 이 시나리오에서는 이러한 인스턴스를 중지하고 작업 기간이 끝날 때(기본적으로 5분) 다시 시작합니다.
- EC2 스트레스: 디스크 - 디스크 사용률 증가가 EC2 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 부착된 현재 리전의 EC2 인스턴스를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EC2 인스턴스에 주입되는 디스크 사용량 증가를 사용자 지정할 수 있습니다(기본적으로 각 디스크 스트레스 작업에 대해 5분).
- EC2 스트레스: CPU - CPU 증가가 EC2 기반 애플리케이션에 미치는 영향을 살펴봅니다.

이 시나리오에서는 특정 태그가 부착된 현재 리전의 EC2 인스턴스를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EC2 인스턴스에 주입되는 CPU 스트레스 양 증가를 사용자 지정할 수 있습니다(기본적으로 각 CPU 스트레스 작업에 대해 5분).

- EC2 스트레스: 메모리 - 메모리 사용률 증가가 EC2 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 부착된 현재 리전의 EC2 인스턴스를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EC2 인스턴스에 주입되는 메모리 스트레스 양 증가를 사용자 지정할 수 있습니다(기본적으로 각 메모리 스트레스 작업에 대해 5분).

- EC2 스트레스: 네트워크 지연 시간 - 네트워크 지연 시간 증가가 EC2 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 부착된 현재 리전의 EC2 인스턴스를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EC2 인스턴스에 주입되는 네트워크 지연 시간 증가를 사용자 지정할 수 있습니다(기본적으로 각 지연 시간 작업에 대해 5분).

AWS FIS는 다음과 같은 Amazon EKS 시나리오를 지원합니다. 이 시나리오는 Kubernetes 애플리케이션 레이블을 사용하는 EKS 포드를 대상으로 합니다. 자체 레이블을 사용하거나 시나리오에 포함된 기본 레이블을 사용할 수 있습니다. FIS를 사용하는 EKS에 대한 자세한 내용은 EKS 포드 작업 단원을 참조하세요.

- EKS 스트레스: 포드 삭제 - 하나 이상의 포드를 삭제하여 EKS 포드 오류의 영향을 살펴보세요.

  이 시나리오에서는 애플리케이션 레이블과 연결된 현재 리전의 포드를 대상으로 합니다. 이 시나리오에서는 일치하는 모든 포드를 종료합니다. 포드의 재생성은 kubernetes 구성에 의해 제어됩니다.

- EKS 스트레스: CPU - CPU 증가가 EKS 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 애플리케이션 레이블과 연결된 현재 리전의 포드를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EKS 포드에 주입되는 CPU 스트레스 양 증가를 사용자 지정할 수 있습니다(기본적으로 각 CPU 스트레스 작업에 대해 5분).

- EKS 스트레스: 디스크 - 디스크 사용률 증가가 EKS 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 애플리케이션 레이블과 연결된 현재 리전의 포드를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EKS 포드에 주입되는 디스크 스트레스 양 증가를 사용자 지정할 수 있습니다(기본적으로 각 CPU 스트레스 작업에 대해 5분).

- EKS 스트레스: 메모리 - 메모리 사용률 증가가 EKS 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 애플리케이션 레이블과 연결된 현재 리전의 포드를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EKS 포드에 주입되는 메모리 스트레스 양 증가를 사용자 지정할 수 있습니다(기본적으로 각 메모리 스트레스 작업에 대해 5분).
- EKS 스트레스: 네트워크 지연 시간 - 네트워크 지연 시간 증가가 EKS 기반 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 애플리케이션 레이블과 연결된 현재 리전의 포드를 대상으로 합니다. 이 시나리오에서는 작업 기간 동안 대상 EKS 포드에 주입되는 네트워크 지연 시간 증가를 사용자 지정할 수 있습니다(기본적으로 각 지연 시간 작업에 대해 5분).

AWS FIS는 단일 AZ, 다중 AZ 및 다중 리전 애플리케이션에 대해 다음 시나리오를 지원합니다. 이러한 시나리오는 여러 리소스 유형을 대상으로 합니다.

- AZ Availability: Power Interruption - 가용 영역(AZ)에서 전원이 완전히 중단되었을 때 예상되는 증상을 입력합니다. AZ Availability: Power Interruption에 대해 자세히 알아보세요.
- AZ: Application Slowdown - 단일 가용 영역(AZ) 내의 리소스 간에 지연 시간을 추가하여 애플리케이션 속도를 늦춥니다. AZ: Application Slowdown에 대해 자세히 알아보세요.
- Cross-AZ: Traffic Slowdown - 패킷 손실을 주입하여 가용 영역(AZs. Cross-AZ: Traffic Slowdown에 대해 자세히 알아보세요.
- Cross-Region: Connectivity - 실험 리전에서 대상 리전으로의 애플리케이션 네트워크 트래픽을 차단하고 리전 간 데이터 복제를 일시 중지합니다. Cross-Region: Connectivity를 사용하는 방법에 대해 자세히 알아보세요.

AWS FIS 는 Amazon EBS 볼륨에 대해 다음 시나리오를 지원합니다. 이러한 시나리오는 태그를 사용하여 볼륨을 대상으로 합니다. 자체 태그를 사용하거나 시나리오에 포함된 기본 태그를 사용할 수 있습니다. 대상 볼륨은 동일한 가용 영역에 있어야 합니다. 자세한 내용은 Amazon EBS에서 결함 테스트를 참조하세요.

- EBS: Sustained Latency - 영구 I/O 지연 시간이 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 연결된 현재 가용 영역의 볼륨을 대상으로 합니다. 이 시나리오에서는 15분 동안 단일 지연 시간 작업을 사용하여 볼륨에 대한 읽기 작업의 50%와 쓰기 작업의 100%에서 500ms의 일정한 지연 시간을 주입합니다. 이 시나리오에서는 주입된 지연 시간, 주입된 I/O의 백분율 및 작업 기간을 사용자 지정할 수 있습니다.

- EBS: Increasing Latency - 애플리케이션에 대한 I/O 지연 시간 증가의 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 연결된 현재 가용 영역의 볼륨을 대상으로 합니다. 이 시나리오에서는 15분 동안 5개의 지연 시간 작업을 사용하여 볼륨에 대한 읽기 작업의 10%와 쓰기 작업의 25%에서 50ms, 200ms, 700ms, 1초 및 15초의 지연 시간을 증가시킵니다. 이 시나리오에서는 각 지연 시간 작업에 대해 주입된 지연 시간, 주입된 I/O의 백분율 및 작업 기간을 사용자 지정할 수 있습니다.
- EBS: Intermittent Latency - 간헐적인 I/O 지연 시간 스파이크가 애플리케이션에 미치는 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 연결된 현재 가용 영역의 볼륨을 대상으로 합니다. 이 시나리오에서는 볼륨에 대한 읽기 및 쓰기 I/O 작업의 0.1%에 30초, 10초 및 20초의 급격한 간헐적 지연 시간 스파이크 3개를 주입합니다.이 경우 3개의 지연 시간 작업을 사용하고 15분 동안 각 스파이크 사이에 복구 간격을 둡니다. 이 시나리오에서는 각 지연 시간 작업에 대해 주입된 지연 시간, 주입된 I/O의 백분율 및 작업 기간을 사용자 지정할 수 있습니다.
- EBS: Decreasing Latency - 애플리케이션에 대한 I/O 지연 시간 감소의 영향을 살펴봅니다.

  이 시나리오에서는 특정 태그가 연결된 현재 가용 영역의 볼륨을 대상으로 합니다. 이 시나리오에서는 15분 동안 5개의 지연 시간 작업을 사용하여 볼륨에 대한 읽기 및 쓰기 작업의 10%에서 20초, 5초, 900ms, 300ms 및 40ms의 지연 시간을 줄입니다. 이 시나리오에서는 각 지연 시간 작업에 대해 주입된 지연 시간, 주입된 I/O의 백분율 및 작업 기간을 사용자 지정할 수 있습니다.

AZ Availability: Power Interruption

AZ Availability: Power Interruption 시나리오를 사용하여 가용 영역(AZ)에서 전원이 완전히 중단될 때 예상되는 증상을 유도할 수 있습니다.

이 시나리오는 한 번의 완전한 AZ 전원 중단 시 다중 AZ 애플리케이션이 예상대로 작동하는 것을 입증하는 데 사용할 수 있습니다. 여기에는 영역 컴퓨팅(Amazon EC2, EKS 및 ECS) 손실, AZ에서 컴퓨팅 크기 조정 없음, 서브넷 연결 손실, RDS 장애 조치, ElastiCache 장애 조치, S3 Express One Zone 디렉터리 버킷에 대한 액세스 손상, 응답하지 않는 EBS 볼륨이 포함됩니다. 기본적으로 대상이 발견되지 않는 작업은 건너뛰게 됩니다.

작업

다음 작업을 함께 취하면 단일 AZ에서 완전한 정전 시 예상되는 여러 가지 증상이 나타납니다. AZ 가용성: 전원 중단은 단일 AZ 전원 중단 중에 영향을 받을 것으로 예상되는 서비스에만 영향을 줍니다. 기본적으로 시나리오는 30분 동안 전원 중단 증상을 주입한 다음 추가로 30분 동안 복구 중에 발생할 수 있는 증상을 주입합니다.

Stop-Instances

AZ 전원이 중단되는 동안 영향을 받는 AZ의 EC2 인스턴스는 종료됩니다. 전원이 복구되면 인스턴스가 재부팅됩니다. AZ Availability: Power Interruption에는 중단 기간 동안 영향을 받는 AZ의 모든 인스턴스를 중지하는 aws:ec2:stop-instances가 포함됩니다. 기간이 지나면 인스턴스가 다시 시작됩니다. Amazon EKS에서 관리하는 EC2 인스턴스를 중지하면 종속된 EKS 포드가 삭제됩니다. Amazon ECS에서 관리하는 EC2 인스턴스를 중지하면 종속된 ECS 태스크가 중지됩니다.

이 작업은 영향을 받는 AZ에서 실행 중인 EC2 인스턴스를 대상으로 합니다. 기본적으로 이 작업은 값이 `StopInstances`인 `AzImpairmentPower`라는 이름의 태그가 있는 인스턴스를 대상으로 합니다. 이 태그를 인스턴스에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 인스턴스를 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

Stop-ASG-Instances

AZ 전원이 중단되는 동안 영향을 받는 AZ의 Auto Scaling 그룹에서 관리하는 EC2 인스턴스가 종료됩니다. 전원이 복구되면 인스턴스가 재부팅됩니다. AZ Availability: Power Interruption에는 중단 기간 동안 영향을 받는 AZ에서 Auto Scaling으로 관리되는 인스턴스를 포함한 모든 인스턴스를 중지하는 aws:ec2:stop-instances가 포함됩니다. 기간이 지나면 인스턴스가 다시 시작됩니다.

이 작업은 영향을 받는 AZ에서 실행 중인 EC2 인스턴스를 대상으로 합니다. 기본적으로 이 작업은 값이 `IceAsg`인 `AzImpairmentPower`라는 이름의 태그가 있는 인스턴스를 대상으로 합니다. 이 태그를 인스턴스에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 인스턴스를 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

인스턴스 시작 일시 중지

AZ 전원이 중단되는 동안에는 AZ에서 용량을 프로비저닝하기 위한 EC2 API 호출이 실패합니다. 특히 `ec2:StartInstances` `ec2:CreateFleet` `ec2:RunInstances` 등의 API가 영향을 받습니다. AZ Availability: Power Interruption includes에는 영향을 받는 AZ에서 새 인스턴스가 프로비저닝되지 않도록 하기 위해 aws:ec2:api-insufficient-instance-capacity-error가 포함됩니다.

이 작업은 인스턴스 프로비저닝에 사용되는 IAM 역할을 대상으로 합니다. 이러한 역할은 ARN을 사용하여 타겟팅해야 합니다. 기본적으로 유효한 IAM 역할을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

ASG 스케일링 일시 중지

AZ 전원 중단 시 Auto Scaling 컨트롤 플레인이 AZ에서 손실된 용량을 복구하기 위해 호출하는 EC2 API는 실패합니다. 특히 `ec2:StartInstances` `ec2:CreateFleet` `ec2:RunInstances` 등의 API가 영향을 받습니다. AZ Availability: Power Interruption에는 영향을 받는 AZ에서 새 인스턴스를 프로

비저닝할 수 없도록 하는 aws:ec2:asg-insufficient-instance-capacity-error가 포함됩니다. 또한 영향을 받는 AZ에서 Amazon EKS 및 Amazon ECS를 확장할 수 없습니다.

이 작업은 Auto Scaling 그룹을 대상으로 합니다. 기본적으로 이 작업은 값이 `IceAsg`인 `AzImpairmentPower`라는 이름의 태그가 있는 Auto Scaling 그룹을 대상으로 합니다. 이 태그를 Auto Scaling 그룹에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 Auto Scaling 그룹을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

네트워크 연결 일시 중지

AZ 전원이 중단되는 동안에는 AZ의 네트워킹을 사용할 수 없습니다. 이 경우 일부 AWS 서비스는 영향을 받는 AZ의 프라이빗 엔드포인트를 사용할 수 없음을 반영하여 DNS를 업데이트하는 데 최대 몇 분 정도 걸릴 수 있습니다. 이 시간 동안 DNS 조회에서 액세스할 수 없는 IP 주소가 반환될 수 있습니다. AZ Availability: Power Interruption에는 영향을 받는 AZ의 모든 서브넷에 대한 모든 네트워크 연결을 2분 동안 차단하는 aws:network:disrupt-connectivity가 포함됩니다. 이렇게 하면 대부분의 애플리케이션에서 시간 초과 및 DNS 새로 고침이 강제로 실행됩니다. 2분 후에 이 작업을 종료하면 AZ를 계속 사용할 수 없는 동안 리전 서비스 DNS를 나중에 복구할 수 있습니다.

이 작업은 서브넷을 대상으로 합니다. 기본적으로 이 작업은 값이 `DisruptSubnet`인 `AzImpairmentPower`라는 이름의 태그가 있는 클러스터를 대상으로 합니다. 이 태그를 서브넷에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 서브넷을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

장애 조치 RDS

AZ 전원이 중단되는 동안 영향을 받는 AZ의 RDS 노드는 종료됩니다. 영향을 받는 AZ의 단일 AZ RDS 노드는 완전히 사용할 수 없게 됩니다. 다중 AZ 클러스터의 경우, 쓰기 노드는 영향을 받지 않는 AZ로 장애 조치되고 영향을 받는 AZ의 읽기 노드는 사용할 수 없게 됩니다. 다중 AZ 클러스터의 경우, AZ Availability: Power Interruption에는 쓰기기가 영향을 받는 AZ에 있는 경우 장애 조치할 aws:rds:failover-db-cluster가 포함됩니다.

이 작업은 RDS 클러스터를 대상으로 합니다. 기본적으로 이 작업은 값이 `DisruptRds`인 `AzImpairmentPower`라는 이름의 태그가 있는 클러스터를 대상으로 합니다. 이 태그를 클러스터에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 클러스터를 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

ElastiCache 복제 그룹 일시 중지

AZ 전원 중단 중에는 AZ의 ElastiCache 노드를 사용할 수 없습니다. AZ Availability: Power Interruption에는 영향을 받는 AZ에서 ElastiCache 노드를 종료하기 위한 aws:elasticache:replicationgroup-

interrupt-az-power가 포함되어 있습니다. 중단 기간 동안 영향을 받는 AZ에 새 인스턴스가 프로비저닝되지 않으므로 복제 그룹은 감소된 용량을 유지합니다.

이 작업은 ElastiCache 복제 그룹을 대상으로 합니다. 기본적으로 이름이 이고 값이 인 태그가 `AzImpairmentPower` 있는 복제 그룹을 대상으로 합니다`ElasticacheImpact`. 이 태그를 복제 그룹에 추가하거나 실험 템플릿에서 기본 태그를 자체 태그로 바꿀 수 있습니다. 기본적으로 유효한 복제 그룹을 찾을 수 없는 경우이 작업은 건너뜁니다. 영향을 받는 AZ에 노드가 있는 복제 그룹만 유효한 대상으로 간주됩니다.

ARC 영역 자동 전환 시작

AZ 전원 중단이 시작된 후 5분이 지나면 복구 작업은 남은 25분 동안 리소스 트래픽을 지정된 AZ에서 `aws:arc:start-zonal-autoshift` 자동으로 이동합니다. 이 기간이 지나면 트래픽이 원래 AZ로 다시 이동합니다. 실제 AZ 전력 중단 중에는 자동 전환이 활성화된 경우 장애가 AWS 감지되고 리소스 트래픽이 이동합니다. 이 교대 근무 시기는 다르지만 장애 발생 시점으로부터 5분 후에 발생할 것으로 예상됩니다.

이 작업은 Amazon Application Recovery Controller(ARC) 자동 전환 지원 리소스를 대상으로 합니다. 기본적으로 태그 키`AzImpairmentPower`와 값를 사용하여 리소스를 대상으로 합니다`RecoverAutoshiftResources`. 이 태그를 리소스에 추가하거나 실험 템플릿에서 기본 태그를 자체 태그로 바꿀 수 있습니다. 예를 들어 애플리케이션별 태그를 사용할 수 있습니다. 기본적으로 유효한 리소스를 찾을 수 없는 경우이 작업은 건너뜁니다.

EBS I/O 일시 중지

AZ 전원 중단 후 전원이 복구되면 극히 일부 인스턴스에서 EBS 볼륨이 응답하지 않는 현상이 발생할 수 있습니다. AZ Availability: Power Interruption에는 aws:ebs:pause-io가 포함되어 EBS 볼륨 1개를 응답하지 않는 상태로 남겨둡니다.

기본적으로 인스턴스가 종료된 후에도 지속되도록 설정된 볼륨만 대상이 됩니다. 이 작업은 값이 `APIPauseVolume`인 `AzImpairmentPower`라는 태그가 있는 볼륨을 대상으로 합니다. 이 태그를 볼륨에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 볼륨을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

S3 Express One Zone 디렉터리 버킷에 대한 연결 중단

AZ 전원 중단 중에는 AZ의 S3 Express One Zone 디렉터리 버킷에 저장된 데이터에 액세스할 수 없습니다. AZ 가용성: 전원 중단에는 실험 기간 동안 영향을 받는 AZ의 서브넷과 One Zone 디렉터리 버킷 간의 연결을 방해하는 aws:network:disrupt-connectivity가 포함되어 있어 영역 엔드포인트 데이터 영역 API 작업에 시간 초과가 발생합니다. 컴퓨팅이 AZ의 스토리지와 함께 배치될 때 중단을 테스트하려면 이 작업을 사용합니다.

이 작업은 서브넷을 대상으로 합니다. 기본적으로 값이 인 태그AzImpairmentPower가 있는 서브넷을 대상으로 합니다DisruptSubnet. 이 태그를 서브넷에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 서브넷을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

제한 사항

- 이 시나리오에는 중지 조건이 포함되어 있지 않습니다. 애플리케이션에 맞는 올바른 중지 조건을 실험 템플릿에 추가해야 합니다.
- 대상 AZ에서는 EC2에서 실행되는 Amazon EKS 포드가 EC2 워커 노드와 함께 종료되고 새 EC2 노드의 시작이 차단됩니다. 하지만 AWS Fargate에서 실행되는 Amazon EKS 포드는 지원되지 않습니다.
- 대상 AZ에서는 EC2에서 실행되는 Amazon ECS 태스크가 EC2 워커 노드와 함께 종료되고 새 EC2 노드의 시작이 차단됩니다. 하지만 AWS Fargate에서 실행되는 Amazon ECS 태스크는 지원되지 않습니다.
- 읽기 가능한 대기 DB 인스턴스 2개가 있는 Amazon RDS 다중AZ는 지원되지 않습니다. 이 경우 인스턴스가 종료되고, RDS가 장애 조치되며, 영향을 받는 AZ에서 용량이 즉시 다시 프로비저닝됩니다. 영향을 받는 AZ의 읽기 가능한 대기는 계속 사용할 수 있습니다.

