Guía del usuario

AWS Servicio de inyección de averías

AWS Servicio de inyección de averías: Guía del usuario

Table of Contents

¿Qué es AWS Fault Injection Service?

AWS Fault Injection Service (AWS FIS) es un servicio administrado que le permite realizar experimentos de inyección de errores en sus cargas de trabajo de AWS. La inyección de errores se basa en los principios de la ingeniería del caos. Estas pruebas hacen que una aplicación se esfuerce al crear eventos disruptivos para que pueda observar cómo responde su aplicación. A continuación, puede utilizar esta información para mejorar el rendimiento y la resiliencia de sus aplicaciones para que se comporten como se espera.

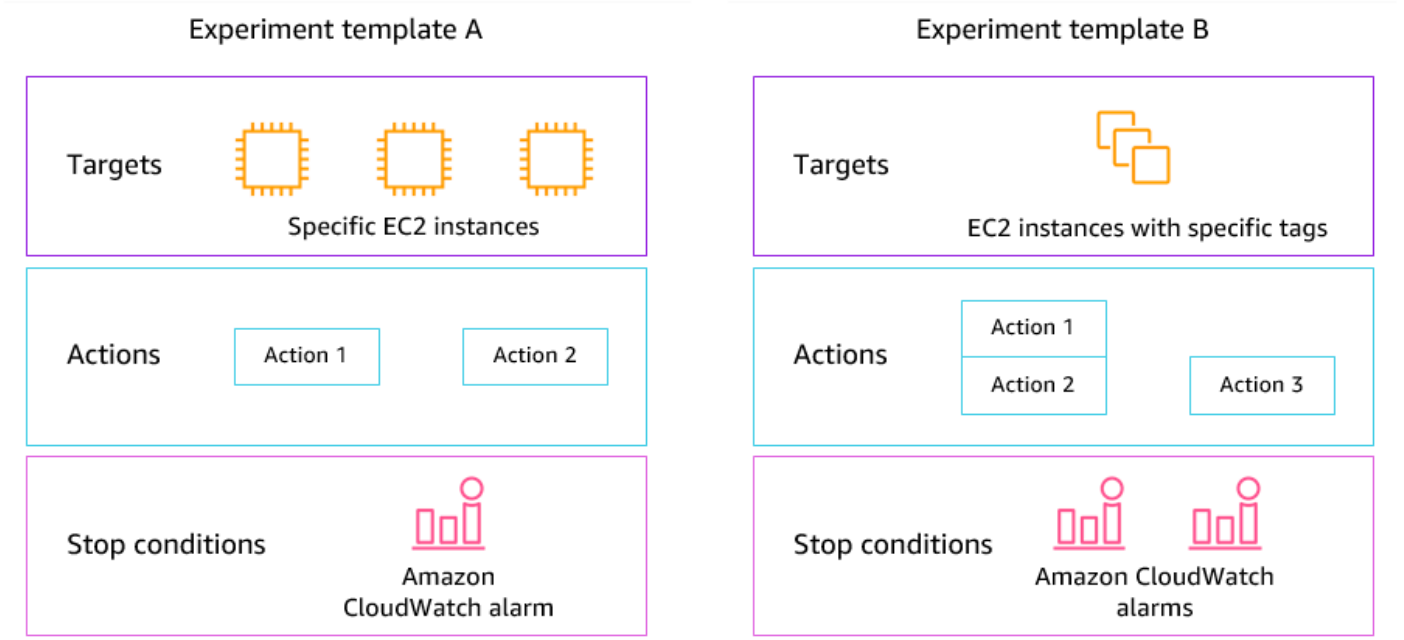
Para utilizar AWS FIS, debe configurar y ejecutar experimentos que le ayuden a crear las condiciones reales necesarias para descubrir problemas en las aplicaciones que, de otro modo, serían difíciles de detectar. AWS FIS proporciona plantillas que generan interrupciones y los controles y las barreras de protección que se necesitan para ejecutar experimentos en producción, como revertir o detener automáticamente el experimento si se cumplen condiciones específicas.

> **Important**
>
> AWS FIS lleva a cabo acciones reales con recursos de AWS reales de su sistema. Por lo tanto, antes de utilizar AWS FIS para realizar experimentos en producción, le recomendamos encarecidamente que complete una fase de planificación y ejecute los experimentos en un entorno de preproducción.

Para obtener más información sobre cómo planificar el experimento, consulte Comprobar la fiabilidad y Planificación de sus experimentos de AWS FIS. Para obtener más información acerca de AWS FIS, consulte AWS Fault Injection Service.

Conceptos de AWS FIS

Para utilizar AWS FIS, ejecute experimentos en sus recursos de AWS para poner a prueba su teoría sobre el rendimiento de una aplicación o un sistema en caso de error. Para ejecutar experimentos, primero debe crear una plantilla de experimento. Una plantilla de experimento es el esquema del experimento. Contiene las acciones, los destinos y las condiciones de detención del experimento. Después de crear una plantilla de experimento, puede utilizarla para ejecutar un experimento. Mientras se ejecuta el experimento, puede hacer un seguimiento de su progreso y ver su estado. Un experimento se completa cuando se han ejecutado todas las acciones del experimento.



Acciones

Una acción es una actividad que AWS FIS realiza en un recurso de AWS durante un experimento. AWS FIS proporciona un conjunto de acciones preconfiguradas en función del tipo de recurso AWS. Cada acción se ejecuta durante un tiempo específico durante un experimento o hasta que lo detenga. Las acciones se pueden ejecutar de forma secuencial o simultánea (en paralelo).

Destinos

Un destino es uno o más recursos de AWS sobre los que AWS FIS realiza una acción durante un experimento. Puede elegir recursos específicos o seleccionar un grupo de recursos en función de criterios específicos, como las etiquetas o el estado.

Condiciones de detención

AWS FIS proporciona los controles y las barreras de protección que necesita para ejecutar experimentos de forma segura en sus cargas de trabajo de AWS. Una condición de parada es un mecanismo para detener un experimento si alcanza un umbral que tú defines como CloudWatch alarma de Amazon. Si se activa una condición de detención mientras se está ejecutando el experimento, AWS FIS detiene el experimento.

Servicios de AWS admitidas

AWS FIS proporciona acciones preconfiguradas para tipos específicos de destinos en todos los servicios de AWS. AWS FIS admite acciones para recursos de destino para los siguientes Servicios de AWS:

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

Para los experimentos de una sola cuenta, los recursos de destino deben estar en la misma Cuenta de AWS que el experimento. Puede ejecutar experimentos de AWS FIS que se dirijan a los recursos de una cuenta de Cuenta de AWS diferente que utilice experimentos de varias cuentas de AWS FIS.

Para obtener más información, consulte Acciones para AWS FIS.

Acceso a AWS FIS

Puede trabajar con AWS FIS de cualquiera de las siguientes formas:

- AWS Management Console: proporciona una interfaz web que se puede utilizar para obtener acceso a AWS FIS. Para obtener más información, consulte Trabajar con la AWS Management Console.
- AWS Command Line Interface (AWS CLI): proporciona comandos para numerosos servicios de AWS, incluido AWS FIS, y es compatible con Windows, macOS y Linux. Para obtener más información, consulte AWS Command Line Interface. Para obtener más información sobre los comandos de AWS FIS, consulte fis en Referencia de comandos de la AWS CLI.

- AWS CloudFormation: crea plantillas que describen sus recursos de AWS. Las plantillas se utilizan para aprovisionar y administrar estos recursos como una única unidad. Para obtener más información, consulte Referencia de tipos de recursos de AWS Fault Injection Service.
- AWS SDK: proporcionan API específicas de cada lenguaje y se encargan de muchos de los detalles de conexión, tales como, el cálculo de firmas, el control de reintentos de solicitudes y el control de errores. Para obtener más información, consulte SDK de AWS.
- API de HTTPS: proporciona acciones de API de nivel bajo a las que puede llamar mediante solicitudes HTTPS. Para obtener más información, consulte Referencia de la API de AWS Fault Injection Service.

Precios de AWS FIS

Se le cobrará por minuto de ejecución de una acción, de principio a fin, en función del número de cuentas objetivo del experimento. Para obtener más información, consulte Precios de AWS FIS.

Planificación de sus experimentos de AWS FIS

La inyección de errores es el proceso de sobrecarga de una aplicación en entornos de prueba o producción creando eventos disruptivos, como interrupciones del servidor o limitaciones de la API. Al observar cómo responde el sistema, se pueden implementar mejoras. Realizar experimentos en su sistema puede ayudarle a identificar las debilidades sistémicas de forma controlada, antes de que esas debilidades afecten a los clientes que dependen de su sistema. De este modo, podrá abordar los problemas de forma proactiva para evitar resultados impredecibles.

Antes de empezar a realizar experimentos de inyección de errores con AWS FIS, le recomendamos que se familiarice con los siguientes principios y directrices.

> **Important**
>
> AWS FIS lleva a cabo acciones reales con recursos de AWS reales de su sistema. Por lo tanto, antes de empezar a utilizar AWS FIS para realizar experimentos, le recomendamos encarecidamente que complete primero una fase de planificación y una prueba en un entorno de preproducción o de prueba.

Contenido

Principios y directrices básicos

Antes de iniciar los experimentos con AWS FIS, siga los pasos siguientes:

1. Identifique la implementación de destino del experimento: comience por identificar la implementación de destino. Si este es su primer experimento, le recomendamos que comience en un entorno de preproducción o de prueba.
2. Revise la arquitectura de la aplicación: asegúrese de haber identificado todos los componentes, las dependencias y los procedimientos de recuperación de cada componente de la aplicación. Comience por revisar la arquitectura de la aplicación. Según la aplicación, consulte Marco de AWS Well-Architected.

3. Defina el comportamiento de estado estable: defina el comportamiento de estado estable de su sistema en términos de métricas técnicas y empresariales importantes, como la latencia, la carga de la CPU, los inicios de sesión fallidos por minuto, el número de reintentos o la velocidad de carga de la página.
4. Formule una hipótesis: formule una hipótesis sobre cómo espera que cambie el comportamiento del sistema durante el experimento. La definición de una hipótesis sigue este formato:

   Si se lleva a cabo una *acción de inyección de errores*, el *impacto de las métricas empresariales o técnicas* no debe superar *valor*.

   Por ejemplo, una hipótesis para un servicio de autenticación podría ser la siguiente: “Si la latencia de la red aumenta un 10 %, los errores de inicio de sesión aumentan menos del 1 %”. Una vez finalizado el experimento, debe evaluar si la resiliencia de la aplicación se ajusta a sus expectativas empresariales y técnicas.

También recomendamos seguir estas pautas cuando trabaje con AWS FIS:

- Comience siempre a experimentar con AWS FIS en un entorno de prueba. Nunca comience con un entorno de producción. A medida que avance en sus experimentos de inyección de errores, podrá experimentar en otros entornos controlados además del entorno de prueba.
- Aumente la confianza de su equipo en la resiliencia de sus aplicaciones empezando con experimentos pequeños y sencillos, como ejecutar la acción aws:ec2:stop-instances en un destino.
- La inyección de errores puede causar problemas reales. Proceda con precaución y asegúrese de que sus primeras inyecciones de errores se realicen en instancias de prueba para que ningún cliente se vea afectado.
- Pruebe, pruebe y pruebe un poco más. La inyección de errores está destinada a implementarse en un entorno controlado con experimentos bien planificados. Esto le permite aumentar la confianza en la capacidad de su aplicación y sus herramientas para soportar condiciones turbulentas.
- Le recomendamos encarecidamente que cuente con un excelente programa de monitorización y alertas antes de empezar. Sin él, no podrá comprender ni medir el impacto de sus experimentos, lo cual es fundamental para que las prácticas de inyección de errores sean sostenibles.

Directrices de planificación de experimentos

Con AWS FIS, puede realizar experimentos en sus recursos de AWS para poner a prueba su teoría sobre el rendimiento de una aplicación o un sistema en caso de error.

A continuación, las directrices recomendadas para planificar sus experimentos de AWS FIS.

- Revise el historial de interrupciones: revise las interrupciones y eventos anteriores de su sistema. Esto puede ayudarle a hacerse una idea del estado general y la capacidad de recuperación de su sistema. Antes de empezar a realizar experimentos en el sistema, debe abordar los problemas y puntos débiles conocidos del sistema.
- Identifique los servicios que tienen el mayor impacto: revise sus servicios e identifique los que tienen el mayor impacto en sus usuarios finales o clientes si dejan de funcionar o no funcionan correctamente.
- Identifique el sistema de destino: el sistema de destino es el sistema en el que realizará los experimentos. Si es la primera vez que utiliza AWS FIS o nunca ha realizado experimentos de inyección de errores, le recomendamos que empiece por realizar los experimentos en un sistema de preproducción o de prueba.
- Consulte con su equipo: pregunte qué es lo que les preocupa. Puede formular una hipótesis para probar o refutar sus preocupaciones. También puede preguntarle a su equipo qué es lo que no les preocupa. Esta pregunta puede revelar dos falacias comunes: la falacia del costo irrecuperable y la falacia del sesgo de confirmación. Formular una hipótesis basada en las respuestas de su equipo puede ayudar a proporcionar más información sobre la realidad del estado de su sistema.
- Revise la arquitectura de su aplicación: realice una revisión de su sistema o aplicación, y asegúrese de haber identificado todos los componentes, las dependencias y los procedimientos de recuperación de cada componente de la aplicación.

  Le recomendamos que revise Marco de AWS Well-Architected. El marco puede ayudarle a crear infraestructuras seguras, de alto rendimiento, resistentes y eficientes para las aplicaciones y cargas de trabajo. Para obtener más información, consulte AWS Well-Architected.
- Identifica las métricas aplicables: puedes monitorizar el impacto de un experimento en tus AWS recursos mediante CloudWatch las métricas de Amazon. Puede utilizar estas métricas para determinar el punto de referencia o el “estado estable” en el que su aplicación tiene un rendimiento óptimo. A continuación, puede monitorizar estas métricas durante el experimento o después de él para determinar el impacto. Para obtener más información, consulte Monitorización de métricas de uso de AWS FIS con Amazon CloudWatch.
- Defina un umbral de rendimiento aceptable para su sistema: identifique la métrica que represente un estado estable aceptable para su sistema. Utilizará esta métrica para crear una o más CloudWatch alarmas que representen una condición de interrupción del experimento. Si se activa la alarma, el experimento se detiene automáticamente. Para obtener más información, consulte Condiciones de detención para AWS FIS.

Tutoriales sobre el servicio de inyección de AWS fallas

Los siguientes tutoriales muestran cómo crear y ejecutar experimentos con el servicio de inyección de AWS fallos (AWS FIS).

Tutoriales

- Tutorial: Prueba de detención e inicio de instancias con AWS FIS
- Tutorial: Ejecutar esfuerzo de la CPU en una instancia con AWS FIS
- Tutorial: Prueba de las interrupciones de instancias de spot con AWS FIS.
- Tutorial: Simulación de un evento de conectividad
- Tutorial: Programación de un experimento recurrente

Tutorial: Prueba de detención e inicio de instancias con AWS FIS

Puede usar el servicio AWS Fault Injection Service (AWS FIS) para probar cómo gestionan sus aplicaciones la detención y el inicio de las instancias. Utilice este tutorial para crear una plantilla de experimento que utilice la acción `aws:ec2:stop-instances` de AWS FIS para detener una instancia y, a continuación, una segunda instancia.

Requisitos previos

Para completar este tutorial, haga lo siguiente:

- Lance dos instancias EC2 de prueba en su cuenta. Después de lanzar las instancias, tenga en cuenta los ID de ambas instancias.
- Cree un rol de IAM que permita al servicio AWS FIS realizar la acción `aws:ec2:stop-instances` en su nombre. Para obtener más información, consulte Roles de IAM para los experimentos de AWS FIS.
- Asegúrese de tener acceso a AWS FIS. Para obtener más información, consulte Ejemplos de política de AWS FIS.

Paso 1: Crear una plantilla de experimento

Cree la plantilla de experimento con la consola AWS FIS. En la plantilla, especifique dos acciones que se ejecutarán secuencialmente durante tres minutos cada una. La primera acción detiene una de

las instancias de prueba, que AWS FIS elige al azar. La segunda acción detiene ambas instancias de prueba.

Para crear una plantilla de experimento

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Elija Crear plantilla de experimento.
4. En Descripción y nombre, escriba un nombre y una descripción para la plantilla.
5. En Actions (Acciones), haga lo siguiente:
   a. Seleccione Agregar acción.
   b. Escriba un nombre para la acción. Por ejemplo, escriba **`stopOneInstance`**.
   c. En Tipo de acción, elija aws:ec2:stop-instances.
   d. En Destino, mantenga el destino que AWS FIS crea automáticamente.
   e. En Parámetros de acción, Iniciar instancias después de la duración, especifique 3 minutos (PT3M).
   f. Seleccione Guardar.
6. En Targets (Destinos), haga lo siguiente:
   a. Elija Editar en el destino que AWS FIS creó automáticamente en el paso anterior.
   b. Sustituya el nombre por defecto por un nombre más descriptivo. Por ejemplo, escriba **`oneRandomInstance`**.
   c. Compruebe que Tipo de recurso sea aws:ec2:instance.
   d. En Método de destino, elija ID de recurso y, a continuación, elija los ID de las dos instancias de prueba.
   e. En Modo de selección, elija Recuento. En Cantidad de recursos, escriba **1**.
   f. Seleccione Guardar.
7. Elija Agregar destino y haga lo siguiente:
   a. Escriba un nombre para el destino. Por ejemplo, escriba **`bothInstances`**.
   b. En Tipo de recurso, elija aws:ec2:instance.
   c. En Método de destino, elija ID de recurso y, a continuación, elija los ID de las dos instancias de prueba.

d. En Modo de selección, elija Todos.

e. Seleccione Guardar.

8. En la sección Acciones, elija Agregar acción. Haga lo siguiente:

   a. En Nombre, escriba un nombre para la acción. Por ejemplo, escriba **`stopBothInstances`**.

   b. En Tipo de acción, elija aws:ec2:stop-instances.

   c. En Comenzar después, elija la primera acción que haya agregado (**`stopOneInstance`**).

   d. En Destino, elija el segundo destino que haya agregado (**`bothInstances`**).

   e. En Parámetros de acción, Iniciar instancias después de la duración, especifique 3 minutos (PT3M).

   f. Seleccione Guardar.

9. En Acceso al servicio, elija Usar un rol de IAM existente y, a continuación, elija el rol de IAM que creó, tal como se describe en los requisitos previos de este tutorial. Si su rol no aparece, compruebe que tiene la relación de confianza requerida. Para obtener más información, consulte the section called “Rol de experimento”.

10. (Opcional) En Etiquetas, elija Agregar nueva etiqueta y especifique una clave y un valor de etiqueta. Las etiquetas que agregue se aplican a la plantilla de experimento, no a los experimentos que se ejecutan con la plantilla.

11. Elija Crear plantilla de experimento. Cuando se le solicite confirmación, ingrese **`create`** y luego, elija Creación de la plantilla de experimento.

(Opcional) Para ver la plantilla de experimento JSON

Elija la pestaña Exportar. A continuación, verá un ejemplo del JSON creado por el procedimiento de consola anterior.

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

Paso 2: Iniciar el experimento

Cuando haya terminado de crear la plantilla de experimento, podrá utilizarla para iniciar un experimento.

Para iniciar un experimento

1. Debería estar en la página de detalles de la plantilla de experimento que acaba de crear. De lo contrario, elija Plantillas de experimento y, a continuación, seleccione el ID de la plantilla de experimento para abrir la página de detalles.
2. Elija Start experiment (Iniciar experimento).
3. (Opcional) Para agregar una etiqueta a su experimento, elija Agregar nueva etiqueta e ingrese una clave y un valor de etiqueta.
4. Elija Start experiment (Iniciar experimento). Cuando se le pida que confirme, ingrese **`start`** y elija Iniciar experimento.

Paso 3: Hacer un seguimiento del progreso del experimento

Puede hacer un seguimiento del progreso de un experimento en ejecución hasta que se complete, se detenga o falle.

Para hacer un seguimiento del progreso de un experimento

1. Debería estar en la página de detalles del experimento que acaba de iniciar. De lo contrario, elija Experimentos y, a continuación, seleccione el ID del experimento para abrir la página de detalles.
2. Para ver el estado del experimento, seleccione Estado en el panel Detalles. Para obtener más información, consulte Estados de experimento.
3. Vaya al siguiente paso cuando el estado del experimento sea En ejecución.

Paso 4: Verificar el resultado del experimento

Puede comprobar que el experimento detuvo e inició las instancias tal y como se esperaba.

Para verificar el resultado del experimento

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/ en una nueva pestaña o ventana del navegador. Esto le permite seguir el progreso del experimento en la consola de AWS FIS mientras ve el resultado del experimento en la consola de Amazon EC2.
2. En el panel de navegación, seleccione Instancias.
3. Cuando el estado de la primera acción cambia de Pendiente a En ejecución (consola de AWS FIS), el estado de una de las instancias de destino cambia de En ejecución a Detenida (consola de Amazon EC2).
4. Transcurridos tres minutos, el estado de la primera acción cambia a Finalizada, el estado de la segunda acción cambia a En ejecución y el estado de la otra instancia de destino cambia a Detenida.
5. Transcurridos tres minutos, el estado de la segunda acción cambia a Finalizada, el estado de las instancias de destino cambian a En ejecución y el estado del experimento cambia a Finalizado.

Paso 5: Eliminar

Si ya no necesita las instancias EC2 de prueba que creó para este experimento, puede terminarlas.

Para terminar las instancias

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, seleccione Instancias.
3. Seleccione ambas instancias de prueba y elija Instance state (Estado de la instancia) y Terminate instance (Terminar instancia).
4. Cuando se le indique que confirme, elija Terminate (Rescindir).

Si ya no necesita la plantilla de experimento, puede eliminarla.

Para eliminar una plantilla de experimento con la consola de AWS FIS

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Eliminar plantilla de experimento.
4. Cuando se le solicite confirmación, ingrese **`delete`** y luego, elija Eliminar plantilla de experimento.

Tutorial: Ejecutar esfuerzo de la CPU en una instancia con AWS FIS

Puede usar AWS Fault Injection Service (AWS FIS) para probar cómo gestionan sus aplicaciones el esfuerzo de la CPU. Usa este tutorial para crear una plantilla de experimento que use AWS FIS para ejecutar un documento de SSM preconfigurado que ejecute esfuerzo de la CPU en una instancia. El tutorial utiliza una condición de detención para detener el experimento cuando la utilización de la CPU de la instancia supera un umbral configurado.

Para obtener más información, consulte the section called “Documentos AWS FIS SSM preconfigurados”.

Requisitos previos

Antes de poder utilizar AWS FIS para ejecutar esfuerzo de la CPU, se deben completar los siguientes requisitos previos:

Crear un rol de IAM

Cree un rol y adjunte una política que permita a AWS FIS utilizar la acción `aws:ssm:send-command` en su nombre. Para obtener más información, consulte Roles de IAM para los experimentos de AWS FIS.

Verificación del acceso a AWS FIS

Asegúrese de tener acceso a AWS FIS. Para obtener más información, consulte Ejemplos de política de AWS FIS.

Preparación de una instancia EC2 de prueba

- Lance una instancia EC2 con Amazon Linux 2 o Ubuntu, tal y como exigen los documentos de SSM preconfigurados.
- SSM debe administrar la instancia. Para comprobar que SSM administra la instancia, abra la consola de Fleet Manager. Si SSM no administra la instancia, compruebe que el agente SSM esté instalado y que la instancia tenga una función de IAM asociada a la política de AmazonSSM. ManagedInstanceCore Para verificar el SSM Agent instalado, conéctese a la instancia y ejecute el siguiente comando.

  Amazon Linux 2

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

Ubuntu

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

- Habilite la monitorización detallada para la instancia. Esto proporciona datos en periodos de 1 minuto por un cargo adicional. Seleccione la instancia y elija Acciones, Monitoreo y solución de problemas, Administrar el monitoreo detallado.

Paso 1: Cree una alarma para una condición de parada CloudWatch

Configure una CloudWatch alarma para poder detener el experimento si la utilización de la CPU supera el umbral que especifique. El siguiente procedimiento establece el umbral en un 50 % de uso de la CPU para la instancia de destino. Para obtener más información, consulte Condiciones de detención.