요구 사항

- AWS FIS 실험 역할에 필요한 권한을 추가합니다.
- 리소스 태그는 실험의 대상이 되는 리소스에 적용해야 합니다. 자체 태그 지정 규칙 또는 시나리오에 정의된 기본 태그를 사용할 수 있습니다.

권한

ARC 영역 자동 전환은 IAM 서비스 연결 역할을 AWSServiceRoleForZonalAutoshiftPracticeRun 사용하여 사용자를 대신하여 영역 전환을 수행합니다. 이 역할은 IAM 관리형 정책를 사용합니다AWSZonalAutoshiftPracticeRunSLRPolicy. 역할을 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS SDK의 AZ 전원 중단 시나리오에서 실험 템플릿을 생성하면 ARC가 서비스 연결 역할을 생성합니다. 자세한 내용은 ARC에서 영역 자동 전환에 서비스 연결 역할 사용을 참조하세요.

다음 정책은 AZ Availability: Power Interruption 시나리오로 실험을 실행하는 데 필요한 권한을 AWS FIS에 부여합니다. 이 정책은 실험 역할에 연결되어야 합니다.

JSON

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

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

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

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

```
        },
        {
            "Sid": "DescribeAsg",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

시나리오 콘텐츠

다음 콘텐츠는 시나리오를 정의합니다. 이 JSON을 저장하여 AWS Command Line Interface(AWS CLI)에서 create-experiment-template 명령을 사용하여 실험 템플릿을 만드는 데 사용할 수 있습니다. 최신 버전의 시나리오를 보려면 FIS 콘솔의 시나리오 라이브러리를 방문하세요.

```
{
    "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."
}
```

AZ: Application Slowdown

AZ: Application Slowdown 시나리오를 사용하여 단일 가용 영역(AZ) 내의 리소스 간에 추가 지연 시간을 도입할 수 있습니다. 이 지연 시간으로 인해 애플리케이션 속도가 느려지는 많은 증상, 즉 부분 중단이 발생하며, 회색 장애라고도 합니다. 대상 리소스 간의 네트워크 흐름에 지연 시간을 추가합니다. 네트워크 흐름은 요청, 응답 및 서버, 컨테이너 및 서비스 간의 기타 통신을 전달하는 데이터 패킷인 컴퓨팅 리소스 간의 트래픽을 나타냅니다. 시나리오는 관찰성 설정을 검증하고, 경보 임계값을 조정하고, 속도 저하에 대한 애플리케이션 민감도를 검색하고, AZ 대피와 같은 중요한 운영 결정을 연습하는 데 도움이 될 수 있습니다.

기본적으로 시나리오는 선택한 AZ 내의 대상 리소스 간 네트워크 흐름의 100%에 30분 동안 200ms의 지연 시간을 추가합니다. AWS FIS 콘솔의 공유 파라미터 편집 대화 상자를 사용하여 시나리오 수준에서 다음 파라미터를 조정한 다음 기본 작업에 적용할 수 있습니다.

- 가용 영역 - 시나리오에서 손상시킬 AZ를 선택할 수 있습니다.
- 밀리초(ms) 지연 시간 - 애플리케이션의 민감도와 요구 사항에 따라 이를 조정합니다. 예를 들어 더 민감한 애플리케이션의 경우 지연 시간을 더 낮게 설정하거나 제한 시간 처리를 테스트하도록 더 높게 설정할 수 있습니다. 현재 애플리케이션 지연 시간의 배수를 기준으로 사용하는 것이 좋습니다.
- 흐름 백분율 - 트래픽의 하위 집합을 손상하도록 줄입니다. 예를 들어 네트워크 흐름의 25%에 영향을 미치는 200ms 지연 시간을 추가하여 훨씬 더 미세한 테스트를 수행할 수 있습니다.

- 기간 - 실험이 실행되는 기간을 설정합니다. 더 빠른 테스트를 위해 단축하거나 더 오래 지속되는 테스트를 실행할 수 있습니다. 예를 들어 손상된 조건에서 복구 메커니즘을 테스트하려면 기간을 2시간으로 설정합니다.
- 리소스 대상 지정 - 태그(EC2 인스턴스 또는 EC2 또는 Fargate의 ECS 작업의 경우) 또는 레이블(ECEC2의 EKS 포드의 경우)을 사용하여 전체 시나리오에 대한 대상 리소스를 정의할 수 있습니다. 자체 태그와 레이블을 지정하거나 시나리오에 제공된 기본값을 사용할 수 있습니다. 태그 또는 레이블을 사용하지 않으려면 다른 파라미터를 지정하여 대상 리소스에 대한 작업을 편집할 수 있습니다.
- 사용자 지정 - EC2 또는 ECS 리소스를 대상으로 지정하지 않으려면 작업을 기본 태그와 함께 그대로 둘 수 있습니다. 실험에서 대상으로 지정할 리소스를 찾지 못하며 작업을 건너뜁니다. 그러나 EKS 리소스를 대상으로 지정하지 않으려면 EKS 클러스터 식별자를 제공해야 하므로 시나리오에서 EKS 작업 및 대상을 완전히 제거해야 합니다. 보다 세분화된 사용자 지정을 위해 실험 템플릿에서 개별 작업을 직접 수정할 수 있습니다.

작업

다음 작업을 함께 수행하면 네트워크 흐름에 추가 지연 시간을 도입하여 단일 AZ에서 애플리케이션 속도 저하의 많은 증상을 생성한 다음 애플리케이션을 통해 전파할 수 있습니다. 이러한 작업은 병렬로 실행되며, 각 작업은 기본적으로 30분 동안 200ms의 지연 시간을 추가합니다. 이 기간이 지나면 지연 시간이 정상 수준으로 돌아갑니다. 시나리오를 실행하려면 EC2 인스턴스, ECS 작업 또는 EKS 포드 중 하나 이상의 리소스 유형이 필요합니다.

ECS 네트워크 지연 시간

AZ: Application Slowdown에는 ECS 작업에 지연 시간을 도입하기 위한 aws:ecs:task-network-latency가 포함되어 있습니다. 작업은 선택한 AZ의 작업을 대상으로 합니다. 기본적으로 태그 https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-using-tags.html 이름이 이고 값이 인 태스크AZApplicationSlowdown를 대상으로 합니다LatencyForECS. 기본 태그를 자체 태그로 바꾸거나 태스크에 시나리오 태그를 추가할 수 있습니다. 유효한 작업을 찾을 수 없는 경우이 작업은 건너뜁니다. ECS에서 실험을 실행하기 전에 ECS 작업 작업에 대한 설정 단계를 따라야 합니다.

EKS 네트워크 지연 시간

AZ: Application Slowdown에는 EKS 포드의 지연 시간을 도입하기 위한 aws:eks:eks:pod-network-latency가 포함되어 있습니다. 작업은 선택한 AZ의 포드를 대상으로 합니다. 기본적으로 key=value 형식의 레이블이 있는 클러스터 내의 포드를 대상으로 합니다. 제공된 기본 레이블은 입니다AZApplicationSlowdown=LatencyForEKS. 기본 레이블을 자체 레이블로 바꾸거나 포드에이 레이블을 추가할 수 있습니다. 유효한 포드를 찾을 수 없는 경우이 작업은 건너뜁니다. EKS에서 실험을 실행하기 전에 EKS 포드 작업에 대한 설정 단계를 따라야 합니다.

EC2 네트워크 지연 시간

AZ: Application Slowdown은 aws:ssm:send-command 작업을 사용하여 AWSFIS-Run-Network-Latency-Sources 문서를 실행하여 EC2 인스턴스에 지연 시간을 도입합니다. 작업은 선택한 AZ의 인스턴스를 대상으로 합니다. 기본적으로 값이 인 태그AZApplicationSlowdown가 있는 인스턴스를 대상으로 합니다LatencyForEC2. 기본 태그를 자체 태그로 바꾸거나이 태그를 인스턴스에 추가할 수 있습니다. 유효한 인스턴스를 찾을 수 없는 경우이 작업은 건너뜁니다. SSM을 사용하여 EC2에서 실험을 실행하기 전에 AWS Systems Manager 에이전트를 구성해야 합니다.

제한 사항

- 이 시나리오에는 중지 조건이 포함되어 있지 않습니다. 애플리케이션에 맞는 올바른 중지 조건을 실험 템플릿에 추가해야 합니다.

요구 사항

- AWS FIS 실험 역할에 필요한 권한을 추가합니다.
- 선택한 AZ 내에서 EC2 인스턴스, ECS 작업 또는 EKS 포드의 세 가지 유형 중 하나 이상의 리소스를 대상으로 지정해야 합니다.
- 시나리오의 모든 대상은 동일한 VPC에 있어야 합니다.

권한

이 시나리오를 실행하려면 FIS가 실험에서 대상으로 하는 리소스 유형에 대한 역할 및 관리형 정책을 수임하도록 허용하는 신뢰 정책이 있는 IAM 역할이 필요합니다. EC2, ECS 및 EKS. AZ: Application Slowdown 시나리오에서 실험 템플릿을 생성하면 FIS는 신뢰 정책 및 다음 AWS 관리형 정책을 사용하여 역할을 생성합니다.

- AWSFaultInjectionSimulatorEC2Access
- AWSFaultInjectionSimulatorECSAccess
- AWSFaultInjectionSimulatorEKSAccess

기존 IAM 역할을 사용하여 AZ: Application Slowdown 시나리오를 실행하는 경우 다음 정책을 연결하여 AWS FIS에 필요한 권한을 부여할 수 있습니다.

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

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

```
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ssm:*:*:managed-instance/*",
                "arn:aws:ssm:*:*:document/*"
            ]
        },
        {
            "Sid": "ListCommands",
            "Effect": "Allow",
            "Action": [
                "ssm:ListCommands"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CancelCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:CancelCommand"
            ],
            "Resource": "*"
        }
    ]
}
```

시나리오 콘텐츠

다음 콘텐츠는 시나리오를 정의합니다. 이 JSON을 저장하여 AWS Command Line Interface(AWS CLI)에서 create-experiment-template 명령을 사용하여 실험 템플릿을 만드는 데 사용할 수 있습니다. 시나리오의 최신 버전을 보려면 FIS 콘솔에서 시나리오 라이브러리를 방문하여 콘텐츠 탭으로 이동합니다.

```
{
    "tags": {
        "Name": "AZ: Application Slowdown"
    },
    "description": "Add latency between resources within a single AZ.",
    "actions": {
        "LatencyForEKS": {
            "actionId": "aws:eks:pod-network-latency",
            "parameters": {
                "delayMilliseconds": "200",
                "duration": "PT30M",
                "flowsPercent": "100",
```

```
                "interface": "DEFAULT",
                "kubernetesServiceAccount": "fis-service-account",
                "sources": "us-east-1a"
            },
            "targets": {
                "Pods": "TargetsForEKS"
            }
        },
        "LatencyForEC2": {
            "actionId": "aws:ssm:send-command",
            "parameters": {
                "duration": "PT30M",
                "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Latency-Sources",
                "documentParameters": "{\"DelayMilliseconds\":\"200\",\"Sources\":\"us-
east-1a\",\"Interface\":\"DEFAULT\",\"TrafficType\":\"egress\",\"DurationSeconds\":
\"1800\",\"FlowsPercent\":\"100\",\"InstallDependencies\":\"True\"}"
            },
            "targets": {
                "Instances": "TargetsForEC2"
            }
        },
        "LatencyForECS": {
            "actionId": "aws:ecs:task-network-latency",
            "parameters": {
                "delayMilliseconds": "200",
                "duration": "PT30M",
                "flowsPercent": "100",
                "installDependencies": "true",
                "sources": "us-east-1a",
                "useEcsFaultInjectionEndpoints": "true"
            },
            "targets": {
                "Tasks": "TargetsForECS"
            },
            "startAfter": []
        }
    },
    "targets": {
        "TargetsForEKS": {
            "parameters": {
                "availabilityZoneIdentifier": "us-east-1a",
                "clusterIdentifier": "",
                "namespace": "default",
```

```
                "selectorType": "labelSelector",
                "selectorValue": "AZApplicationSlowdown=LatencyForEKS"
            },
            "resourceType": "aws:eks:pod",
            "selectionMode": "ALL"
        },
        "TargetsForEC2": {
            "filters": [
                {
                    "path": "Placement.AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "resourceTags": {
                "AZApplicationSlowdown": "LatencyForEC2"
            },
            "resourceType": "aws:ec2:instance",
            "selectionMode": "ALL"
        },
        "TargetsForECS": {
            "filters": [
                {
                    "path": "AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "resourceTags": {
                "AZApplicationSlowdown": "LatencyForECS"
            },
            "resourceType": "aws:ecs:task",
            "selectionMode": "ALL"
        }
    },
    "experimentOptions": {
        "accountTargeting": "single-account",
        "emptyTargetResolutionMode": "skip"
    },
    "stopConditions": [
        {
            "source": "none"
```

```
        }
    ]
}
```

Cross-AZ: Traffic Slowdown

교차 AZ: 트래픽 속도 저하 시나리오를 사용하여 패킷 손실을 주입하여 가용 영역(AZs. 패킷 손실은 회색 장애라고도 하는 부분 중단인 교차 AZ 통신을 손상시킵니다. 대상 리소스 간의 네트워크 흐름에 패킷 손실을 주입합니다. 네트워크 흐름은 요청, 응답 및 서버, 컨테이너 및 서비스 간의 기타 통신을 전달하는 데이터 패킷인 컴퓨팅 리소스 간의 트래픽을 나타냅니다. 시나리오는 관찰성 설정을 검증하고, 경보 임계값을 조정하고, 교차 AZ 통신에서 애플리케이션 민감도 및 종속성을 검색하고, AZ 대피와 같은 중요한 운영 결정을 연습하는 데 도움이 될 수 있습니다.

기본적으로 시나리오는 30분 동안 선택한 AZ의 대상 리소스에 대한 아웃바운드 네트워크 흐름의 100%에 15% 패킷 손실을 주입합니다. AWS FIS 콘솔의 공유 파라미터 편집 대화 상자를 사용하여 시나리오 수준에서 다음 파라미터를 조정한 다음 기본 작업에 적용할 수 있습니다.

- 가용 영역 - 손상시킬 AZ를 선택할 수 있으며 패킷 손실이 해당 AZ에서 리전 내의 다른 AZs로 주입됩니다.
- 패킷 손실 - 5% 이상과 같은 미세한 중단 테스트의 경우 패킷 손실을 낮게 조정하여 총 연결 영향의 경우 50% 또는 100%와 같은 심각한 통신 성능 저하 및 복구 메커니즘을 테스트합니다.
- 흐름 백분율 - 트래픽의 하위 집합을 손상하도록 줄입니다. 예를 들어 네트워크 흐름의 25%에 영향을 미치는 15% 패킷 손실을 주입하여 훨씬 더 미세한 테스트를 수행할 수 있습니다.
- 기간 - 실험이 실행되는 기간을 설정합니다. 더 빠른 테스트를 위해 단축하거나 더 오래 지속되는 테스트를 실행할 수 있습니다. 예를 들어 손상된 조건에서 복구 메커니즘을 테스트하는 데 도움이 되도록 기간을 2시간으로 설정합니다.
- 리소스 대상 지정 - 태그(EC2 인스턴스 또는 EC2 또는 Fargate의 ECS 작업의 경우) 또는 레이블(ECEC2의 EKS 포드의 경우)을 사용하여 전체 시나리오에 대한 대상 리소스를 정의할 수 있습니다. 자체 태그와 레이블을 지정하거나 시나리오에 제공된 기본값을 사용할 수 있습니다. 태그 또는 레이블을 사용하지 않으려면 다른 파라미터를 지정하여 대상 리소스에 대한 작업을 편집할 수 있습니다.
- 사용자 지정 - EC2 또는 ECS 리소스를 대상으로 지정하지 않으려면 작업을 기본 태그와 함께 그대로 둘 수 있습니다. 실험에서 대상으로 지정할 리소스를 찾지 못하며 작업을 건너뜁니다. 그러나 EKS 리소스를 대상으로 지정하지 않으려면 EKS 클러스터 식별자를 제공해야 하므로 시나리오에서 EKS 작업 및 대상을 완전히 제거해야 합니다. 보다 세분화된 사용자 지정을 위해 실험 템플릿에서 개별 작업을 직접 수정할 수 있습니다.

작업

다음 작업은 대상 AZs에서 네트워크 계층의 리전에 있는 다른 AZ로의 아웃바운드 통신에 패킷 손실을 도입하여 AZs 간 트래픽 속도 저하의 증상을 생성하는 데 도움이 됩니다. 이러한 작업은 병렬로 실행되며, 각 작업은 기본적으로 30분 동안 15% 패킷 손실을 주입합니다. 이 기간이 지나면 통신이 정상으로 돌아갑니다. 시나리오를 실행하려면 선택한 AZ에 EC2 인스턴스, ECS 작업 또는 EKS 포드 중 하나 이상의 리소스 유형이 필요합니다.

ECS 네트워크 패킷 손실

교차 AZ: 트래픽 속도 저하에는 ECS 작업에 대한 패킷 손실을 주입하기 위한 aws:ecs:task-network-packet-loss가 포함됩니다. 작업은 선택한 AZ의 작업을 대상으로 하며 리전의 다른 모든 AZs 대한 아웃바운드 통신을 손상시킵니다. 작업을 편집하고 `Sources` 필드에서 AZs를 추가하거나 제거하여 영향 범위를 추가로 사용자 지정할 수 있습니다. 기본적으로 태그 https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-using-tags.html 이름이 이고 값이 인 작업을 대상으로 `CrossAZTrafficSlowdown` 합니다`PacketLossForECS`. 기본 태그를 자체 태그로 바꾸거나 태스크에 시나리오 태그를 추가할 수 있습니다. 유효한 작업을 찾을 수 없는 경우이 작업은 건너뜁니다. ECS에서 실험을 실행하기 전에 ECS 작업 작업에 대한 설정 단계를 따라야 합니다.

EKS 네트워크 패킷 손실

교차 AZ: 트래픽 속도 저하에는 EKS 포드의 패킷 손실을 주입하기 위한 aws:eks:eks:pod-network-packet-loss가 포함됩니다. 작업은 선택한 AZ의 포드를 대상으로 하며 리전의 다른 모든 AZs에 대한 아웃바운드 통신을 손상시킵니다. 작업을 편집하고 `Sources` 필드에서 AZs를 추가하거나 제거하여 영향 범위를 추가로 사용자 지정할 수 있습니다. 기본적으로 key=value 형식의 레이블이 있는 클러스터 내의 포드를 대상으로 합니다. 제공된 기본 레이블은 입니다`CrossAZTraffic=PacketLossForEKS`. 기본 레이블을 자체 레이블로 바꾸거나 포드에이 레이블을 추가할 수 있습니다. 유효한 포드를 찾을 수 없는 경우이 작업은 건너뜁니다. EKS에서 실험을 실행하기 전에 EKS 포드 작업에 대한 설정 단계를 따라야 합니다.

EC2 네트워크 패킷 손실

Cross-AZ: Traffic Slowdown은 aws:ssm:send-command 작업을 사용하여 AWSFIS-Run-Network-Packet-Loss-Sources 문서를 실행하여 EC2 인스턴스에 대한 패킷 손실을 주입하고 리전의 다른 모든 AZs에 대한 아웃바운드 통신을 손상시킵니다. 작업을 편집하고 `Sources` 필드에서 AZs를 추가하거나 제거하여 영향 범위를 추가로 사용자 지정할 수 있습니다. 작업은 선택한 AZ의 인스턴스를 대상으로 합니다. 기본적으로 값이 인 태그`CrossAZTrafficSlowdown`가 있는 인스턴스를 대상으로 합니다`PacketLossForEC2`. 기본 태그를 자체 태그로 바꾸거나이 태그를 인스턴스에 추가할 수 있습니다. 유효한 인스턴스를 찾을 수 없는 경우이 작업은 건너뜁니다. SSM을 사용하여 EC2에서 실험을 실행하기 전에 AWS Systems Manager 에이전트를 구성해야 합니다.

제한 사항

- 이 시나리오에는 중지 조건이 포함되어 있지 않습니다. 애플리케이션에 맞는 올바른 중지 조건을 실험 템플릿에 추가해야 합니다.

요구 사항

- AWS FIS 실험 역할에 필요한 권한을 추가합니다.
- 선택한 AZ 내에서 EC2 인스턴스, ECS 작업 또는 EKS 포드의 세 가지 유형 중 하나 이상의 리소스를 대상으로 지정해야 합니다.
- 시나리오의 모든 대상은 동일한 VPC에 있어야 합니다.

권한

이 시나리오를 실행하려면 FIS가 실험에서 대상으로 하는 리소스 유형에 대한 역할 및 관리형 정책을 수임하도록 허용하는 신뢰 정책이 있는 IAM 역할이 필요합니다. EC2, ECS 및 EKS. 교차 AZ: 트래픽 속도 저하 시나리오에서 실험 템플릿을 생성하면 FIS는 신뢰 정책 및 다음 AWS 관리형 정책을 사용하여 역할을 생성합니다.

- AWSFaultInjectionSimulatorEC2Access
- AWSFaultInjectionSimulatorECSAccess
- AWSFaultInjectionSimulatorEKSAccess

기존 IAM 역할을 사용하여 교차 AZ: 트래픽 속도 저하 시나리오를 실행하는 경우 다음 정책을 연결하여 AWS FIS에 필요한 권한을 부여할 수 있습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DescribeTasks",
            "Effect": "Allow",
            "Action": "ecs:DescribeTasks",
            "Resource": "*"
        },
        {
            "Sid": "DescribeContainerInstances",
            "Effect": "Allow",
```

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

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "CancelCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:CancelCommand"
            ],
            "Resource": "*"
        }
    ]
}
```

시나리오 콘텐츠

다음 콘텐츠는 시나리오를 정의합니다. 이 JSON을 저장하여 AWS Command Line Interface(AWS CLI)에서 create-experiment-template 명령을 사용하여 실험 템플릿을 만드는 데 사용할 수 있습니다. 시나리오의 최신 버전을 보려면 FIS 콘솔에서 시나리오 라이브러리를 방문하여 콘텐츠 탭으로 이동합니다.