Para crear una alarma que indique cuándo el uso de la CPU supera un umbral

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, seleccione Instancias.
3. Seleccione la instancia de destino y elija Acciones, Supervisar y solucionar problemas y Administrar CloudWatch alarmas.
4. En Notificación de alarma, active la opción para desactivar las notificaciones de Amazon SNS.
5. En Umbrales de alarma, utilice los siguientes ajustes:
   - Agrupar muestras por: Máximo
   - Tipo de datos para la muestra: Utilización de la CPU
   - Porcentaje: **`50`**
   - Período: **`1 Minute`**
6. Cuando haya terminado de configurar la alarma, elija Crear.

Paso 2: Crear una plantilla de experimento

Cree la plantilla de experimento con la consola AWS FIS. En la plantilla, debe especificar la siguiente acción para ejecutarla: AWSFISaws:ssm:send-command/ -Run-CPU-Stress.

Para crear una plantilla de experimento

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Elija Crear plantilla de experimento.
4. En Descripción y nombre, escriba un nombre y una descripción para la plantilla.
5. En Actions (Acciones), haga lo siguiente:
    a. Seleccione Agregar acción.
    b. Escriba un nombre para la acción. Por ejemplo, escriba **`runCpuStress`**.
    c. Como Tipo de acción, elija AWSFIS aws:ssm:send-command/ -Run-CPU-Stress. Esto agrega automáticamente el ARN del documento de SSM al ARN de documento.
    d. En Destino, mantenga el destino que AWS FIS crea automáticamente.
    e. En Parámetros de acción, Parámetros del documento, ingrese lo siguiente:

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

    f. En Parámetros de acción, Duración, especifique 5 minutos (PT5M).
    g. Seleccione Guardar.
6. En Targets (Destinos), haga lo siguiente:
    a. Elija Editar en el destino que AWS FIS creó automáticamente en el paso anterior.
    b. Sustituya el nombre por defecto por un nombre más descriptivo. Por ejemplo, escriba **`testInstance`**.
    c. Compruebe que Tipo de recurso sea aws:ec2:instance.
    d. En Método de destino, elija ID de recurso y, a continuación, elija el ID de la instancia de prueba.
    e. En Modo de selección, elija Todos.
    f. Seleccione Guardar.

7. En Acceso al servicio, elija Usar un rol de IAM existente y, a continuación, elija el rol de IAM que creó, tal como se describe en los requisitos previos de este tutorial. Si su rol no aparece, compruebe que tiene la relación de confianza requerida. Para obtener más información, consulte the section called “Rol de experimento”.
8. Para las condiciones de parada, seleccione la alarma que creó en el paso 1. CloudWatch
9. (Opcional) En Etiquetas, elija Agregar nueva etiqueta y especifique una clave y un valor de etiqueta. Las etiquetas que agregue se aplican a la plantilla de experimento, no a los experimentos que se ejecutan con la plantilla.
10. Elija Crear plantilla de experimento.

(Opcional) Para ver la plantilla de experimento JSON

Elija la pestaña Exportar. A continuación, verá un ejemplo del JSON creado por el procedimiento de consola anterior.

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

Paso 3: Iniciar el experimento

Cuando haya terminado de crear la plantilla de experimento, podrá utilizarla para iniciar un experimento.

Para iniciar un experimento

1. Debería estar en la página de detalles de la plantilla de experimento que acaba de crear. De lo contrario, elija Plantillas de experimento y, a continuación, seleccione el ID de la plantilla de experimento para abrir la página de detalles.
2. Elija Start experiment (Iniciar experimento).
3. (Opcional) Para agregar una etiqueta a su experimento, elija Agregar nueva etiqueta e ingrese una clave y un valor de etiqueta.
4. Elija Start experiment (Iniciar experimento). Cuando se le solicite confirmación, ingrese **`start`**. Elija Start experiment (Iniciar experimento).

Paso 4: Hacer un seguimiento del progreso del experimento

Puede hacer un seguimiento del progreso de un experimento en ejecución hasta que se complete, se detenga o falle.

Para hacer un seguimiento del progreso de un experimento

1. Debería estar en la página de detalles del experimento que acaba de iniciar. De lo contrario, seleccione Experimentos y, a continuación, seleccione el ID del experimento para abrir la página de detalles del experimento.
2. Para ver el estado del experimento, seleccione Estado en el panel Detalles. Para obtener más información, consulte Estados de experimento.

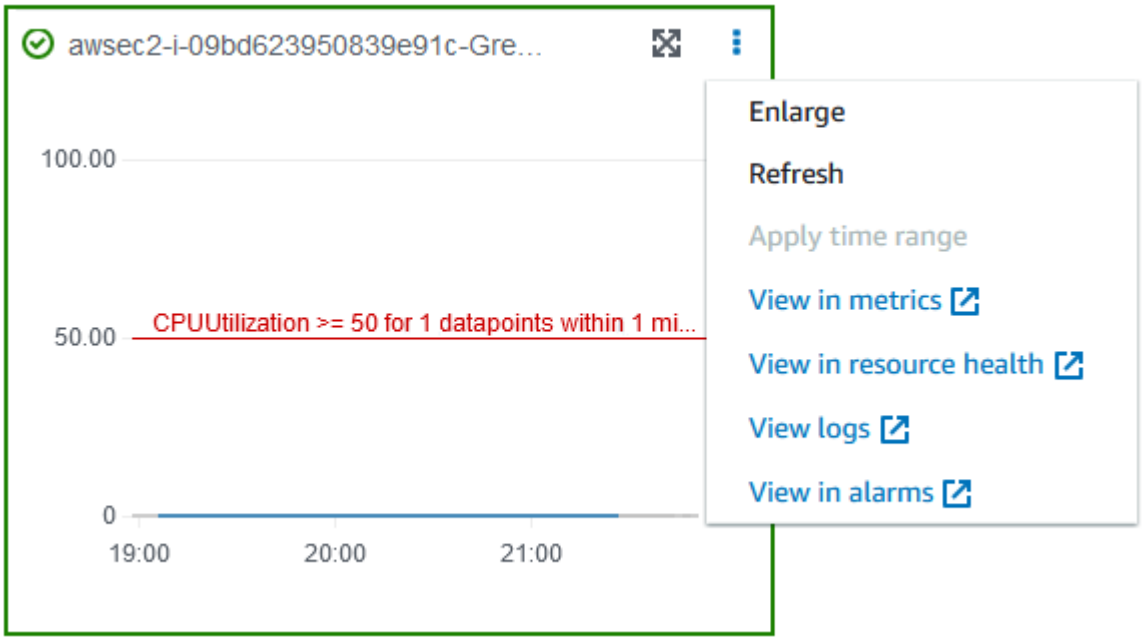
3. Cuando el estado del experimento sea En ejecución, pase al siguiente paso.

Paso 5: Verificar los resultados del experimento

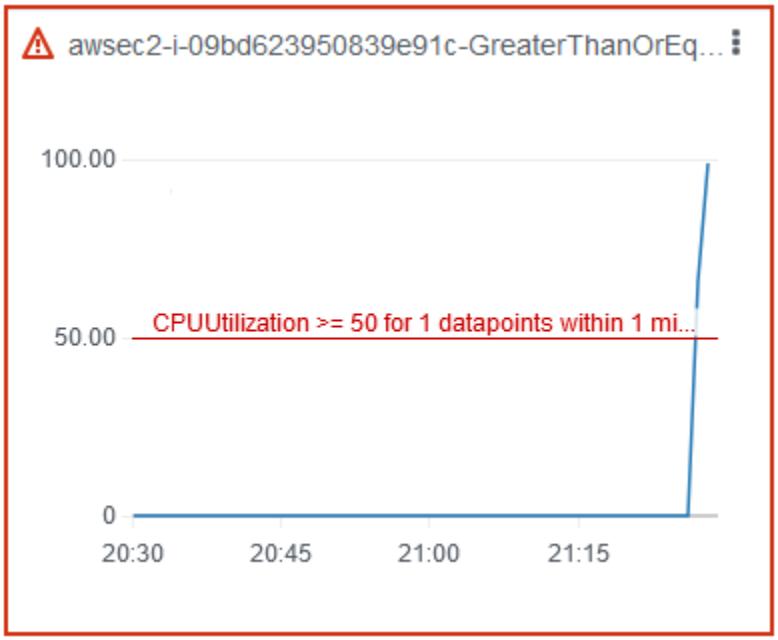
Puede monitorizar el uso de la CPU de la instancia mientras se ejecuta el experimento. Cuando el uso de la CPU alcanza el umbral, se activa la alarma y el experimento se detiene cuando se produce la condición de detención.

Para verificar los resultados del experimento

1. Seleccione la pestaña Condiciones de detención. El borde y el icono de marca de verificación verdes indican que el estado inicial de la alarma es `OK`. La línea roja indica el umbral de alarma. Si prefiere un gráfico más detallado, elija Ampliar en el menú del widget.



2. Cuando el uso de la CPU supera el umbral, el borde y el icono de signo de exclamación rojos de la pestaña Condiciones de detención indican que el estado de alarma ha cambiado a `ALARM`. En el panel Detalles, el estado del experimento es Detenido. Si selecciona el estado, el mensaje que aparece es “El experimento se ha detenido por la condición de detención”.



3. Cuando el uso de la CPU cae por debajo del umbral, el borde y el icono de marca de verificación verdes indican que el estado de alarma ha cambiado a 0K.
4. (Opcional) Seleccione Ver en alarmas en el menú del widget. Esto abre la página de detalles de la alarma en la CloudWatch consola, donde puede obtener más detalles sobre la alarma o editar la configuración de la alarma.

Paso 6: Limpiar

Si ya no necesita la instancia EC2 de prueba que creó para este experimento, puede terminarla.

Para terminar la instancia

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, seleccione Instancias.
3. Seleccione las instancias de prueba y elija Estado de instancia y Terminar instancia.
4. Cuando se le indique que confirme, elija Terminate (Rescindir).

Si ya no necesita la plantilla de experimento, puede eliminarla.

Para eliminar una plantilla de experimento con la consola de AWS FIS

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.

2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Eliminar plantilla de experimento.
4. Cuando se le solicite confirmación, ingrese **`delete`** y luego, elija Eliminar plantilla de experimento.

Tutorial: Prueba de las interrupciones de instancias de spot con AWS FIS.

Las instancias de spot utilizan la capacidad sobrante de EC2 que está disponible, con un descuento de hasta un 90 % en comparación con los precios bajo demanda. Sin embargo, Amazon EC2 puede interrumpir sus instancias de spot cuando necesite su capacidad. Cuando utilice instancias de spot, debe estar preparado para las posibles interrupciones. Para obtener más información, consulte Interrupciones de instancias de spot en la Guía del usuario de Amazon EC2.

Puede usar AWS Fault Injection Service (AWS FIS) para probar cómo gestionan sus aplicaciones una interrupción de instancia de spot. Utilice este tutorial para crear una plantilla de experimento que utilice la acción `aws:ec2:send-spot-instance-interruptions` de AWS FIS para interrumpir una instancia y, a continuación, una de sus instancias de spot.

Como alternativa, para iniciar el experimento con la consola de Amazon EC2, consulte Inicio de una interrupción de instancias de spot en la Guía del usuario de Amazon EC2.

Requisitos previos

Antes de utilizar AWS FIS para interrumpir una instancia de spot, complete los siguientes requisitos previos.

1. Crear un rol de IAM

Cree un rol y asocie una política que permita a AWS FIS realizar la acción `aws:ec2:send-spot-instance-interruptions` en su nombre. Para obtener más información, consulte Roles de IAM para los experimentos de AWS FIS.

2. Verificación del acceso a AWS FIS

Asegúrese de tener acceso a AWS FIS. Para obtener más información, consulte Ejemplos de política de AWS FIS.

3. (Opcional) Crear una solicitud de instancia de spot

Si desea utilizar una nueva instancia de spot en este experimento, utilice el comando run-instances para solicitar una instancia de spot. El comportamiento predeterminado es terminar las instancias de spot cuando se interrumpen. Si establece el comportamiento de interrupción en `stop`, también debe establecer el tipo en `persistent`. Para este tutorial, no defina el comportamiento de interrupción en `hibernate`, ya que el proceso de hibernación comenzará inmediatamente.

```
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
```

A continuación se muestra un ejemplo del archivo `spot-options.json`.

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

La opción `--query` del comando de ejemplo permite que el comando devuelva solo el ID de instancia de la instancia de spot. A continuación, se muestra un ejemplo del resultado.

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

4. Agregar una etiqueta para que AWS FIS pueda identificar la instancia de spot de destino

Use el comando create-tags para agregar la etiqueta Name=interruptMe a la instancia de spot de destino.

```
aws ec2 create-tags \
```

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

Paso 1: Crear una plantilla de experimento

Cree la plantilla de experimento con la consola AWS FIS. En la plantilla, especifique la acción que se ejecutará. La acción interrumpe la instancia de spot con la etiqueta especificada. Si hay más de una instancia de spot con la etiqueta, AWS FIS elige una de ellas al azar.

Para crear una plantilla de experimento

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Elija Crear plantilla de experimento.
4. En Descripción y nombre, escriba un nombre y una descripción para la plantilla.
5. En Actions (Acciones), haga lo siguiente:
    a. Seleccione Agregar acción.
    b. Escriba un nombre para la acción. Por ejemplo, escriba **`interruptSpotInstance`**.
    c. En Tipo de acción, elija aws:ec2:. send-spot-instance-interruptions
    d. En Destino, mantenga el destino que AWS FIS crea automáticamente.
    e. En Parámetros de acción, Duración antes de la interrupción, especifique 2 minutos (PT2M).
    f. Seleccione Guardar.
6. En Targets (Destinos), haga lo siguiente:
    a. Elija Editar en el destino que AWS FIS creó automáticamente en el paso anterior.
    b. Sustituya el nombre por defecto por un nombre más descriptivo. Por ejemplo, escriba **`oneSpotInstance`**.
    c. Compruebe que Tipo de recurso sea aws:ec2:spot-instance.
    d. En Método de destino, elija Etiquetas, filtros y parámetros de recursos.
    e. En Etiquetas de recursos, elija Agregar nueva etiqueta e ingrese la clave y el valor de la etiqueta. Utilice la etiqueta que ha agregado a la instancia de spot que se va a interrumpir, tal y como se describe en Requisitos previos de este tutorial.
    f. En Filtros de recursos, elija Agregar nuevo filtro e ingrese **`State.Name`** como ruta y **`running`** como valor.

g. En Modo de selección, elija Recuento. En Cantidad de recursos, escriba **1**.

h. Seleccione Guardar.

7. En Acceso al servicio, elija Usar un rol de IAM existente y, a continuación, elija el rol de IAM que creó, tal como se describe en los requisitos previos de este tutorial. Si su rol no aparece, compruebe que tiene la relación de confianza requerida. Para obtener más información, consulte the section called “Rol de experimento”.

8. (Opcional) En Etiquetas, elija Agregar nueva etiqueta y especifique una clave y un valor de etiqueta. Las etiquetas que agregue se aplican a la plantilla de experimento, no a los experimentos que se ejecutan con la plantilla.

9. Elija Crear plantilla de experimento. Cuando se le solicite confirmación, ingrese **`create`** y luego, elija Creación de la plantilla de experimento.

(Opcional) Para ver la plantilla de experimento JSON

Elija la pestaña Exportar. A continuación, verá un ejemplo del JSON creado por el procedimiento de consola anterior.

```
{
    "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"
    }
}
```

Paso 2: Iniciar el experimento

Cuando haya terminado de crear la plantilla de experimento, podrá utilizarla para iniciar un experimento.

Para iniciar un experimento

1. Debería estar en la página de detalles de la plantilla de experimento que acaba de crear. De lo contrario, elija Plantillas de experimento y, a continuación, seleccione el ID de la plantilla de experimento para abrir la página de detalles.
2. Elija Start experiment (Iniciar experimento).
3. (Opcional) Para agregar una etiqueta a su experimento, elija Agregar nueva etiqueta e ingrese una clave y un valor de etiqueta.
4. Elija Start experiment (Iniciar experimento). Cuando se le pida que confirme, ingrese **`start`** y elija Iniciar experimento.

Paso 3: Hacer un seguimiento del progreso del experimento

Puede hacer un seguimiento del progreso de un experimento en ejecución hasta que se complete, se detenga o falle.

Para hacer un seguimiento del progreso de un experimento

1. Debería estar en la página de detalles del experimento que acaba de iniciar. De lo contrario, elija Experimentos y, a continuación, seleccione el ID del experimento para abrir la página de detalles.
2. Para ver el estado del experimento, seleccione Estado en el panel Detalles. Para obtener más información, consulte Estados de experimento.
3. Vaya al siguiente paso cuando el estado del experimento sea En ejecución.

Paso 4: Verificar el resultado del experimento

Cuando la acción de este experimento se complete, ocurre lo siguiente:

- La instancia de spot de destino recibe una recomendación de reequilibrio de instancias.
- Se emite un aviso de interrupción de instancia de spot dos minutos antes de que Amazon EC2 termine o detenga su instancia.
- Cuando pasan dos minutos, la instancia de spot se termina o detiene.
- Una instancia de spot que detuvo AWS FIS permanece detenida hasta que la reinicie.

Para verificar que el experimento interrumpió la instancia

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, abra Spot Requests (Solicitudes de spot) e Instances (Instancia[s]) en pestañas o ventanas separadas del navegador.
3. En Spot Requests (Solicitudes de spot), seleccione la solicitud de instancia de spot. El estado inicial es.`fulfilled` Una vez completado el experimento, el estado cambia de la siguiente manera:
   - `terminate`: el estado cambia a `instance-terminated-by-experiment`.
   - `stop`: el estado cambia a `marked-for-stop-by-experiment` y, a continuación, a `instance-stopped-by-experiment`.
4. En Instances (Instancia[s]), seleccione la instancia de spot. El estado inicial es.`Running` Dos minutos después de recibir el aviso de interrupción de la instancia de spot, el estado cambia de la siguiente forma:
   - `stop`: el estado cambia a `Stopping` y, a continuación, a `Stopped`.

- `terminate`: el estado cambia a `Shutting-down` y, a continuación, a `Terminated`.

Paso 5: Eliminar

Si creó la instancia de spot de prueba para este experimento con un comportamiento de interrupción de `stop` y ya no la necesita, puede cancelar la solicitud de instancia de spot y terminarla.

Para cancelar la solicitud y terminar la instancia con la AWS CLI

1. Utilice el cancel-spot-instance-requestscomando para cancelar la solicitud de instancia puntual.

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

2. Utilice el comando terminate-instances para terminar la instancia.

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

Si ya no necesita la plantilla de experimento, puede eliminarla.

Para eliminar una plantilla de experimento con la consola de AWS FIS

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Eliminar plantilla de experimento.
4. Cuando se le solicite confirmación, ingrese **`delete`** y luego, elija Eliminar plantilla de experimento.

Tutorial: Simulación de un evento de conectividad

Puede utilizar AWS Fault Injection Service (AWS FIS) para simular una variedad de eventos de conectividad. AWS FIS simula eventos de conectividad bloqueando las conexiones de red de una de las siguientes maneras:

- `all`: niega todo el tráfico entrante y saliente de la subred. Tenga en cuenta que esta opción permite el tráfico intrasubred, incluido el tráfico hacia las interfaces de red de la subred y desde ellas.

- `availability-zone`: niega el tráfico intraVPC hacia subredes y desde ellas en otras zonas de disponibilidad.
- `dynamodb`: niega el tráfico hacia el punto de conexión regional y desde él para DynamoDB en la región actual.
- `prefix-list`: niega el tráfico hacia la lista de prefijos especificada y desde ella.
- `s3`: niega el tráfico hacia el punto de conexión regional y desde él para Amazon S3 en la región actual.
- `vpc`: niega el tráfico entrante y saliente de la VPC.

Utilice este tutorial para crear una plantilla de experimento que utilice la acción `aws:network:disrupt-connectivity` de AWS FIS para introducir la pérdida de conectividad con Amazon S3 en una subred de destino.

Temas

Requisitos previos

Antes de comenzar este tutorial, necesita un rol con los permisos adecuados en su Cuenta de AWS y una instancia de Amazon EC2 de prueba:

Un rol con permisos en su Cuenta de AWS

Cree un rol y asocie una política que permita a AWS FIS realizar la acción `aws:network:disrupt-connectivity` en su nombre.

Su rol de IAM requiere la siguiente política:

- AWSFaultInjectionSimulatorNetworkAccess— Otorga el permiso de servicio AWS FIS en la red Amazon EC2 y otros servicios necesarios para AWS realizar acciones de FIS relacionadas con la infraestructura de red.

> **Note**
>
> Para simplificar el proceso, este tutorial utiliza una política administrada por AWS. Para uso en producción, le recomendamos que otorgue solo los permisos mínimos necesarios para su caso de uso.
> Para obtener más información sobre cómo crear un rol de IAM, consulte Roles de IAM para experimentos de AWS FIS (AWS CLI) o Creación de un rol de IAM (consola) en la Guía del usuario de IAM.

Una instancia de Amazon EC2 de prueba

Lance una instancia de Amazon EC2 de prueba y conéctese a ella. Puede usar el siguiente tutorial para lanzar una instancia de Amazon EC2 y conectarse a ella: Tutorial: Introducción a las instancias Linux de Amazon EC2 en la Guía del usuario de Amazon EC2 para instancias Linux.

Paso 1: Crear una plantilla de experimento de AWS FIS

Cree la plantilla de experimento con la AWS Management Console de AWS FIS. Una plantilla de AWS FIS se compone de acciones, destinos, condiciones de detención y un rol de experimento. Para obtener más información acerca del funcionamiento de las plantillas, consulte Plantillas de experimento para AWS FIS.

Antes de empezar, asegúrese de que tiene preparado lo siguiente:

- Un rol de IAM con los permisos correctos.
- Una instancia de Amazon EC2.
- El ID de subred de la instancia de Amazon EC2.

Para crear una plantilla de experimento

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación izquierdo, elija Plantillas de experimento.

3. Elija Crear plantilla de experimento.
4. Escriba una descripción para la plantilla, como `Amazon S3 Network Disrupt Connectivity`.
5. En Acciones, elija Agregar acción.
   a. En Nombre, escriba `disruptConnectivity`.
   b. En Tipo de acción, seleccione aws:network:disrupt-connectivity.
   c. En Parámetros de acción, defina Duración en `2 minutes`.
   d. En Ámbito, seleccione s3.
   e. En la parte superior, seleccione Guardar.
6. En Destinos, debería ver el destino que se ha creado automáticamente. Elija Editar.
   a. Compruebe que Tipo de recurso sea `aws:ec2:subnet`.
   b. En Método de destino, seleccione ID de recursos y, a continuación, elija la subred que utilizó al crear la instancia de Amazon EC2 en los pasos de Requisitos previos.
   c. Compruebe que Modo de selección sea Todos.
   d. Seleccione Guardar.
7. En Acceso al servicio, seleccione el rol de IAM que creó, tal y como se describe en Requisitos previos de este tutorial. Si su rol no aparece, compruebe que tiene la relación de confianza requerida. Para obtener más información, consulte the section called “Rol de experimento”.
8. (Opcional) En condiciones de parada, puede seleccionar una CloudWatch alarma para detener el experimento en caso de que se produzca esa condición. Para obtener más información, consulte Condiciones de detención para AWS FIS.
9. (Opcional) En Logs, puedes seleccionar un bucket de Amazon S3 o enviar los registros CloudWatch para tu experimento.
10. Elija Crear plantilla de experimento y cuando se le solicite confirmación, ingrese `create`. A continuación, elija Crear plantilla de experimento.

Paso 2: Ejecutar ping en un punto de conexión de Amazon S3

Compruebe que su instancia de Amazon EC2 puede llegar a un punto de conexión de Amazon S3.

1. Conecte con la instancia de Amazon EC2 que ha creado en los pasos de Requisitos previos.

Para solucionar problemas, consulte Solución de problemas con la conexión a la instancia en la Guía del usuario de Amazon EC2 para instancias de Linux.

2. Compruebe la Región de AWS donde se encuentra la instancia. Puede hacerlo en la consola de Amazon EC2 o ejecutando el siguiente comando.

```
hostname
```

Por ejemplo, si lanzó una instancia de Amazon EC2 en `us-west-2`, verá el siguiente resultado.

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

3. Ejecute ping en un punto de conexión de Amazon S3 en su Región de AWS. Reemplace la *Región de AWS* por su región.

```
ping -c 1 s3.Región de AWS.amazonaws.com
```

En el resultado, debería ver un ping correcto con una pérdida de paquetes del 0 %, tal como se muestra en el siguiente ejemplo.

```
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
```

Paso 3: Iniciar su experimento de AWS FIS

Comience un experimento con la plantilla de experimento que acaba de crear.