```
{
    "tags": {
        "Name": "Cross-AZ: Traffic Slowdown"
    },
    "description": "Inject packet loss to disrupt and slow down traffic between AZs.",
    "actions": {
        "PacketLossForEC2": {
            "actionId": "aws:ssm:send-command",
            "parameters": {
                "duration": "PT30M",
                "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Packet-Loss-Sources",
                "documentParameters": "{\"Sources\":\"us-east-1b,us-east-1c,us-
east-1d,us-east-1e,us-east-1f\",\"LossPercent\":\"15\",\"Interface\":\"DEFAULT\",
\"TrafficType\":\"egress\",\"DurationSeconds\":\"1800\",\"FlowsPercent\":\"100\",
\"InstallDependencies\":\"True\"}"
            },
            "targets": {
                "Instances": "TargetsForEC2"
            }
        },
        "PacketLossForECS": {
```

```
            "actionId": "aws:ecs:task-network-packet-loss",
            "parameters": {
                "sources": "us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f",
                "lossPercent": "15",
                "duration": "PT30M",
                "flowsPercent": "100",
                "installDependencies": "true",
                "useEcsFaultInjectionEndpoints": "true"
            },
            "targets": {
                "Tasks": "TargetsForECS"
            }
        },
        "PacketLossForEKS": {
            "actionId": "aws:eks:pod-network-packet-loss",
            "parameters": {
                "sources": "us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f",
                "lossPercent": "15",
                "duration": "PT30M",
                "flowsPercent": "100",
                "interface": "DEFAULT",
                "kubernetesServiceAccount": "fis-service-account"
            },
            "targets": {
                "Pods": "TargetsForEKS"
            }
        }
    },
    "targets": {
        "TargetsForEC2": {
            "filters": [
                {
                    "path": "Placement.AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "resourceTags": {
                "CrossAZTrafficSlowdown": "PacketLossForEC2"
            },
            "resourceType": "aws:ec2:instance",
            "selectionMode": "ALL"
        },
```

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

Cross-Region: Connectivity

Cross-Region: Connectivity 시나리오를 사용하여 실험 리전에서 대상 리전으로의 애플리케이션 네트워크 트래픽을 차단하고 Amazon S3 및 Amazon DynamoDB 다중 리전 글로벌 테이블에 대한 리전 간 복제를 일시 중지할 수 있습니다. 교차 리전: 연결은 실험을 실행하는 리전(실험 리전)에서의 아웃바운

드 애플리케이션 트래픽에 영향을 미칩니다. 실험 리전(대상 리전)에서 격리하려는 리전으로부터의 상태 비저장 인바운드 트래픽은 차단되지 않을 수 있습니다. WS 관리형 서비스의 트래픽은 차단되지 않을 수 있습니다.

이 시나리오는 실험 리전에서 대상 리전의 리소스에 액세스할 수 없는 경우 다중 리전 애플리케이션이 예상대로 작동하는지 시연하는 데 사용할 수 있습니다. 여기에는 전송 게이트웨이와 라우트 테이블을 대상으로 실험 리전에서 대상 리전으로의 네트워크 트래픽을 차단하는 것이 포함됩니다. 또한 S3 및 DynamoDB 글로벌 테이블에 대한 교차 리전 복제를 일시 중지합니다. 기본적으로 대상이 발견되지 않는 작업은 건너뛰게 됩니다.

작업

다음 작업은 함께 포함된 AWS 서비스에 대한 교차 리전 연결을 차단합니다. 작업은 병렬로 실행됩니다. 기본적으로 시나리오는 3시간 동안 트래픽을 차단하며, 최대 12시간까지 늘릴 수 있습니다.

Transit Gateway 연결 중단

Cross Region: Connectivity에는 실험 리전의 VPC에서 전송 게이트웨이로 연결된 대상 리전의 VPC로의 교차 리전 네트워크 트래픽을 차단하기 위한 aws:network:transit-gateway-disrupt-cross-region-connectivity가 포함됩니다. 이 작업은 실험 리전 내의 VPC 엔드포인트에 대한 액세스에는 영향을 미치지 않지만, 실험 리전에서 대상 리전의 VPC 엔드포인트로 향하는 트래픽을 차단합니다.

이 작업은 실험 리전과 대상 리전을 연결하는 전송 게이트웨이를 대상으로 합니다. 기본적으로 이 작업은 값이 `Allowed`인 `DisruptTransitGateway`라는 이름의 태그가 있는 전송 게이트웨이를 대상으로 합니다. 이 태그를 전송 게이트웨이에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 전송 게이트웨이를 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

서브넷 연결 중단

Cross Region: Connectivity에는 실험 리전의 VPC에서 대상 리전의 퍼블릭 AWS IP 블록으로의 교차 리전 네트워크 트래픽을 차단하기 위한 aws:network:route-table-disrupt-cross-region-connectivity가 포함됩니다. 이러한 퍼블릭 IP 블록에는 대상 리전의 AWS 서비스 엔드포인트(예: S3 리전 엔드포인트)와 관리형 서비스용 AWS IP 블록(예: 로드 밸런서 및 Amazon API Gateway에 사용되는 IP 주소)이 포함됩니다. 이 작업은 실험 리전에서 대상 리전으로의 교차 리전 VPC 피어링 연결을 통한 네트워크 연결도 차단합니다. 이는 실험 리전 내의 VPC 엔드포인트에 대한 액세스에는 영향을 미치지 않지만, 실험 리전에서 대상 리전의 VPC 엔드포인트로 향하는 트래픽을 차단합니다.

이 작업은 실험 리전의 서브넷을 대상으로 합니다. 기본적으로 이 작업은 값이 `Allowed`인 `DisruptSubnet`라는 이름의 태그가 있는 서브넷을 대상으로 합니다. 이 태그를 서브넷에 추가하거나

실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 서브넷을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

VPC 엔드포인트 연결 중단

Cross Region: Connectivity 에는 대상 VPC 엔드포인트와 연결된 서비스에 대한 aws:network:disrupt-vpc-endpoint 중단 연결이 포함됩니다. 예를 들어 VPC 엔드포인트가 com.amazonaws.us-east-1.ec2 에 대한 프라이빗 링크를 생성하면 해당 서비스에 대한 연결이 중단됩니다.

이 작업은 실험 리전의 VPC 엔드포인트를 대상으로 합니다. 기본적으로 값이 인 DisruptVpcEndpoint 라는 태그가 있는 인터페이스 VPC 엔드포인트를 대상으로 합니다`Allowed`. 이 태그를 VPC 엔드포인트에 추가하거나 실험 템플릿에서 기본 태그를 자체 태그로 바꿀 수 있습니다. 기본적으로 유효한 VPC 엔드포인트를 찾을 수 없는 경우이 작업은 건너뜁니다.

S3 복제 일시 중지

Cross Region: Connectivity에는 대상 버킷에 대해 실험 리전에서 대상 리전으로의 S3 복제를 일시 중지하는 aws:s3:bucket-pause-replication이 포함됩니다. 대상 리전에서 실험 리전으로의 복제는 영향을 받지 않습니다. 시나리오가 종료되면 일시 중지된 시점부터 버킷 복제가 다시 시작됩니다. 복제가 모든 객체를 동기화 상태로 유지하는 데 걸리는 시간은 실험 기간과 버킷에 대한 객체 업로드 속도에 따라 달라질 수 있습니다.

이 작업은 대상 리전의 S3 버킷에 교차 리전 복제(CPR)를 사용하도록 설정한 실험 리전의 S3 버킷을 대상으로 합니다. 기본적으로 이 작업은 값이 `Allowed`인 `DisruptS3`라는 이름의 태그가 있는 버킷을 대상으로 합니다. 이 태그를 버킷에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 버킷을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

DynamoDB 복제 일시 중지

Cross-Region: Connectivity에는 실험 리전과 대상 리전을 포함한 다른 모든 리전 간의 복제를 일시 중지하는 aws:dynamodb:global-table-pause-replication이 포함됩니다. 이렇게 하면 실험 리전 안팎으로의 복제는 방지되지만 다른 리전 간의 복제에는 영향을 미치지 않습니다. 시나리오가 종료되면 일시 중지된 시점부터 테이블 복제가 다시 시작됩니다. 복제가 모든 데이터를 동기화 상태로 유지하는 데 걸리는 시간은 실험 기간과 테이블 변경 속도에 따라 달라질 수 있습니다.

이 작업은 DynamoDB 다중 리전을 강력하게 대상으로 하며 실험 리전에서 최종적으로 일관된 전역 테이블을 모두 대상으로 합니다. 기본적으로 이 작업은 값이 `Allowed`인 `DisruptDynamoDb`라는 이름의 태그가 있는 테이블을 대상으로 합니다. 이 태그를 테이블에 추가하거나 실험 템플릿에서 기본 태그를 사용자 지정 태그로 바꿀 수 있습니다. 기본적으로 유효한 글로벌 테이블을 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

MemoryDB 다중 리전 복제 일시 중지

Cross-Region: Connectivity 에는 실험 리전의 리전 멤버 클러스터에서 대상 다중 리전 클러스터의 나머지 클러스터로의 복제를 일시 중지하는 aws:memorydb:multi-region-cluster-pause-replication이 포함되어 있습니다. 다른 리전 멤버 클러스터 간의 복제는 영향을 받지 않습니다. 시나리오가 종료되면 일시 중지된 시점부터 복제가 재개됩니다. 복제가 멤버 클러스터 간에 데이터를 동기화하는 시간은 실험 기간과 클러스터에 기록된 데이터 속도에 따라 달라집니다.

이 작업은 실험 리전에 리전 멤버가 있는 MemoryDB 다중 리전 클러스터를 대상으로 합니다. 기본적으로 값이 인 태그가 있는 다중 리전 클러스터DisruptMemoryDB를 대상으로 합니다Allowed. 이 태그를 다중 리전 클러스터에 추가하거나 실험 템플릿에서 기본 태그를 자체 태그로 바꿀 수 있습니다. 기본적으로 유효한 클러스터를 찾을 수 없는 경우 이 작업은 건너뛰게 됩니다.

제한 사항

- 이 시나리오에는 중지 조건이 포함되어 있지 않습니다. 애플리케이션에 맞는 올바른 중지 조건을 실험 템플릿에 추가해야 합니다.

요구 사항

- AWS FIS 실험 역할에 필요한 권한을 추가합니다.
- 리소스 태그는 실험의 대상이 되는 리소스에 적용해야 합니다. 자체 태그 지정 규칙 또는 시나리오에 정의된 기본 태그를 사용할 수 있습니다.

권한

다음 정책은 Cross-Region: Connectivity 시나리오로 실험을 실행하는 데 필요한 권한을 AWS FIS에 부여합니다. 이 정책은 실험 역할에 연결되어야 합니다.

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

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

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

```
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity9",
            "Effect": "Allow",
            "Action": "ec2:DeleteNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity10",
            "Effect": "Allow",
            "Action": "ec2:CreateManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity11",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteManagedPrefixList",
                "ec2:ModifyManagedPrefixList"
            ],
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "EC2DescribeResources",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
```

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

```
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity18",
            "Effect": "Allow",
            "Action": "ec2:DisassociateRouteTable",
            "Resource": "arn:aws:ec2:*:*:subnet/*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity19",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity1",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateTransitGatewayRouteTable",
                "ec2:AssociateTransitGatewayRouteTable"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:transit-gateway-route-table/*",
                "arn:aws:ec2:*:*:transit-gateway-attachment/*"
            ]
        },
        {
            "Sid": "S3CrossRegion1",
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion3",
            "Effect": "Allow",
            "Action": "s3:PauseReplication",
            "Resource": "arn:aws:s3:::*",
            "Condition": {
```

```
                "StringLike": {
                    "s3:DestinationRegion": "*"
                }
            }
        },
        {
            "Sid": "S3CrossRegion4",
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:PutReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "BoolIfExists": {
                    "s3:isReplicationPauseRequest": "true"
                }
            }
        },
        {
            "Sid": "DynamoDbPauseReplication",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:PutResourcePolicy",
                "dynamodb:GetResourcePolicy",
                "dynamodb:DeleteResourcePolicy"
            ],
            "Resource": [
                "arn:aws:dynamodb:*:*:table/*"
            ]
        },
        {
            "Sid": "DynamoDbMrscPauseReplication",
            "Effect": "Allow",
            "Action": [
                "dynamodb:InjectError"
            ],
            "Resource": ["*"]
        },
        {
            "Sid": "ResolveResourcesViaTags",
            "Effect": "Allow",
            "Action": "tag:GetResources",
```

```
            "Resource": "*"
        },
        {
            "Sid": "MemDbCrossRegion",
            "Effect": "Allow",
            "Action": [
                "memorydb:DescribeMultiRegionClusters",
                "memorydb:PauseMultiRegionClusterReplication"
            ],
            "Resource": [
                "arn:aws:memorydb::*:multiregioncluster/*"
            ]
        },
        {
            "Sid": "DisruptVPCE1",
            "Effect": "Allow",
            "Action": "ec2:CreateSecurityGroup",
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Sid": "DisruptVPCE2",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateSecurityGroup",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "DisruptVPCE3",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteSecurityGroup",
                "ec2:RevokeSecurityGroupEgress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "DisruptVPCE4",
            "Effect": "Allow",
            "Action": "vpce:AllowMultiRegion",
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*"
        },
        {
            "Sid": "ModifyVPCE",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        }
    ]
}
```

시나리오 콘텐츠

다음 콘텐츠는 시나리오를 정의합니다. 이 JSON을 저장하여 AWS Command Line Interface(AWS CLI)에서 create-experiment-template 명령을 사용하여 실험 템플릿을 만드는 데 사용할 수 있습니다. 최신 버전의 시나리오를 보려면 FIS 콘솔의 시나리오 라이브러리를 방문하세요.

```
{
        "targets": {
                "Transit-Gateway": {
                        "resourceType": "aws:ec2:transit-gateway",
                        "resourceTags": {
                                "TgwTag": "TgwValue"
                        },
                        "selectionMode": "ALL"
                },
                "Subnet": {
                        "resourceType": "aws:ec2:subnet",
                        "resourceTags": {
                                "SubnetKey": "SubnetValue"
                        },
                        "selectionMode": "ALL",
```

```
                        "parameters": {}
                },
                "VPC-Endpoint": {
                    "resourceType": "aws:ec2:vpc-endpoint",
                    "resourceTags": {
                        "DisruptPrivateLink": "Allowed"
                    },
                    "selectionMode": "ALL"
                },
                "S3-Bucket": {
                        "resourceType": "aws:s3:bucket",
                        "resourceTags": {
                                "S3Impact": "Allowed"
                        },
                        "selectionMode": "ALL"
                },
                "DynamoDB-Global-Table": {
                        "resourceType": "aws:dynamodb:global-table",
                        "resourceTags": {
                                "DisruptDynamoDb": "Allowed"
                        },
                        "selectionMode": "ALL"
                },
                "MemoryDB-Multi-Region-Cluster": {
                    "resourceType": "aws:memorydb:multi-region-cluster",
                    "resourceTags": {
                        "DisruptMemoryDb": "Allowed"
                    },
                    "selectionMode": "ALL"
                }
        },
        "actions": {
                "Disrupt-Transit-Gateway-Connectivity": {
                        "actionId": "aws:network:transit-gateway-disrupt-cross-region-
connectivity",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "TransitGateways": "Transit-Gateway"
                        }
                },
                "Disrupt-Subnet-Connectivity": {
```

```
                        "actionId": "aws:network:route-table-disrupt-cross-region-
connectivity",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "Subnets": "Subnet"
                        }
                },
                "Disrupt-Vpc-Endpoint": {
                        "actionId": "aws:network:disrupt-vpc-endpoint",
                        "parameters": {
                                "duration": "PT3H"
                        },
                        "targets": {
                                "VPCEndpoints": "VPC-Endpoint"
                        }
                },
                "Pause-S3-Replication": {
                        "actionId": "aws:s3:bucket-pause-replication",
                        "parameters": {
                                "duration": "PT3H",
                                "region": "eu-west-1"
                        },
                        "targets": {
                                "Buckets": "S3-Bucket"
                        }
                },
                "Pause-DynamoDB-Replication": {
                        "actionId": "aws:dynamodb:global-table-pause-replication",
                        "parameters": {
                                "duration": "PT3H"
                        },
                        "targets": {
                                "Tables": "DynamoDB-Global-Table"
                        }
                },
                "Pause-MemoryDB-Multi-Region-Cluster-Replication": {
                    "actionId": "aws:memorydb:multi-region-cluster-pause-replication",
                    "parameters": {
                        "duration": "PT3H",
                        "region": "eu-west-1"
                    },
```

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

에 대한 다중 계정 실험 작업 AWS FIS

AWS FIS 콘솔 또는 명령줄을 사용하여 다중 계정 실험 템플릿을 생성하고 관리할 수 있습니다. 계정 대상 실험 옵션을 `"multi-account"`으로 지정하고 대상 계정 구성을 추가하여 다중 계정 실험을 만듭니다. 실험 템플릿을 만든 후 이를 사용하여 다중 계정 실험을 실행할 수 있습니다.

다중 계정 실험을 사용하면 리전 내 여러 AWS 계정에 걸쳐 있는 애플리케이션에서 실제 장애 시나리오를 설정하고 실행할 수 있습니다. 여러 대상 계정의 리소스에 영향을 미치는 오케스트레이터 계정에서 다중 계정 실험을 실행합니다.

다중 계정 실험을 실행하면 영향을 받는 리소스가 있는 대상 계정은 AWS Health Dashboard를 통해 알림을 받게 되며, 대상 계정의 사용자에게 알림을 제공합니다. 다중 계정 실험을 통해 다음을 수행할 수 있습니다:

- 에서 제공하는 중앙 제어 및 가드레일을 사용하여 여러 계정에 걸쳐 있는 애플리케이션에서 실제 장애 시나리오를 실행합니다 AWS FIS .
- 각 대상의 범위를 정의하는 세분화된 권한과 태그가 있는 IAM 역할을 사용하여 다중 계정 실험의 효과를 제어하세요.
- AWS Management Console 및 AWS FIS 로그를 통해 각 계정에서 AWS FIS 수행한 작업을 중앙에서 확인합니다.
- AWS CloudTrail을 사용하여 각 계정에서 AWS FIS API 호출을 모니터링하고 감사합니다.

이 섹션은 다중 계정 실험을 시작하는 데 도움이 됩니다.

주제

다중 계정 실험에 관한 개념

다음은 다중 계정 실험의 핵심 개념입니다.

- 오케스트레이터 계정 - 오케스트레이터 계정은 AWS FIS 콘솔에서 실험을 구성 및 관리하고 로깅을 중앙 집중화하는 중앙 계정 역할을 합니다. 오케스트레이터 계정은 AWS FIS 실험 템플릿과 실험을 소유합니다.
- 대상 계정 - 대상 계정은 AWS FIS 다중 계정 실험의 영향을 받을 수 있는 리소스가 있는 개별 AWS 계정입니다.
- 대상 계정 구성 - 실험 템플릿에 대상 계정 구성을 추가하여 실험의 일부가 되는 대상 계정을 정의합니다. 대상 계정 구성은 다중 계정 실험에 필요한 실험 템플릿의 요소입니다. 계정 ID, IAM 역할 및 선택적 설명을 설정하여 각 대상 AWS 계정에 대해 하나를 정의합니다.

다중 계정 실험에 관한 모범 사례

다중 계정 실험에 관한 모범 사례는 다음과 같습니다.

- 다중 계정 실험의 대상을 구성할 때는 모든 대상 계정에서 일관된 리소스 태그를 사용하여 타겟팅하는 것이 좋습니다. AWS FIS 실험은 각 대상 계정에서 일관된 태그가 있는 리소스를 확인합니다. `emptyTargetResolutionMode`가 `skip`로 설정된 실험을 제외하고, 작업은 모든 대상 계정에서 하나 이상의 대상 리소스를 확인해야 하며 그렇지 않으면 실패합니다. 작업 할당량은 계정별로 적용됩니다. 리소스 ARN을 기준으로 리소스를 타깃팅하려는 경우에는 작업당 동일한 단일 계정 제한이 적용됩니다.
- 파라미터나 필터를 사용하여 하나 이상의 가용성 영역에서 리소스를 타깃팅하는 경우에는 AZ 이름이 아닌 AZ ID를 지정해야 합니다. AZ ID는 계정 전체의 가용 영역에 대한 고유하고 일관된 식별자입니다. 계정의 가용 영역에 대한 AZ ID를 찾는 방법을 알아보려면 AWS 리소스에 대한 가용 영역 ID를 참조하세요.

다중 계정 실험을 위한 사전 조건

다중 계정 실험에 중지 조건을 사용하려면 먼저 교차 계정 경보를 구성해야 합니다. IAM 역할은 다중 계정 실험 템플릿을 만들 때 정의됩니다. 템플릿을 생성하기 전에 필요한 IAM 역할을 생성할 수 있습니다.

내용

- 다중 계정 실험에 대한 권한
- 다중 계정 실험의 중지 조건(선택 사항)
- 다중 계정 실험에 대한 안전 레버(선택 사항)

다중 계정 실험에 대한 권한

멀티-계정 실험은 IAM 역할 체인을 사용하여 대상 계정의 리소스에 대한 작업을 수행할 수 있는 권한을 AWS FIS 에 부여합니다. 다중 계정 실험의 경우, 각 대상 계정과 오케스트레이터 계정에서 IAM 역할을 설정합니다. 이러한 IAM 역할을 사용하려면 대상 계정과 오케스트레이터 계정 및 오케스트레이터 계정과 AWS FIS간의 신뢰 관계가 필요합니다.

대상 계정의 IAM 역할에는 리소스에 대한 작업을 수행하는 데 필요한 권한이 포함되어 있으며, 대상 계정 구성을 추가하여 실험 템플릿에 대해 생성됩니다. 대상 계정의 역할을 맡을 수 있는 권한이 있는 오케스트레이터 계정에 대한 IAM 역할을 생성하고 AWS FIS와 신뢰 관계를 설정합니다. 이 IAM 역할은 실험 템플릿의 `roleArn`으로 사용됩니다.

역할 체인에 대해 자세히 알아보려면 IAM 사용 설명서에서 역할 용어 및 개념을 참조하세요.

다음 예에서는 오케스트레이터 계정 A가 대상 계정 B에서 `aws:ebs:pause-volume-io`로 실험을 실행할 수 있도록 권한을 설정합니다.

1. 계정 B에서 작업을 실행하는 데 필요한 권한이 있는 IAM 역할을 만듭니다. 각 작업에 필요한 권한은 작업 참조를 참조하세요. 다음 예는 EBS 볼륨 IO 일시 중지 작업 the section called “aws:ebs:pause-volume-io”를 실행하기 위해 대상 계정에서 부여하는 권한을 보여줍니다.

   JSON

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

```
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
}
```

2. 다음으로 계정 A와 신뢰 관계를 만드는 신뢰 정책을 계정 B에 추가합니다. 3단계에서 만들 계정 A에 대한 IAM 역할의 이름을 선택합니다.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "AccountIdA"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike": {
                    "sts:ExternalId":
 "arn:aws:fis:region:accountIdA:experiment/*"
                },
                "ArnEquals": {
                    "aws:PrincipalArn":
 "arn:aws:iam::111122223333:role/role_name"
                }
            }
        }
    ]
}
```

3. 계정 A에서 IAM 역할을 생성합니다. 이 역할 이름은 2단계의 신뢰 정책에서 지정한 역할과 일치해야 합니다. 여러 계정을 타겟팅하려면 오케스트레이터에게 각 역할을 맡을 수 있는 권한을 부여합니다. 다음 예는 계정 A가 계정 B를 가정할 수 있는 권한을 보여 줍니다. 대상 계정이 더 있는 경우 이 정책에 역할 ARN을 추가합니다. 대상 계정당 하나의 역할 ARN만 가질 수 있습니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/role_name"
            ]
        }
    ]
}
```

4. 계정 A에 대한 이 IAM 역할은 실험 템플릿의 `roleArn`으로 사용됩니다. 다음 예제에서는 오케스트레이터 계정인 계정 A를 수임할 수 있는 AWS FIS 권한을 부여하는 IAM 역할에 필요한 신뢰 정책을 보여줍니다.

   JSON

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

Stacksets를 사용하여 한 번에 여러 IAM 역할을 프로비저닝할 수도 있습니다. CloudFormation StackSets를 사용하려면 AWS 계정에서 필요한 StackSet 권한을 설정해야 합니다. 자세한 내용은 AWS CloudFormation StackSets 사용을 참조하세요.

다중 계정 실험의 중지 조건(선택 사항)

중지 조건은 경보로 정의한 임계값에 도달할 경우 실험을 중지하는 메커니즘입니다. 다중 계정 실험에 대한 중지 조건을 설정하려면 교차 계정 경보를 사용할 수 있습니다. 오케스트레이터 계정에서 읽기 전용 권한을 사용하여 경보를 사용할 수 있도록 하려면 각 대상 계정에서 공유를 활성화해야 합니다. 공유가 완료되면 지표 수식을 사용하여 서로 다른 대상 계정의 지표를 결합할 수 있습니다. 그런 다음 이 경보를 실험의 중지 조건으로 추가할 수 있습니다.

교차 계정 대시보드에 대해 자세히 알아보려면 CloudWatch에서 교차 계정 기능 활성화를 참조하세요.

다중 계정 실험에 대한 안전 레버(선택 사항)

안전 레버는 실행 중인 모든 실험을 중지하고 새 실험이 시작되지 않도록 방지하는 데 사용됩니다. 안전 레버를 사용하여 특정 기간 동안이나 애플리케이션 상태 경보에 대응하여 FIS 실험을 방지할 수 있습니다. 모든 AWS 계정에는 당 안전 레버가 있습니다 AWS 리전. 안전 레버가 작동하면 안전 레버와 동일한 계정 및 리전에서 실행되는 모든 실험에 영향을 미칩니다. 다중 계정 실험을 중지 및 방지하려면 실험이 실행 중인 계정과 리전에서 안전 레버를 작동해야 합니다.

다중 계정 실험 템플릿 생성

를 통해 실험 템플릿을 생성하는 방법을 알아보려면 AWS Management Console

실험 템플릿 만들기을(를) 참조하세요.

CLI를 사용하여 실험 템플릿을 만들려면

1. 를 엽니다. AWS Command Line Interface
2. 계정 대상 실험 옵션이 "multi-account"로 설정된 저장된 JSON 파일(예: my-template.json)에서 실험을 만들려면 ####로 된 자리 표시자 값을 사용자 고유의 값으로 바꾼 다음 다음 create-experiment-template 명령을 실행합니다.

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

   그러면 응답에 실험 템플릿이 반환됩니다. 응답에서 실험 템플릿의 ID인 id를 복사합니다.
3. create-target-account-configuration 명령을 실행하여 실험 템플릿에 대상 계정 구성을 추가합니다. ####로 표시된 자리 표시자 값을 사용자 고유의 값으로 바꾸고, 2단계의 id를--experiment-template-id 파라미터의 값으로 사용한 후 다음을 실행합니다. --description 파라미터는 선택 항목입니다. 각 대상 계정에 대해 이 단계를 반복합니다.

```
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. 특정 대상 계정 구성에 대한 세부 정보를 검색하려면 get-target-account-configuration 명령을 실행합니다.

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

5. 모든 대상 계정 구성을 추가한 후에는 list-target-account-configurations 명령을 실행하여 대상 계정 구성이 생성되었는지 확인할 수 있습니다.

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

또한 get-experiment-template 명령을 실행하여 대상 계정 구성을 추가했는지 확인할 수도 있습니다. 이 템플릿은 실험 템플릿에 있는 모든 대상 계정 구성의 개수를 나타내는 읽기 전용 필드 `targetAccountConfigurationsCount`를 반환합니다.

6. 준비가 되면 start-experiment 명령을 사용하여 실험 템플릿을 실행할 수 있습니다.

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

대상 계정 구성 업데이트

계정에 대한 역할 ARN 또는 설명을 변경하려는 경우 기존 대상 계정 구성을 업데이트할 수 있습니다. 대상 계정 구성을 업데이트하면 템플릿을 사용하는 실행 중인 실험에는 변경 사항이 영향을 미치지 않습니다.

를 사용하여 대상 계정 구성을 업데이트하려면 AWS Management Console

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 업데이트를 선택합니다.
4. 사이드 패널에서 3단계, 서비스 액세스 구성을 선택합니다.
5. 대상 계정 구성을 수정하고 실험 템플릿 업데이트를 선택합니다.
6. 5단계, 검토 및 생성을 선택합니다.

CLI를 사용하여 대상 계정 구성을 업데이트하려면 다음을 수행합니다.

update-target-account-configuration 명령을 실행하여 ####로 표시된 자리 표시자 값을 사용자 지정 값으로 바꿉니다. --role-arn 및 --description 파라미터는 선택 사항이며, 포함하지 않으면 업데이트되지 않습니다.

```
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"
```

대상 계정 구성 삭제

대상 계정 구성이 더 이상 필요하지 않은 경우 삭제할 수 있습니다. 대상 계정 구성을 삭제해도 템플릿을 사용하여 실행 중인 모든 실험은 영향을 받지 않습니다. 실험은 완료되거나 중지될 때까지 계속 실행됩니다.

를 사용하여 대상 계정 구성을 삭제하려면 AWS Management Console

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 업데이트를 선택합니다.
4. 사이드 패널에서 3단계, 서비스 액세스 구성을 선택합니다.
5. 대상 계정 구성에서 삭제하려는 대상 계정 역할 ARN에 대해 제거를 선택합니다.
6. 5단계, 검토 및 생성을 선택합니다.
7. 템플릿을 검토하고 실험 템플릿 업데이트를 선택합니다. 확인 메시지가 표시되면를 update 입력하고 실험 템플릿 업데이트를 선택합니다.

CLI를 사용하여 대상 계정 구성을 삭제하려면 다음을 수행합니다.

delete-target-account-configuration 명령을 실행하여 ####로 표시된 자리 표시자 값을 사용자 지정 값으로 바꿉니다.

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

실험 일정 예약

AWS Fault Injection Service(FIS)를 통해 AWS 워크로드에 대한 오류 주입 실험을 수행할 수 있습니다. 이러한 실험은 지정된 대상에서 실행할 하나 이상의 작업이 포함된 템플릿에서 실행됩니다. 이제 FIS 콘솔에서 기본적으로 실험을 일회성 작업 또는 반복 작업으로 예약할 수 있습니다. 이제 FIS는 예약된 규칙 외에도 새로운 일정 예약 기능을 제공합니다. FIS는 이제 EventBridge 스케줄러와 통합되어 자동으로 규칙을 생성합니다. EventBridge Scheduler는 하나의 중앙 관리형 서비스에서 작업을 생성, 실행 및 관리할 수 있는 서버리스 스케줄러입니다.

> **Important**
>
> 를 사용하는 실험 스케줄러 AWS Fault Injection Service 는 AWS GovCloud(미국 동부) 및 AWS GovCloud(미국 서부)에서 사용할 수 없습니다.

주제

스케줄러 역할 생성

실행 역할은 EventBridge 스케줄러와 상호 작용하고 Event Bridge 스케줄러가 FIS 실험을 시작하기 위해가 AWS FIS 수임하는 IAM 역할입니다. 이 역할에 권한 정책을 추가하여 EventBridge 스케줄러에 FIS 실험을 간접 호출할 수 있는 액세스 권한을 부여합니다. 다음 단계에서는 EventBridge가 실험을 시작하도록 허용하는 새 실행 역할과 정책을 생성하는 방법을 설명합니다.

AWS CLI를 사용하여 스케줄러 역할 생성

이는 Event Bridge가 고객을 대신하여 실험 일정을 잡을 수 있도록 하는 데 필요한 IAM 역할입니다.

1. 다음 역할 수임 JSON 정책을 복사하고 로컬에 `fis-execution-role.json`로 저장하세요. 이 신뢰 정책은 EventBridge 스케줄러가 사용자를 대신하여 역할을 맡을 수 있도록 합니다.

JSON

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

2. AWS Command Line Interface(AWS CLI)를 사용하여 다음 명령을 입력하여 새 역할을 생성합니다. `FisSchedulerExecutionRole`를 이 역할에 부여하려는 이름으로 바꾸세요.

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

성공하면 다음과 같은 결과가 출력됩니다.

```
{
    "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. EventBridge 스케줄러가 실험을 간접 호출할 수 있도록 허용하는 새 정책을 생성하려면 다음 JSON을 복사하고 로컬에 `fis-start-experiment-permissions.json`로 저장합니다. 다음 정책은 EventBridge 스케줄러가 사용자 계정의 모든 실험 템플릿에서 `fis:StartExperiment` 작업을 직접 호출하도록 허용합니다. 역할을 단일 실험 템플릿으로 제한하려면 `"arn:aws:fis:*:*:experiment-template/*"` 끝에 있는 `*`를 실험 템플릿의 ID로 바꾸세요.

   JSON

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

4. 다음 명령을 실행하여 새 권한 정책을 생성합니다. `FisSchedulerPolicy`를 이 정책에 부여하려는 이름으로 바꾸세요.

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

   성공하면 다음과 같은 결과가 출력됩니다. 정책 ARN을 기록합니다. 다음 단계에서 이 ARN을 사용하여 정책을 실행 역할에 연결합니다.

```
{
    "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. 실행 역할에 정책을 연결하려면 다음 명령을 실행하세요. `your-policy-arn`을 이전 단계에서 생성한 정책의 ARN으로 변경합니다. `FisSchedulerExecutionRole`을 실행 역할의 이름으로 바꿉니다.

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

   이 `attach-role-policy` 작업은 명령줄에서 응답을 반환하지 않습니다.

6. 특정 태그 값이 있는 AWS FIS 실험 템플릿만 실행하도록 스케줄러를 제한할 수 있습니다. 예를 들어 다음 정책은 모든 AWS FIS 실험에 대한 `fis:StartExperiment` 권한을 부여하지만 스케줄러가 태그가 지정된 실험 템플릿만 실행하도록 제한합니다 `Purpose=Schedule`.

   JSON

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

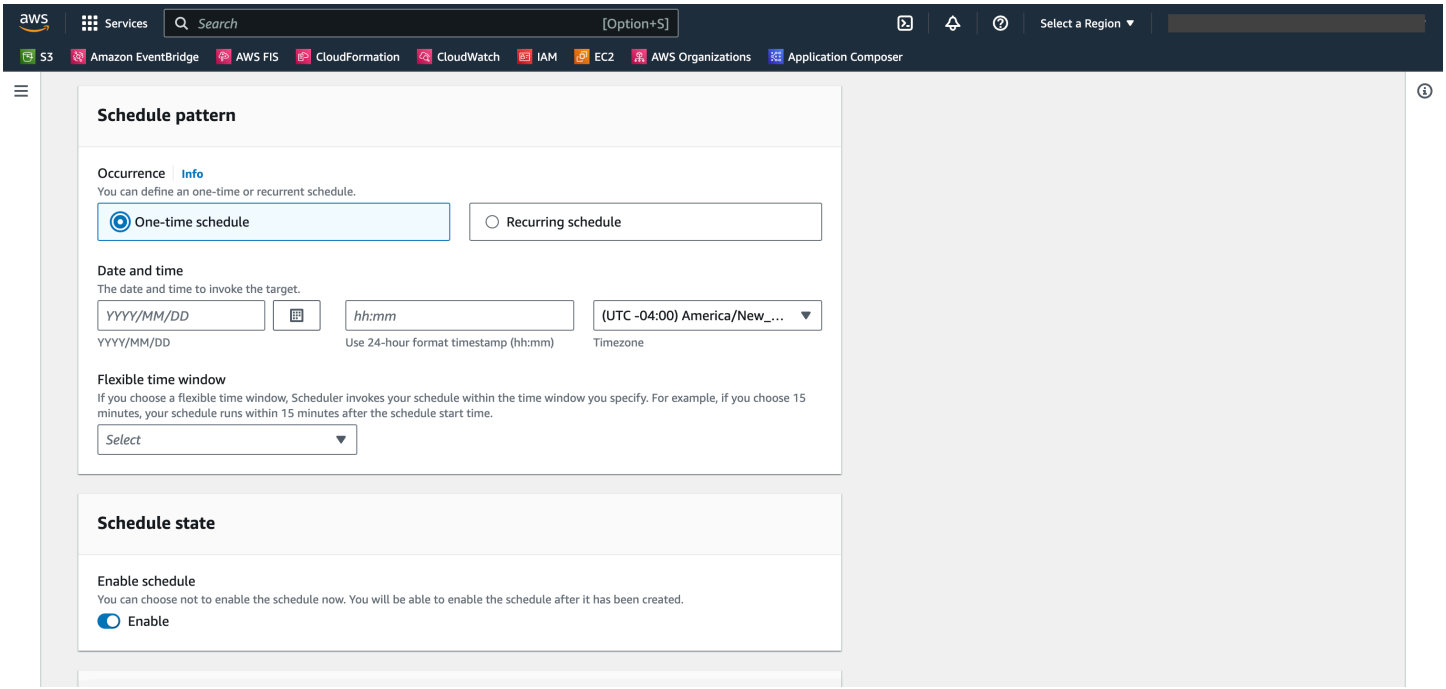
```
            }
        }
    ]
}
```

실험 일정 생성

실험 일정을 잡기 전에 일정이 간접적으로 호출할 하나 이상의 실험 템플릿 구성 요소이 필요합니다. 기존의 AWS 리소스를 사용하거나 새로 생성할 수 있습니다.