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación izquierdo, elija Plantillas de experimento.
3. Seleccione el ID de la plantilla de experimento que ha creado para abrir la página de detalles.
4. Elija Start experiment (Iniciar experimento).
5. (Opcional) En la página de confirmación, agregue etiquetas para el experimento.

6. En la página de confirmación, seleccione Iniciar experimento.

Paso 4: Hacer un seguimiento del progreso del experimento de AWS FIS

Puede hacer un seguimiento del progreso de un experimento en ejecución hasta que se complete, se detenga o falle.

1. Debería estar en la página de detalles del experimento que acaba de iniciar. Si no lo está, elija Experimentos y, a continuación, seleccione el ID del experimento para abrir su página de detalles.
2. Para ver el estado del experimento, seleccione Estado en el panel de detalles. Para obtener más información, consulte Estados de experimento.
3. Vaya al siguiente paso cuando el estado del experimento sea En ejecución.

Paso 5: Verificar la interrupción de la red de Amazon S3

Puede validar el progreso del experimento ejecutando ping en el punto de conexión de Amazon S3.

- Desde su instancia de Amazon EC2, ejecute ping en el punto de conexión de Amazon S3 de su Región de AWS. Reemplace la *Región de AWS* por su región.

  ```
  ping -c 1 s3.Región de AWS.amazonaws.com
  ```

  En el resultado, debería ver un ping fallido con una pérdida de paquetes del 100 %, tal como se muestra en el siguiente ejemplo.

  ```
  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
  ```

Paso 5: Eliminar

Si ya no necesita la instancia de Amazon EC2 que ha creado para este experimento o la plantilla de AWS FIS, puede eliminarlos.

Para eliminar la instancia de Amazon EC2

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación, seleccione Instancias.
3. Seleccione la instancia de prueba, elija Estado de la instancia y luego, Terminar instancia.
4. Cuando se le indique que confirme, elija Terminate (Rescindir).

Para eliminar la plantilla de experimento con la consola de AWS FIS

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y, a continuación, elija Acciones, Eliminar plantilla de experimento.
4. Cuando se le solicite confirmación, ingrese `delete` y luego, elija Eliminar plantilla de experimento.

Tutorial: Programación de un experimento recurrente

Con AWS Fault Injection Service (AWS FIS), puede realizar experimentos de inyección de errores en sus cargas de trabajo de AWS. Estos experimentos se ejecutan en plantillas que contienen una o más acciones para ejecutarse en destinos específicos. Si también usa Amazon EventBridge, puede programar sus experimentos como tareas únicas o recurrentes.

Utilice este tutorial para crear un EventBridge cronograma que ejecute una plantilla de experimento AWS FIS cada 5 minutos.

Tareas

- Requisitos previos
- Paso 1: Crear una política y un rol de IAM
- Paso 2: Crear un Programador de Amazon EventBridge
- Paso 3: Comprobar el experimento
- Paso 4: Limpiar

Requisitos previos

Antes de comenzar este tutorial, debe tener una plantilla de experimento de AWS FIS que desee ejecutar según una programación. Si ya tiene una plantilla de experimento funcional, anote el ID de la plantilla y la Región de AWS. Si no, puede crear una plantilla siguiendo las instrucciones de the section called “Prueba de detención e inicio de instancias” y, a continuación, volver a este tutorial.

Paso 1: Crear una política y un rol de IAM

Para crear una política y un rol de IAM

1. Abra la consola de IAM en https://console.aws.amazon.com/iam/.
2. En el panel de navegación, seleccione Roles y, a continuación, seleccione Crear rol.
3. Elija Política de confianza personalizada y, a continuación, inserte el siguiente fragmento para permitir que Programador de Amazon EventBridge asuma el rol en su nombre.

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

   Elija Siguiente.
4. En Agregar permisos, elija Crear política.
5. Elija JSON y, a continuación, inserte la siguiente política. Sustituya el *your-experiment-template-id*valor por el ID de plantilla de su experimento de los pasos previos.

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

```
            "Effect": "Allow",
            "Action": "fis:StartExperiment",
            "Resource": [
                "arn:aws:fis:*:*:experiment-template/your-experiment-template-id",
                "arn:aws:fis:*:*:experiment/*"
            ]
        }
    ]
}
```

Puede restringir el programador para que solo ejecute experimentos de AWS FIS que tengan un valor de etiqueta específico. Por ejemplo, la siguiente política concede el permiso `StartExperiment` para todas las plantillas de experimento de AWS FIS, pero restringe al programador a ejecutar únicamente los experimentos que estén etiquetados como `Purpose=Schedule`.

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

Elija Siguiente: Etiquetas.

6. Elija Siguiente: Revisar.
7. En Revisar política, asigne un nombre a la política `FIS_RecurringExperiment` y, a continuación, elija Crear política.

8. En Agregar permisos, agregue la nueva política `FIS_RecurringExperiment` a su rol y, a continuación, seleccione Siguiente.
9. En la página Asignar nombre, revisar y crear, asigne al rol el nombre `FIS_RecurringExperiment_role` y, a continuación, elija Crear rol.

Paso 2: Crear un Programador de Amazon EventBridge

Para crear un Programador de Amazon EventBridge

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Programaciones.
3. Compruebe que utiliza la misma Región de AWS que su plantilla de experimento de AWS FIS.
4. Elija Crear programación y rellene lo siguiente:
   - En Nombre de la programación, inserte `FIS_recurring_experiment_tutorial`.
   - En Patrón de programación, seleccione Programación periódica.
   - En Tipo de programación, seleccione Programación basada en frecuencia.
   - En Expresión de frecuencia, seleccione 5 minutos.
   - En Intervalo de tiempo flexible, seleccione Desactivado.
   - (Opcional) En Periodo, seleccione su zona horaria.
   - Elija Siguiente.
5. En Seleccionar destino, elija Todas las API y, a continuación, busque AWS FIS.
6. Elija AWSFIS y, a continuación, seleccione StartExperiment.
7. En Entrada, inserte la siguiente carga útil de JSON. Sustituya el *`your-experiment-template-id`*valor por el ID de plantilla de su experimento. `ClientToken` es un identificador único para el programador. En este tutorial, usaremos una palabra clave contextual permitida por el Programador de Amazon EventBridge. Para obtener más información, consulte Añadir atributos de contexto en la Guía del EventBridge usuario de Amazon.

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

Elija Siguiente.

8. (Opcional) En Configuración, puede establecer la política de reintentos, la cola de mensajes fallidos (DLQ) y los ajustes de Cifrado. También puede mantener los valores predeterminados.
9. En Permisos, seleccione Usar rol existente y, a continuación, busque `FIS_RecurringExperiment_role`.
10. Elija Siguiente.
11. En Revisar y crear una programación, revise los detalles de su programador y, a continuación, seleccione Crear programación.

Paso 3: Comprobar el experimento

Para comprobar que el experimento de AWS FIS se llevó a cabo según la programación

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación de la izquierda, elija Experimentos.
3. Cinco minutos después de crear la programación, debería ver el experimento ejecutándose.

Paso 4: Limpiar

Para deshabilitar su Programador de Amazon EventBridge

1. Abre la EventBridge consola de Amazon en https://console.aws.amazon.com/events/.
2. En el panel de navegación izquierdo, elija Programaciones.
3. Seleccione el planificador recién creado y, a continuación, seleccione Deshabilitar.

Acciones para AWS FIS

Una acción es la actividad de inyección de errores que se ejecuta en un objetivo mediante AWS Fault Injection Service (AWS FIS). AWS FIS proporciona acciones preconfiguradas para tipos específicos de objetivos en todos los AWS servicios. Agregue acciones a una plantilla de experimento, que luego se utilizan para ejecutar experimentos.

Contenido

Identificadores de acciones

Cada AWS FIS acción tiene un identificador con el siguiente formato:

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

Por ejemplo, la siguiente acción detiene las instancias de destino de Amazon EC2:

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

Para ver la lista completa de acciones, consulte AWS FIS referencia de acciones. Para obtener la lista mediante AWS CLI, consulteEnumeración de las acciones.

Parámetros de acciones

Algunas AWS FIS acciones tienen parámetros adicionales que son específicos de la acción. Estos parámetros se utilizan para pasar información al AWS FIS momento de ejecutar la acción.

AWS FIS admite tipos de error personalizados mediante la `aws:ssm:send-command` acción, que utiliza el agente SSM y un documento de comandos SSM para crear la condición de error en las instancias de destino. La acción `aws:ssm:send-command` incluye un parámetro `documentArn` que toma el nombre de recurso de Amazon (ARN) de un documento de SSM como valor. Los valores de los parámetros se especifican al agregar la acción a la plantilla de experimento.

Para obtener más información acerca de cómo especificar parámetros para la acción `aws:ssm:send-command`, consulte Uso de la acción aws:ssm:send-command.

Siempre que sea posible, puede introducir una configuración de reversión (también denominada acción posterior) en los parámetros de la acción. Una acción posterior devuelve el destino al estado en el que se encontraba antes de que se ejecutara la acción. La acción posterior se ejecuta después del tiempo especificado en la duración de la acción. No todas las acciones admiten acciones posteriores. Por ejemplo, si la acción termina una instancia de Amazon EC2, no podrá recuperarla una vez terminada.

Destinos de acciones

Una acción se ejecuta en los recursos de destino que especifique. Tras definir un destino, puede especificar su nombre al definir una acción.

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

AWS FIS las acciones admiten los siguientes tipos de recursos para los objetivos de acción:

- Grupos de escalado automático: grupos de Amazon EC2 Auto Scaling
- Buckets: buckets de Amazon S3
- Clúster: clústeres de Amazon EKS
- Clústeres: clústeres de Amazon ECS o clústeres de base de datos de Amazon Aurora
- Instancias de base de datos: instancias de base de datos de Amazon RDS
- Tablas globales cifradas: tablas globales de Amazon DynamoDB cifradas con una clave administrada por el cliente
- Instancias: instancias de Amazon EC2
- Grupos de nodos: grupos de nodos de Amazon EKS

- Pods: pods de Kubernetes en Amazon EKS
- ReplicationGroups— Grupos ElastiCache de replicación de Redis
- Roles: roles de IAM
- SpotInstances— Instancias puntuales de Amazon EC2
- Subredes: subredes de VPC
- Tareas: tareas de Amazon ECS
- TransitGateways— Pasarelas de tránsito
- Volúmenes: volúmenes de Amazon EBS

Para ver ejemplos, consulte the section called “Acciones de ejemplo”.

AWS FIS referencia de acciones

En esta referencia se describen las acciones habituales AWS FIS, incluida la información sobre los parámetros de la acción y los permisos de IAM necesarios. También puede enumerar AWS FIS las acciones compatibles mediante la AWS FIS consola o el comando list-actions de (). AWS Command Line Interface AWS CLI

Para más información, consulte Acciones para AWS FIS y Cómo funciona el servicio de inyección de AWS fallos con IAM.

Acciones

- Acciones de inyección de errores
- Acción de espera
- CloudWatch Acciones de Amazon
- Acciones de Amazon DynamoDB
- Acciones de Amazon EBS
- Acciones de Amazon EC2
- Acciones de Amazon ECS
- Acciones de Amazon EKS
- ElastiCache Acciones de Amazon
- Acciones de red

- Acciones de Amazon RDS
- Acciones de Amazon S3
- Acciones de Systems Manager

Acciones de inyección de errores

AWS FIS admite las siguientes acciones de inyección de errores.

Acciones

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

aws:fis:inject-api-internal-error

Inyecta errores internos en las solicitudes realizadas por el rol de IAM de destino.

Tipo de recurso

- aws:iam:role

Parámetros

- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- service— El espacio de nombres de la AWS API de destino. El valor admitido es `ec2`.
- percentage: el porcentaje (del 1 al 100) de llamadas en las que se debe inyectar el error.
- operations: las operaciones en las que se inyecta el error, separadas por comas. Para obtener una lista de las acciones de la API para el espacio de nombres de `ec2`, consulte Acciones en Referencia de API de Amazon EC2.

Permisos

- `fis:InjectApiInternalError`

aws:fis:inject-api-throttle-error

Inyecta errores de limitación en las solicitudes realizadas por el rol de IAM de destino.

Tipo de recurso

- aws:iam:role

Parámetros

- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- service— El espacio de nombres de la AWS API de destino. El valor admitido es `ec2`.
- percentage: el porcentaje (del 1 al 100) de llamadas en las que se debe inyectar el error.
- operations: las operaciones en las que se inyecta el error, separadas por comas. Para obtener una lista de las acciones de la API para el espacio de nombres de `ec2`, consulte Acciones en Referencia de API de Amazon EC2.

Permisos

- `fis:InjectApiThrottleError`

aws:fis:inject-api-unavailable-error

Inyecta errores de no disponibilidad en las solicitudes realizadas por el rol de IAM de destino.

Tipo de recurso

- aws:iam:role

Parámetros

- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- service— El espacio de nombres de la AWS API de destino. El valor admitido es `ec2`.

- percentage: el porcentaje (del 1 al 100) de llamadas en las que se debe inyectar el error.
- operations: las operaciones en las que se inyecta el error, separadas por comas. Para obtener una lista de las acciones de la API para el espacio de nombres de `ec2`, consulte Acciones en Referencia de API de Amazon EC2.

Permisos

- `fis:InjectApiUnavailableError`

Acción de espera

AWS FIS admite la siguiente acción de espera.

aws:fis:wait

Ejecuta la acción de AWS FIS espera.

Parámetros

- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

- Ninguna

CloudWatch Acciones de Amazon

AWS FIS admite la siguiente CloudWatch acción de Amazon.

aws:cloudwatch:assert-alarm-state

Verifica que las alarmas especificadas estén en uno de los estados de alarma especificados.

Tipo de recurso

- Ninguna

Parámetros

- alarmArns: los ARN de las alarmas, separados por comas. Puede especificar hasta cinco alarmas.
- alarmStates: los estados de alarma, separados por comas. Los posibles estados de alarma son `OK`, `ALARM` e `INSUFFICIENT_DATA`.

Permisos

- `cloudwatch:DescribeAlarms`

Acciones de Amazon DynamoDB

AWS FIS admite la siguiente acción de Amazon DynamoDB.

aws:dynamodb:encrypted-global-table-pause-replication

Hace una pausa en la replicación global de tablas de Amazon DynamoDB para las tablas AWS Key Management Service cifradas con claves administradas por el cliente (CMK). Esta acción elimina los permisos para que el rol vinculado al servicio de replicación de DynamoDB acceda a AWS KMS la clave utilizada para proteger los datos de la tabla global de DynamoDB de destino. Las tablas pueden seguir replicándose durante un máximo de 5 minutos desde que comienza la acción.

La siguiente declaración se añadirá dinámicamente a la política de la AWS KMS clave utilizada para proteger los datos en las tablas globales de DynamoDB de destino:

```
{
    "Sid": "DO_NOT_MODIFY_FIS_DDB_PAUSE_REPLICATION-EXP123456789012345",
    "Effect": "Deny",
    "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
```

```
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:dynamodb:tableName": [
                "transactions-global-table",
                "inventory-global-table"
            ]
        }
    }
}
```

La instrucción de la política anterior elimina los permisos para que el rol vinculado al servicio de DynamoDB replique datos con origen y destino en las tablas que se muestran en la clave de contexto `kms:EncryptionContext:aws:dynamodb:tableName`. En el ejemplo anterior, la replicación se detendría para las tablas globales de DynamoDB con los nombres `transaction-global-table`, `inventory-global-table`.

Tipo de recurso

- aws:dynamodb:encrypted-global-table

Parámetros

- duration— En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

- `kms:DescribeKey`
- `kms:GetKeyPolicy`
- `kms:PutKeyPolicy`
- `dynamodb:DescribeTable`
- `dynamodb:DescribeGlobalTable`
- `tag:GetResources`

Para ver una política de ejemplo, consulte Ejemplo: experimentar un rol con permisos para ejecutar `aws:dynamodb:encrypted-global-table-pause-replication`.

> **Note**
>
> AWS FIS utiliza `kms:PutKeyPolicy` para denegar el acceso a DynamoDB; a la clave AWS KMS administrada por el cliente, lo que detiene la replicación. Se recomienda utilizar el rol solo cuando se esté realizando activamente un experimento con esta acción; de lo contrario, se recomienda eliminarlo. Al eliminar el rol, se eliminan los permisos de FIS para `kms:PutKeyPolicy`. Una vez finalizado el experimento, busque el rol en los detalles de la plantilla de experimento. Elija el enlace al rol de IAM en la consola de IAM y elija eliminar. Tras eliminar el rol, vaya a la AWS KMS consola y busque la AWS KMS clave utilizada para proteger los datos en la tabla de DynamoDB de destino. Compruebe que la política AWS KMS clave coincide con sus expectativas. Ya no deberías ver una AWS FIS declaración (por ejemplo,`FIS_DDB_PAUSE_REPLICATION-EXP123456789012345_DO_NOT_MODIFY`).

Acciones de Amazon EBS

AWS FIS admite la siguiente acción de Amazon EBS.

aws:ebs:pause-volume-io

Detiene las operaciones de E/S en los volúmenes de EBS de destino. Los volúmenes de destino deben estar en la misma zona de disponibilidad y deben estar asociados a instancias creadas en Nitro System. Los volúmenes no se pueden asociar a instancias de un Outpost.

Para iniciar el experimento con la consola de Amazon EC2, consulte Pruebas de fallos en Amazon EBS en la Guía del usuario de Amazon EC2.

Tipo de recurso

- aws:ec2:ebs-volume

Parámetros

- duration: la duración, de un segundo a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto, PT5S representa cinco segundos y PT6H representa seis horas. En la AWS FIS consola, se introduce el número de segundos, minutos u horas. Si la duración es corta, como en el caso de PT5S, la E/S se pausa durante el tiempo especificado, pero es posible que el experimento tarde más en completarse debido al tiempo que tarda en inicializarse el experimento.

Permisos

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

Acciones de Amazon EC2

AWS FIS admite las siguientes acciones de Amazon EC2.

Acciones

- 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 también admite acciones de inyección de errores a través del agente AWS Systems Manager SSM. Systems Manager usa un documento de SSM que define las acciones que se deben realizar en las instancias EC2. Puede utilizar su propio documento para inyectar errores personalizados o puede utilizar documentos de SSM preconfigurados. Para obtener más información, consulte the section called “Uso de documentos de SSM”.

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

Inyecta respuestas de error `InsufficientInstanceCapacity` en las solicitudes realizadas por los roles de IAM de destino. Las operaciones compatibles son RunInstances, CreateCapacityReservation StartInstances, CreateFleet llamadas. No se admiten solicitudes que incluyan peticiones de capacidad en varias zonas de disponibilidad. Esta acción no permite definir destinos mediante etiquetas, filtros o parámetros de recursos.

Tipo de recurso

- aws:iam:role

Parámetros

- duration— En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- availabilityzoneIdentifiers: lista separada por comas de zonas de disponibilidad. Admite ID de zona (por ejemplo, `"use1-az1, use1-az2"`) y los nombres de zona (por ejemplo, `"us-east-1a"`).
- percentage: el porcentaje (del 1 al 100) de llamadas en las que se debe inyectar el error.

Permisos

- `ec2:InjectApiError` con el valor `ec2:FisActionId` de la clave de condición establecido en `aws:ec2:api-insufficient-instance-capacity-error` y la clave de condición `ec2:FisTargetArns` establecida en roles de IAM de destino.

Para ver una política de ejemplo, consulte Ejemplo: utilice claves de condición para `ec2:InjectApiError`.

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

Inyecta respuestas de error `InsufficientInstanceCapacity` en las solicitudes realizadas por los grupos de escalado automático de destino. Esta acción solo admite grupos de escalado automático que utilizan plantillas de lanzamiento. Para obtener más información sobre los errores de capacidad insuficiente de instancias, consulte la Guía del usuario de Amazon EC2.

Tipo de recurso

- aws:ec2:autoscaling-group

Parámetros

- duration— En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- availabilityzoneIdentifiers: lista separada por comas de zonas de disponibilidad. Admite ID de zona (por ejemplo, `"use1-az1, use1-az2"`) y los nombres de zona (por ejemplo, `"us-east-1a"`).

- percentage: opcional. El porcentaje (del 1 al 100) de las solicitudes de lanzamiento del grupo de escalado automático de destino para inyectar el error. El valor predeterminado es 100.

Permisos

- `ec2:InjectApiError`con la clave de condición ec2: FisActionId valor establecido en `aws:ec2:asg-insufficient-instance-capacity-error` y clave de `ec2:FisTargetArns` condición establecida en los grupos de Auto Scaling de destino.
- `autoscaling:DescribeAutoScalingGroups`

Para ver una política de ejemplo, consulte Ejemplo: utilice claves de condición para `ec2:InjectApiError`.

aws:ec2:reboot-instances

Ejecuta la acción de la API Amazon EC2 RebootInstancesen las instancias EC2 de destino.

Tipo de recurso

- aws:ec2:instance

Parámetros

- Ninguna

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:send-spot-instance-interruptions

Interrumpe las instancias de spot de destino. Envía un aviso de interrupción de instancia de spot a las instancias de spot de destino dos minutos antes de interrumpirlas. El tiempo de interrupción viene determinado por el durationBeforeInterruptionparámetro especificado. Dos minutos después del tiempo de interrupción, las instancias de spot terminan o se detienen, en función de su comportamiento de interrupción. Una instancia de spot que detuvo AWS FIS permanece detenida hasta que la reinicie.

Inmediatamente después de iniciar la acción, la instancia de destino recibe una recomendación de reequilibrio de la instancia EC2. Si lo especificó durationBeforeInterruption, podría haber un retraso entre la recomendación de reequilibrio y el aviso de interrupción.

Para obtener más información, consulte the section called “Prueba de interrupciones de instancias de spot”. Como alternativa, para iniciar el experimento con la consola de Amazon EC2, consulte Inicio de una interrupción de instancias de spot en la Guía del usuario de Amazon EC2.

Tipo de recurso

- aws:ec2:spot-instance

Parámetros

- durationBeforeInterruption: el tiempo de espera antes de interrumpir la instancia, de 2 a 15 minutos. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT2M representa dos minutos. En la AWS FIS consola, se introduce el número de minutos.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:stop-instances

Ejecuta la acción de la API Amazon EC2 StopInstancesen las instancias EC2 de destino.

Tipo de recurso

- aws:ec2:instance

Parámetros

- startInstancesAfterDuration: opcional. El tiempo de espera antes de iniciar la instancia, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la consola de AWS FIS , se introduce el número de segundos, minutos u horas. Si la instancia tiene un volumen de EBS cifrado, debes conceder AWS FIS permiso a la clave de KMS utilizada para cifrar el volumen o añadir la función de experimento a la política de claves de KMS.
- completeIfInstancesTerminated: opcional. Si es verdadero y también `startInstancesAfterDuration` lo es, esta acción no fallará cuando las instancias de EC2 de destino se hayan terminado mediante una solicitud independiente ajena a FIS y no se puedan reiniciar. Por ejemplo, los grupos de escalado automático pueden terminar las instancias de EC2 detenidas bajo su control antes de que se complete esta acción. El valor predeterminado es false.

Permisos

- `ec2:StopInstances`
- `ec2:StartInstances`
- `ec2:DescribeInstances`: opcional. Se requiere con completeIfInstancesTerminated para validar el estado de la instancia al final de la acción.
- `kms:CreateGrant`: opcional. Se requiere junto con startInstancesAfterla duración para reiniciar las instancias con volúmenes cifrados.

AWS política gestionada

- AWSFaultInjectionSimulatorEC2Access

aws:ec2:terminate-instances

Ejecuta la acción de la API Amazon EC2 TerminateInstancesen las instancias EC2 de destino.

Tipo de recurso

- aws:ec2:instance

Parámetros

- Ninguna

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEC2Access

Acciones de Amazon ECS

AWS FIS admite las siguientes acciones de Amazon ECS.

Acciones

- 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

Ejecuta la acción de la API Amazon ECS UpdateContainerInstancesStatepara drenar el porcentaje especificado de instancias de Amazon EC2 subyacentes en los clústeres de destino.

Tipo de recurso

- aws:ecs:cluster

Parámetros

- drainagePercentage: el porcentaje (de 1 a 100).
- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:stop-task

Ejecuta la acción de la API Amazon ECS StopTaskpara detener la tarea de destino.