실험 템플릿이 생성되면 작업을 클릭하고 실험 예약을 선택합니다. 실험 예약 페이지로 리디렉션됩니다. 일정의 이름이 자동으로 채워집니다.

일정 패턴 섹션에 따라 일회성 일정 또는 반복 일정을 선택하세요. 필수 입력 필드를 채우고 권한으로 이동하세요.



일정 상태는 기본적으로 활성화되어 있습니다. 참고: 일정 상태를 비활성화하면 일정을 생성하더라도 실험이 예약되지 않습니다.

AWS FIS 실험 스케줄러는 EventBridge 스케줄러를 기반으로 구축됩니다. 지원되는 다양한 일정 유형에 대한 설명서를 참조할 수 있습니다.

콘솔을 사용하여 일정을 업데이트하려면

1. AWS FIS 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 실험 템플릿을 선택합니다.
3. 일정을 생성하려는 실험 템플릿을 선택합니다.
4. 작업을 클릭하고 드롭다운에서 실험 예약을 선택합니다.
   a. 일정 이름 아래에 이름이 자동으로 채워집니다.
   b. 일정 패턴에서 반복 일정을 선택합니다.
   c. 일정 유형에서 요금 기반 일정을 선택할 수 있습니다. 일정 유형을 참조하세요.
   d. rate 표현식에서 실험 실행 시간보다 느린 속도(예: 5분)를 선택합니다.
   e. 기간에서 시간대를 선택합니다.
   f. 시작 날짜 및 시간에서 시작 날짜 및 시간을 지정합니다.
   g. 종료 날짜 및 시간에서 종료 날짜 및 시간을 지정합니다.
   h. 일정 상태에서 일정 활성화 옵션을 토글합니다.
   i. 권한 아래에서 기존 역할 사용을 선택한 다음 `FisSchedulerExecutionRole`을 검색합니다.
   j. 다음을 선택합니다.
5. 일정 검토 및 생성을 선택하고 스케줄러 세부 정보를 검토한 다음 일정 생성을 선택합니다.

실험 일정 업데이트

실험 일정이 적합한 특정 날짜와 시간에 발생하도록 업데이트할 수 있습니다.

콘솔을 사용하여 실험 실행을 업데이트하려면

1. Amazon FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 일정이 이미 생성된 리소스 유형: 실험 템플릿을 선택합니다.
4. 템플릿의 실험 ID를 클릭합니다. 그런 다음 일정 탭으로 이동합니다.
5. 실험과 관련된 기존 일정이 있는지 확인하세요. 관련 일정을 선택하고 일정 업데이트 버튼을 클릭합니다.

실험 일정 비활성화 또는 삭제

실험이 일정에 따라 실행되거나 실행되지 않도록 하려면 규칙을 삭제하거나 비활성화할 수 있습니다. 다음 단계에서는 AWS 콘솔을 사용하여 실험 실행을 삭제하거나 비활성화하는 방법을 안내합니다.

규칙을 삭제하거나 비활성화하려면

1. Amazon FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 일정이 이미 생성된 리소스 유형: 실험 템플릿을 선택합니다.
4. 템플릿의 실험 ID를 클릭합니다. 그런 다음 일정 탭으로 이동합니다.
5. 실험과 관련된 기존 일정이 있는지 확인하세요. 관련 일정을 선택하고 일정 업데이트 버튼을 클릭합니다.
6. 다음 중 하나를 수행하세요.
   a. 일정을 삭제하려면 일정 삭제 규칙 옆의 버튼을 선택합니다. `delete`을 입력하고 일정 삭제 버튼을 클릭합니다.
   b. 일정을 비활성화하려면 일정 비활성화 규칙 옆의 버튼을 선택합니다. `disable`을 입력하고 일정 비활성화 버튼을 클릭합니다.

에 대한 안전 레버 AWS FIS

안전 레버는 실행 중인 모든 실험을 중지하고 새 실험이 시작되지 않도록 방지하는 데 사용됩니다. 안전 레버를 사용하여 특정 기간 동안이나 애플리케이션 상태 경보에 대응하여 FIS 실험을 방지할 수 있습니다. 모든 AWS 계정에는 당 안전 레버가 있습니다 AWS 리전.

진행 중인 실험이 안전 레버에 의해 중지된 경우, 실험이 중지되기 전에 실행된 작업 기간에 대해서만 비용을 지불하면 됩니다. 시작되지 않도록 방지된 실험에는 비용이 발생하지 않습니다. 다음 섹션에서는 안전 레버의 사용을 시작하는 방법을 알려줍니다.

주제

안전 레버의 개념

안전 레버는 작동하거나 해제할 수 있습니다.

- 안전 레버가 해제되면 FIS 실험이 허용됩니다. 기본적으로 안전 레버는 해제됩니다.
- 안전 레버가 작동하면 진행 중인 실험이 중지되고 새 실험을 시작할 수 없습니다.

안전 레버의 영향을 받는 실험은 다음 상태 중 하나로 종료됩니다.

- 중지됨: 안전 레버가 작동했을 때 실험이 실행 중이었던 경우
- 취소됨: 안전 레버가 이미 작동했을 때 실험이 시작된 경우

중지되거나 취소된 실험은 재개하거나 다시 실행할 수 없습니다. 하지만 안전 레버가 해제되면 동일한 실험 템플릿을 사용하여 새 실험을 시작할 수 있습니다.

안전 레버 리소스

안전 레버는 Amazon 리소스 이름(ARN)으로 정의되는 리소스입니다. 안전 레버에는 다음 파라미터가 포함됩니다.

- 상태: 작동 또는 작동 해제입니다.
- 사유: 안전 레버 상태가 변경된 이유를 로깅하기 위해 사용자가 입력하는 문자열입니다.

안전 레버 작업

이 섹션에서는 AWS FIS 콘솔 또는 명령줄을 사용하여 안전 레버를 보고, 연결하고, 해제하는 방법을 자세히 설명합니다.

안전 레버 보기

아래 단계에 따라 계정 및 리전의 안전 레버 상태를 볼 수 있습니다.

콘솔을 사용하여 안전 레버를 보려면

1. AWS FIS 콘솔 열기
2. 탐색 창에서 실험을 선택합니다.
3. 안전 레버가 작동하는 경우 페이지 상단에 알림 배너가 표시됩니다. 알림 배너가 없으면 안전 레버가 해제된 것입니다.

CLI를 사용하여 안전 레버를 보려면

- 다음 명령을 사용합니다.

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

안전 레버는 다음 상태 중 하나일 수 있습니다.

- 작동 해제 - 안전 레버가 실험에 영향을 주지 않습니다. 실험을 자유롭게 실행할 수 있습니다. 기본적으로 안전 레버는 해제됩니다.
- 작동 준비 중 - 안전 레버가 작동 해제에서 작동 상태로 바뀌는 중입니다. 아직 중지되지 않은 실험이 있을 수 있습니다. 이 상태에서는 안전 레버를 변경할 수 없습니다.
- 작동 - 안전 레버가 활성 상태이고 실행 중인 실험이 없습니다. 안전 레버가 작동하는 동안 시작하려고 시도하는 모든 새 실험은 취소됩니다.

안전 레버 작동

콘솔을 사용하여 안전 레버를 작동하려면

1. AWS FIS 콘솔 열기

2. 탐색 창에서 실험을 선택합니다.
3. 모든 실험 중지 버튼을 선택합니다.
4. 안전 레버를 작동하는 이유를 입력합니다.
5. 확인을 선택합니다.

CLI를 사용하여 안전 레버를 작동하려면

- 다음 명령을 사용합니다. 사유 필드에 해당 응답을 입력합니다.

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

안전 레버 작동 해제

콘솔을 사용하여 안전 레버를 해제하려면

1. AWS FIS 콘솔 열기
2. 탐색 창에서 실험을 선택합니다.
3. 안전 레버 분리 버튼을 선택합니다.
4. 안전 레버를 해제하는 이유를 입력합니다.
5. 확인을 선택합니다.

CLI를 사용하여 안전 레버를 해제하려면

- 다음 명령을 사용합니다.

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

AWS FIS 실험 모니터링

다음 도구를 사용하여 AWS Fault Injection Service(AWS FIS) 실험의 진행 상황과 영향을 모니터링할 수 있습니다.

AWS FIS 콘솔 및 AWS CLI

AWS FIS 콘솔 또는 AWS CLI 를 사용하여 실행 중인 실험의 진행 상황을 모니터링합니다. 실험의 각 작업 상태와 각 작업의 결과를 볼 수 있습니다. 자세한 내용은 the section called “실험 보기” 단원을 참조하십시오.

CloudWatch 사용량 지표 및 경보

CloudWatch 사용량 지표를 사용하여 계정의 리소스 사용량에 대한 가시성을 제공합니다. AWS FIS 사용량 지표는 AWS 서비스 할당량에 해당합니다. 사용량이 서비스 할당량에 가까워지면 경고하는 경보를 구성할 수 있습니다. 자세한 내용은 CloudWatch를 사용한 모니터링 단원을 참조하십시오.

실험이 범위를 벗어나는 시점을 정의하는 CloudWatch 경보를 생성하여 AWS FIS 실험에 대한 중지 조건을 생성할 수도 있습니다. 경보가 트리거되면 실험이 중지됩니다. 자세한 내용은 중지 조건 단원을 참조하십시오. CloudWatch 경보 생성에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 정적 임계값을 기반으로 CloudWatch 경보 생성 및 이상 탐지를 기반으로 CloudWatch 경보 생성을 참조하세요.

AWS FIS 실험 로깅

실험 로깅을 활성화하여 실행 중인 실험에 대한 세부 정보를 캡처할 수 있습니다. 자세한 정보는 실험 로깅 섹션을 참조하세요.

실험 상태 변경 이벤트

Amazon EventBridge를 사용하면 시스템 이벤트 또는 리소스 변경에 자동으로 응답할 수 있습니다. AWS FIS는 실험 상태가 변경될 때 알림을 보냅니다. 이벤트가 규칙과 일치할 때 수행할 자동화된 작업을 지정하는 규칙을 관심 있는 이벤트에 대해 생성할 수 있습니다. Amazon SNS 주제에 알림을 보내거나 Lambda 함수를 호출하는 경우를 예로 들 수 있습니다. 자세한 내용은 EventBridge를 사용하여 모니터링 단원을 참조하십시오.

CloudTrail 로그

AWS CloudTrail 를 사용하여 AWS FIS API 호출에 대한 자세한 정보를 캡처하고 Amazon S3에 로그 파일로 저장합니다. 또한 CloudTrail은 실험을 실행 중인 리소스의 서비스 API에 대한 호출을 기

록합니다. 이러한 CloudTrail 로그를 사용하여 어떤 요청이 이루어졌는지, 어떤 소스 IP 주소에서 요청을 했는지, 누가 언제 요청했는지 등을 확인할 수 있습니다.

AWS 상태 대시보드 알림

AWS Health는 리소스 성능과 AWS 서비스 및 계정의 가용성에 대한 지속적인 가시성을 제공합니다. 실험을 시작하면 AWS FIS가 AWS 상태 대시보드에 알림을 보냅니다. 이 알림은 다중 계정 실험을 포함하여 실험 대상 리소스가 포함된 각 계정에서 실험이 진행되는 동안 표시됩니다. `aws:ssm:start-automation-execution` 및 `aws:fis:wait`와 같이 대상이 포함되지 않은 작업만 있는 다중 계정 실험은 알림을 보내지 않습니다. 실험을 허용하는 데 사용된 역할에 대한 정보는 영향을 받는 리소스 아래에 나열됩니다. AWS Health Dashboard에 대해 자세히 알아보려면 AWS Health 사용 설명서에서 AWS Health Dashboard를 참조하세요.

> **Note**
>
> AWS Health는 최선을 다해 이벤트를 제공합니다.

Amazon CloudWatch를 사용하여 AWS FIS 사용량 지표 모니터링

Amazon CloudWatch를 사용하여 AWS FIS 실험이 대상에 미치는 영향을 모니터링할 수 있습니다. AWS FIS 사용량을 모니터링할 수도 있습니다.

실험 상태 보기에 대한 자세한 내용은 실험 보기 섹션을 참조하세요.

AWS FIS 실험 모니터링

AWS FIS 실험을 계획할 때 실험의 대상 리소스 유형에 대한 기준 또는 "정상 상태"를 식별하는 데 사용할 수 있는 CloudWatch 지표를 식별합니다. 실험을 시작한 후 실험 템플릿을 통해 선택한 대상의 CloudWatch 지표를 모니터링할 수 있습니다.

AWS FIS에서 지원하는 대상 리소스 유형에 사용할 수 있는 CloudWatch 지표에 대한 자세한 내용은 다음을 참조하세요.

- CloudWatch를 사용하여 인스턴스 모니터링
- Amazon ECS CloudWatch 지표
- CloudWatch를 사용한 Amazon RDS 지표 모니터링
- CloudWatch를 사용하여 명령 실행 지표 모니터링

AWS FIS 사용량 지표

CloudWatch 사용량 지표를 사용하여 계정의 리소스 사용량을 확인할 수 있습니다. 이러한 지표를 사용하여 CloudWatch 그래프 및 대시보드에서 현재 서비스 사용량을 시각화합니다.

AWS FIS 사용량 지표는 AWS 서비스 할당량에 해당합니다. 사용량이 서비스 할당량에 가까워지면 경고하는 경보를 구성할 수 있습니다. CloudWatch 경보에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

AWS FIS는 AWS/Usage 네임스페이스에 다음 지표를 게시합니다.

지표	설명
ResourceCount	계정에서 실행 중인 지정된 리소스의 총 수입니다. 리소스는 지표와 연결된 차원으로 정의됩니다.

다음 차원은 AWS FIS에서 게시하는 사용량 지표를 구체화하는 데 사용됩니다.

차원	설명
Service	리소스가 포함된 AWS 서비스의 이름입니다. AWS FIS 사용량 지표의 경우이 차원의 값은 입니다FIS.
Type	보고되는 엔터티의 유형입니다. 현재 AWS FIS 사용량 지표에 유효한 값은 입니다Resource.
Resource	실행 중인 리소스의 유형입니다. 가능한 값은 실험 템플릿의 ExperimentTemplates , 진행 중인 실험의 ActiveExperiments 입니다.
Class	이 차원은 나중에 사용하기 위해 예약되어 있습니다.

Amazon EventBridge를 사용하여 AWS FIS 실험 모니터링

실험 상태가 변경되면 AWS FIS는 알림을 보냅니다. 이러한 알림은 Amazon EventBridge(이전에는 CloudWatch Events)를 통해 이벤트로 제공됩니다. AWS FIS는 이러한 이벤트를 최대한으로 내보냅니다. 이벤트는 거의 실시간으로 EventBridge로 전송됩니다.

EventBridge를 사용하면 이벤트에 대한 응답으로 프로그래밍 동작을 트리거하는 규칙을 생성할 수 있습니다. 예를 들어 SNS 주제를 호출하여 이메일 알림을 보내거나 Lambda 함수를 호출하여 조치를 취하는 규칙을 구성할 수 있습니다.

EventBridge에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 시작하기를 참조하세요.

다음은 실험 상태 변경 이벤트의 구문입니다.

```
{
    "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`

상태가 변경된 실험의 ID입니다.

`experiment-template-id`

실험에 사용된 실험 템플릿의 ID입니다.

`new_value`

실험의 새로운 상태입니다. 가능한 값은 다음과 같습니다.

- `completed`
- `failed`
- `initiating`
- `running`
- `stopped`
- `stopping`

`old_value`

실험의 이전 상태입니다. 가능한 값은 다음과 같습니다.

- `initiating`
- `pending`
- `running`
- `stopping`

AWS FIS에 대한 실험 로깅

실험 로깅을 사용하여 실행 중인 실험에 대한 세부 정보를 캡처할 수 있습니다.

각 로그 대상 유형과 관련된 비용을 기준으로 실험 로깅 요금이 부과됩니다. 자세한 내용은 Amazon CloudWatch 요금(유료 티어, 로그, 벤딩 로그 아래) 및 Amazon S3 요금을 참조하세요.

권한

구성하는 각 로그 대상으로 로그를 전송할 수 있는 AWS FIS 권한을 부여해야 합니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 다음 내용을 참조하세요.

- CloudWatch Logs로 전송된 로그
- Amazon S3로 보낸 로그

로그 스키마

다음은 실험 로깅에 사용되는 스키마입니다. 현재 스키마 버전은 2입니다. `details`의 필드는 `log_type`의 값에 따라 달라집니다. `resolved_targets`의 필드는 `target_type`의 값에 따라 달라집니다. 자세한 내용은 the section called "로그 레코드 예" 단원을 참조하십시오.

```
{
    "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

    }
}
```

릴리스 정보

- 버전 2에서는 다음 사항이 도입됩니다.
  - `target_type` 필드를 변경하고 `resolved_targets` 필드를 ARN 목록에서 객체 목록으로 변경합니다. `resolved_targets` 객체의 유효한 필드는 대상의 리소스 유형인 `target_type`의 값에 따라 달라집니다.
  - `account_id` 필드를 추가하는 `action-error` 및 `target-resolution-detail` 이벤트 유형입니다.
- 버전 1이 초기 릴리스입니다.

로그 대상

AWS FIS는 다음 대상으로 로그 전송을 지원합니다.

- Amazon S3 버킷
- Amazon CloudWatch Logs 로그 그룹

S3 로그 전달

로그는 다음 위치로 전달됩니다.

```
bucket-and-optional-prefix/AWSLogs/account-id/fis/region/experiment-
id/YYYY/MM/DD/account-id_awsfislogs_region_experiment-id_YYYYMMDDHHMMZ_hash.log
```

로그가 버킷으로 전달되려면 몇 분 정도 걸릴 수 있습니다.

CloudWatch Logs 로그 전달

로그는 /aws/fis/*experiment-id*라는 로그 스트림으로 전송됩니다.

로그는 1분 이내에 로그 그룹에 전달됩니다.

로그 레코드 예

다음은 무작위로 선택한 EC2 인스턴스에서 aws:ec2:reboot-instances 작업을 실행하는 실험의 예제 로그 기록입니다.

레코드

- experiment-start
- target-resolution-start
- target-resolution-detail
- target-resolution-end
- action-start
- action-end
- action-error
- experiment-end

experiment-start

다음은 `experiment-start` 이벤트의 예시 레코드입니다.

```
{
    "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

다음은 target-resolution-start 이벤트의 예시 레코드입니다.

```
{
    "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

다음은 target-resolution-detail 이벤트의 예시 레코드입니다. 대상 확인에 실패할 경우 기록에는 error_information 필드도 포함됩니다.

```
{
    "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

대상 확인에 실패할 경우 기록에는 error_information 필드도 포함됩니다. total_pages가 1보다 크면 확인된 대상 수가 한 레코드의 크기 제한을 초과했습니다. 나머지 확인된 대상이 포함된 추가 target-resolution-end 레코드가 있습니다.

다음은 EC2 작업에 대한 target-resolution-end 이벤트의 예시 레코드입니다.

```
{
    "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
    }
}
```

다음은 EKS 작업에 대한 `target-resolution-end` 이벤트의 예시 레코드입니다.

```
{
    "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

다음은 `action-start` 이벤트의 예시 레코드입니다. 실험 템플릿에 작업 파라미터가 지정되어 있는 경우 레코드에는 `parameters` 필드도 포함됩니다.