Tipo de recurso

- aws:ecs:task

Parámetros

- Ninguna

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorECSAccess

aws:ecs:task-cpu-stress

Ejecuta esfuerzo de la CPU en las tareas de destino. Utiliza el documento SSM AWSFIS-Run-CPU-Stress. Las tareas deben ser gestionadas por. AWS Systems Manager Para obtener más información, consulte Uso de las acciones de las tareas de ECS.

Tipo de recurso

- aws:ecs:task

Parámetros

- duration: la duración de la prueba de esfuerzo, en formato ISO 8601.
- percent: opcional. El porcentaje de carga de destino, de 0 (sin carga) a 100 (carga completa). El valor predeterminado es 100.
- workers: opcional. El número de factores de esfuerzo que se van a utilizar. El valor predeterminado es 0, que utiliza todos los factores de esfuerzo.
- installDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en el contenedor asociado de SSM Agent, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es stress-ng.

Permisos

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

- `ssm:CancelCommand`

aws:ecs:task-io-stress

Ejecuta esfuerzo de E/S en las tareas de destino. Utiliza el documento AWSFISSSM -Run-IO-Stress. Las tareas deben ser gestionadas por. AWS Systems Manager Para obtener más información, consulte Uso de las acciones de las tareas de ECS.

Tipo de recurso

- aws:ecs:task

Parámetros

- duration: la duración de la prueba de esfuerzo, en formato ISO 8601.
- percent: opcional. El porcentaje de espacio libre en el sistema de archivos que se utilizará durante la prueba de esfuerzo. El valor predeterminado es 80 %.
- workers: opcional. Número de procesos de trabajo Los procesos de trabajo realizan operaciones de lectura/escritura secuenciales, aleatorias y asignadas a memoria, sincronizaciones forzadas y pérdida de memoria caché. Varios procesos secundarios realizan diferentes operaciones de E/S en el mismo archivo. El valor predeterminado de es 1.
- installDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en el contenedor asociado de SSM Agent, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es stress-ng.

Permisos

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

aws:ecs:task-kill-process

Detiene el proceso especificado en las tareas con el comando killall. Utiliza el documento AWSFISSSM -Run-Kill-Process. La definición de la tarea debe tener `pidMode` establecido en `task`.

Las tareas deben ser gestionadas por. AWS Systems Manager Para obtener más información, consulte Uso de las acciones de las tareas de ECS.

Tipo de recurso

- aws:ecs:task

Parámetros

- processName: el nombre del proceso que se va a detener.
- signal: opcional. La señal que se va a enviar junto con el comando. Los valores posibles son `SIGTERM` (que el receptor puede elegir ignorar) y `SIGKILL` (que no se pueden ignorar). El valor predeterminado es `SIGTERM`.
- installDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en el contenedor asociado de SSM Agent, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es killall.

Permisos

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

aws:ecs:task-network-blackhole-port

Elimina el tráfico entrante o saliente del protocolo y el puerto especificados. Utiliza el documento SSM AWSFIS-Run-Network-Blackhole-Port. La definición de la tarea debe tener `pidMode` establecido en `task`. Las tareas deben ser gestionadas por. AWS Systems Manager No se puede establecer `networkMode` en `bridge` en la definición de la tarea. Para obtener más información, consulte Uso de las acciones de las tareas de ECS.

Tipo de recurso

- aws:ecs:task

Parámetros

- duration: la duración de la prueba, en formato ISO 8601.
- port: el número de puerto.
- trafficType: el tipo de tráfico. Los valores posibles son `ingress` y `egress`.
- protocol: opcional. El protocolo. Los valores posibles son `tcp` y `udp`. El valor predeterminado es `tcp`.
- installDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en el contenedor asociado de SSM Agent, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd, dig e iptables.

Permisos

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

aws:ecs:task-network-latency

Agrega latencia y fluctuación a la interfaz de red con la herramienta tc para el tráfico hacia fuentes específicas o desde ellas. Utiliza el documento SSM AWSFIS-Run-Network-Latency-Sources. La definición de la tarea debe tener `pidMode` establecido en `task`. Las tareas deben ser gestionadas por. AWS Systems Manager No se puede establecer `networkMode` en `bridge` en la definición de la tarea. Para obtener más información, consulte Uso de las acciones de las tareas de ECS.

Tipo de recurso

- aws:ecs:task

Parámetros

- duration: la duración de la prueba, en formato ISO 8601.
- interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- delayMilliseconds: opcional. El retraso, en milisegundos. El valor predeterminado es 200.
- jitterMilliseconds: opcional. La fluctuación, en milisegundos. El valor predeterminado es 10.

- sources: opcional. Las fuentes, separadas por comas. Los valores posibles son: una dirección IPv4, un bloque CIDR IPv4, un nombre de dominio, `DYNAMODB` y `S3`. Si especifica `DYNAMODB` o `S3`, solo se aplicará al punto de conexión regional de la región actual. El valor predeterminado es 0.0.0.0/0, que coincide con todo el tráfico IPv4.
- installDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en el contenedor asociado de SSM Agent, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd, dig, jq y tc.

Permisos

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

aws:ecs:task-network-packet-loss

Agrega la pérdida de paquetes a la interfaz de red con la herramienta tc. Utiliza el documento SSM AWSFIS-Run-Network-Packet-Loss-Sources. La definición de la tarea debe tener `pidMode` establecido en `task`. Las tareas deben ser gestionadas por. AWS Systems Manager No se puede establecer `networkMode` en `bridge` en la definición de la tarea. Para obtener más información, consulte Uso de las acciones de las tareas de ECS.

Tipo de recurso

- aws:ecs:task

Parámetros

- duration: la duración de la prueba, en formato ISO 8601.
- interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- lossPercent: opcional. El porcentaje de pérdida de paquetes. El valor predeterminado es 7 %.
- sources: opcional. Las fuentes, separadas por comas. Los valores posibles son: una dirección IPv4, un bloque CIDR IPv4, un nombre de dominio, `DYNAMODB` y `S3`. Si especifica `DYNAMODB` o `S3`, solo se aplicará al punto de conexión regional de la región actual. El valor predeterminado es 0.0.0.0/0, que coincide con todo el tráfico IPv4.

- installDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en el contenedor asociado de SSM Agent, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd, dig, jq y tc.

Permisos

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

Acciones de Amazon EKS

AWS FIS admite las siguientes acciones de Amazon EKS.

Acciones

- 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

Ejecuta un experimento ChaosMesh o un experimento de Litmus en un único clúster objetivo. Debes instalar ChaosMesh Litmus en el clúster de destino.

Al crear una plantilla de experimento y definir un tipo de destino `aws:eks:cluster`, debe dirigir esta acción a un único nombre de recurso de Amazon (ARN). Esta acción no permite definir destinos mediante etiquetas, filtros o parámetros de recursos.

Al realizar la instalación ChaosMesh, debes especificar el tiempo de ejecución del contenedor adecuado. A partir de la versión 1.23 de Amazon EKS, el tiempo de ejecución predeterminado cambió de Docker a containerd. A partir de la versión 1.24, se eliminó Docker.

Tipo de recurso

- aws:eks:cluster

Parámetros

- kubernetesApiVersion: la versión de la API del recurso personalizado de Kubernetes. Los valores posibles son `chaos-mesh.org/v1alpha1` | `litmuschaos.io/v1alpha1`.
- kubernetesKind: el tipo de recurso personalizado de Kubernetes. El valor depende de la versión de la API.
  - `chaos-mesh.org/v1alpha1`: los valores posibles son `AWSChaos` | `DNSChaos` | `GCPChaos` | `HTTPChaos` | `IOChaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodIOChaos` | `PodNetworkChaos` | `Schedule` | `StressChaos` | `TimeChaos` |
  - `litmuschaos.io/v1alpha1`: el valor posible es `ChaosEngine`.
- kubernetesNamespace: el espacio de nombres de Kubernetes.
- kubernetesSpec: la sección `spec` del recurso personalizado de Kubernetes, en formato JSON.
- maxDuration: el tiempo máximo permitido para que se complete la ejecución de la automatización, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

No se requieren permisos de AWS Identity and Access Management (IAM) para realizar esta acción. Kubernetes controla los permisos necesarios para usar esta acción mediante la autorización de RBAC. Para obtener más información, consulte Utilización de la autorización de RBAC en la documentación oficial de Kubernetes. Para obtener más información sobre Chaos Mesh, consulte la documentación oficial de Chaos Mesh. Para obtener más información sobre Litmus, consulta la documentación oficial de Litmus.

aws:eks:pod-cpu-stress

Ejecuta esfuerzo de la CPU en los pods de destino. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- duration: la duración de la prueba de esfuerzo, en formato ISO 8601.
- percent: opcional. El porcentaje de carga de destino, de 0 (sin carga) a 100 (carga completa). El valor predeterminado es 100.
- workers: opcional. El número de factores de esfuerzo que se van a utilizar. El valor predeterminado es 0, que utiliza todos los factores de esfuerzo.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. De forma predeterminada, se utilizan las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-delete

Elimina los pods de destino. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- gracePeriodSeconds: opcional. La duración, en segundos, para esperar a que el pod termine correctamente. Si el valor es 0, realizamos la acción de inmediato. Si el valor es nulo, utilizamos el período de gracia predeterminado para el pod.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. El valor predeterminado es utilizar las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-io-stress

Ejecuta esfuerzo de E/S en los pods de destino. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- duration: la duración de la prueba de esfuerzo, en formato ISO 8601.
- workers: opcional. Número de procesos de trabajo Los procesos de trabajo realizan operaciones de lectura/escritura secuenciales, aleatorias y asignadas a memoria, sincronizaciones forzadas y pérdida de memoria caché. Varios procesos secundarios realizan diferentes operaciones de E/S en el mismo archivo. El valor predeterminado de es 1.
- percent: opcional. El porcentaje de espacio libre en el sistema de archivos que se utilizará durante la prueba de esfuerzo. El valor predeterminado es 80 %.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. El valor predeterminado es utilizar las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-memory-stress

Ejecuta esfuerzo de la memoria en los pods de destino. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- duration: la duración de la prueba de esfuerzo, en formato ISO 8601.
- workers: opcional. El número de factores de esfuerzo que se van a utilizar. El valor predeterminado de es 1.
- percent: opcional. El porcentaje de memoria virtual que se utilizará durante la prueba de esfuerzo. El valor predeterminado es 80 %.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. El valor predeterminado es utilizar las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-blackhole-port

Elimina el tráfico entrante o saliente del protocolo y el puerto especificados. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- duration: la duración de la prueba, en formato ISO 8601.
- protocol: opcional. El protocolo. Los valores posibles son `tcp` y `udp`. El valor predeterminado es `tcp`.
- trafficType: el tipo de tráfico. Los valores posibles son `ingress` y `egress`.
- port: el número de puerto.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. El valor predeterminado es utilizar las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-latency

Agrega latencia y fluctuación a la interfaz de red con la herramienta tc para el tráfico hacia fuentes específicas o desde ellas. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- duration: la duración de la prueba, en formato ISO 8601.
- interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- delayMilliseconds: opcional. El retraso, en milisegundos. El valor predeterminado es 200.
- jitterMilliseconds: opcional. La fluctuación, en milisegundos. El valor predeterminado es 10.
- sources: opcional. Las fuentes, separadas por comas. Los valores posibles son: una dirección IPv4, un bloque CIDR IPv4, un nombre de dominio, `DYNAMODB` y `S3`. Si especifica `DYNAMODB` o `S3`, solo se aplicará al punto de conexión regional de la región actual. El valor predeterminado es 0.0.0.0/0, que coincide con todo el tráfico IPv4.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. El valor predeterminado es utilizar las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:pod-network-packet-loss

Agrega la pérdida de paquetes a la interfaz de red con la herramienta tc. Para obtener más información, consulte Uso de las acciones de pod de EKS.

Tipo de recurso

- aws:eks:pod

Parámetros

- duration: la duración de la prueba, en formato ISO 8601.
- interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- lossPercent: opcional. El porcentaje de pérdida de paquetes. El valor predeterminado es 7 %.
- sources: opcional. Las fuentes, separadas por comas. Los valores posibles son: una dirección IPv4, un bloque CIDR IPv4, un nombre de dominio, `DYNAMODB` y `S3`. Si especifica `DYNAMODB` o `S3`, solo se aplicará al punto de conexión regional de la región actual. El valor predeterminado es 0.0.0.0/0, que coincide con todo el tráfico IPv4.
- kubernetesServiceAccount: cuentas de servicio de Kubernetes. Para obtener más información acerca de los permisos necesarios, consulte the section called “Configurar la cuenta de servicio de Kubernetes”.
- fisPodContainerImage: opcional. La imagen del contenedor utilizada para crear el pod del inyector de errores. El valor predeterminado es utilizar las imágenes proporcionadas por AWS FIS. Para obtener más información, consulte the section called “Imágenes de contenedor de pods”.
- maxErrorsPercent: opcional. El porcentaje de destinos que pueden fallar antes de que falle la inyección de errores. El valor predeterminado es 0.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

aws:eks:terminate-nodegroup-instances

Ejecuta la acción de la API Amazon EC2 TerminateInstancesen el grupo de nodos de destino.

Tipo de recurso

- aws:eks:nodegroup

Parámetros

- instanceTerminationPercentage: el porcentaje (del 1 al 100) de instancias que se van a terminar.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEKSAccess

ElastiCache Acciones de Amazon

AWS FIS apoya la siguiente ElastiCache acción.

aws:elasticache:interrupt-cluster-az-power

Interrumpe el suministro eléctrico a los nodos de la zona de disponibilidad especificada para grupos de replicación de Redis de destino. Cuando el objetivo es un nodo principal, la réplica de lectura correspondiente con el menor retraso de replicación se promociona al elemento principal. Los reemplazos de réplicas de lectura en la zona de disponibilidad especificada se bloquean mientras

dura esta acción, lo que significa que los grupos de replicación de destino funcionan con capacidad reducida.

Tipo de recurso

- aws:elasticache:redis-replicationgroup

Parámetros

- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

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

Acciones de red

AWS FIS admite las siguientes acciones de red.

Acciones

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

aws:network:disrupt-connectivity

Niega el tráfico especificado a las subredes de destino.

Tipo de recurso

- aws:ec2:subnet

Parámetros

- scope: el tipo de tráfico que se va a negar. Los valores posibles son:
  - `all`: niega todo el tráfico entrante y saliente de la subred. Tenga en cuenta que esta opción permite el tráfico intrasubred, incluido el tráfico hacia las interfaces de red de la subred y desde ellas.
  - `availability-zone`: niega el tráfico intraVPC hacia subredes y desde ellas en otras zonas de disponibilidad.
  - `dynamodb`: niega el tráfico hacia el punto de conexión regional y desde él para DynamoDB en la región actual.
  - `prefix-list`: niega el tráfico hacia la lista de prefijos especificada y desde ella.
  - `s3`: niega el tráfico hacia el punto de conexión regional y desde él para Amazon S3 en la región actual.
  - `vpc`: niega el tráfico entrante y saliente de la VPC.
- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- prefixListIdentifier: si el alcance es `prefix-list`, es el identificador de la lista de prefijos gestionada por el cliente. Puede especificar un nombre, un ID o un ARN. La lista de prefijos puede tener un máximo de 10 entradas.

Permisos

- `ec2:CreateNetworkAcl`: crea la ACL de red con la etiqueta managedByFIS=true.
- `ec2:CreateNetworkAclEntry`: la ACL de red debe tener la etiqueta managedByFIS=true.
- `ec2:CreateTags`
- `ec2:DeleteNetworkAcl`: la ACL de red debe tener la etiqueta managedByFIS=true.
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:GetManagedPrefixListEntries`
- `ec2:ReplaceNetworkAclAssociation`

AWS política gestionada

- AWSFaultInjectionSimulatorNetworkAccess

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

Bloquea el tráfico que se origina en las subredes de destino y está destinado a la región especificada.

Tipo de recurso

- aws:ec2:subnet

Parámetros

- `region`: código de la región que desea aislar (por ejemplo, eu-west-1).
- `duration`: tiempo que dura la acción. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

- `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

† Se acota mediante la etiqueta managedByFIS=true.

AWS política gestionada

- AWSFaultInjectionSimulatorNetworkAccess

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

Bloquea el tráfico procedente de las vinculaciones de interconexiones de la puerta de enlace de tránsito de destino que está destinado a la región especificada.

Tipo de recurso

- aws:ec2:transit-gateway

Parámetros

- region: código de la región que desea aislar (por ejemplo, eu-west-1).
- duration: tiempo que dura la acción. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

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

- `ec2:DisassociateTransitGatewayRouteTable`

AWS política gestionada

- AWSFaultInjectionSimulatorNetworkAccess

Acciones de Amazon RDS

AWS FIS admite las siguientes acciones de Amazon RDS.

Acciones

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

aws:rds:failover-db-cluster

Ejecuta la acción FailoverDBCluster de la API de Amazon RDS en el clúster de base de datos Aurora de destino.

Tipo de recurso

- aws:rds:cluster

Parámetros

- Ninguna

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorRDSAccess

aws:rds:reboot-db-instances

Ejecuta la acción RebootDBInstance de la API de Amazon RDS en las instancias de base de datos de destino.

Tipo de recurso

- aws:rds:db

Parámetros

- forceFailover: opcional. Si el valor es true y las instancias son Multi-AZ, fuerza la conmutación por error de una zona de disponibilidad a otra. El valor predeterminado es false.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorRDSAccess

Acciones de Amazon S3

AWS FIS admite la siguiente acción de Amazon S3.

Acciones

- aws:s3:bucket-pause-replication

aws:s3:bucket-pause-replication

Pausa la replicación de buckets de origen objetivo en buckets de destino. Los buckets de destino pueden estar en diferentes regiones de AWS o dentro de la misma región que el bucket de origen. Los objetos existentes pueden seguir replicándose hasta una hora después de que comience la

acción. Esta acción solo puede llevarse a cabo mediante etiquetas. Para obtener más información sobre la replicación de Amazon S3, consulte la Guía del usuario de Amazon S3.

Tipo de recurso

- aws:s3:bucket

Parámetros

- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.
- region: región de AWS en la que se encuentran los buckets de destino.
- destinationBuckets: opcional. Lista separada por comas de los buckets de S3 de destino.
- prefixes: opcional. Lista separada por comas de prefijos de claves de objetos de S3 procedentes de filtros de reglas de replicación. Se pausarán las reglas de replicación de los buckets de destino con un filtro basado en los prefijos.

Permisos

- `S3:PutReplicationConfiguration` con la clave de condición `S3:IsReplicationPauseRequest` establecida en `True`
- `S3:GetReplicationConfiguration` con la clave de condición `S3:IsReplicationPauseRequest` establecida en `True`
- `S3:PauseReplication`
- `S3:ListAllMyBuckets`
- `tag:GetResources`

Para ver una política de ejemplo, consulte Ejemplo: utilizar claves de condición para `aws:s3:bucket-pause-replication`.

Acciones de Systems Manager

AWS FIS admite las siguientes acciones de Systems Manager.

Acciones

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

aws:ssm:send-command

Ejecuta la acción de la API de Systems Manager SendCommanden las instancias EC2 de destino. Un documento de Systems Manager (documento de SSM) define las acciones que Systems Manager realiza en sus instancias. Para obtener más información, consulte Uso de la acción aws:ssm:send-command.

Tipo de recurso

- aws:ec2:instance

Parámetros

- documentArn: el nombre de recurso de Amazon (ARN) del documento. En la consola, este parámetro se completa automáticamente si elige un valor del tipo de acción que corresponda a uno de los documentos AWS FIS SSM preconfigurados.
- documentVersion: opcional. La versión del documento. Si está vacía, se ejecuta la versión predeterminada.
- documentParameters: condicional. Los parámetros obligatorios y opcionales que el documento acepta. El formato es un objeto JSON con claves que son cadenas, y valores que son cadenas o matrices de cadenas.
- duration: la duración, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

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

AWS política gestionada

- AWSFaultInjectionSimulatorEC2Access

aws:ssm:start-automation-execution

Ejecuta la acción de la API de Systems Manager StartAutomationExecution.

Tipo de recurso

- Ninguna

Parámetros

- documentArn: el nombre de recurso de Amazon (ARN) del documento de automatización.
- documentVersion: opcional. La versión del documento. Si está vacía, se ejecuta la versión predeterminada.
- documentParameters: condicional. Los parámetros obligatorios y opcionales que el documento acepta. El formato es un objeto JSON con claves que son cadenas, y valores que son cadenas o matrices de cadenas.
- maxDuration: el tiempo máximo permitido para que se complete la ejecución de la automatización, de un minuto a 12 horas. En la AWS FIS API, el valor es una cadena en formato ISO 8601. Por ejemplo, PT1M representa un minuto. En la AWS FIS consola, se introduce el número de segundos, minutos u horas.

Permisos

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `iam:PassRole`: opcional. Obligatorio si el documento de automatización asume un rol.

AWS política gestionada

- AWSFaultInjectionSimulatorSSMAccess

Utilice los documentos SSM de Systems Manager con AWS FIS

AWS El FIS admite tipos de errores personalizados a través del agente AWS Systems Manager SSM y la acción del FIS. AWS aws:ssm:send-command Los documentos SSM de Systems Manager preconfigurados (documentos SSM) que se pueden utilizar para crear acciones comunes de inyección de errores están disponibles como AWS documentos públicos que comienzan con el AWSFIS prefijo -.

SSM Agent es software de Amazon que se puede instalar y configurar en instancias de Amazon EC2, en servidores en las instalaciones o en máquinas virtuales (VM). Esto posibilita que Systems Manager administre estos recursos. Agent procesa las solicitudes de Systems Manager y, a continuación, las ejecuta como se especifica en la solicitud. Puede incluir su propio documento de SSM para inyectar errores personalizados o hacer referencia a uno de los documentos públicos propiedad de Amazon.

Requisitos

Para las acciones que requieren que SSM Agent ejecute la acción en el destino, debe asegurarse de lo siguiente:

- Agent está instalado en el destino. De forma predeterminada, SSM Agent se instala en algunas imágenes de máquina de Amazon (AMI). Si no, puede instalar SSM Agent en sus instancias. Para obtener más información, consulte Instalación manual de SSM Agent en instancias EC2 en la Guía del usuario de AWS Systems Manager .
- Systems Manager tiene permiso para realizar acciones en sus instancias. Puede otorgar acceso con un perfil de instancia de IAM. Para obtener más información, consulte Creación de un perfil de instancia de IAM para Systems Manager y Adjuntar un perfil de instancia de IAM a una instancia EC2 en la Guía del usuario de AWS Systems Manager .

Uso de la acción aws:ssm:send-command

Un documento de SSM define las acciones que Systems Manager realiza en las instancias administradas. Systems Manager incluye una serie de documentos preconfigurados, o también puede crear los suyos propios. Para obtener más información sobre cómo crear su propio documento de SSM, consulte Creación de documentos de Systems Manager en la Guía del usuario de AWS Systems Manager . Para obtener más información sobre los documentos de SSM en general, consulte Documentos de AWS Systems Manager en la Guía del usuario de AWS Systems Manager .

AWS El FIS proporciona documentos SSM preconfigurados. Puede ver los documentos SSM preconfigurados en la sección Documentos de la consola: https://console.aws.amazon.com/systems-manager/documents. AWS Systems Manager También puede elegir entre una selección de documentos preconfigurados en la AWS consola FIS. Para obtener más información, consulte Documentos AWS FIS SSM preconfigurados.

Para utilizar un documento SSM en sus experimentos con el AWS FIS, puede utilizar la acción. aws:ssm:send-command Esta acción recupera el documento de SSM especificado en las instancias de destino y lo ejecuta.

Al utilizar la acción `aws:ssm:send-command` en la plantilla de experimento, debe especificar parámetros adicionales para la acción, incluidos los siguientes:

- documentArn: obligatorio. El nombre de recurso de Amazon (ARN) del documento de SSM.
- documentParameters: condicional. Los parámetros obligatorios y opcionales que el documento de SSM acepta. El formato es un objeto JSON con claves que son cadenas, y valores que son cadenas o matrices de cadenas.
- documentVersion: opcional. La versión del documento de SSM que se va a ejecutar.

Puede ver la información de un documento de SSM (incluidos los parámetros del documento) con la consola de Systems Manager o la línea de comandos.

Para ver información acerca de un documento de SSM con la consola

1. Abre la AWS Systems Manager consola en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Documentos.
3. Seleccione el documento y elija la pestaña Detalles.

Para ver información acerca de un documento de SSM con la línea de comandos

Utilice el comando describe-document de SSM.

Documentos AWS FIS SSM preconfigurados

Puede utilizar documentos FIS SSM AWS preconfigurados con la acción en las `aws:ssm:send-command` plantillas de sus experimentos.

Requisitos

- Los documentos SSM preconfigurados proporcionados por el AWS FIS solo son compatibles con los siguientes sistemas operativos:
  - Amazon Linux 2023, Amazon Linux 2, Amazon Linux
  - Ubuntu
  - RHEL 7, 8, 9
  - CentOS 7, 8, 9
- Los documentos SSM preconfigurados proporcionados por la AWS FIS solo son compatibles con las instancias EC2. No se admiten en otros tipos de nodos gestionados, como servidores en las instalaciones.

Para utilizar estos documentos de SSM en experimentos sobre tareas de ECS, utilice las the section called “Acciones de Amazon ECS” correspondientes. Por ejemplo, la acción aws:ecs:task-cpu-stress usa el documento AWSFIS-Run-CPU-Stress.

Documentos

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

AWSFIS-Run-CPU-Stress

Ejecuta esfuerzo de la CPU en una instancia con la herramienta stress-ng. Utiliza el documento SSM -Run-CPU-StressAWSFIS.

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- DurationSeconds: obligatorio. La duración de la prueba de esfuerzo de la CPU, en segundos.
- CPU: opcional. El número de factores de esfuerzo de la CPU que se van a utilizar. El valor predeterminado es 0, que utiliza todos los factores de esfuerzo de la CPU.
- LoadPercent: opcional. El porcentaje de carga de la CPU de destino, de 0 (sin carga) a 100 (carga completa). El valor predeterminado es 100.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es stress-ng.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Disk-Fill

Asigna espacio en disco en el volumen raíz de una instancia para simular un error de disco lleno. Utiliza el documento SSM AWSFIS-Run-Disk-Fill.

Si el experimento en el que se ha introducido este error se detiene, ya sea manualmente o mediante una condición de parada, el AWS FIS intenta revertirlo cancelando el documento SSM en ejecución. Sin embargo, si el disco está lleno al 100 %, ya sea debido al error o al error más la actividad de la aplicación, es posible que Systems Manager no pueda completar la operación de cancelación. Por lo tanto, si necesita detener el experimento, asegúrese de que el disco no se llene al 100 %.

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- DurationSeconds: obligatorio. La duración de la prueba de llenado de disco, en segundos.
- Percent: opcional. El porcentaje del disco que se debe asignar durante la prueba de llenado de disco. El valor predeterminado es 95 %.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd y fallocate.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-IO-Stress

Ejecuta esfuerzo de E/S en una instancia con la herramienta stress-ng. Utiliza el documento SSM - Run-IO-StressAWSFIS.

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- DurationSeconds: obligatorio. La duración de la prueba de esfuerzo de E/S, en segundos.
- Workers: opcional. El número de procesos de trabajo que realizan operaciones de lectura/escritura secuenciales, aleatorias y asignadas a memoria, sincronizaciones forzadas y pérdida de memoria caché. Varios procesos secundarios realizan diferentes operaciones de E/S en el mismo archivo. El valor predeterminado de es 1.
- Percent: opcional. El porcentaje de espacio libre en el sistema de archivos que se utilizará durante la prueba de esfuerzo de E/S. El valor predeterminado es 80 %.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es stress-ng.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Kill-Process

Detiene el proceso especificado en la instancia con el comando killall. Utiliza el documento SSM - Run-Kill-Process. AWSFIS

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- ProcessName: obligatorio. El nombre del proceso que se va a detener.
- Signal: opcional. La señal que se va a enviar junto con el comando. Los valores posibles son `SIGTERM` (que el receptor puede elegir ignorar) y `SIGKILL` (que no se pueden ignorar). El valor predeterminado es `SIGTERM`.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es killall.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Memory-Stress

Ejecuta esfuerzo de la memoria en una instancia con la herramienta stress-ng. Utiliza el documento SSM -Run-Memory-Stress. AWSFIS

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- DurationSeconds: obligatorio. La duración de la prueba de esfuerzo de la memoria, en segundos.
- Workers: opcional. El número de factores de esfuerzo de la memoria virtual. El valor predeterminado de es 1.
- Percent: obligatorio. El porcentaje de memoria virtual que se utilizará durante la prueba de esfuerzo de la memoria.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. La dependencia es stress-ng.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Network-Blackhole-Port

Elimina el tráfico entrante o saliente del protocolo y el puerto con la herramienta iptables. Utiliza el documento SSM -Run-Network-Blackhole-Port. AWSFIS

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- Protocol: obligatorio. El protocolo. Los valores posibles son `tcp` y `udp`.
- Port: obligatorio. El número de puerto.
- TrafficType: opcional. El tipo de tráfico. Los valores posibles son `ingress` y `egress`. El valor predeterminado es `ingress`.

- DurationSeconds: obligatorio. La duración de la prueba de agujero negro de red, en segundos.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd, dig e iptables.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Network-Latency

Agrega latencia a la interfaz de red con la tc herramienta. Utiliza el AWSFISdocumento SSM -Run-Network-Latency.

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- Interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- DelayMilliseconds: opcional. El retraso, en milisegundos. El valor predeterminado es 200.
- DurationSeconds: obligatorio. La duración de la prueba de latencia de red, en segundos.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd, dig e tc.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Network-Latency-Sources

Agrega latencia y fluctuación a la interfaz de red con la herramienta tc para el tráfico hacia fuentes específicas o desde ellas. Utiliza el documento SSM -Run-Network-Latency-Sources. AWSFIS

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- Interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- DelayMilliseconds: opcional. El retraso, en milisegundos. El valor predeterminado es 200.
- JitterMilliseconds: opcional. La fluctuación, en milisegundos. El valor predeterminado es 10.
- Sources: obligatorio. Las fuentes, separadas por comas. Los valores posibles son: una dirección IPv4, un bloque CIDR IPv4, un nombre de dominio, `DYNAMODB` y `S3`. Si especifica `DYNAMODB` o `S3`, solo se aplicará al punto de conexión regional de la región actual.
- TrafficType: opcional. El tipo de tráfico. Los valores posibles son `ingress` y `egress`. El valor predeterminado es `ingress`.
- DurationSeconds: obligatorio. La duración de la prueba de latencia de red, en segundos.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino, si aún no están instaladas. El valor predeterminado es `True`. Las dependencias son atd, dig, jq y tc.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

```
{"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

Agrega la pérdida de paquetes a la interfaz de red con la herramienta tc. Utiliza el AWSFISdocumento SSM -Run-Network-Packet-Loss.

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- Interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- LossPercent: opcional. El porcentaje de pérdida de paquetes. El valor predeterminado es 7 %.
- DurationSeconds: obligatorio. La duración de la prueba de pérdida de paquetes, en segundos.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino. El valor predeterminado es `True`. Las dependencias son atd, dig e tc.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

AWSFIS-Run-Network-Packet-Loss-Sources

Agrega pérdida de paquetes a la interfaz de red con la herramienta tc para el tráfico hacia fuentes específicas o desde ellas. Utiliza el documento SSM AWSFIS-Run-Network-Packet-Loss-Sources.

Tipo de acción (solo en la consola)

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

ARN

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

Parámetros de documento

- Interface: opcional. Interfaz de red. El valor predeterminado es `eth0`.
- LossPercent: opcional. El porcentaje de pérdida de paquetes. El valor predeterminado es 7 %.

- Sources: obligatorio. Las fuentes, separadas por comas. Los valores posibles son: una dirección IPv4, un bloque CIDR IPv4, un nombre de dominio, `DYNAMODB` y `S3`. Si especifica `DYNAMODB` o `S3`, solo se aplicará al punto de conexión regional de la región actual.
- TrafficType: opcional. El tipo de tráfico. Los valores posibles son `ingress` y `egress`. El valor predeterminado es `ingress`.
- DurationSeconds: obligatorio. La duración de la prueba de pérdida de paquetes, en segundos.
- InstallDependencies: opcional. Si este valor es `True`, Systems Manager instala las dependencias necesarias en las instancias de destino. El valor predeterminado es `True`. Las dependencias son atd, dig, jq y tc.

A continuación, un ejemplo de la cadena que se puede introducir en la consola.

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

Ejemplos

Para obtener un ejemplo de plantilla de experimento, consulte the section called “Ejecución de un documento de SSM de AWS FIS preconfigurado”.

Para ver un tutorial de ejemplo, consulte Ejecutar esfuerzo de la CPU en una instancia.

Solución de problemas

Use el siguiente procedimiento para solucionar problemas.

Para solucionar problemas con los documentos de SSM

1. Abra la AWS Systems Manager consola en https://console.aws.amazon.com/systems-manager/.
2. En el panel de navegación, elija Administración de nodos, Run Command.
3. En la pestaña Historial de comandos, utilice los filtros para localizar la ejecución del documento.
4. Elija el ID del comando para abrir la página de detalles.
5. Elija el ID de la instancia. Revise el resultado y los errores de cada paso.

Utilice las acciones AWS aws:ecs:task del FIS

Puede utilizar las acciones aws:ecs:task para inyectar errores en las tareas de Amazon ECS.

Estas acciones utilizan un agente de SSM como contenedor sidecar para ejecutar documentos de SSM que realizarán la inyección de errores y registrarán tareas de Amazon ECS como instancias gestionadas por SSM a través del contenedor sidecar. Para utilizar estas acciones tendrá que actualizar las definiciones de tareas de Amazon ECS para agregar el agente de SSM como contenedor sidecar para que registre la tarea en la que se está ejecutando como instancia gestionada por SSM. Cuando ejecuta la segmentación de un experimento de AWS FIS`aws:ecs:task`, el AWS FIS asigna las tareas de Amazon ECS de destino que especifique en una plantilla de experimento de AWS FIS a un conjunto de instancias gestionadas por SSM mediante una etiqueta de recurso que se añade a la instancia gestionada. `ECS_TASK_ARN` El valor de la etiqueta es el ARN de la tarea de Amazon ECS asociada en la que se deben ejecutar los documentos de SSM, por lo que no se debe eliminar al ejecutar el experimento.

Acciones

- 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”

Limitaciones

- Las siguientes acciones no funcionan con: AWS Fargate
  - aws:ecs:task-kill-process
  - aws:ecs:task-network-blackhole-port
  - aws:ecs:task-network-latency
  - aws:ecs:task-network-packet-loss
- Si ha habilitado ECS Exec, debe deshabilitarlo para poder utilizar estas acciones.

Requisitos

- Añada los siguientes permisos a la función de experimento AWS de la FIS:
  - `ssm:SendCommand`

  - `ssm:ListCommands`
  - `ssm:CancelCommand`
- Agregue los siguientes permisos al rol de IAM de tareas de Amazon ECS.
  - `ssm:CreateActivation`
  - `ssm:AddTagsToResource`
  - `iam:PassRole`

  Tenga en cuenta que puede especificar el ARN del rol de instancia administrada como recurso para el `iam:PassRole`.
- Cree un rol de IAM de ejecución de tareas de Amazon ECS y añada la política gestionada de TaskExecutionRolePolicyAmazonECS.
- Agregue los siguientes permisos al rol de instancia administrada asociada a las tareas registradas como instancias administradas:
  - `ssm:DeleteActivation`
  - `ssm:DeregisterManagedInstance`
- Agregue la política ManagedInstanceCore administrada de AmazonSSM a la función de instancia administrada asociada a las tareas registradas como instancias administradas.
- Defina la variable de entorno `MANAGED_INSTANCE_ROLE_NAME` con el nombre del rol de instancia administrada.
- Agregue un contenedor de SSM Agent a la definición de la tarea de ECS. El script de comandos registra las tareas de ECS como instancias administradas.

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

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

```
    "dnsSearchDomains": [],
    "extraHosts": [],
    "dockerSecurityOptions": [],
    "dockerLabels": {},
    "ulimits": [],
    "logConfiguration": {},
    "systemControls": []
}
```

Para obtener una versión más legible del script, consulte the section called “Versión de referencia del script”.

- Al utilizar las acciones `aws:ecs:task-network-blackhole-port`, `aws:ecs:task-network-latency` y `aws:ecs:task-network-packet-loss`, debe actualizar el contenedor de SSM Agent en la definición de la tarea de ECS con una de las siguientes opciones.
  - Opción 1: Agregar la capacidad específica de Linux.

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

  - Opción 2: Agregar todas las capacidades de Linux.

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

- Al utilizar las accionesaws:ecs:task-kill-process, `aws:ecs:task-network-blackhole-port`, `aws:ecs:task-network-latency` y `aws:ecs:task-network-packet-loss`, la definición de la tarea de ECS debe tener `pidMode` configurado en `task`.

Versión de referencia del script

La siguiente es una versión más legible del script en la sección Requisitos, para su consulta.

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

# This is the activation script used to register ECS tasks as Managed Instances in SSM
```

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

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

yum upgrade -y
yum install jq procps awscli -y

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

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

  kill -SIGTERM $SSM_AGENT_PID
}
trap term_handler SIGTERM SIGINT

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

# check if the agent is already running (it will be if ECS Exec is enabled)
```

```
if ! ps ax | grep amazon-ssm-agent | grep -v grep > /dev/null; then

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

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

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

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

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

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

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

```
    fi

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

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

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

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

Ejemplo de plantilla de experimento

A continuación, se muestra un ejemplo de plantilla de experimento para la acción the section called "aws:ecs:task-cpu-stress".

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

Utilice las acciones aws:eks:pod del AWS FIS

Puede usar las acciones aws:eks:pod para inyectar errores en los pods de Kubernetes que se ejecutan en sus clústeres de EKS.

Acciones

- 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”

Limitaciones

- Las siguientes acciones no funcionan con: AWS Fargate
  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- Las siguientes acciones no admiten el modo de red `bridge`:

  - aws:eks:pod-network-blackhole-port
  - aws:eks:pod-network-latency
  - aws:eks:pod-network-packet-loss
- No puede identificar destinos del tipo aws:eks:pod en la plantilla de experimento con los ARN de los recursos o las etiquetas de los recursos. Debe identificar los destinos con los parámetros de recursos necesarios.
- Las acciones `aws:eks:pod-network-latency` y `aws:eks:pod-network-packet-loss` no deben ejecutarse en paralelo y tener como objetivo el mismo pod. Según el valor del parámetro `maxErrors` que especifique, la acción puede terminar en estado completado o de error:
  - Si `maxErrorsPercent` es 0 (predeterminado), la acción finalizará en estado de error.
  - De lo contrario, el error se sumará al presupuesto de `maxErrorsPercent`. Si el número de inyecciones de error no alcanza los `maxErrors` indicados, la acción terminará en estado completado.
  - Puede identificar estos errores a partir de los registros del contenedor efímero inyectado en el pod de destino. Producirá error con `Exit Code: 16`.
- La acción `aws:eks:pod-network-blackhole-port` no debe ejecutarse en paralelo con otras acciones que tengan como objetivo el mismo pod y utilicen el mismo `trafficType`. Se admiten acciones paralelas que utilicen diferentes tipos de tráfico.
- FIS solo puede monitorizar el estado de la inyección de error cuando el `securityContext` de los pods de destino está configurado en `readOnlyRootFilesystem: false`. Sin esta configuración, todas las acciones del pod de EKS producirán error.

Requisitos

- Instálelo AWS CLI en su ordenador. Esto solo es necesario si va a utilizar la AWS CLI para crear roles de IAM. Para obtener más información, consulte Instalación o actualización de la AWS CLI.
- Instale kubectl en su equipo. Esto solo es necesario para interactuar con el clúster de EKS a fin de configurar o monitorizar la aplicación de destino. Para obtener más información, consulte https://kubernetes.io/docs/tasks/tools/.
- La versión de EKS mínima compatible es 1.23.

Creación de un rol de servicio para la cuenta de servicio de Kubernetes

Cree un rol de IAM para usar un rol de servicio. Para obtener más información, consulte the section called “Rol de experimento”.

Configurar la cuenta de servicio de Kubernetes

Configure una cuenta de servicio de Kubernetes para ejecutar experimentos con destinos en el espacio de nombres de Kubernetes especificado. En el siguiente ejemplo, la cuenta de servicio es *myserviceaccount* y el espacio de nombres es *default*. Tenga en cuenta que default es uno de los espacios de nombres estándar de Kubernetes.

Para configurar la cuenta de servicio de Kubernetes

1. Cree un archivo llamado `rbac.yaml` y agregue lo siguiente.

```
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. Ejecute el siguiente comando de la .

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

Asignación de su rol de experimento al usuario de Kubernetes

Utilice el siguiente comando para crear una asignación de identidades. Para obtener más información, consulte Administrar usuarios y roles de IAM en la documentación de eksctl.

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

Imágenes de contenedor de pods

Las imágenes del contenedor de pods proporcionadas por AWS FIS están alojadas en Amazon ECR. Al hacer referencia a una imagen de Amazon ECR, debe usar el URI de imagen completo.

Región de AWS	URI de imagen
US East (Ohio)	`051821878176.dkr.ecr.us-east-2.amazonaws.com/aws-fis-pod:0.1`
Este de EE. UU. (Norte de Virginia)	`731367659002.dkr.ecr.us-east-1.amazonaws.com/aws-fis-pod:0.1`
Oeste de EE. UU. (Norte de California)	`080694859247.dkr.ecr.us-west-1.amazonaws.com/aws-fis-pod:0.1`
Oeste de EE. UU. (Oregón)	`864386544765.dkr.ecr.us-west-2.amazonaws.com/aws-fis-pod:0.1`
África (Ciudad del Cabo)	`056821267933.dkr.ecr.af-south-1.amazonaws.com/aws-fis-pod:0.1`
Asia-Pacífico (Hong Kong)	`246405402639.dkr.ecr.ap-east-1.amazonaws.com/aws-fis-pod:0.1`
Asia-Pacífico (Bombay)	`524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1`
Asia-Pacífico (Seúl)	`526524659354.dkr.ecr.ap-northeast-2.amazonaws.com/aws-fis-pod:0.1`
Asia-Pacífico (Singapur)	`316401638346.dkr.ecr.ap-southeast-1.amazonaws.com/aws-fis-pod:0.1`
Asia-Pacífico (Sídney)	`488104106298.dkr.ecr.ap-southeast-2.amazonaws.com/aws-fis-pod:0.1`
Asia-Pacífico (Tokio)	`635234321696.dkr.ecr.ap-northeast-1.amazonaws.com/aws-fis-pod:0.1`
Canadá (centro)	`490658072207.dkr.ecr.ca-central-1.amazonaws.com/aws-fis-pod:0.1`

Región de AWS	URI de imagen
Europa (Fráncfort)	`713827034473.dkr.ecr.eu-central-1.amazonaws.com/aws-fis-pod:0.1`
Europa (Irlanda)	`205866052826.dkr.ecr.eu-west-1.amazonaws.com/aws-fis-pod:0.1`
Europa (Londres)	`327424803546.dkr.ecr.eu-west-2.amazonaws.com/aws-fis-pod:0.1`
Europa (Milán)	`478809367036.dkr.ecr.eu-south-1.amazonaws.com/aws-fis-pod:0.1`
Europa (París)	`154605889247.dkr.ecr.eu-west-3.amazonaws.com/aws-fis-pod:0.1`
Europa (Estocolmo)	`263175118295.dkr.ecr.eu-north-1.amazonaws.com/aws-fis-pod:0.1`
Medio Oriente (Baréin)	`065825543785.dkr.ecr.me-south-1.amazonaws.com/aws-fis-pod:0.1`
América del Sur (São Paulo)	`767113787785.dkr.ecr.sa-east-1.amazonaws.com/aws-fis-pod:0.1`
AWS GovCloud (Este de EE. UU.)	`246533647532.dkr.ecr.us-gov-east-1.amazonaws.com/aws-fis-pod:0.1`
AWS GovCloud (Estados Unidos-Oeste)	`246529956514.dkr.ecr.us-gov-west-1.amazonaws.com/aws-fis-pod:0.1`

Ejemplo de plantilla de experimento

A continuación, se muestra un ejemplo de plantilla de experimento para la acción the section called “aws:eks:pod-network-latency”.

```
{
```

```
    "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"
    }
}
```

Enumere las AWS FIS acciones utilizando el AWS CLI

Puede usar AWS Command Line Interface (AWS CLI) para ver información sobre las acciones que AWS FIS admite.

Requisito previo

Instálelo AWS CLI en su ordenador. Para empezar, consulte la AWS Command Line Interface Guía del usuario de . Para obtener más información sobre los comandos de AWS FIS, consulte fis en la Referencia de AWS CLI comandos.

Ejemplo: Enumeración de los nombres de todas las acciones

Puede enumerar los nombres de todas las acciones con el comando list-actions de la siguiente manera.

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

A continuación, se muestra un ejemplo del resultado.

```
aws:cloudwatch:assert-alarm-state
aws:dynamodb:encrypted-global-table-pause-replication
aws:ebs:pause-volume-io
aws:ec2:api-insufficient-instance-capacity-error
aws:ec2:asg-insufficient-instance-capacity-error
aws:ec2:reboot-instances
aws:ec2:send-spot-instance-interruptions
aws:ec2:stop-instances
aws:ec2:terminate-instances
aws:ecs:drain-container-instances
aws:ecs:stop-task
aws:eks:inject-kubernetes-custom-resource
aws:eks:terminate-nodegroup-instances
aws:elasticache:interrupt-cluster-az-power
aws:fis:inject-api-internal-error
aws:fis:inject-api-throttle-error
aws:fis:inject-api-unavailable-error
aws:fis:wait
aws:network:disrupt-connectivity
aws:network:route-table-disrupt-cross-region-connectivity
aws:network:transit-gateway-disrupt-cross-region-connectivity
aws:rds:failover-db-cluster
```

```
aws:rds:reboot-db-instances
aws:s3:bucket-pause-replication
aws:ssm:send-command
aws:ssm:start-automation-execution
```

Ejemplo: Visualización de la información sobre una acción

Una vez que tenga el nombre de una acción, puede ver información detallada sobre la acción con el comando get-action de la siguiente manera.

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

A continuación, se muestra un ejemplo del resultado.

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

Plantillas de experimentos para AWS FIS

Una plantilla de experimento contiene una o más acciones que se ejecutan en destinos específicos durante un experimento. También contiene las condiciones de detención que impiden que el experimento se salga de los límites. Después de crear una plantilla de experimento, puede utilizarla para ejecutar un experimento.

Componentes de plantilla

Utilizará los siguientes componentes para crear plantillas de experimento:

Conjunto de acciones

: Las acciones de AWS FIS que desea ejecutar. Las acciones se pueden ejecutar en el orden establecido que especifique o simultáneamente. Para obtener más información, consulte Conjunto de acciones.

Destinos

: Los AWS recursos sobre los que se lleva a cabo una acción específica. Para obtener más información, consulte Destinos.

Condiciones de detención

: Las CloudWatch alarmas que definen un umbral en el que el rendimiento de la aplicación no es aceptable. Si se activa una condición de parada mientras se está ejecutando un experimento, el AWS FIS lo detiene. Para obtener más información, consulte Condiciones de detención.

Rol de experimento

: Un rol de IAM que otorga a la AWS FIS los permisos necesarios para que pueda realizar experimentos en tu nombre. Para obtener más información, consulte Rol de experimento.

Opciones de experimento

: Opciones de la plantilla de experimento. Para obtener más información, consulte Opciones de experimento.

Su cuenta tiene cuotas relacionadas con el FIS. AWS Por ejemplo, hay una cuota en el número de acciones por plantilla de experimento. Para obtener más información, consulte Cuotas y limitaciones.

Sintaxis de plantilla

A continuación, se muestra la sintaxis de una plantilla de experimento.

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

Para ver ejemplos, consulte Plantillas de ejemplo.

Introducción

Para crear una plantilla de experimento utilizando el AWS Management Console, consulteCreación de una plantilla de experimento.

Para crear una plantilla de experimento con AWS CLI, consulteEjemplo de plantillas de experimento de AWS FIS.

Conjunto de acciones para AWS FIS

Para crear una plantilla de experimento, debe definir una o más acciones para formar el conjunto de acciones. Para obtener una lista de las acciones predefinidas proporcionadas por el AWS FIS, consulte. Acciones

Puede ejecutar una acción solo una vez durante un experimento. Para ejecutar la misma acción del AWS FIS más de una vez en el mismo experimento, agréguela a la plantilla varias veces con nombres diferentes.