```
{
    "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

다음은 `action-error` 이벤트의 예시 레코드입니다. 작업이 실패할 때만 반환되는 이벤트입니다. 작업이 실패한 각 계정에 대해 반환됩니다.

```
{
    "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

다음은 action-end 이벤트의 예시 레코드입니다.

```
{
    "id": "EXPhjAXCGY78HV2a4A",
    "log_type": "action-end",
    "event_timestamp": "2023-05-31T18:50:56Z",
    "version": "2",
    "details": {
        "action_name": "Reboot",
        "action_id": "aws:ec2:reboot-instances",
        "action_end_time": "2023-05-31T18:50:56Z",
        "action_state": {
            "status": "completed",
            "reason": "Action was completed."
        }
    }
}
```

experiment-end

다음은 experiment-end 이벤트의 예시 레코드입니다.

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

실험 로깅 활성화

실험 로깅은 기본적으로 비활성화되어 있습니다. 실험에 대한 실험 로그를 받으려면 로깅이 활성화된 실험 템플릿에서 실험을 생성해야 합니다. 이전에 로깅에 사용하지 않은 대상을 사용하도록 구성된 실

험을 처음 실행하면 이 대상으로의 로그 전달을 구성하기 위해 실험이 지연되며, 약 15초가 소요됩니다.

콘솔을 이용하여 실험 로그를 활성화하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 업데이트를 선택합니다.
4. 로그의 경우 대상 옵션을 구성합니다. S3 버킷으로 로그를 보내려면 Amazon S3 버킷으로 전송을 선택하고 버킷 이름과 접두사를 입력합니다. 로그를 CloudWatch Logs로 보내려면 CloudWatch Logs로 전송을 선택하고 로그 그룹을 입력합니다.
5. 실험 템플릿 업데이트를 선택합니다.

를 사용하여 실험 로깅을 활성화하려면 AWS CLI

update-experiment-template 명령을 사용하고 로그 구성을 지정합니다.

실험 로깅 비활성화

실험에 대한 로그를 더 이상 받지 않으려면 실험 로깅을 비활성화할 수 있습니다.

콘솔을 이용하여 실험 로깅을 비활성화하려면

1. https://console.aws.amazon.com/fis/ AWS FIS 콘솔을 엽니다.
2. 탐색 창에서 실험 템플릿을 선택합니다.
3. 실험 템플릿을 선택하고 작업, 실험 템플릿 업데이트를 선택합니다.
4. 로그의 경우 Amazon S3 버킷으로 전송 및 CloudWatch Logs로 전송 선택을 취소하세요.
5. 실험 템플릿 업데이트를 선택합니다.

를 사용하여 실험 로깅을 비활성화하려면 AWS CLI

update-experiment-template 명령을 사용하고 빈 로그 구성을 지정합니다.

를 사용하여 API 호출 로깅 AWS CloudTrail

AWS Fault Injection Service(AWS FIS)는 AWS FIS에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 AWS FIS에 대

한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 AWS FIS 콘솔의 호출과 AWS FIS API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하는 경우 AWS FIS 이벤트를 포함하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 AWS FIS에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요.

CloudTrail 사용

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. AWS FIS에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 정보는 CloudTrail 이벤트 기록을 사용하여 이벤트 보기를 참조하세요.

AWS FIS에 대한 이벤트를 AWS 계정포함하여에서 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 추적을 생성하면 추적이 모든 AWS 리전에 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- AWS 계정에 대한 추적 생성
- CloudTrail 지원 서비스 및 통합
- CloudTrail에서 Amazon SNS 알림 구성
- 여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기

모든 AWS FIS 작업은 CloudTrail에서 로깅되며 AWS Fault Injection Service API 참조에 문서화됩니다. 대상 리소스에서 수행된 실험 작업은 리소스를 소유한 서비스의 API 참조 설명서를 참조하세요. 예를 들어, Amazon EC2 인스턴스에서 수행되는 작업은 Amazon EC2 API 참조를 참조하세요.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부입니다.

자세한 설명은 CloudTrail userIdentity 요소를 참조하세요.

AWS FIS 로그 파일 항목 이해

추적이란 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출의 주문 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

다음은 AWS FIS `StopExperiment` 작업에 대한 호출에 대한 CloudTrail 로그 항목의 예입니다.

```
{
  "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"

}
```

다음은 AWS FIS 작업이 포함된 실험의 일부로 `aws:ssm:send-command` AWS FIS가 호출한 API 작업에 대한 CloudTrail 로그 항목의 예입니다. `userIdentity` 요소는 역할을 맡아 얻은 임시 보안 인증 정보로 이루어진 요청을 반영합니다. 위임된 역할의 이름은 `userName`에 표시됩니다. 실험의 ID인 EXP21nT17WMzA6dnUgz는 맡은 역할의 ARN의 일부 및 `principalId`에 나타납니다.

```
{
    "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"
}
```

문제 해결 AWS FIS

오류를 해결하기 위해는 GetExperiment API 및 FIS 실험 로그에서 자세한 오류를 AWS FIS 반환합니다. 실험의 상태가 실패이면 실험 상태의 일부로 오류가 반환됩니다. 여러 작업이 실패하는 경우 첫 번째 실패한 작업이 실험 오류로 반환됩니다. FIS 실험 로그에서 다른 오류가 있는지 검토할 수 있습니다. AWS FIS 실험을 로깅하고 모니터링하는 방법을 알아보려면 섹션을 참조하세요AWS FIS 실험 모니터링.

실패 유형에 따라 아래와 같은 오류 중 하나가 발생할 수 있습니다.

- 사유: 특정 실패에 대한 자세한 설명입니다. 사유 값은 변경될 수 있으므로 자동화에 사용해서는 안 됩니다.
- 코드: 실패 유형입니다. 아래 표에 달리 명시되지 않는 한 코드 값은 변경될 수 있으므로 자동화에 사용해서는 안 됩니다.
- 위치: 작업 또는 대상과 같이 실패한 실험 템플릿 섹션에 대한 컨텍스트입니다.
- 계정 ID: 장애가 발생한 AWS 계정입니다.

오류 코드

오류 코드	코드 설명
ConfigurationFailure	작업, 대상, 실험 또는 로그가 올바르게 구성되지 않았습니다. 오류 location를 확인하고 파라미터와 구성이 올바른지 확인하세요.
DependentServiceFailure	다른 AWS 서비스에서 장애가 발생했습니다. 실험을 다시 실행해 보세요.
InternalFailure	실험을 실행할 때 내부 오류가 발생했습니다. 이 오류 코드를 기반으로 자동화할 수 있습니다.
InvalidTarget	대상을 확인하는 동안 또는 작업을 시작할 때 대상을 확인할 수 없습니다. 이유는 다음 중 하나 때문일 수 있습니다. • 대상이 존재하지 않습니다(예: 대상이 삭제되었거나 ARN이 잘못됨).

오류 코드	코드 설명
	• 대상에 대해 리소스를 확인하지 않는 태그가 있습니다. • 대상에 연결되지 않은 작업이 있습니다. 문제를 해결하려면 로그를 검토하여 확인되지 않는 대상을 식별합니다. 모든 작업이 대상에 연결되어 있고 리소스 ID 또는 태그가 존재하며 철자가 틀리지 않았는지 확인합니다.
`AuthorizationFailure`	권한 오류로 인한 실험 실패에는 두 가지 주요 원인이 있습니다. • 대상 IAM 역할에 대상을 확인하거나 리소스에 대한 조치를 취할 수 있는 적절한 권한이 없습니다. 이 오류를 해결하려면 FIS 작업 참조에서 작업에 필요한 권한을 검토하고 실험 IAM 역할에 추가하세요. • FIS에 대한 AWS 서비스 연결 역할(SLR) 생성이 조직의 서비스 제어 정책(SCP)에 의해 거부되었습니다. FIS는 SLR을 사용하여 실험에 대한 모니터링 및 리소스 선택을 관리합니다. 자세한 내용은 AWS FIS에 대한 서비스 연결 역할 권한 단원을 참조하십시오.
`QuotaExceededFailure`	해당 리소스 유형의 할당량이 초과되었습니다. 할당량을 늘릴 수 있는지 확인하려면 AWS Fault Injection Service의 할당량 및 제한 사항 섹션을 참조하세요.

AWS Fault Injection Service의 보안

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. AWS Fault Injection Service에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스 규정 준수 프로그램 .
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS FIS를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 AWS FIS를 구성하는 방법을 보여줍니다. 또한 AWS FIS 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

내용

AWS Fault Injection Service의 데이터 보호

AWS 공동 책임 모델 AWS Fault Injection Service의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 관한 자세한 내용은

데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 AWS FIS 또는 기타 AWS 서비스 에서 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함시켜서는 안 됩니다.

저장된 데이터 암호화

AWS FIS는 항상 저장 데이터를 암호화합니다. AWS FIS의 데이터는 투명한 서버 측 암호화를 사용하여 저장 시 암호화됩니다. 이를 사용하면 중요한 데이터 보호와 관련된 운영 부담 및 복잡성을 줄일 수 있습니다. 유휴 시 암호화를 사용하면 암호화 규정 준수 및 규제 요구 사항이 필요한, 보안에 민감한 애플리케이션을 구축할 수 있습니다.

전송 중 암호화

AWS FIS는 서비스와 기타 통합 AWS 서비스 간에 전송 중인 데이터를 암호화합니다. AWS FIS와 통합 서비스 간에 전달되는 모든 데이터는 전송 계층 보안(TLS)을 사용하여 암호화됩니다. 다른 통합 AWS 서비스에 대한 자세한 내용은 섹션을 참조하세요지원됨 AWS 서비스.

AWS Fault Injection Service의 ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 누가 AWS FIS 리소스를 사용할 수 있는 인증(로그인) 및 권한(권한 있음)을 받을 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

내용

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 AWS FIS에서 수행하는 작업에 따라 다릅니다.

서비스 사용자 - AWS FIS 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 AWS FIS 기능을 사용하여 작업을 수행하게 되면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다.

서비스 관리자 - 회사에서 AWS FIS 리소스를 책임지고 있는 경우 AWS FIS에 대한 전체 액세스 권한을 가지고 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 AWS FIS 기능과 리소스를 결정합니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하세요.

IAM 관리자 - IAM 관리자인 경우 AWS FIS에 대한 액세스를 관리하는 정책을 작성하는 방법에 대한 세부 정보를 알고 싶을 수 있습니다.

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증해야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 연동을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS Fault Injection Service가 IAM과 작동하는 방식

IAM을 사용하여 AWS FIS에 대한 액세스를 관리하기 전에 AWS FIS에서 사용할 수 있는 IAM 기능에 대해 알아봅니다.

AWS Fault Injection Service와 함께 사용할 수 있는 IAM 기능

IAM 특성	AWS FIS 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
엔터티 권한	예
서비스 역할	예
서비스 연결 역할	예

AWS FIS 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방식을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

AWS FIS에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

AWS FIS에 대한 자격 증명 기반 정책 예제

AWS FIS 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS Fault Injection Service 정책 예제.

AWS FIS 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM에서 교차 계정 리소스 액세스를 참조하세요.

AWS FIS에 대한 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

AWS FIS 작업 목록을 보려면 서비스 승인 참조의 AWS Fault Injection Service에서 정의한 작업을 참조하세요.

AWS FIS의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
fis
```

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

```
"Action": [
    "fis:action1",
    "fis:action2"
```

```
]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `List`라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "fis:List*"
```

AWS FIS에 대한 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

일부 AWS FIS API 작업은 여러 리소스를 지원합니다. 단일 문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

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

AWS FIS 리소스 유형 및 해당 ARNs 목록을 보려면 서비스 승인 참조의 AWS Fault Injection Service에서 정의한 리소스 유형을 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 AWS Fault Injection Service에서 정의한 작업을 참조하세요.

AWS FIS에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수

있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

AWS FIS 조건 키 목록을 보려면 서비스 승인 참조의 AWS Fault Injection Service에 사용되는 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 AWS Fault Injection Service에서 정의한 작업을 참조하세요.

AWS FIS 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요AWS Fault Injection Service 정책 예제.

AWS FISACLs

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

AWS FIS를 사용한 ABAC

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

리소스의 태그를 기반으로 리소스에 대한 액세스를 제한하는 자격 증명 기반 정책의 예제는 예: 태그를 사용하여 리소스 사용 제어에서 확인할 수 있습니다.

AWS FIS에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는 AWS 서비스 섹션을 참조하세요.

AWS FIS에 대한 교차 서비스 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

AWS FIS에 대한 서비스 역할

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

AWS FIS에 대한 서비스 연결 역할

서비스 연결 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

AWS FIS 서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 섹션을 참조하세요AWS Fault Injection Service에 서비스 연결 역할 사용.

AWS Fault Injection Service 정책 예제

기본적으로 사용자 및 역할에는 AWS FIS 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARNs 형식을 포함하여 AWS FIS에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 권한 부여 참조의 AWS Fault Injection Service에 사용되는 작업, 리소스 및 조건 키를 참조하세요.

내용

정책 모범 사례

자격 증명 기반 정책에 따라 계정에서 사용자가 AWS FIS 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서

비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.

- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

예: AWS FIS 콘솔 사용

AWS Fault Injection Service 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은에서 AWS FIS 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게는 최소 콘솔 권한을 허용할 필요가 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

다음 예제 정책은 AWS FIS 콘솔을 사용하여 모든 AWS FIS 리소스를 나열하고 볼 수 있는 권한을 부여하지만 생성, 업데이트 또는 삭제할 수는 없습니다. 또한 실험 템플릿에서 지정할 수 있는 모든 AWS FIS 작업에 사용되는 사용 가능한 리소스를 볼 수 있는 권한을 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FISReadOnlyActions",
            "Effect": "Allow",
            "Action": [
                "fis:List*",
                "fis:Get*"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "AdditionalReadOnlyActions",
            "Effect": "Allow",
            "Action": [
                "ssm:Describe*",
                "ssm:Get*",
                "ssm:List*",
                "ec2:DescribeInstances",
                "rds:DescribeDBClusters",
                "ecs:DescribeClusters",
                "ecs:ListContainerInstances",
                "eks:DescribeNodegroup",
                "cloudwatch:DescribeAlarms",
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PermissionsToCreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "fis.amazonaws.com"
                }
            }
        }
    ]
}
```

예: 사용 가능한 AWS FIS 작업 나열

다음 정책은 사용 가능한 AWS FIS 작업을 나열할 수 있는 권한을 부여합니다.

JSON

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

```
        {
            "Effect": "Allow",
            "Action": [
                "fis:ListActions"
            ],
            "Resource": "arn:aws:fis:*:*:action/*"
        }
    ]
}
```

예: 특정 작업을 위한 실험 템플릿 만들기

다음 정책은 `aws:ec2:stop-instances` 작업에 사용할 실험 템플릿을 만들 수 있는 권한을 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PolicyExample",
            "Effect": "Allow",
            "Action": [
                "fis:CreateExperimentTemplate"
            ],
            "Resource": [
                "arn:aws:fis:*:*:action/aws:ec2:stop-instances",
                "arn:aws:fis:*:*:experiment-template/*"
            ]
        },
        {
            "Sid": "PolicyPassRoleExample",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/role-name"
            ]
        }
```

```
    ]
}
```

예: 실험 시작

다음 정책은 지정된 IAM 역할 및 실험 템플릿을 사용하여 실험을 시작할 수 있는 권한을 부여합니다. 또한 AWS FIS는 사용자를 대신하여 서비스 연결 역할을 생성할 수 있습니다. 자세한 내용은 AWS Fault Injection Service에 서비스 연결 역할 사용 단원을 참조하십시오.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "fis:StartExperiment"
      ],
      "Resource": [
        "arn:aws:fis:*:*:experiment-template/experiment-template-id",
        "arn:aws:fis:*:*:experiment/*"
      ]
    },
    {
        "Sid": "PolicyExampleforServiceLinkedRole",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": "fis.amazonaws.com"
            }
        }
    }
  ]
}
```

예: 태그를 사용하여 리소스 사용 제어

다음 정책은 Purpose=Test 태그가 있는 실험 템플릿에서 실험을 실행할 권한을 부여합니다. 실험 템플릿을 생성 또는 수정하거나 지정된 태그가 없는 템플릿을 사용하여 실험을 실행할 수 있는 권한은 부여하지 않습니다.

JSON

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

예: 특정 태그가 포함된 실험 템플릿 삭제

다음 정책은 Purpose=Test 태그가 있는 실험 템플릿을 삭제할 권한을 부여합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fis:DeleteExperimentTemplate"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

예: 사용자가 자신의 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

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

예: **ec2:InjectApiError**의 조건 키 사용

다음 예제 정책은 ec2:FisTargetArns 조건 키를 사용하여 대상 리소스의 범위를 지정합니다. 이 정책은 AWS FIS 작업 aws:ec2:api-insufficient-instance-capacity-error 및를 허용합니다aws:ec2:asg-insufficient-instance-capacity-error.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:FisActionId": [
            "aws:ec2:api-insufficient-instance-capacity-error",
            "aws:ec2:asg-insufficient-instance-capacity-error"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "ForAllValues:ArnLike": {
          "ec2:FisTargetArns": [

 "arn:aws:autoscaling:*:*:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
          ]
        }
      }
    },
```

```
    {
      "Effect": "Allow",
      "Action": "autoscaling:DescribeAutoScalingGroups",
      "Resource": "*"
    }
  ]
}
```

예: **aws:s3:bucket-pause-replication**의 조건 키 사용

다음 예제 정책은 S3:IsReplicationPauseRequest 조건 키를 사용하여 AWS FIS 작업의 컨텍스트에서 AWS FIS가 사용하는 GetReplicationConfiguration 경우에만 PutReplicationConfiguration 및를 허용합니다aws:s3:bucket-pause-replication.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "S3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "StringEquals": {
                    "s3:DestinationRegion": "region"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "S3:PutReplicationConfiguration",
                "S3:GetReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "BoolIfExists": {
                    "s3:IsReplicationPauseRequest": "true"
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "S3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
    }
```

AWS Fault Injection Service에 서비스 연결 역할 사용

AWS Fault Injection Service는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 AWS FIS에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 AWS FIS에서 사전 정의하며, 서비스에서 사용자 대신 다른 AWS 서비스를 직접적으로 호출하기 위해 필요한 모든 권한을 포함합니다.

서비스 연결 역할을 사용하면 실험에 대한 모니터링 및 리소스 선택을 관리하는 데 필요한 권한을 수동으로 추가할 필요가 없으므로 AWS FIS를 더 쉽게 설정할 수 있습니다. AWS FIS는 서비스 연결 역할의 권한을 정의하며, 달리 정의되지 않은 한 AWS FIS만 해당 역할을 수임할 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

서비스 연결 역할 외에도 실험 템플릿에서 대상으로 지정한 리소스를 수정할 수 있는 권한을 부여하는 IAM 역할도 지정해야 합니다. 자세한 내용은 AWS FIS 실험을 위한 IAM 역할 단원을 참조하십시오.

먼저 관련 리소스를 삭제한 후에만 서비스 링크 역할을 삭제할 수 있습니다. 이렇게 하면 리소스에 대한 액세스 권한을 실수로 제거할 수 없기 때문에 AWS FIS 리소스가 보호됩니다.

AWS FIS에 대한 서비스 연결 역할 권한

AWS FIS는 AWSServiceRoleForFIS라는 서비스 연결 역할을 사용하여 실험에 대한 모니터링 및 리소스 선택을 관리할 수 있습니다.

AWSServiceRoleForFIS 서비스 연결 역할은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `fis.amazonaws.com`

AWSServiceRoleForFIS 서비스 연결 역할은 관리형 정책 AmazonFISServiceRolePolicy를 사용합니다. 이 정책을 통해 AWS FIS는 실험에 대한 모니터링 및 리소스 선택을 관리할 수 있습니다. 자세한 내용은 AWS 관리형 정책 참조의 AmazonFISServiceRolePolicy를 참조하세요.

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. AWSServiceRoleForFIS 서비스 연결 역할을 성공적으로 생성하려면 AWS FIS를 사용하는 IAM 자격 증명에 필요한 권한이 있어야 합니다. 필수 권한을 부여하려면 다음 정책을 IAM ID에 연결하세요.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "fis.amazonaws.com"
                }
            }
        }
    ]
}
```

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

AWS FIS에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console, AWS CLI또는 AWS API에서 AWS FIS 실험을 시작하면 AWS FIS가 서비스 연결 역할을 생성합니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. AWS FIS 실험을 시작하면 AWS FIS가 서비스 연결 역할을 다시 생성합니다.