Contenido

Sintaxis de acción

A continuación, se presenta la sintaxis de un conjunto de acciones:

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

Cuando se define un destino, se proporciona lo siguiente:

action_name

Un nombre para la acción.

actionId

El identificador de la acción.

description

Una descripción opcional.

parameters

Cualquier parámetro de acción.

startAfter

Cualquier acción que deba completarse antes de que se pueda iniciar esta acción. De lo contrario, la acción se ejecuta al inicio del experimento.

targets

Cualquier destino de acción.

Para ver ejemplos, consulte the section called “Acciones de ejemplo”.

Duración de la acción

Si una acción incluye un parámetro que se puede usar para especificar la duración de la acción, la acción se considera completa de forma predeterminada solo después de que haya transcurrido la duración especificada. Si ha establecido la opción de experimento de `emptyTargetResolutionMode` en `skip`, la acción se completará inmediatamente con el estado 'omitida' cuando no se hayan resuelto destinos. Por ejemplo, si especificas una duración de 5 minutos, el AWS FIS considerará que la acción se ha completado al cabo de 5 minutos. A continuación, inicia la siguiente acción, hasta que se hayan completado todas las acciones.

La duración puede ser el tiempo durante el que se mantiene una condición de acción o el tiempo durante el que se monitorizan las métricas. Por ejemplo, la latencia se inyecta durante el tiempo especificado. Para los tipos de acción casi instantáneos, como la terminación de una instancia, las condiciones de detención se monitorizan durante el tiempo especificado.

Si una acción incluye una acción posterior en los parámetros de acción, la acción posterior se ejecuta una vez completada la acción. El tiempo que se tarda en completar la acción posterior puede provocar un retraso entre la duración de la acción especificada y el comienzo de la siguiente acción (o el final del experimento, si se han completado todas las demás acciones).

Acciones de ejemplo

A continuación, se muestran algunos ejemplos.

Ejemplos

- Detener instancias EC2
- Interrumpir instancias de spot
- Interrumpir el tráfico de red
- Terminar procesos de trabajo de EKS

Ejemplo: Detener instancias EC2

La siguiente acción detiene las instancias EC2 identificadas con el destino denominado *`targetInstances`*. Después de dos minutos, reinicia las instancias de destino.

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

Ejemplo: Interrumpir instancias de spot

La siguiente acción detiene las instancias puntuales identificadas mediante el objetivo indicado. `targetSpotInstances` Espera dos minutos antes de interrumpir la instancia de spot.

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

Ejemplo: Interrumpir el tráfico de red

La siguiente acción deniega el tráfico entre las subredes de destino y las subredes de otras zonas de disponibilidad.

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

```
        },
        "targets": {
            "Subnets": "targetSubnets"
        }
    }
}
```

Ejemplo: Terminar procesos de trabajo de EKS

La siguiente acción cierra el 50% de las instancias de EC2 del clúster de EKS identificadas con el objetivo nombrado. *targetNodeGroups*

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

Objetivos de la AWS FIS

Un objetivo es uno o más AWS recursos en los que el Servicio de Inyección de AWS Fallos (AWS FIS) realiza una acción durante un experimento. Los objetivos pueden estar en la misma cuenta de AWS que el experimento o en una cuenta diferente utilizando un experimento con varias cuentas. Para obtener más información sobre cómo recurrir a recursos de otra cuenta, consulte Experimentos con varias cuentas.

Los destinos se definen al crear una plantilla de experimento. Puede utilizar el mismo destino para varias acciones en la plantilla de experimento.

AWS El FIS identifica todos los objetivos al inicio del experimento, antes de iniciar cualquiera de las acciones del conjunto de acciones. AWS El FIS utiliza los recursos objetivo que selecciona para todo el experimento. Si no se encuentra ningún destino, el experimento falla.

Contenido

Sintaxis de destino

A continuación, se presenta la sintaxis de un destino.

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

Cuando se define un destino, se proporciona lo siguiente:

target_name

Un nombre para el destino.

resourceType

El tipo de recurso.

resourceArns

Los nombres de recurso de Amazon (ARN) de recursos específicos.

resourceTags

Las etiquetas aplicadas a recursos específicos.

parameters

Los parámetros que identifican los destinos mediante atributos específicos.

filters

Los filtros de recursos se centran en los recursos de destino identificados mediante atributos específicos.

selectionMode

El modo de selección de los recursos identificados.

Para ver ejemplos, consulte the section called “Ejemplos de destinos”.

Tipos de recurso

Cada acción AWS del FIS se realiza en un tipo de AWS recurso específico. Cuando se define un destino, se debe especificar exactamente un tipo de recurso. Al especificar un destino para una acción, el destino debe ser el tipo de recurso compatible con la acción.

El AWS FIS admite los siguientes tipos de recursos:

- aws:dynamodb: encrypted-global-table — Tabla global cifrada con una clave gestionada por el cliente
- aws:ec2:autoscaling-group: grupo de Amazon EC2 Auto Scaling
- aws:ec2:ebs-volume: un volumen de Amazon EBS
- aws:ec2:instance: una instancia de Amazon EC2

- aws:ec2:spot-instance: una instancia de spot de Amazon EC2
- aws:ec2:subnet: una subred de Amazon VPC
- aws:ec2:transit-gateway: puerta de enlace de tránsito
- aws:ecs:cluster: un clúster de Amazon ECS
- aws:ecs:task: una tarea de Amazon ECS
- aws:eks:cluster: un clúster de Amazon EKS
- aws:eks:nodegroup: un grupo de nodos de Amazon EKS
- aws:eks:pod: un pod de Kubernetes
- aws:elasticache:redis-replicationgroup: un grupo de replicación de Redis ElastiCache
- aws:iam:role: un rol de IAM
- aws:rds:cluster: un clúster de base de datos de Amazon Aurora
- aws:rds:db: una instancia de base de datos de Amazon RDS
- aws:s3:bucket: bucket de Amazon S3

Identificación de recursos de destino

Al definir un objetivo en la consola FIS, puede elegir recursos específicos (de un tipo AWS de recurso específico) a los que dirigirse. AWS O bien, puede permitir que el AWS FIS identifique un grupo de recursos en función de los criterios que proporcione.

Para identificar sus recursos de destino, puede especificar lo siguiente:

- Identificadores de recursos: los identificadores de AWS recursos específicos. Todos los ID de recursos deben representar el mismo tipo de recurso.
- Etiquetas de recursos: las etiquetas que se aplican a AWS recursos específicos.
- Filtros de recursos: la ruta y los valores que representan los recursos con atributos específicos. Para obtener más información, consulte Filtros de recursos.
- Parámetros de recursos: los parámetros que representan los recursos que cumplen criterios específicos. Para obtener más información, consulte Parámetros de recursos.

Consideraciones

- No se puede especificar a la vez un ID de recurso y una etiqueta de recurso para el mismo destino.
- No se puede especificar a la vez un ID de recurso y un filtro de recurso para el mismo destino.

- Si especifica una etiqueta de recurso con un valor de etiqueta vacío, no equivale a un comodín. Coincide con recursos que tienen una etiqueta con la clave de etiqueta especificada y un valor de etiqueta vacío.

Filtros de recursos

Los filtros de recursos son consultas que identifican los recursos de destino según atributos específicos. AWS El FIS aplica la consulta al resultado de una acción de API que contiene la descripción canónica del AWS recurso, según el tipo de recurso que se especifique. Los recursos que tienen atributos que coinciden con la consulta se incluyen en la definición de destino.

Cada filtro se expresa como una ruta de atributos y valores posibles. Una ruta es una secuencia de elementos, separados por puntos, que describen la ruta para llegar a un atributo en el resultado de la acción Describe de un recurso. Cada elemento debe expresarse en tipo Pascal, incluso aunque el resultado de la acción Describir de un recurso esté en formato camel. Por ejemplo, debe utilizar `AvailabilityZone`, no `availablityZone` como elemento de atributo.

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

La siguiente tabla incluye las acciones y los AWS CLI comandos de la API que puedes usar para obtener las descripciones canónicas de cada tipo de recurso. AWS El FIS ejecuta estas acciones en tu nombre para aplicar los filtros que especifiques. La documentación correspondiente describe los recursos que se incluyen en los resultados de forma predeterminada. Por ejemplo, la documentación de los estados DescribeInstances indica que las instancias finalizadas recientemente podrían aparecer en los resultados.

Tipo de recurso	Acción de la API	AWS CLI comando
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes

Tipo de recurso	Acción de la API	AWS CLI comando
aws:ec2:instance	DescribeInstances	describe-instances
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ecs:cluster	DescribeClusters	describe-clusters
aws:ecs:task	DescribeTasks	describe-tasks
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	describe-nodegroup
aws:elasticache:redis-replicationgroup	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	list-roles
aws:rds:cluster	DescribeDBClusters	describe-db-clusters
aws:rds:db	DescribeDBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets

La siguiente lógica se aplica a todos los filtros de recursos:

- Valores dentro de un filtro: `OR`
- Valores entre filtros: `AND`

Para ver ejemplos, consulte the section called “Ejemplos de filtros”.

Parámetros de recursos

Los parámetros de recursos identifican los recursos de destino según criterios específicos.

El siguiente tipo de recurso admite parámetros.

aws:ec2:ebs-volume

- `availabilityZoneIdentifier`: el código (por ejemplo, us-east-1a) de la zona de disponibilidad que contiene los volúmenes de destino.

aws:ec2:subnet

- `availabilityZoneIdentifier`: el código (por ejemplo, us-east-1a) o ID de AZ (por ejemplo, use1-az1) de la zona de disponibilidad que contiene las subredes de destino.
- `vpc`: la VPC que contiene las subredes de destino. No admite más de una VPC por cuenta.

aws:ecs:task

- `cluster`: el clúster que contiene las tareas de destino.
- `service`: el servicio que contiene las tareas de destino.

aws:eks:pod

- `availabilityZoneIdentifier`: opcional. La zona de disponibilidad que contiene los pods de destino. Por ejemplo, `us-east-1d`. Determinamos la zona de disponibilidad de un pod comparando su IP del host y el CIDR de la subred del clúster.
- `clusterIdentifier`: obligatorio. El nombre o ARN del clúster de destino de EKS.
- `namespace`: obligatorio. El espacio de nombres de Kubernetes de los pods de destino.
- `selectorType`: obligatorio. El tipo de selector. Los valores posibles son `labelSelector`, `deploymentName` y `podName`.
- `selectorValue`: obligatorio. El valor del selector. Este valor depende del valor de `selectorType`.
- `targetContainerName`: opcional. El nombre del contenedor de destino tal y como se especifica en la especificación de pod. El valor predeterminado es el primer contenedor definido en la especificación de cada pod de destino.

aws:rds:cluster

- `writerAvailabilityZoneIdentifiers`: opcional. Las zonas de disponibilidad del escritor del clúster de base de datos. Los valores posibles son: una lista de identificadores de zonas de disponibilidad separados por comas, `all`.

aws:rds:db

- `availabilityZoneIdentifiers`: opcional. Las zonas de disponibilidad de la instancia de base de datos que se van a ver afectadas. Los valores posibles son: una lista de identificadores de zonas de disponibilidad separados por comas, `all`.

aws:elasticache:redis-replicationgroup

- `availabilityZoneIdentifier`: obligatorio. El código (por ejemplo, us-east-1a) o ID de AZ (por ejemplo, use1-az1) de la zona de disponibilidad que contiene los nodos de destino.

Modo de selección

Para determinar el alcance de los recursos identificados, especifique un modo de selección. AWS El FIS admite los siguientes modos de selección:

- `ALL`: ejecuta la acción en todos los destinos.
- `COUNT(n)`: ejecuta la acción en el número especificado de destinos, elegidos de entre los destinos identificados al azar. Por ejemplo, COUNT(1) selecciona uno de los destinos identificados.
- `PERCENT(n)`: ejecuta la acción en el porcentaje especificado de destinos, elegidos de entre los destinos identificados al azar. Por ejemplo, PERCENT(25) selecciona el 25 % de los destinos identificados.

Si tiene un número impar de recursos y especifica el 50%, el AWS FIS lo redondea a la baja. Por ejemplo, si añade cinco instancias de Amazon EC2 como objetivos y el alcance hasta el 50%, AWS FIS redondea a la baja a dos instancias. No puede especificar un porcentaje inferior a un recurso. Por ejemplo, si añade cuatro instancias de Amazon EC2 y el alcance AWS es del 5%, FIS no podrá seleccionar ninguna instancia.

Si define varios objetivos con el mismo tipo de recurso de destino, AWS FIS puede seleccionar el mismo recurso varias veces.

Independientemente del modo de selección que utilice, si el alcance que especifique no identifica ningún recurso, el experimento fallará.

Ejemplos de destinos

A continuación, se muestran algunos ejemplos de destinos.

Ejemplos

- Instancias de la VPC especificada con las etiquetas especificadas
- Tareas con los parámetros especificados

Ejemplo: Instancias de la VPC especificada con las etiquetas especificadas

Los posibles destinos de este ejemplo son las instancias de Amazon EC2 en la VPC especificada con la etiqueta env=prod. El modo de selección especifica que el AWS FIS elige uno de estos objetivos al azar.

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

Ejemplo: Tareas con los parámetros especificados

Los posibles destinos de este ejemplo son las tareas de Amazon ECS con el clúster y el servicio especificados. El modo de selección especifica que el AWS FIS elija uno de estos objetivos al azar.

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

```
}
```

Ejemplos de filtros

A continuación, se muestran algunos ejemplos.

Ejemplos

- Instancias EC2
- Clústeres de base de datos

Ejemplo: Instancias EC2

Al especificar un filtro para una acción que admite el tipo de recurso aws:ec2:instance, AWS FIS utiliza el describe-instances comando Amazon EC2 y aplica el filtro para identificar los objetivos.

El comando describe-instances devuelve el resultado JSON en el que cada instancia es una estructura en `Instances`. El siguiente es un resultado parcial que incluye los campos marcados con *`cursiva.`* Proporcionaremos ejemplos en los que se utilizan estos campos para especificar una ruta de atributos a partir de la estructura de la salida de JSON.

```
{
    "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-00bbbbbbbbbbbbbbb",
                    ...
                },
                ...
                {
                    ...
                }
            ],
            "OwnerId": "123456789012",
            "ReservationId": "r-aaaaaabbbbb111111"
        },
        ...
    ]
}
```

Para seleccionar instancias en una zona de disponibilidad específica mediante un filtro de recursos, especifique la ruta de atributos de la `AvailabilityZone` y el código de la zona de disponibilidad como valor. Por ejemplo:

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

Para seleccionar instancias en una subred específica mediante un filtro de recursos, especifique la ruta de atributos de `SubnetId` y el ID de la subred como valor. Por ejemplo:

```
"filters": [
    {
```

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

Para seleccionar instancias que estén en un estado de instancia específico, especifique la ruta de atributos de `Name` y uno de los siguientes nombres de estado como valor: `pending` | `running` | `shutting-down` | `terminated` | `stopping` | `stopped`. Por ejemplo:

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

Ejemplo: Clúster de Amazon RDS (clúster de base de datos)

Al especificar un filtro para una acción que admite el tipo de recurso aws:rds:cluster, FIS AWS ejecuta el describe-db-clusters comando Amazon RDS y aplica el filtro para identificar los objetivos.

El comando describe-db-clusters devuelve un resultado JSON similar al siguiente para cada clúster de base de datos. El siguiente es un resultado parcial que incluye los campos marcados con *`cursiva`*. Proporcionaremos ejemplos en los que se utilizan estos campos para especificar una ruta de atributos a partir de la estructura de la salida de JSON.

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

Para aplicar un filtro de recursos que devuelva solo los clústeres de base de datos que utilizan un motor de base de datos específico, especifique la ruta de atributos como `Engine` y el valor como `aurora-postgresql`, tal y como se muestra en el siguiente ejemplo.

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

Para aplicar un filtro de recursos que devuelva solo los clústeres de base de datos de una zona de disponibilidad específica, especifique la ruta de atributos y el valor tal y como se muestra en el siguiente ejemplo.

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

Condiciones de detención para AWS FIS

AWS Fault Injection Service (AWS FIS) proporciona controles y barreras de protección para que pueda ejecutar experimentos de forma segura en cargas de trabajo de AWS. Una condición de parada es un mecanismo para detener un experimento si alcanza un umbral que tú defines como CloudWatch alarma de Amazon. Si se activa una condición de detención durante un experimento, AWS FIS detiene el experimento. No se puede reanudar un experimento detenido.

Para crear una condición de detención, defina primero el estado estable de la aplicación o el servicio. El estado estable es cuando la aplicación tiene un rendimiento óptimo, definido en términos de métricas empresariales o técnicas. Por ejemplo, la latencia, la carga de la CPU o el número de reintentos. Puedes usar el estado estable para crear una CloudWatch alarma que puedas usar para detener un experimento si tu aplicación o servicio alcanza un estado en el que su rendimiento no sea aceptable. Para obtener más información, consulta Uso de CloudWatch alarmas de Amazon en la Guía del CloudWatch usuario de Amazon.

Su cuenta tiene una cuota de condiciones de detención que se pueden especificar en una plantilla de experimento. Para obtener más información, consulte Cuotas y limitaciones del servicio de inyección de AWS averías.

Sintaxis de condiciones de detención

Al crear una plantilla de experimento, se especifican una o más condiciones de parada especificando las CloudWatch alarmas que se han creado.

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

El siguiente ejemplo indica que la plantilla de experimento no especifica una condición de detención.

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

Más información

Para ver un tutorial que muestra cómo crear una CloudWatch alarma y añadir una condición de parada a una plantilla de experimento, consulteEjecutar esfuerzo de la CPU en una instancia.

Para obtener más información sobre las CloudWatch métricas disponibles para los tipos de recursos compatibles con el AWS FIS, consulte lo siguiente:

- Supervise sus instancias mediante CloudWatch
- CloudWatch Métricas de Amazon ECS
- Supervisión de las métricas de Amazon RDS mediante CloudWatch
- Monitorización de las métricas de Run Command mediante CloudWatch

Roles de IAM para los experimentos de AWS FIS

AWS Identity and Access Management (IAM) es un servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos de AWS. Para utilizar AWS FIS, debe crear un rol de IAM que conceda a AWS FIS los permisos necesarios para que AWS FIS pueda realizar experimentos en su nombre. Este rol de experimento se especifica al crear una plantilla de experimento. En el caso de un experimento con una sola cuenta, la política de IAM del rol de experimento debe conceder permiso para modificar los recursos que especifique como destinos de la plantilla de experimento. En el caso de un experimento con varias cuentas, el rol del experimento debe conceder permiso al rol de orquestador para que asuma el rol de IAM para cada cuenta de destino. Para obtener más información, consulte Permisos para experimentos con varias cuentas.

Le recomendamos que siga la práctica de seguridad estándar para conceder privilegios mínimos. Puede hacerlo especificando ARN o etiquetas de recursos específicos en sus políticas.

Para ayudarle a empezar a utilizar AWS FIS rápidamente, le ofrecemos políticas administradas por AWS que puede especificar al crear un rol de experimento. Como alternativa, también puede utilizar estas políticas como modelo al crear sus propios documentos de políticas insertadas.

Contenido

Requisitos previos

Antes de comenzar, instale la AWS CLI y cree la política de confianza requerida.

Instalar la AWS CLI

Antes de comenzar, instale y configure la AWS CLI. Cuando configure la AWS CLI, se le solicitarán credenciales de AWS. En los ejemplos de este procedimiento se asume que configuró una región predeterminada. De lo contrario, agregue la opción `--region` para cada comando. Para obtener más información, consulte Installing or updating the AWS CLI and Configuring the AWS CLI (Instalación o actualización de la CLI y Configuración de la CLI).

Creación de una política de relación de confianza

Un rol de experimento debe tener una relación de confianza que permita al servicio AWS FIS asumir el rol. Cree un archivo de texto denominado `fis-role-trust-policy.json` y añada la siguiente política de relación de confianza.

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

Le recomendamos que utilice las claves de condición `aws:SourceAccount` y `aws:SourceArn` para protegerse contra el problema del suplente confuso. La cuenta de origen es la propietaria del experimento, y el ARN de origen es el ARN del experimento. Por ejemplo, debería agregar el siguiente bloque de condición a su política de confianza:

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

Agregue permisos para asumir roles de cuentas de destino (solo en experimentos con varias cuentas)

En el caso de los experimentos con varias cuentas, se necesitan permisos que autoricen a la cuenta del orquestador a asumir roles de cuentas de destino. Puede modificar el siguiente ejemplo y agregarlo como documento de política integrado para asumir roles de cuentas de destino:

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

Opción 1: Crear un rol de experimento y asociar una política administrada por AWS

Utilice una de las políticas administradas por AWS de AWS FIS para empezar rápidamente.

Para crear un rol de experimento y asociar una política administrada por AWS

1. Compruebe que haya una política administrada para las acciones de AWS FIS en su experimento. De lo contrario, tendrá que crear su propio documento de política insertada. Para obtener más información, consulte the section called “AWS políticas gestionadas”.
2. Utilice el siguiente comando create-role para crear un rol y agregar la política de confianza que ha creado en los requisitos previos.

   ```
   aws iam create-role --role-name my-fis-role --assume-role-policy-document
    file://fis-role-trust-policy.json
   ```
3. Utilice el siguiente attach-role-policycomando para adjuntar la política AWS gestionada.

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

   Dónde *fis-policy-arn*está uno de los siguientes:

   - 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

Opción 2: Crear un rol de experimento y agregar un documento de política insertada

Use esta opción para las acciones que no tengan una política administrada o para incluir solo los permisos necesarios para su experimento específico.

Para crear un rol de experimento y agregar un documento de política insertada

1. Utilice el siguiente comando create-role para crear un rol y agregar la política de confianza que ha creado en los requisitos previos.

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

2. Cree un archivo de texto denominado `fis-role-permissions-policy.json` y agregar una política de permisos. Como ejemplo que puede utilizar como punto de partida, consulte lo siguiente.

   - Acciones de inyección de errores: comience con la siguiente política.

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

```
    ]
}
```

- Acciones de Amazon EBS: comience con la siguiente política.

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

- Acciones de Amazon EC2: comience por la AWSFaultInjectionSimulatorEC2Accesspolítica.
- Acciones de Amazon ECS: comience por la AWSFaultInjectionSimulatorECSAccesspolítica.
- Acciones de Amazon EKS: comience por la AWSFaultInjectionSimulatorEKSAccesspolítica.
- Acciones de red: comience desde la AWSFaultInjectionSimulatorNetworkAccesspolítica.
- Acciones de Amazon RDS: comience por la AWSFaultInjectionSimulatorRDSAccesspolítica.
- Acciones de Systems Manager: comience desde la AWSFaultInjectionSimulatorSSMAccesspolítica.

3. Use el siguiente put-role-policycomando para agregar la política de permisos que creó en el paso anterior.

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

Opciones de experimento

Las opciones de experimento son ajustes opcionales de un experimento. Puede definir determinadas opciones de experimento en la plantilla de experimento. Las opciones de experimento adicionales se configuran al comenzar el experimento.

La siguiente es la sintaxis de las opciones de experimento que se definen en la plantilla del experimento.

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

Si no especifica ninguna opción de experimento al crear la plantilla de experimento, se utilizará la opción predeterminada para cada opción.

La siguiente es la sintaxis de las opciones de experimento que se configuran al comenzar el experimento.

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

Si no especifica ninguna opción de experimento al comenzar el experimento, `run-all` se utilizará la opción predeterminada.

Contenido

Segmentación de cuentas

Si tienes varias AWS cuentas con recursos a los que quieres dirigirte en un experimento, puedes definir un experimento con varias cuentas mediante la opción de experimento de segmentación de cuentas. Los experimentos con varias cuentas se ejecutan desde una cuenta de orquestador que afecta a los recursos de varias cuentas de destino. La cuenta del orquestador es la propietaria de la plantilla del AWS FIS experimento y del experimento. Una cuenta de destino es una cuenta de AWS individual con recursos que pueden verse afectados por un AWS FIS experimento. Para obtener más información, consulte Experimentos con varias cuentas para AWS FIS.

La segmentación de cuentas se utiliza para indicar la ubicación de los recursos de destino. Puede proporcionar dos valores para segmentación de cuentas:

- single-account: predeterminada. El experimento solo se destinará a los recursos de la AWS cuenta en la que se ejecute el AWS FIS experimento.
- multi-account: el objetivo del experimento pueden ser recursos de varias cuentas de AWS.

Configuraciones de cuentas de destino

Para realizar un experimento con varias cuentas debe definir una o más configuraciones de cuenta de destino. La configuración de una cuenta de destino especifica accountId, roleArn y la descripción de cada cuenta con los recursos segmentados en el experimento. Los ID de cuenta de las configuraciones de cuenta de destino de una plantilla de experimento deben ser únicos.

Al crear una plantilla de experimento con varias cuentas, la plantilla de experimento devolverá un campo de solo lectura, `targetAccountConfigurationsCount`, que es un recuento de todas las configuraciones de cuenta de destino de la plantilla de experimento.

A continuación, se presenta la sintaxis de una configuración de cuentas de destino.

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

Cuando crea una configuración de cuenta de destino debe proporcionar lo siguiente:

accountId

ID de cuenta de AWS de 12 dígitos de la cuenta de destino.

roleArn

Un rol de IAM que otorga AWS FIS permisos para realizar acciones en la cuenta de destino.

description

Una descripción opcional.

Para obtener más información acerca de cómo trabajar con configuraciones de cuenta de destino, consulte the section called “Trabajo en experimentos con varias cuentas”.

Modo de resolución de destino vacío

Este modo ofrece la opción de permitir que los experimentos se completen incluso cuando un recurso de destino no está resuelto.

- fail: predeterminado. Si no se ha resuelto ningún recurso para el destino, el experimento se termina inmediatamente con un estado de `failed`.
- skip: si no se ha resuelto ningún recurso para el destino, el experimento continuará y se omitirán las acciones que no tengan destinos resueltos. No se pueden omitir las acciones con destinos definidos mediante identificadores únicos, como ARN. Si no se encuentra un destino definido mediante un identificador único, el experimento se termina inmediatamente con un estado de `failed`

Modo de acciones

El modo Acciones es un parámetro opcional que puede especificar al iniciar un experimento. Puede configurar el modo de acciones `skip-all` para generar una vista previa del objetivo antes de introducir errores en los recursos objetivo. La vista previa de destino le permite verificar lo siguiente:

- Que ha configurado la plantilla de experimento para destinarla a los recursos que espera. Los recursos reales a los que se dirige al iniciar este experimento pueden ser diferentes de los de la vista previa, ya que los recursos se pueden eliminar, actualizar o muestrear de forma aleatoria.
- Que las configuraciones de registro estén configuradas correctamente.
- Que para los experimentos con varias cuentas, ha configurado correctamente un rol de IAM para cada una de las configuraciones de su cuenta de destino.

> **Note**
>
> Este `skip-all` modo no te permite comprobar que tienes los permisos necesarios para ejecutar el AWS FIS experimento y realizar acciones con tus recursos.

El parámetro del modo de acciones acepta los siguientes valores:

- `run-all`- (Predeterminado) El experimento realizará acciones con los recursos objetivo.
- `skip-all`- El experimento omitirá todas las acciones relacionadas con los recursos objetivo.

Para obtener más información sobre cómo configurar el parámetro del modo de acciones al iniciar un experimento, consulteGenera una vista previa de un objetivo a partir de una plantilla de experimento.

Trabaje con plantillas de experimentos del AWS FIS

Puede crear y gestionar plantillas de experimentos mediante la consola AWS FIS o la línea de comandos. Después de crear una plantilla de experimento, puede utilizarla para ejecutar un experimento.

Tareas

- Creación de una plantilla de experimento
- Visualización de plantillas de experimento
- Genera una vista previa de un objetivo a partir de una plantilla de experimento
- Inicio de un experimento a partir de una plantilla
- Actualización de una plantilla de experimento
- Etiquetado de plantillas de experimento
- Eliminación de una plantilla de experimento

Creación de una plantilla de experimento

Antes de empezar, complete las siguientes tareas:

- Planifique su experimento.

- Cree un rol de IAM que conceda permiso al servicio AWS FIS para realizar acciones en su nombre. Para obtener más información, consulte Roles de IAM para los experimentos de AWS FIS.
- Asegúrese de tener acceso al FIS. AWS Para obtener más información, consulte Ejemplos de política de AWS FIS.

Para crear una plantilla de experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Elija Crear plantilla de experimento.
4. (Opcional) Para Segmentación de cuentas, seleccione Varias cuentas para configurar una plantilla de experimento con varias cuentas.
5. Para Segmentación de cuentas, seleccione Confirmar.
6. En Descripción y nombre, escriba un nombre y una descripción para la plantilla.
7. En Acciones, especifique el conjunto de acciones de la plantilla. Para cada acción, elija Agregar acción y complete lo siguiente:

   - En Nombre, escriba un nombre para la acción.

     Se permiten caracteres alfanuméricos, guiones (-) y guiones bajos (_). El nombre debe comenzar por una letra. No se permiten espacios. El nombre de cada acción debe ser único en esta plantilla.
   - (Opcional) En Descripción, ingrese una descripción para la acción. La longitud máxima es de 512 caracteres.
   - (Opcional) En Comenzar después, seleccione otra acción definida en esta plantilla que debe completarse antes de que comience la acción actual. De lo contrario, la acción se ejecuta al inicio del experimento.
   - En Tipo de acción, elija la acción AWS FIS.
   - En Destino, elija un destino que haya definido en la sección Destinos. Si aún no ha definido un objetivo para esta acción, el AWS FIS crea uno nuevo para usted.
   - En Parámetros de acción, especifique los parámetros de la acción. Esta sección aparece solo si la acción del AWS FIS tiene parámetros.
   - Seleccione Guardar.

8. En Destinos, defina los recursos de destino en los que se van a llevar a cabo las acciones. Debe especificar al menos un ID de recurso o una etiqueta de recurso como destino. Seleccione Editar para editar el objetivo que AWS FIS creó para usted en el paso anterior, o bien elija Agregar objetivo. En cada destino, haga lo siguiente:

   - En Nombre, escriba un nombre para el destino.

     Se permiten caracteres alfanuméricos, guiones (-) y guiones bajos (_). El nombre debe comenzar por una letra. No se permiten espacios. Cada nombre de destino debe ser único en esta plantilla.

   - En Tipo de recurso, elija un tipo de recurso que sea compatible con la acción.
   - En Método de destino, realice una de las siguientes acciones:
     - Elija ID de recursos y, a continuación, elija los ID de recursos.
     - Elija las etiquetas, los filtros y los parámetros de los recursos y, a continuación, agregue las etiquetas y los filtros que necesite. Para obtener más información, consulte the section called “Identificación de recursos de destino”.
   - En Modo de selección, elija Recuento para ejecutar la acción en el número especificado de destinos identificados o elija Porcentaje para ejecutar la acción en el porcentaje especificado de los destinos identificados. De forma predeterminada, la acción se ejecuta en todos los destinos identificados.
   - Seleccione Guardar.

9. Para actualizar una acción con el destino que ha creado, busque la acción en Acciones, elija Editar y, a continuación, actualice Destino. Puede utilizar el mismo destino para varias acciones.
10. (Solo experimentos con varias cuentas) En el caso de Configuraciones de cuentas de destino, agregue un ARN de rol y una descripción opcional para cada cuenta de destino. Para cargar los ARN de roles de cuentas de destino con un archivo CSV, seleccione Cargar ARN de roles para todas las cuentas de destino y, a continuación, seleccione Elegir un archivo CSV
11. En Acceso al servicio, elija Usar un rol de IAM existente y, a continuación, elija el rol de IAM que creó, tal como se describe en los requisitos previos de este tutorial. Si su rol no aparece, compruebe que tiene la relación de confianza requerida. Para obtener más información, consulte the section called “Rol de experimento”.
12. (Opcional) Para las condiciones de parada, selecciona las CloudWatch alarmas de Amazon para las condiciones de parada. Para obtener más información, consulte Condiciones de detención para AWS FIS.

13. (Opcional) En Registros, configure la opción de destino. Para enviar registros a un bucket de S3, seleccione Enviar a un bucket de Amazon S3 y escriba el nombre y el prefijo del bucket. Para enviar registros a CloudWatch registros, seleccione Enviar a CloudWatch registros e introduzca el grupo de registros.
14. (Opcional) En Etiquetas, elija Agregar nueva etiqueta y especifique una clave y un valor de etiqueta. Las etiquetas que agregue se aplican a la plantilla de experimento, no a los experimentos que se ejecutan con la plantilla.
15. Elija Crear plantilla de experimento. Cuando se le solicite confirmación, ingrese **`create`** y elija Crear plantilla de experimento.

Para crear una plantilla de experimento con la CLI

Utilice el comando create-experiment-template.

Puede cargar una plantilla de experimento desde un archivo JSON.

Utilice el parámetro `--cli-input-json`.

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

Para obtener más información, consulte Generar una plantilla de esqueleto de la CLI en la Guía del usuario de la AWS Command Line Interface . Para ver ejemplos de plantillas, consulte Ejemplo de plantillas de experimento de AWS FIS.

Visualización de plantillas de experimento

Puede ver las plantillas de experimento que ha creado.

Para ver una plantilla de experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Para ver la información de una plantilla específica, seleccione el ID de plantilla de experimento.
4. En la sección Detalles, puede ver la descripción y las condiciones de detención de la plantilla.
5. Para ver las acciones de la plantilla de experimento, elija Acciones.
6. Para ver los destinos de la plantilla de experimento, elija Destinos.

7. Para ver las etiquetas de la plantilla de experimento, elija Etiquetas.

Para ver una plantilla de experimento con la CLI

Utilice el list-experiment-templatescomando para obtener una lista de plantillas de experimentos y utilice el get-experiment-templatecomando para obtener información sobre una plantilla de experimento específica.

Genera una vista previa de un objetivo a partir de una plantilla de experimento

Antes de iniciar un experimento, puede generar una vista previa del objetivo para comprobar que la plantilla del experimento está configurada para destinarse a los recursos esperados. Los recursos a los que se dirige cuando comienzas el experimento propiamente dicho pueden ser diferentes de los de la vista previa, ya que los recursos se pueden eliminar, actualizar o muestrear de forma aleatoria. Al generar una vista previa de un objetivo, se inicia un experimento en el que se omiten todas las acciones.

> **Note**
>
> La generación de una vista previa de destino no le permite comprobar que dispone de los permisos necesarios para realizar acciones con sus recursos.

Para iniciar una vista previa de un destino mediante la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Para ver los destinos de la plantilla de experimento, elija Destinos.
4. Para verificar los recursos de destino para la plantilla del experimento, elija Generar vista previa. Cuando ejecutes un experimento, esta vista previa del objetivo se actualizará automáticamente con los objetivos del experimento más reciente.

Para iniciar una vista previa de destino mediante la CLI

- Ejecute el siguiente comando start-experiment. Sustituya los valores en cursiva por sus propios valores.

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

Inicio de un experimento a partir de una plantilla

Después de crear una plantilla de experimento, puede iniciar los experimentos con esa plantilla.

Al iniciar un experimento, creamos una instantánea de la plantilla especificada y la utilizamos para ejecutar el experimento. Por lo tanto, si la plantilla de experimento se actualiza o elimina mientras el experimento está en ejecución, esos cambios no afectarán al experimento en ejecución.

Al iniciar un experimento, el AWS FIS crea un rol vinculado al servicio en tu nombre. Para obtener más información, consulte Utilice funciones vinculadas al servicio para el servicio de inyección de errores AWS.

Tras iniciar el experimento, puede detenerlo en cualquier momento. Para obtener más información, consulte Detener un experimento.

Para iniciar un experimento con la consola

1. Abra la consola de la AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. (Opcional) Para generar una vista previa para verificar sus objetivos:
   - Elige Objetivos.
   - Seleccione Generar vista previa.
4. Seleccione la plantilla de experimento y elija Iniciar experimento.
5. (Opcional) Para agregar una etiqueta a su experimento, elija Agregar nueva etiqueta e ingrese una clave y un valor de etiqueta.
6. Elija Start experiment (Iniciar experimento). Cuando se le pida que confirme, ingrese **`start`** y elija Iniciar experimento.

Para iniciar un experimento con la CLI

Utilice el comando start-experiment.

Actualización de una plantilla de experimento

Puede actualizar una plantilla de experimento existente. Al actualizar una plantilla de experimento, los cambios no afectan a ningún experimento en ejecución que utilice la plantilla.

Para actualizar una plantilla de experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Actualizar plantilla de experimento.
4. Modifique los detalles de la plantilla según sea necesario y elija Actualizar plantilla de experimento.

Para actualizar una plantilla de experimento con la CLI

Utilice el comando update-experiment-template.

Etiquetado de plantillas de experimento

Puede aplicar sus propias etiquetas a las plantillas de experimento para ayudarle a organizarlos. También puede implementar políticas de IAM basadas en etiquetas para controlar el acceso a las plantillas de experimento.

Para etiquetar una plantilla de experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla experimento y elija Acciones, Administrar etiquetas.
4. Para agregar una nueva etiqueta, elija Agregar nueva etiqueta y, a continuación, especifique una clave y un valor.

   Para eliminar una etiqueta, elija Eliminar de la etiqueta.
5. Seleccione Guardar.

Para etiquetar una plantilla de experimento con la CLI

Utilice el comando tag-resource.

Eliminación de una plantilla de experimento

Si ya no necesita una plantilla de experimento, puede eliminarla. Al eliminar una plantilla de experimento, no se ve afectado ningún experimento en ejecución que utilice la plantilla. El experimento continúa ejecutándose hasta que se complete o detenga. Sin embargo, las plantillas de experimento que se eliminen no estarán disponibles para su visualización en la página Experimentos de la consola.

Para eliminar una plantilla de experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Eliminar plantilla de experimento.
4. Cuando se le pida confirmación, ingrese **`delete`** y elija Eliminar plantilla de experimento.

Para eliminar una plantilla de experimento con la CLI

Utilice el comando delete-experiment-template.

Ejemplo de plantillas de experimento de AWS FIS

Si utilizas la API AWS FIS o una herramienta de línea de comandos para crear una plantilla de experimento, puedes crear la plantilla en JavaScript Object Notation (JSON). Para obtener más información acerca de los componentes de una plantilla de experimento, consulte Componentes de plantilla.

Para crear un experimento con una de las plantillas de ejemplo, guárdala en un archivo JSON (por ejemplo,`my-template.json`), sustituye los valores de los marcadores de posición en *`cursiva`* por tus propios valores y, a continuación, ejecuta el siguiente comando. create-experiment-template

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

Plantillas de ejemplo

- Detención de instancias EC2 en función de los filtros
- Detención de número específico de instancias EC2
- Ejecución de un documento de SSM de AWS FIS preconfigurado
- Ejecución de un manual de procedimientos de Automation
- Limitación de las acciones de la API en las instancias EC2 con el rol de IAM de destino
- Prueba de esfuerzo de la CPU de los pods de un clúster de Kubernetes

Detención de instancias EC2 en función de los filtros

En el siguiente ejemplo, se detienen todas las instancias de Amazon EC2 de la región especificada con la etiqueta especificada en la VPC especificada. Las reinicia después de dos minutos.

```
{
    "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"
}
```

Detención de número específico de instancias EC2

En el siguiente ejemplo, se detienen tres instancias con la etiqueta especificada. AWS FIS selecciona las instancias específicas para detenerlas al azar. Reinicia estas instancias después de dos minutos.

```
{
    "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"
}
```

Ejecución de un documento de SSM de AWS FIS preconfigurado

En el siguiente ejemplo, se ejecuta una inyección de errores de CPU durante 60 segundos en la instancia EC2 especificada mediante un documento AWS FIS SSM preconfigurado, -Run-CPU-Stress. AWSFIS AWS FIS monitoriza el experimento durante dos minutos.

```
{
```

```
    "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"
}
```

Ejecución de un manual de procedimientos de Automation

En el siguiente ejemplo, se publica una notificación para Amazon SNS con un manual de procedimientos que proporciona Systems Manager, AWS-PublishSNSNotification. El rol debe tener permisos para publicar notificaciones en el tema de SNS especificado.

```
{
    "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"
}
```

Limitación de las acciones de la API en las instancias EC2 con el rol de IAM de destino

En el siguiente ejemplo, se limita el 100 % de las llamadas a las acciones de la API especificadas en las instancias EC2 con el rol de IAM especificado.

```
{
    "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"
}
```

Prueba de esfuerzo de la CPU de los pods de un clúster de Kubernetes

En el siguiente ejemplo, se utiliza Chaos Mesh para realizar una prueba de esfuerzo de la CPU de los pods en un clúster de Kubernetes de Amazon EKS durante un minuto.

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

En el siguiente ejemplo, se utiliza Litmus para realizar una prueba de esfuerzo de la CPU de los pods en un clúster de Kubernetes de Amazon EKS durante un minuto.

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

Experimentos con varias cuentas para AWS FIS

Con un experimento con varias cuentas, puedes configurar y ejecutar escenarios de fallo reales en una aplicación que abarque varias AWS cuentas de una región. Los experimentos con varias cuentas se ejecutan desde una cuenta de orquestador que afecta a los recursos de varias cuentas de destino.

Cuando ejecute un experimento con varias cuentas, se notificará a las cuentas de destino con recursos afectados a través de sus paneles de AWS Health, lo que servirá para informar a los usuarios de las cuentas de destino. En los experimentos con varias cuentas, puede:

- Ejecute situaciones de fallo reales en aplicaciones que abarquen varias cuentas con los controles centrales y las barreras que proporciona. AWS FIS
- Controlar los efectos de un experimento con varias cuentas utilizando roles de IAM con permisos y etiquetas detallados para definir el alcance de cada destino.
- Visualice de forma centralizada las acciones que se llevan a AWS FIS cabo en cada cuenta desde AWS Management Console y hasta AWS FIS los registros.
- Supervise y AWS FIS audite las llamadas a la API realizadas en cada cuenta con AWS CloudTrail.

Esta sección ayuda a familiarizarse con los experimentos de varias cuentas.

Temas

Conceptos para experimentos con varias cuentas

A continuación se indican los conceptos clave de los experimentos con varias cuentas:

Cuenta del orquestador

La cuenta del orquestador actúa como una cuenta central para configurar y administrar el experimento en la AWS FIS consola, así como para centralizar el registro. La cuenta del orquestador es la propietaria de la plantilla del AWS FIS experimento y del experimento.

Cuentas de destino

Una cuenta de destino es una cuenta de AWS individual con recursos que pueden verse afectados por un experimento con AWS FIS varias cuentas.

Configuraciones de cuentas de destino

Para definir las cuentas de destino que forman parte de un experimento, agregue configuraciones de cuenta de destino a la plantilla de experimento. Una configuración de cuentas de destino es un elemento de la plantilla de experimento necesario para los experimentos con varias cuentas. Para definir una para cada cuenta de destino, debe establecer un ID de AWS cuenta, una función de IAM y una descripción opcional.

Requisitos previos de los experimentos con varias cuentas

Para utilizar las condiciones de parada en un experimento con varias cuentas, primero debe configurar las alarmas entre cuentas. Los roles de IAM se definen al crear una plantilla de experimento con varias cuentas. Puede crear los roles de IAM necesarios antes de crear la plantilla.

Contenidos

Permisos para experimentos con varias cuentas

Los experimentos con varias cuentas utilizan el encadenamiento de roles de IAM para conceder permisos para que AWS FIS realice acciones con los recursos de cuentas de destino. Para experimentos con varias cuentas, deberá configurar los roles de IAM en cada cuenta de destino y en la cuenta del orquestador. Estos roles de IAM requieren una relación de confianza entre las cuentas de destino y la cuenta del orquestador, y entre la cuenta del orquestador y AWS FIS.

Los roles de IAM de las cuentas de destino contienen los permisos necesarios para realizar acciones con los recursos y se crean para una plantilla de experimento agregando configuraciones de cuenta de destino. Cree un rol de IAM para la cuenta del orquestador con permiso para asumir los roles de cuentas de destino y establecer una relación de confianza con AWS FIS. Este rol de IAM se utiliza como `roleArn` para la plantilla de experimento.

Para obtener más información sobre el encadenamiento de roles, consulte Términos y conceptos de roles en la Guía del usuario de IAM

En el siguiente ejemplo, configurará permisos para una cuenta del orquestador A para ejecutar un experimento con `aws:ebs:pause-volume-io` en la cuenta de destino B.

1. En la cuenta B, cree un rol de IAM con los permisos necesarios para ejecutar la acción. Para conocer los permisos necesarios para cada acción, consulte the section called “Referencia de las acciones”. El siguiente ejemplo muestra los permisos que concede una cuenta de destino para ejecutar la acción de E/S de volumen de pause de EBS the section called “aws:ebs:pause-volume-io”.

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

2. A continuación, agregue una política de confianza a la cuenta B que cree una relación de confianza con la cuenta A. Elija un nombre para el rol de IAM de la cuenta A, que creará en el paso 3.

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

3. En la cuenta A, cree un rol de IAM. El nombre de este rol debe coincidir con el rol que especificó en la política de confianza del paso 2. Para utilizar varias cuentas, conceda al orquestador permisos para asumir cada rol. El siguiente ejemplo muestra los permisos para que la cuenta A asuma la cuenta B. Si tiene cuentas de destino adicionales, agregará ARN de rol adicionales a esta política. Solo puede tener un ARN de rol por cuenta de destino.

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

4. Este rol de IAM de la cuenta A se utiliza como `roleArn` para la plantilla de experimento. El siguiente ejemplo muestra la política de confianza requerida en el rol de IAM que concede AWS FIS permisos para asumir la cuenta A, la cuenta de orquestador.

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

También puede usar Stacksets para aprovisionar varios roles de IAM al mismo tiempo. Para CloudFormation StackSets utilizarla, tendrás que configurar los StackSet permisos necesarios en tus AWS cuentas. Para obtener más información, consulte Trabajar con AWS CloudFormation StackSets.

Condiciones de detención para experimentos con varias cuentas (opcional)

Una condición de detención es un mecanismo para detener un experimento si alcanza un umbral que se define como alarma. Para configurar una condición de parada para un experimento con varias cuentas puede emplear alarmas entre cuentas. Debe habilitar el uso compartido en cada cuenta de destino para que la alarma esté disponible en la cuenta del orquestador con permisos de solo lectura. Una vez compartidas, puede combinar métricas de diferentes cuentas de destino mediante Metric Math. A continuación, puede agregar esta alarma como condición de parada para el experimento.

Para obtener más información sobre los paneles de control multicuenta, consulte Habilitar la funcionalidad multicuenta en. CloudWatch

Trabajo en experimentos con varias cuentas

Puede crear y gestionar plantillas de experimentos con varias cuentas mediante la AWS FIS consola o la línea de comandos. Para crear un experimento con varias cuentas, especifique la opción de experimento de segmentación de cuentas como `"multi-account"` y agregue configuraciones de cuenta de destino. Después de crear una plantilla de experimento con varias cuentas, puede utilizarla para ejecutar un experimento.

Contenidos

Prácticas recomendadas para experimentos con varias cuentas

A continuación, se indican las prácticas recomendadas para utilizar experimentos con varias cuentas:

- Cuando configura destinos para experimentos con varias cuentas se recomienda segmentar con etiquetas de recursos coherentes en todas las cuentas de destino. Un AWS FIS experimento resolverá los recursos con etiquetas consistentes en cada cuenta de destino. Una acción debe resolver al menos un recurso de destino en cualquier cuenta de destino o fallará, salvo en el caso de los experimentos con `emptyTargetResolutionMode` establecido en `skip`. Las cuotas de acción se aplican por cuenta. Si desea segmentar recursos por ARN de recurso se aplica el mismo límite de cuenta única por acción.
- Cuando segmenta recursos de una o varias zonas de disponibilidad mediante parámetros o filtros, debe especificar un ID de AZ, no un nombre de AZ. El ID de AZ es un identificador único y coherente para una zona de disponibilidad entre cuentas. Para obtener información sobre cómo encontrar el ID de AZ de las zonas de disponibilidad de su cuenta, consulte Availability Zone IDs for your AWS resources.

Creación de una plantilla de experimento con varias cuentas

Para obtener información sobre cómo crear una plantilla de experimento mediante el AWS Management Console

Consulte Creación de una plantilla de experimento.

Para crear una plantilla de experimento con la CLI

1. Abre el AWS Command Line Interface
2. Para crear un experimento a partir de un archivo JSON guardado con la opción de experimento de segmentación de cuentas establecida en `"multi-account"` (por ejemplo,`my-template.json`), sustituye los valores de los marcadores de posición en *`cursiva`* por tus propios valores y, a continuación, ejecuta el siguiente create-experiment-templatecomando.