AWS FIS에 대한 서비스 연결 역할 편집

AWS FIS에서는 AWSServiceRoleForFIS 서비스 연결 역할을 편집할 수 없습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

AWS FIS에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제하는 것이 좋습니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 엔터티가 없도록 합니다. 단, 서비스 연결 역할에 대한 리소스를 먼저 정리해야 수동으로 삭제할 수 있습니다.

> **Note**
>
> 리소스를 정리하려고 할 때 AWS FIS 서비스가 역할을 사용하는 경우 정리가 실패할 수 있습니다. 이 문제가 발생하면 몇 분 기다렸다가 작업을 다시 시도하세요.

AWSServiceRoleForFIS에서 사용하는 AWS FIS 리소스를 정리하려면

현재 실행 중인 실험이 없는지 확인합니다. 필요한 경우 실험을 중단하세요. 자세한 내용은 실험 중지 단원을 참조하십시오.

IAM을 사용하여 수동으로 서비스 연결 역할을 삭제하려면 다음을 수행하세요.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForFIS 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하십시오.

AWS FIS 서비스 연결 역할에 지원되는 리전

AWS FIS는 서비스를 사용할 수 있는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 AWS Fault Injection Service 엔드포인트 및 할당량을 참조하세요.

AWS AWS Fault Injection Service에 대한 관리형 정책

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 미칩니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AmazonFISServiceRolePolicy

이 정책은 AWS FIS가 실험에 대한 모니터링 및 리소스 선택을 관리할 수 있도록 AWSServiceRoleForFIS라는 서비스 연결 역할에 연결됩니다. 자세한 내용은 AWS Fault Injection Service에 서비스 연결 역할 사용 단원을 참조하십시오.

AWS 관리형 정책: AWSFaultInjectionSimulatorEC2Access

실험 역할에서이 정책을 사용하여 Amazon EC2에 대한 AWS FIS 작업을 사용하는 실험을 실행할 수 있는 권한을 FIS에 부여합니다. AWS Amazon EC2 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSFaultInjectionSimulatorEC2Access를 확인하세요.

AWS 관리형 정책: AWSFaultInjectionSimulatorECSAccess

실험 역할에서이 정책을 사용하여 Amazon ECS에 대한 AWS FIS 작업을 사용하는 실험을 실행할 수 있는 권한을 FIS에 부여합니다. AWS 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSFaultInjectionSimulatorECSAccess를 확인하세요.

AWS 관리형 정책: AWSFaultInjectionSimulatorEKSAccess

실험 역할에서이 정책을 사용하여 Amazon EKS에 AWS 대한 FIS 작업을 사용하는 실험을 실행할 수 있는 권한을 FIS에 부여합니다. AWS 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSFaultInjectionSimulatorEKSAccess를 확인하세요.

AWS 관리형 정책: AWSFaultInjectionSimulatorNetworkAccess

실험 역할에서이 정책을 사용하여 AWS FIS AWS 네트워킹 작업을 사용하는 실험을 실행할 수 있는 권한을 FIS에 부여합니다. 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSFaultInjectionSimulatorNetworkAccess를 확인하세요.

AWS 관리형 정책: AWSFaultInjectionSimulatorRDSAccess

실험 역할에서이 정책을 사용하여 Amazon RDS에 AWS 대한 FIS 작업을 사용하는 실험을 실행할 수 있는 권한을 FIS에 부여합니다. AWS 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSFaultInjectionSimulatorRDSAccess를 확인하세요.

AWS 관리형 정책: AWSFaultInjectionSimulatorSSMAccess

실험 역할에서이 정책을 사용하여 Systems Manager에 대한 AWS FIS 작업을 사용하는 실험을 실행할 수 있는 권한을 FIS에 부여합니다. AWS 자세한 내용은 the section called “실험 역할” 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSFaultInjectionSimulatorSSMAccess를 확인하세요.

AWS AWS 관리형 정책에 대한 FIS 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 AWS FIS의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다.

변경	설명	Date
AWSFaultInjectionSimulatorEC2Access - 기존 정책에 대한 업데이트	"AZ: Application Slowdown" 및 "Cross-AZ: Traffic Slowdown" 시나리오에 필요한 권한이 추가되었습니다. 권한은 ec2:DescribeSubnets.	2025년 11월 12일
AWSFaultInjectionSimulatorECSAccess - 기존 정책에 대한 업데이트	"AZ: Application Slowdown" 및 "Cross-AZ: Traffic Slowdown" 시나리오에 필요한 권한이 추가되었습니다. 권한은 ecs:DescribeContainerInstances, ec2:DescribeSubnets 및 ec2:DescribeInstances.	2025년 11월 12일
AWSFaultInjectionSimulatorECSAccess - 기존 정책에 대한 업데이트	AWS FIS가 ECS 대상을 확인할 수 있는 권한을 추가했습니다.	2024년 1월 25일
AWSFaultInjectionSimulatorNetworkAccess – 기존 정책 업데이트	AWS FIS가 aws:network:route-table-disrupt-cross-region-connectivity 및 aws:network:transit-gateway-disrupt-cross-region-connectivity 작업을 사용하여 실험을 실행할 수 있는 권한을 추가했습니다.	2024년 1월 25일
AWSFaultInjectionSimulatorEC2Access - 기존 정책에 대한 업데이트	AWS FIS가 EC2 인스턴스를 확인할 수 있는 권한을 추가했습니다.	2023년 11월 13일
AWSFaultInjectionSimulatorEKSAccess - 기존 정책에 대한 업데이트	AWS FIS가 EKS 대상을 확인할 수 있는 권한을 추가했습니다.	2023년 11월 13일
AWSFaultInjectionSimulatorRDSAccess - 기존 정책에 대한 업데이트	AWS FIS가 RDS 대상을 확인할 수 있는 권한을 추가했습니다.	2023년 11월 13일

변경	설명	Date
AWSFaultInjectionSimulatorEC2Access - 기존 정책에 대한 업데이트	AWS FIS가 EC2 인스턴스에서 SSM 문서를 실행하고 EC2 인스턴스를 종료할 수 있는 권한을 추가했습니다.	2023년 6월 2일
AWSFaultInjectionSimulatorSSMAccess - 기존 정책에 대한 업데이트	AWS FIS가 EC2 인스턴스에서 SSM 문서를 실행할 수 있는 권한을 추가했습니다.	2023년 6월 2일
AWSFaultInjectionSimulatorECSAccess - 기존 정책에 대한 업데이트	AWS FIS가 새 aws:ecs:task 작업을 사용하여 실험을 실행할 수 있는 권한을 추가했습니다.	2023년 6월 1일
AWSFaultInjectionSimulatorEKSAccess - 기존 정책에 대한 업데이트	AWS FIS가 새 aws:eks:pod 작업을 사용하여 실험을 실행할 수 있는 권한을 추가했습니다.	2023년 6월 1일
AWSFaultInjectionSimulatorEC2Access – 새 정책	AWS FIS가 Amazon EC2에 AWS FIS 작업을 사용하는 실험을 실행할 수 있도록 허용하는 정책이 추가되었습니다.	2022년 10월 26일
AWSFaultInjectionSimulatorECSAccess – 새 정책	AWS FIS가 Amazon ECS에 AWS FIS 작업을 사용하는 실험을 실행할 수 있도록 허용하는 정책이 추가되었습니다.	2022년 10월 26일
AWSFaultInjectionSimulatorEKSAccess - 새 정책	AWS FIS가 Amazon EKS에 AWS FIS 작업을 사용하는 실험을 실행할 수 있도록 허용하는 정책이 추가되었습니다.	2022년 10월 26일
AWSFaultInjectionSimulatorNetworkAccess – 새 정책	AWS FIS가 AWS FIS 네트워킹 작업을 사용하는 실험을 실행할 수 있도록 허용하는 정책이 추가되었습니다.	2022년 10월 26일

변경	설명	Date
AWSFaultInjectionSimulatorRDSAccess – 새 정책	AWS FIS가 Amazon RDS에 AWS FIS 작업을 사용하는 실험을 실행할 수 있도록 허용하는 정책이 추가되었습니다.	2022년 10월 26일
AWSFaultInjectionSimulatorSSMAccess – 새 정책	AWS FIS가 Systems Manager에 대해 AWS FIS 작업을 사용하는 실험을 실행할 수 있도록 허용하는 정책이 추가되었습니다.	2022년 10월 26일
AmazonFISServiceRolePolicy - 기존 정책에 대한 업데이트	AWS FIS가 서브넷을 설명할 수 있도록 권한을 추가했습니다.	2022년 10월 26일
AmazonFISServiceRolePolicy - 기존 정책에 대한 업데이트	AWS FIS가 EKS 클러스터를 설명할 수 있는 권한을 추가했습니다.	2022년 7월 7일
AmazonFISServiceRolePolicy - 기존 정책에 대한 업데이트	AWS FIS가 클러스터의 작업을 나열하고 설명할 수 있는 권한을 추가했습니다.	2022년 2월 7일
AmazonFISServiceRolePolicy - 기존 정책에 대한 업데이트	`events:DescribeRule` 작업에서 `events:ManagedBy` 조건을 제거했습니다.	2022년 1월 6일
AmazonFISServiceRolePolicy - 기존 정책에 대한 업데이트	AWS FIS가 중지 조건에 사용되는 CloudWatch 경보에 대한 기록을 검색할 수 있는 권한을 추가했습니다.	2021년 6월 30일
AWS FIS에서 변경 사항 추적 시작	AWS FIS가 AWS 관리형 정책에 대한 변경 사항 추적 시작	2021년 3월 1일

AWS Fault Injection Service의 인프라 보안

관리형 서비스인 AWS Fault Injection Service는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참

조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 AWS FIS에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

인터페이스 VPC 엔드포인트를 사용하여 AWS FIS에 액세스(AWS PrivateLink)

인터페이스 VPC 엔드포인트를 생성하여 VPC와 AWS Fault Injection Service 간에 프라이빗 연결을 설정할 수 있습니다. VPC 엔드포인트는 인터넷 게이트웨이AWS PrivateLink, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결 없이 비공개로 AWS FIS APIs에 액세스할 수 있는 기술로 구동됩니다. VPC의 인스턴스는 AWS FIS APIs와 통신하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 탄력적 네트워크 인터페이스로 표현됩니다.

자세한 내용은 AWS PrivateLink 가이드의를 AWS 서비스 통한 액세스를 AWS PrivateLink 참조하세요.

AWS FIS VPC 엔드포인트에 대한 고려 사항

AWS FIS용 인터페이스 VPC 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의 인터페이스 VPC 엔드포인트를 AWS 서비스 사용하여에 액세스를 검토하세요.

AWS FIS는 VPC에서 모든 API 작업을 호출할 수 있도록 지원합니다.

AWS FIS용 인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 사용하여 AWS FIS 서비스에 대한 VPC 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 정보는 AWS PrivateLink 가이드의 VPC 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 AWS FIS용 VPC 엔드포인트를 생성합니다com.amazonaws.*region*.fis.

엔드포인트에 대해 프라이빗 DNS를 활성화하면 리전의 기본 DNS 이름, 예를 들어를 사용하여 AWS FIS에 API 요청을 할 수 있습니다`fis.us-east-1.amazonaws.com`.

AWS FIS에 대한 VPC 엔드포인트 정책 생성

AWS FIS에 대한 액세스를 제어하는 VPC 엔드포인트에 엔드포인트 정책을 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스

자세한 정보는 AWS PrivateLink 가이드의 엔드포인트 정책을 사용하여 VPC 엔드포인트에 대한 액세스 제어를 참조하세요.

예: 특정 AWS FIS 작업에 대한 VPC 엔드포인트 정책

다음 VPC 엔드포인트 정책은 모든 리소스에 대해 나열된 AWS FIS 작업에 대한 액세스 권한을 모든 보안 주체에게 부여합니다.

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "fis:ListExperimentTemplates",
            "fis:StartExperiment",
            "fis:StopExperiment",
            "fis:GetExperiment"
         ],
         "Resource":"*",
         "Principal":"*"
      }
   ]
}
```

예: 특정의 액세스를 거부하는 VPC 엔드포인트 정책 AWS 계정

다음 VPC 엔드포인트 정책은 모든 작업 및 리소스에 대한 지정된 AWS 계정 액세스를 거부하지만 모든 작업 및 리소스에 대한 다른 모든 AWS 계정 액세스 권한을 부여합니다.

```
{
   "Statement":[
      {
         "Effect": "Allow",
         "Action": "*",
         "Resource": "*",
         "Principal": "*"
      },
      {
         "Effect":"Deny",
         "Action": "*",
         "Resource": "*",
         "Principal": {
           "AWS": [ "123456789012" ]
         }
      }
   ]
}
```

AWS FIS 리소스에 태그 지정

태그는 사용자 또는 AWS 가 AWS 리소스에 할당하는 메타데이터 레이블입니다. 각 태그는 키와 값으로 구성됩니다. 사용자가 할당하는 태그에 대해 키와 값을 정의합니다. 예를 들어 키를 `purpose`로 정의하고 리소스 값을 `test`로 정의할 수 있습니다.

태그는 다음을 지원합니다.

- AWS 리소스를 식별하고 구성합니다. 많은 AWS 서비스가 태그 지정을 지원하므로 서로 다른 서비스의 리소스에 동일한 태그를 할당하여 리소스가 관련이 있음을 나타낼 수 있습니다.
- AWS 리소스에 대한 액세스를 제어합니다. 자세한 내용은 IAM 사용 설명서의 태그를 사용한 액세스 제어를 참조하세요.

태그 지정 제한

AWS FIS 리소스의 태그에는 다음과 같은 기본 제한이 적용됩니다.

- 리소스에 할당할 수 있는 최대 태그 수: 50
- 최대 키 길이: 유니코드 128자
- 최대 값 길이: 유니코드 256자
- 키 및 값에 사용할 수 있는 문자 - a-z, A-Z, 0-9, 공백, _ . : / = + - 및 @ 문자
- 키와 값은 대/소문자를 구분합니다
- 키 접두사로 `aws:`를 사용하지 마세요. AWS 사용을 위해 예약되어 있습니다.

태그 작업

다음 AWS Fault Injection Service(AWS FIS) 리소스는 태그 지정을 지원합니다.

- 작업
- 실험
- 실험 템플릿

콘솔을 사용하여 실험용 태그 및 실험 템플릿으로 작업할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 실험에 태그 지정
- 실험 템플릿에 태그 지정

다음 AWS CLI 명령을 사용하여 작업, 실험 및 실험 템플릿에 대한 태그를 사용할 수 있습니다.

- tag-resource – 리소스에 태그를 추가합니다.
- untag-resource – 리소스에서 태그를 제거합니다.
- list-tags-for-resource – 특정 리소스에 대한 태그를 나열합니다.

AWS Fault Injection Service의 할당량 및 제한 사항

AWS 계정 에는 각 AWS 서비스에 대해 이전에 제한이라고 하는 기본 할당량이 있습니다. 다르게 표시되지 않는 한, 리전별로 각 할당량이 적용됩니다. 아래 표에서 조정 가능으로 표시된 할당량에 대한 증가를 요청할 수 있습니다.

계정에서 AWS FIS에 대한 할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택하고 AWS Fault Injection Service를 선택합니다. 자동으로 승인된 할당량까지의 값은 즉시 적용됩니다. 자동으로 승인된 할당량은 아래 표의 설명 열에 요약되어 있습니다. 자동으로 승인된 한도를 초과하는 할당량이 필요한 경우 요청을 제출하세요. 자동 승인된 한도를 초과하는 값은 고객 지원에서 검토하고 가능한 경우 승인합니다.

할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

AWS 계정 에는 AWS FIS와 관련된 다음과 같은 할당량이 있습니다.

이름	기본값	조정 가능	설명
작업 기간 (시간)	지원되는 각 리전: 12	아니요	현재 리전에서 이 계정으로 하나의 작업을 실행할 수 있는 최대 시간.
aws:dynamodb:global-table-pause-replication 작업에 대한 다중 리전당 작업 분 (MRSC) 전역 테이블	지원되는 각 리전: 5,040	예	aws:dynamodb:global-table-pause-replication이 7일 롤링 기간에 대상으로 지정할 수 있는 MRSC 글로벌 테이블당 최대 누적 작업 분입니다.
실험 템플릿별 작업	지원되는 각 리전: 20	아니요	현재 리전에서 이 계정의 실험 템플릿에 만들 수 있는 작업 최대 수.

이름	기본값	조정 가능	설명
적극적인 실험	지원되는 각 리전: 5개	아니요	현재 리전에서 이 계정으로 동시에 실행할 수 있는 활성 실험 최대 수.
완료된 실험 데이터 보존 기간(일)	지원되는 각 리전: 120	아니요	AWS FIS가 현재 리전의 이 계정에서 완료된 실험에 대한 데이터를 유지할 수 있는 최대 일수입니다.
실험 기간(시간)	지원되는 각 리전: 12	아니요	현재 리전에서 이 계정으로 하나의 실험을 실행할 수 있는 최대 시간.
실험 템플릿	지원되는 각 리전: 500개	아니요	현재 리전의 이 계정에서 생성할 수 있는 실험 템플릿 최대 수.
aws: network:route-table-disrupt-cross-region-connectivity의 관리 접두사 목록 최대 개수	지원되는 각 리전: 15	아니요	작업당 aws:network:route-table-disrupt-cross-region-connectivity가 허용하는 관리되는 접두사 목록의 최대 개수.
aws:network:route-table-disrupt-cross-region-connectivity의 최대 라우팅 테이블 수	지원되는 각 리전: 10	아니요	작업당 aws:network:route-table-disrupt-cross-region-connectivity가 허용하는 라우팅 테이블의 최대 개수.
aws:network:route-table-disrupt-cross-region-connectivity의 최대 라우팅 수	지원되는 각 리전: 200	아니요	작업당 aws:network:route-table-disrupt-cross-region-connectivity가 허용하는 최대 라우팅 수.

이름	기본값	조정 가능	설명
실험당 병렬 작업	지원되는 각 리전: 10	아니요	현재 리전에서 이 계정으로 실험에 동시에 실행할 수 있는 작업의 최대 수.
실험 템플릿별 중지 조건	지원되는 각 리전: 5개	아니요	현재 리전에서 이 계정의 실험 템플릿에 추가할 수 있는 중지 조건의 최대 수.
aws:dsql:cluster-connection-failure 작업의 대상 Aurora DSQL 클러스터	지원되는 각 리전: 100	예	태그를 사용하여 대상을 식별할 때 실험당 aws:dsql:cluster-connection-failure가 대상으로 지정할 수 있는 최대 Aurora DSQL 클러스터 수입니다.
aws:ec2:asg-insufficient-instance-capacity-error에 대한 오토 스케일링 그룹 타겟팅	지원되는 각 리전: 500	예	태그를 사용하여 대상을 식별할 때 aws:ec2:asg-insufficient-instance-capacity-error가 실험당 타겟팅할 수 있는 최대 오토 스케일링 그룹 수.
aws:s3: bucket-pause-replication에 대한 대상 버킷	지원되는 각 지역: 25	예	태그를 사용하여 대상을 식별할 때 aws:s3:bucket-pause-replication이 실험당 대상으로 삼을 수 있는 최대 S3 버킷 수.

이름	기본값	조정 가능	설명
aws:ecs:드레인 컨테이너 인스턴스의 대상 클러스터	지원되는 각 리전: 100	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:vs:ecs:drain-instances가 대상으로 지정할 수 있는 클러스터의 최대 수.
aws:rds:failover-db-cluster의 대상 클러스터	지원되는 각 리전: 160	예	태그를 사용하여 대상을 식별할 때 실험당 aws:rds:rds:failover-db-cluster가 대상으로 지정할 수 있는 클러스터의 최대 수.
aws:rds:reboot-db-instans의 대상 DB 인스턴스	지원되는 각 리전: 130	예	태그를 사용하여 대상을 식별할 때 실험당 aws:rds:reboot-db-instants가 대상으로 지정할 수 있는 DB 인스턴스의 최대 수.
aws:ec2:reboot-instances의 대상 인스턴스	지원되는 각 지역: 600	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ec2:reboot-instances가 대상으로 지정할 수 있는 인스턴스의 최대 수.
aws:ec2:stop-instances의 대상 인스턴스	지원되는 각 리전: 400	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ec2:stop-instances가 대상으로 지정할 수 있는 인스턴스의 최대 수.

이름	기본값	조정 가능	설명
aws:ec2:terminate-instances의 대상 인스턴스	지원되는 각 리전: 300개	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ec2:terminate-instances가 대상으로 지정할 수 있는 인스턴스의 최대 수.
aws:ssm:send-command 대상 인스턴스	지원되는 각 리전: 200	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ssm:send-command가 타깃팅할 수 있는 최대 인스턴스 수.
aws:kinesis:stream-expired-iterator-exception 작업의 대상 Kinesis 데이터 스트림	지원되는 각 리전: 280	예	태그를 사용하여 대상을 식별할 때 실험당 aws:kinesis:stream-expired-iterator-exception이 대상으로 지정할 수 있는 최대 Kinesis 데이터 스트림 수입니다.
aws:kinesis:stream-provisioned-throughput-exception 작업의 대상 Kinesis 데이터 스트림	지원되는 각 리전: 280	예	태그를 사용하여 대상을 식별할 때 실험당 aws:kinesis:stream-provisioned-throughput-exception이 대상으로 지정할 수 있는 최대 Kinesis 데이터 스트림 수입니다.

이름	기본값	조정 가능	설명
aws:arc:start-zonal-autoshift 작업의 대상 ManagedResources	지원되는 각 리전: 200	예	태그를 사용하여 대상을 식별할 때 실험당 aws:arc:start-zonal-autoshift가 대상으로 지정할 수 있는 관리형 리소스의 최대 수입니다.
aws:eks:terminate-nodegroup-instances 대상 노드 그룹	지원되는 각 리전: 100	예	태그를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:ecs:terminate-nodegroup 인스턴스가 대상으로 지정할 수 있는 노드 그룹의 최대 수.
aws:ex:pod-cpu-스트레스용 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:oks:pod-cpu-s가 대상으로 지정할 수 있는 포드 최대 수.
aws:eks:pod-delete 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:pod-delete가 대상으로 지정할 수 있는 포드 최대 수.
aws:ex:pod-cpu-stress 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:pod-io-stress가 대상으로 지정할 수 있는 포드 최대 수.

이름	기본값	조정 가능	설명
aws:eks:pod-memory-stress 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:oks:pod-memory-s가 대상으로 지정할 수 있는 포드 최대 수.
aws:eks:pod-network-blackhole-port 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:eks:pod-network-blackhole-port가 대상으로 지정할 수 있는 파드의 최대 수.
aws:eks:pod-network-latency 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:oks:pod--network-latency가 대상으로 지정할 수 있는 포드의 최대 수.
aws:eks:pod-network-packet-loss 대상 포드	지원되는 각 리전: 1,000	예	파라미터를 사용하여 대상을 식별할 때 실험당 aws:eks:eks:eks:pod-network-packet-loss가 대상으로 지정할 수 있는 파드의 최대 수.

이름	기본값	조정 가능	설명
aws:elasticache:interrupt-cluster-az-power의 대상 복제 그룹 - 중단 예정	지원되는 각 리전: 5개	예	태그/파라미터를 사용하여 대상을 식별할 때 aws:elasticache:interrupt-cluster-az-power가 실험당 대상으로 삼을 수 있는 최대 복제 그룹 수.
aws:elasticache:replicationgroup-interrupt-az-power의 대상 복제 그룹	지원되는 각 리전: 20개	예	실험당 aws:elasticache:replicationgroup-interrupt-az-power가 대상으로 지정할 수 있는 최대 복제 그룹 수입니다. 복제 그룹 대상 지정에는 일일 제한이 적용됩니다. 자세한 내용을 알아보려면 https://docs.aws.amazon.com/fis/latest/userguide/fis-quotas.html 페이지를 방문하세요.
aws:ec2:send-spot-instance-interruptions에 대한 대상 스팟 인스턴스	지원되는 각 리전: 100	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ec2:send-spot-instance-interruptions가 대상으로 지정할 수 있는 스팟 인스턴스의 최대 수.

이름	기본값	조정 가능	설명
aws:network:disrupt-connectivity 대상 서브넷	지원되는 각 리전: 100	예	태그를 사용하여 대상을 식별할 때 실험당 ws:network:disrupt-connectivity가 대상으로 지정할 수 있는 서브넷의 최대 수. 5를 초과하는 할당량은 scope:all 파라미터에만 적용됩니다. 다른 범위 유형에 대해 더 높은 할당량이 필요한 경우 고객 지원 센터에 문의하세요.
aws:network:route-table-disrupt-cross-region-connectivity의 대상 서브넷	지원되는 각 지역: 50	예	태그를 사용하여 대상을 식별할 때 실험당 aws:network:route-table-disrupt-cross-region-connectivity가 타겟팅할 수 있는 최대 서브넷 수.
aws:ecs:stop-task 대상 태스크	지원되는 각 리전: 500	예	태그를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:stop-task가 대상으로 지정할 수 있는 작업의 최대 수.
aws:ecs:task-cpu-stress 대상 태스크	지원되는 각 리전: 200	예	태그/파라미터를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:ecs:task-cpu-stress가 대상으로 지정할 수 있는 작업의 최대 수.

이름	기본값	조정 가능	설명
aws:ecs:task-io-stress 대상 태스크	지원되는 각 리전: 200	예	태그/파라미터를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:task-io-stress가 대상으로 지정할 수 있는 작업의 최대 수.
aws:ecs:task-kill-process 대상 태스크	지원되는 각 리전: 200	예	태그/파라미터를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:task-kill-process가 대상으로 지정할 수 있는 작업의 최대 수.
aws:ecs:task-network-blackhole-port 대상 태스크	지원되는 각 리전: 200	예	태그/파라미터를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:ecs:task-network-blackhole-port가 대상으로 지정할 수 있는 작업의 최대 수.
aws:ecs:task-network-latency 대상 태스크	지원되는 각 리전: 200	예	태그/파라미터를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:ecs:task-network-latency가 대상으로 지정할 수 있는 작업의 최대 수.

이름	기본값	조정 가능	설명
aws:ecs:task-network-packet-loss 대상 태스크	지원되는 각 리전: 200	예	태그/파라미터를 사용하여 대상을 식별할 때 실험당 aws:ecs:ecs:ecs:eck-network-packet-loss가 대상으로 지정할 수 있는 작업의 최대 수.
aws: network:transit-gateway-disrupt-cross-region-connectivity에 대한 TransitGateway 타겟팅	지원되는 각 지역: 50	예	태그를 사용하여 대상을 식별할 때 실험당 aws:network:transit-gateway-disrupt-cross-region-connectivity가 대상으로 삼을 수 있는 최대 전송 게이트웨이의 수.
aws:directconnect:virtual-interface의 대상 가상 인터페이스	지원되는 각 리전: 5개	예	태그를 사용하여 대상을 식별할 때 실험당 aws:directconnect:virtual-interface가 대상으로 지정할 수 있는 최대 가상 인터페이스 수입니다.
aws:ebs:pause-volume-io의 대상 볼륨	지원되는 각 리전: 160	예	태그를 사용하여 대상을 식별할 때 aws:ebs:pause-volume-io가 실험당 대상으로 삼을 수 있는 최대 볼륨 수.
실험 템플릿별 대상 계정 구성	지원되는 각 리전: 40	예	현재 리전에서 이 계정의 실험 템플릿에 대해 만들 수 있는 최대 대상 계정 구성 수.

이름	기본값	조정 가능	설명
aws:lambda:invocation-add-delay 작업의 대상 함수	지원되는 각 지역: 140	예	태그를 사용하여 대상을 식별할 때 실험당 aws:lambda:invocation-add-delay가 대상으로 지정할 수 있는 최대 Lambda 함수 수입니다.
aws:lambda:invocation-error 작업의 대상 함수	지원되는 각 지역: 140	예	태그를 사용하여 대상을 식별할 때 실험당 aws:lambda:invocation-error가 대상으로 지정할 수 있는 최대 Lambda 함수 수입니다.
aws:lambda:invocation-http-integration-response 작업의 대상 함수	지원되는 각 지역: 140	예	태그를 사용하여 대상을 식별할 때 실험당 aws:lambda:invocation-http-integration-response가 대상으로 지정할 수 있는 최대 Lambda 함수 수입니다.
aws:memorydb:multi-region-cluster-pause-replication 작업의 대상 다중 리전 클러스터	지원되는 각 지역: 50	예	태그를 사용하여 대상을 식별할 때 실험당 aws:memorydb:multi-region-cluster-pause-replication이 대상으로 지정할 수 있는 최대 MemoryDB 다중 리전 클러스터 수입니다. 더 높은 할당량이 필요한 경우 고객 지원 센터에 문의하세요.

이름	기본값	조정 가능	설명
aws:dynamodb:global-table-pause-replication 작업의 대상 테이블	지원되는 각 리전: 60개	예	aws:dynamodb:global-table-pause-replication이 실험당 대상으로 삼을 수 있는 최대 글로벌 테이블 수.

AWS FIS 사용에는 다음과 같은 추가 제한 사항이 적용됩니다.

이름	제한 사항
대상에 대한 일일 제한 aws:elasticache:replicationgroup-interrupt-az-power에 대한 ReplicationGroups	한도는 하루에 리전별로 계정당 대상으로 지정된 ReplicationGroups 20개입니다. AWS Support Center 콘솔에서 지원 사례를 생성하여 증가를 요청할 수 있습니다.

문서 기록

다음 표에서는 AWS Fault Injection Service 사용 설명서의 중요한 설명서 업데이트에 대해 설명합니다.

변경 사항	설명	날짜
에 대한 새 작업AWS Direct Connect	aws:directconnect:virtual-interface 작업을 사용하여 온프레미스 네트워크와 대상 가상 인터페이스와 연결된 피어 간의 Border Gateway 프로토콜 세션을 일시적으로 중단하여 AWS Direct Connect연결 복원력을 테스트할 수 있습니다.	2025년 12월 18일
새로운 시나리오	이제 새로운 "AZ: Application Slowdown" 및 "Cross-AZ: Traffic Slowdown" 시나리오를 사용할 수 있습니다.	2025년 11월 12일
SSM 문서에 대한 새 파라미터	AWSFIS-Run-Network-Latency-Sources 및 AWSFIS-Run-Network-Packet-Loss-Sources SSM 문서는 이제 flowsPercent 파라미터를 지원합니다.	2025년 11월 12일
ECS 및 EKS 작업에 대한 새 파라미터	aws:ecs:task-network-latency, aws:ecs:task-network-packet-loss, aws:eks:pod-network-latency 및 aws:eks:pod-network-packet-loss 작업은 이제 flowsPercent 파라미터를 지원합니다.	2025년 11월 12일

AWS관리형 정책 업데이트	관리형 정책 AWSFaultInjectionSimulatorEC2Access가 업데이트됨: ec2:DescribeSubnets 권한이 추가되었습니다.	2025년 11월 12일
AWS관리형 정책 업데이트	관리형 정책 AWSFaultInjectionSimulatorECSAccess 업데이트: ecs:DescribeContainerInstances, ec2:DescribeSubnets 및 ec2:DescribeInstances 권한이 추가되었습니다.	2025년 11월 12일
Amazon Kinesis Data Streams에 대한 새로운 작업	aws:kinesis:stream 작업을 사용할 수 있습니다.	2025년 10월 15일
Amazon EBS에 대한 새로운 실험	aws:ebs:volume-io-latency 작업을 사용하여 Amazon EBS 볼륨에서 늘어난 I/O 지연 시간을 시뮬레이션할 수 있습니다.	2025년 9월 16일
새 작업 범위 AWSFIS	aws:network:disrupt-connectivity 작업을 사용하여 S3 Express One Zone 디렉터리 버킷에 대한 연결을 중단할 수 있습니다. 이 범위는 이제 AZ 가용성: 전원 중단 시나리오에도 포함됩니다.	2025년 8월 18일
AWSFIS에서 MemoryDB 지원	AWSFIS를 사용하여 Amazon MemoryDB 다중 리전 클러스터가 있는 애플리케이션이 리전 간 네트워크 중단 중에 일시 중지되는 데이터 복제에 어떻게 응답하는지 테스트할 수 있습니다.	2025년 7월 15일

AWSFIS에서 ARC 지원	AWSFIS를 사용하여 AZ 전원 중단 시 ARC 영역 자동 전환이 애플리케이션을 자동으로 복구하는 방법을 테스트할 수 있습니다.	2025년 3월 26일
새로운 실험 보고서 구성	이제 AWSFIS를 활성화하여 CloudWatch 대시보드에서 실험 작업 및 응답을 요약하는 실험에 대한 보고서를 생성할 수 있습니다.	2024년 11월 12일
새 Lambda 작업	이제 aws:lambda:function 작업을 사용하여 Lambda 함수 호출에 오류를 주입할 수 있습니다.	2024년 10월 31일
새로운 안전 레버 기능	AWS이제 FIS는 실행 중인 모든 실험을 빠르게 중지하고 새 실험이 시작되지 않도록 하는 안전 레버를 지원합니다.	2024년 9월 3일
새로 추가된 문제 해결 장	AWSFIS는 실패한 실험에 대한 오류 코드와 컨텍스트가 포함된 문제 해결 가이드를 추가했습니다.	2024년 8월 13일
새로운 작업	이제 `aws:dynamodb:global-table-pause-replication` 작업을 사용하여 대상 글로벌 테이블과 해당 복제본 테이블 간의 데이터 복제를 일시 중지할 수 있습니다. `aws:dynamodb:encrypted-global-table-pause-replication` 작업은 더 이상 지원되지 않습니다.	2024년 4월 24일

새로운 작업 모드 실험 옵션	실험을 실행하기 전에 대상 미리 보기를 생성하도록 작업 모드를 `skip-all`로 설정할 수 있습니다.	2024년 3월 13일
AWS관리형 정책 업데이트	AWSFIS는 기존 관리형 정책을 업데이트했습니다.	2024년 1월 25일
새 작업 및 시나리오	이제 AWSFIS 시나리오 교차 리전: 연결 및 AZ 가용성: 전원 중단을 사용할 수 있습니다.	2023년 11월 30일
새로운 작업	이제 aws:ec2:asg-insufficient-instance-capacity-error 작업을 사용할 수 있습니다.	2023년 11월 30일
새로운 작업	이제 aws:ec2:api-insufficient-instance-capacity-error 작업을 사용할 수 있습니다.	2023년 11월 30일
새로운 작업	이제 aws:network:route-table-disrupt-cross-region-connectivity 작업을 사용할 수 있습니다.	2023년 11월 30일
새로운 작업	이제 aws:network:transit-gateway-disrupt-cross-region-connectivity 작업을 사용할 수 있습니다.	2023년 11월 30일
새로운 작업	이제 aws:dynamodb:encrypted-global-table-pause-replication 작업을 사용할 수 있습니다.	2023년 11월 30일
새로운 작업	이제 aws:s3:bucket-pause-replication 작업을 사용할 수 있습니다.	2023년 11월 30일

새로운 작업	이제 aws:elasticache:interrupt-cluster-az-power 작업을 사용할 수 있습니다.	2023년 11월 30일
새 실험 옵션	이제 계정 대상 지정 및 빈 대상 확인에 AWSFIS 실험 옵션을 사용할 수 있습니다.	2023년 11월 27일
AWSFIS의 이름 변경	서비스 이름을 AWSFault Injection Service로 업데이트했습니다.	2023년 11월 15일
AWS관리형 정책 업데이트	AWSFIS는 기존 관리형 정책을 업데이트했습니다.	2023년 11월 13일
새 시나리오 라이브러리	이제 AWSFIS 시나리오 라이브러리 기능을 사용할 수 있습니다.	2023년 11월 7일
새 실험 스케줄러	이제 AWSFIS 실험 스케줄러 기능을 사용할 수 있습니다.	2023년 11월 7일
AWS관리형 정책 업데이트	AWSFIS는 기존 관리형 정책을 업데이트했습니다.	2023년 6월 2일
새로운 작업	새 aws:ecs:task 및 aws:eks:pod 작업을 사용할 수 있습니다.	2023년 6월 1일
AWS관리형 정책 업데이트	AWSFIS는 기존 관리형 정책을 업데이트했습니다.	2023년 6월 1일
사전 구성된 새 SSM 문서	다음과 같은 사전 구성된 SSM 문서인 AWSFIS-Run-Disk-Fill을 사용할 수 있습니다.	2023년 4월 28일
새로운 작업	aws:ebs:pause-volume-io 작업을 사용하여 대상 볼륨과 대상 볼륨이 연결된 인스턴스 간의 I/O를 일시 중지할 수 있습니다.	2023년 1월 27일

새로운 작업	aws:network:disrupt-connectivity 작업을 사용하여 대상 서브넷으로 향하는 특정 유형의 트래픽을 거부할 수 있습니다.	2022년 10월 26일
새로운 작업	aws:eks:inject-kubernetes-custom-resource 작업을 사용하여 단일 대상 클러스터에서 ChaosMesh 또는 Litmus 실험을 실행할 수 있습니다.	2022년 7월 7일
실험 로깅	실험 활동 로그를 CloudWatch Logs 또는 S3 버킷으로 전송하도록 실험 템플릿을 구성할 수 있습니다.	2022년 2월 28일
새 알림	실험 상태가 변경되면 AWSFIS는 알림을 보냅니다. 이러한 알림은 Amazon EventBridge를 통해 이벤트로 제공됩니다.	2022년 2월 24일
새로운 작업	aws:ecs:stop-task 작업을 사용하여 지정된 작업을 중지할 수 있습니다.	2022년 2월 9일
새로운 작업	aws:cloudwatch:assert-alarm-state 작업을 사용하여 지정된 경보가 지정된 경보 상태 중 하나에 해당하는지 확인할 수 있습니다.	2021년 11월 5일

사전 구성된 새 SSM 문서	사전 구성된 SSM 문서로는 AWSFIS-Run-IO-Stress, AWSFIS-Run-Network-Blackhold-Port, AWSFIS-Run-Network-Latency-Sources, AWSFIS-Run-Network-Packet-Loss, AWSFIS-Run-Network-Packet-Loss-Sources 등을 사용할 수 있습니다.	2021년 11월 4일
새로운 작업	aws:ec2:send-spot-instance-interruptions 작업을 사용하여 대상 스팟 인스턴스에 스팟 인스턴스 중단 알림을 보낸 다음 대상 스팟 인스턴스를 중단할 수 있습니다.	2021년 10월 20일
새로운 작업	aws:ssm:start-automation-execution 작업을 사용하여 자동화 런북의 실행을 시작할 수 있습니다.	2021년 9월 17일
최초 릴리스	AWSFault Injection Service 사용 설명서의 최초 릴리스입니다.	2021년 3월 15일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.