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

   De este modo se devolverá la plantilla de experimento en la respuesta. Copie el `id` de la respuesta, que es el ID de la plantilla de experimento.
3. Ejecuta el create-target-account-configurationcomando para añadir una configuración de cuenta de destino a la plantilla del experimento. Sustituya los valores de los marcadores de posición en *`cursiva`* por sus propios valores, utilizando el `id` del paso 2 como valor para el parámetro `--experiment-template-id` y, a continuación, ejecute lo siguiente. El parámetro `--description` es opcional. Repita este paso para cada cuenta de destino.

   ```
   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. Ejecute el get-target-account-configurationcomando para recuperar los detalles de una configuración de cuenta de destino específica.

   ```
   aws fis get-target-account-configuration --experiment-template-id EXTxxxxxxxxx --
   account-id 111122223333
   ```
5. Una vez que haya agregado todas las configuraciones de la cuenta de destino, puede ejecutar el list-target-account-configurationscomando para comprobar que se han creado las configuraciones de la cuenta de destino.

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

   También puede comprobar que ha agregado las configuraciones de la cuenta de destino ejecutando el get-experiment-templatecomando. La plantilla devolverá un campo de solo lectura

`targetAccountConfigurationsCount` que ofrece un recuento de todas las configuraciones de cuenta de destino de la plantilla de experimento.

6. Cuando esté listo, puede ejecutar la plantilla de experimento mediante el comando start-experiment.

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

Actualización de una configuración de cuenta de destino

Puede actualizar una configuración de cuenta de destino existente si desea cambiar el ARN del rol o la descripción de la cuenta. Al actualizar una configuración de cuenta de destino, los cambios no afectan a ningún experimento en ejecución que utilice la plantilla.

Para actualizar la configuración de una cuenta de destino mediante el AWS Management Console

1. Abra la AWS FIS consola en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Actualizar plantilla de experimento.
4. Modifique las configuraciones de cuenta de destino y elija Actualizar plantilla de experimento.

Para actualizar una configuración de cuenta de destino utilizando la CLI

Ejecute el update-target-account-configurationcomando a comando y sustituya los valores de los marcadores de posición en *cursiva* por sus propios valores. Los parámetros `--role-arn` y `--description` son opcionales y no se actualizarán si no se incluyen.

```
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"
```

Eliminación de una configuración de cuenta de destino

Si ya no necesita una configuración de cuenta de destino, puede eliminarla. Al eliminar una configuración de cuenta de destino, no se ven afectados los experimentos en ejecución que utilizan la plantilla. El experimento continúa ejecutándose hasta que se complete o detenga.

Para eliminar la configuración de una cuenta de destino mediante el AWS Management Console

1. Abra la AWS FIS consola en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Actualizar.
4. En Configuraciones de cuenta de destino, seleccione Eliminar para el ARN del rol de la cuenta de destino que desee eliminar.

Para eliminar una configuración de cuenta de destino utilizando la CLI

Ejecute el delete-target-account-configurationcomando y sustituya los valores de los marcadores de posición en *cursiva* por sus propios valores.

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

Biblioteca de escenarios de AWS FIS

Los escenarios definen eventos o condiciones que los clientes pueden aplicar para probar la resiliencia de sus aplicaciones, como la interrupción de los recursos informáticos en los que se ejecuta la aplicación. AWS crea y posee los escenarios, que minimizan el trabajo pesado indiferenciado al proporcionar un grupo de destinos y acciones de error predefinidos (por ejemplo, detener el 30 % de las instancias de un grupo de escalado automático) para los problemas comunes de las aplicaciones.

Temas

Trabajo con escenarios de AWS FIS

Los escenarios los proporciona una biblioteca de escenarios solo para consola, que se ejecutan mediante una plantilla de experimento de AWS FIS. Para ejecutar un experimento con un escenario, debe seleccionarlo de la biblioteca, especificar los parámetros que coincidan con los detalles de su carga de trabajo y guardarlo como plantilla de experimento en su cuenta.

Temas

Visualización de un escenario

Para visualizar un escenario con la consola:

1. Abra la AWS FIS consola en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Biblioteca de escenarios.

3. Para ver información sobre un escenario específico, seleccione la tarjeta de escenario para abrir un panel dividido.
   - En la pestaña Descripción del panel dividido en la parte inferior de la página, puede ver una breve descripción del escenario. También encontrará un breve resumen de los requisitos previos que contiene un resumen de los recursos de destino necesarios y de las acciones que debe realizar para preparar los recursos con el fin de utilizarlos en el escenario. Por último, también puede ver información adicional sobre los destinos y las acciones del escenario, así como la duración prevista cuando el experimento se ejecute correctamente con la configuración predeterminada.
   - En la pestaña Contenido del panel dividido en la parte inferior de la página, puede previsualizar una versión parcialmente rellenada de la plantilla de experimento que se creará a partir del escenario.
   - En la pestaña Detalles del panel dividido en la parte inferior de la página, encontrará una explicación detallada de cómo se implementa el escenario. Puede contener información detallada sobre cómo se aproximan los aspectos individuales del escenario. Si procede, también puede obtener información sobre qué métricas utilizar como condiciones de detención y proporcionar observabilidad para aprender del experimento. Por último, encontrará recomendaciones sobre cómo ampliar la plantilla de experimento resultante.

Uso de un escenario

Para usar un escenario con la consola:

1. Abra la AWS FIS consola en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Biblioteca de escenarios.
3. Para ver información sobre un escenario específico, seleccione la tarjeta de escenario para abrir un panel dividido
4. Para usar el escenario, seleccione la tarjeta de escenario y elija Crear plantilla con escenario.
5. En la vista Crear plantilla de experimento, rellene los elementos que falten.
   a. Algunos escenarios permiten editar de forma masiva los parámetros que se comparten entre varias acciones o destinos. Esta funcionalidad se deshabilitará una vez que realice cualquier cambio en el escenario, incluidos los cambios mediante la edición masiva de parámetros. Para utilizar esta característica, seleccione el botón Editar parámetros masivamente. Edite los parámetros en la ventana emergente y seleccione el botón Guardar.

b. Es posible que a algunas plantillas de experimentos les falten parámetros de acción o destino, resaltados en cada tarjeta de acción y destino. Seleccione el botón Editar en cada tarjeta, agregue la información que falta y seleccione el botón Guardar de la tarjeta.

c. Todas las plantillas requieren un rol de ejecución Acceso al servicio. Puede elegir un rol existente o crear uno nuevo para esta plantilla de experimento.

d. Recomendamos definir una o más condiciones de parada opcionales seleccionando una CloudWatch alarma de AWS existente. Obtener más información sobre Condiciones de detención para AWS FIS. Si aún no tienes una alarma configurada, puedes seguir las instrucciones de Uso de Amazon CloudWatch Alarms y actualizar la plantilla del experimento más adelante.

e. Recomendamos habilitar los registros de experimentos opcionales en los CloudWatch registros de Amazon o en un bucket de Amazon S3. Obtener más información sobre Registro de experimentos de AWS FIS. Si aún no ha configurado los recursos adecuados, puede actualizar la plantilla de experimento más adelante.

6. En Crear plantilla de experimento, seleccione Crear plantilla de experimento.
7. En la vista Plantillas de experimento de la consola de AWS FIS, seleccione Iniciar experimento. Obtener más información sobre Experimentos para el FIS AWS.

Exportación de un escenario

Los escenarios son una experiencia solo para consolas. Si bien son similares a las plantillas de experimento, los escenarios no son plantillas de experimento completas y no se pueden importar directamente a AWS FIS. Si desea utilizar los escenarios como parte de su propia automatización, puede utilizar una de estas dos rutas:

1. Siga los pasos que se indican en Uso de un escenario para crear una plantilla de experimento de AWS FIS válida y exportarla.
2. Siga los pasos de Visualización de un escenario y en el paso 3, desde la pestaña Contenido, copie el contenido del escenario y guárdelo, a continuación, agregue manualmente los parámetros que falten para crear una plantilla de experimento válida.

Escenarios de la biblioteca de escenarios de AWS FIS

Los escenarios incluidos en la biblioteca de escenarios están diseñados para usar etiquetas siempre que sea posible y cada escenario describe las etiquetas necesarias en las secciones Requisitos

previos y Cómo funciona de la descripción del escenario. Puede etiquetar sus recursos con esas etiquetas predefinidas o puede establecer sus propias etiquetas mediante la experiencia de edición masiva de parámetros (consulte Uso de un escenario).

Esta referencia describe los escenarios comunes de la biblioteca de escenarios de AWS FIS. También puede enumerar los escenarios admitidos mediante la consola AWS FIS.

Para obtener más información, consulte Trabajar con escenarios.

AWS FIS admite los siguientes escenarios de Amazon EC2. El objetivo de estos escenarios son las instancias que utilizan etiquetas. Puede usar sus propias etiquetas o las etiquetas predeterminadas incluidas en el escenario. Algunos de estos escenarios utilizan documentos de SSM.

- Esfuerzo en EC2: error de instancia: explore el efecto del error de instancia deteniendo una o más instancias EC2.

  Céntrese en las instancias de la región actual que tienen una etiqueta específica adjunta. En este escenario, detendremos esas instancias y las reiniciaremos al final de la duración de la acción, que de forma predeterminada es de 5 minutos.
- Esfuerzo en EC2: disco: explore el impacto del aumento de uso del disco en su aplicación basada en EC2.

  En este escenario, nos centraremos en las instancias EC2 de la región actual que tienen una etiqueta específica adjunta. En este escenario, puede personalizar una cantidad cada vez mayor de uso del disco inyectado en las instancias EC2 de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo del disco.
- Esfuerzo en EC2: CPU: explore el impacto del aumento de la CPU en su aplicación basada en EC2.

  En este escenario, nos centraremos en las instancias EC2 de la región actual que tienen una etiqueta específica adjunta. En este escenario, puede personalizar una cantidad cada vez mayor de esfuerzo de la CPU inyectado en las instancias EC2 de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo de la CPU.
- Esfuerzo en EC2: memoria: explore el impacto del aumento de la utilización de la memoria en su aplicación basada en EC2.

  En este escenario, nos centraremos en las instancias EC2 de la región actual que tienen una etiqueta específica adjunta. En este escenario, puede personalizar una cantidad cada vez mayor

de esfuerzo de la memoria inyectada en las instancias EC2 de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo de la memoria.

- Esfuerzo en EC2: latencia de la red: explore el impacto del aumento de la latencia de la red en su aplicación basada en EC2.

  En este escenario, nos centraremos en las instancias EC2 de la región actual que tienen una etiqueta específica adjunta. En este escenario, puede personalizar una cantidad cada vez mayor de latencia de la red inyectada en las instancias EC2 de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo de la latencia de la red.

AWS FIS admite los siguientes escenarios de Amazon EKS. El objetivo de estos escenarios son los pods de EKS mediante etiquetas de aplicación de Kubernetes. Puede utilizar sus propias etiquetas o las etiquetas predeterminadas incluidas en el escenario. Para obtener más información sobre EKS con FIS, consulte Uso de las acciones de pod de EKS.

- Esfuerzo en EKS: eliminación de pods: explore el efecto del error de pods en EKS eliminando uno o más pods.

  En este escenario, nos centraremos en los pods de la región actual que estén asociados a una etiqueta de aplicación. En este escenario, eliminaremos todos los pods coincidentes. La recreación de los pods se controlará mediante la configuración de Kubernetes.
- Esfuerzo en EKS: CPU: explore el impacto del aumento de la CPU en su aplicación basada en EKS.

  En este escenario, nos centraremos en los pods de la región actual que estén asociados a una etiqueta de aplicación. En este escenario, puede personalizar una cantidad cada vez mayor de esfuerzo de la CPU inyectada en los pods de EKS de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo de la CPU.
- Esfuerzo en EKS: disco: explore el impacto del aumento de la utilización del disco en su aplicación basada en EKS.

  En este escenario, nos centraremos en los pods de la región actual que estén asociados a una etiqueta de aplicación. En este escenario, puede personalizar una cantidad cada vez mayor de esfuerzo del disco inyectado en los pods de EKS de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo del disco.
- Esfuerzo en EKS: memoria: explore el impacto del aumento de la utilización de la memoria en su aplicación basada en EKS.

En este escenario, nos centraremos en los pods de la región actual que estén asociados a una etiqueta de aplicación. En este escenario, puede personalizar una cantidad cada vez mayor de esfuerzo de la memoria inyectada en los pods de EKS de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo de la memoria.

- Esfuerzo en EKS: latencia de la red: explore el impacto del aumento de la latencia de la red en su aplicación basada en EKS.

  En este escenario, nos centraremos en los pods de la región actual que estén asociados a una etiqueta de aplicación. En este escenario, puede personalizar una cantidad cada vez mayor de latencia de la red inyectada en los pods de EKS de destino durante la acción, que de forma predeterminada es de 5 minutos por cada acción de esfuerzo de la latencia de la red.

AWS FIS admite los siguientes escenarios para aplicaciones de varias zonas de disponibilidad (AZ) y de varias regiones. El objetivo de estos escenarios son varios tipos de recursos.

- AZ Availability: Power Interruption: inyecte los síntomas esperados de una interrupción total del suministro eléctrico en una zona de disponibilidad (AZ). Obtener más información sobre AZ Availability: Power Interruption.
- Cross-Region: Connectivity: bloquee el tráfico de red de aplicaciones desde la región del experimento a la región de destino y detenga la replicación de datos entre regiones. Obtenga más información sobre el uso de Cross-Region: Connectivity.

AZ Availability: Power Interruption

Puede utilizar el escenario AZ Availability: Power Interruption para provocar los síntomas esperados de una interrupción total del suministro eléctrico en una zona de disponibilidad (AZ).

Este escenario se puede utilizar para demostrar que las aplicaciones con varias AZ funcionan según lo esperado en una interrupción completa del suministro eléctrico en zonas de disponibilidad. Incluye la pérdida de procesamiento zonal (Amazon EC2, EKS y ECS), la falta de cambio de escala del procesamiento en la zona de disponibilidad, la pérdida de conectividad de subred, la conmutación por error de RDS, la conmutación por error ElastiCache y los volúmenes de EBS que no responden. De forma predeterminada, se omiten las acciones para las que no se encuentre ningún objetivo.

Acciones

En conjunto, las siguientes acciones crean muchos de los síntomas esperados en una interrupción total del suministro eléctrico en una única AZ. Disponibilidad AZ: la interrupción del suministro eléctrico solo afecta a los servicios que se espera que se vean afectados en una interrupción del suministro AZ. De forma predeterminada, el escenario inyecta síntomas de interrupción del suministro eléctrico durante 30 minutos y, a continuación, durante otros 30 minutos, inyecta síntomas que pueden presentarse durante la recuperación.

Stop-Instances

En una interrupción del suministro eléctrico AZ, las instancias de EC2 de la AZ afectada se cerrarán. Una vez restablecido el suministro, las instancias se reiniciarán. AZ Availability: Power Interruption incluye aws:ec2:stop-instances para detener todas las instancias de la AZ afectada durante el tiempo que dure la interrupción. Transcurrido ese tiempo, las instancias se reinician. Al detener las instancias de EC2 administradas por Amazon EKS, se eliminan los pods de EKS dependientes. Al detener las instancias de EC2 administradas por Amazon ECS, se detienen las tareas de ECS dependientes.

El objetivo de esta acción son las instancias de EC2 que se ejecutan en la AZ afectada. De forma predeterminada, su objetivo son las instancias con una etiqueta `AzImpairmentPower` que tiene un valor de `StopInstances`. Puede agregar esta etiqueta a sus instancias o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran instancias válidas, se omitirá esta acción.

Stop-ASG-Instances

En una interrupción del suministro eléctrico AZ, las instancias de EC2 administradas por un grupo de escalado automático de la AZ afectada se cerrarán. Una vez restablecido el suministro, las instancias se reiniciarán. AZ Availability: Power Interruption incluye aws:ec2:stop-instances para detener todas las instancias, incluidas las administradas por escalado automático, de la AZ afectada durante el tiempo que dure la interrupción. Transcurrido ese tiempo, las instancias se reinician.

El objetivo de esta acción son las instancias de EC2 que se ejecutan en la AZ afectada. De forma predeterminada, su objetivo son las instancias con una etiqueta `AzImpairmentPower` que tiene un valor de `IceAsg`. Puede agregar esta etiqueta a sus instancias o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran instancias válidas, se omitirá esta acción.

Pausar lanzamientos de instancias

Durante una interrupción del suministro eléctrico AZ se producirán errores en las llamadas a la API de EC2 para aprovisionar capacidad en la AZ. En concreto, se verán afectadas las siguientes API:, y. `ec2:StartInstances` `ec2:CreateFleet` `ec2:RunInstances` AZ Availability: Power Interruption includesincluye aws:ec2: api-insufficient-instance-capacity -error para evitar que se aprovisionen nuevas instancias en la zona de disponibilidad afectada.

El objetivo de esta acción son los roles de IAM que se utilizan para aprovisionar instancias. Debe hacerse referencia a estos mediante un ARN. De forma predeterminada, si no se encuentran roles de IAM válidos, se omitirá esta acción.

Pausar el escalado de ASG

Durante una interrupción del suministro eléctrico AZ fallarán las llamadas a la API de EC2 realizadas por el plano de control de escalado automático para recuperar la capacidad perdida en la AZ. En concreto, se verán afectadas las siguientes API:, y. `ec2:StartInstances` `ec2:CreateFleet` `ec2:RunInstances` AZ Availability: Power Interruptionincluye aws:ec2: asg-insufficient-instance-capacity -error para evitar que se aprovisionen nuevas instancias en la zona de disponibilidad afectada. Esto también impide que Amazon EKS y Amazon ECS se escalen en la AZ afectada.

El objetivo de esta acción son los grupos de escalado automático. De forma predeterminada, su objetivo son los grupos de escalado automático con una etiqueta `AzImpairmentPower` que tiene un valor de `IceAsg`. Puede agregar esta etiqueta a sus grupos de escalado automático o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran grupos de escalado automático válidos, se omitirá esta acción.

Pausar conectividad de red

Durante una interrupción del suministro eléctrico AZ, la red no estará disponible en la AZ. Cuando esto ocurre, algunos servicios de AWS pueden tardar unos minutos en actualizar DNS para reflejar que los puntos de conexión privados de la AZ afectada no están disponibles. En este tiempo es posible que las búsquedas de DNS devuelvan direcciones IP inaccesibles. AZ Availability: Power Interruption incluye aws:network:disrupt-connectivity para bloquear toda la conectividad de red de todas las subredes de la AZ afectada durante 2 minutos. Esto provocará tiempos de espera y actualizaciones de DNS para la mayoría de las aplicaciones. Si se pone fin a la acción transcurridos 2 minutos, se podrá recuperar posteriormente el DNS de servicio regional mientras la AZ sigue sin estar disponible.

El objetivo de esta acción son las subredes. De forma predeterminada, su objetivo son los clústeres con una etiqueta `AzImpairmentPower` que tiene un valor de `DisruptSubnet`. Puede agregar esta etiqueta a sus subredes o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran subredes válidas, se omitirá esta acción.

RDS de conmutación por error

En una interrupción del suministro eléctrico AZ, los nodos RDS de la AZ afectada se cerrarán. Los nodos AZ RDS únicos de la AZ afectada no estarán disponibles por completo. En el caso de clústeres con varias AZ, el nodo escritor realizará una conmutación por error a una AZ no afectada y los nodos lectores de la AZ afectada no estarán disponibles. Para los clústeres con varias zonas de disponibilidad, AZ Availability: Power Interruption incluye aws:rds: para realizar la conmutación por error si el grabador se encuentra en failover-db-cluster la zona de disponibilidad afectada.

El objetivo de esta acción son los clústeres de RDS. De forma predeterminada, su objetivo son los clústeres con una etiqueta `AzImpairmentPower` que tiene un valor de `DisruptRds`. Puede agregar esta etiqueta a sus clústeres o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran clústeres válidos, se omitirá esta acción.

ElastiCache Pausa Redis

Durante una interrupción del suministro eléctrico en la AZ, ElastiCache los nodos de la AZ no están disponibles. AZ Availability: Power Interruptionincluye aws:elasticache: interrupt-cluster-az-power para terminar ElastiCache los nodos de la AZ afectada. Mientras dure la interrupción, no se aprovisionarán nuevas instancias en la AZ afectada, por lo que el clúster se mantendrá con capacidad reducida.

Esta acción se dirige a los clústeres. ElastiCache De forma predeterminada, su objetivo son los clústeres con una etiqueta `AzImpairmentPower` que tiene un valor de `ElasticacheImpact`. Puede agregar esta etiqueta a sus clústeres o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran clústeres válidos, se omitirá esta acción. Tenga en cuenta que solo los clústeres con nodos escritores en la AZ afectada se considerarán objetivos válidos.

Pausar E/S de EBS

Tras una interrupción del suministro eléctrico AZ, una vez restablecido el suministro, es posible que un porcentaje muy pequeño de instancias experimente volúmenes de EBS que no responden. AZ

Availability: Power Interruption incluye aws:ebs:pause-io para dejar 1 volumen de EBS estado de no respuesta.

De forma predeterminada, solo se seleccionan los volúmenes configurados para persistir una vez terminada la instancia. El objetivo de esta acción son los volúmenes con una etiqueta `AzImpairmentPower` que tiene un valor de `APIPauseVolume`. Puede agregar esta etiqueta a sus volúmenes o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran volúmenes válidos, se omitirá esta acción.

Limitaciones

- Este escenario no incluye condiciones de parada. Deben agregarse a la plantilla de experimento las condiciones de parada correctas para su aplicación.
- No se admiten pods de Amazon EKS que se ejecutan en AWS Fargate.
- No se admiten tareas de Amazon ECS que se ejecutan en AWS Fargate.
- No se admite Amazon RDS Multi-AZ con dos instancias de base de datos en espera legibles. En este caso, las instancias se terminarán, RDS realizará conmutación por error y la capacidad se volverá a aprovisionar inmediatamente en la AZ afectada. La espera legible en la AZ afectada seguirá disponible.

Requisitos

- Agregue el permiso necesario al rol de experimento de AWS FIS.
- Se deben aplicar etiquetas de recursos a recursos que no son el objetivo del experimento. Pueden ser etiquetas que usen su propia convención de etiquetado o etiquetas predeterminadas definidas en el escenario.

Permisos

La siguiente política otorga a AWS FIS los permisos necesarios para ejecutar un experimento con el escenario AZ Availability: Power Interruption. Esta política debe estar asociada al rol de experimento.

```
{
    "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": [
```

```
                "*"
            ]
        }
    ]
}
```

Contenido del escenario

El siguiente contenido define el escenario. Este JSON se puede guardar y utilizar para crear una plantilla de experimento mediante el create-experiment-templatecomando de la interfaz de línea de comandos de AWS (AWS CLI). Para ver la versión más reciente del escenario, visite la biblioteca de escenarios de la consola de FIS.

```
{
    "targets": {
        "IAM-role": {
            "resourceType": "aws:iam:role",
            "resourceArns": [],
            "selectionMode": "ALL"
        },
        "EBS-Volumes": {
            "resourceType": "aws:ec2:ebs-volume",
            "resourceTags": {
                "AzImpairmentPower": "ApiPauseVolume"
            },
            "selectionMode": "COUNT(1)",
            "parameters": {
                "availabilityZoneIdentifier": "us-east-1a"
            },
            "filters": [
                {
                    "path": "Attachments.DeleteOnTermination",
                    "values": [
                        "false"
                    ]
                }
            ]
        },
        "EC2-Instances": {
            "resourceType": "aws:ec2:instance",
            "resourceTags": {
                "AzImpairmentPower": "StopInstances"
```

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

```
        },
        "Subnet": {
            "resourceType": "aws:ec2:subnet",
            "resourceTags": {
                "AzImpairmentPower": "DisruptSubnet"
            },
            "filters": [
                {
                    "path": "AvailabilityZone",
                    "values": [
                        "us-east-1a"
                    ]
                }
            ],
            "selectionMode": "ALL",
            "parameters": {}
        },
        "RDS-Cluster": {
            "resourceType": "aws:rds:cluster",
            "resourceTags": {
                "AzImpairmentPower": "DisruptRds"
            },
            "selectionMode": "ALL",
            "parameters": {
                "writerAvailabilityZoneIdentifiers": "us-east-1a"
            }
        },
        "ElastiCache-Cluster": {
            "resourceType": "aws:elasticache:redis-replicationgroup",
            "resourceTags": {
                "AzImpairmentPower": "DisruptElasticache"
            },
            "selectionMode": "ALL",
            "parameters": {
                "availabilityZoneIdentifier": "us-east-1a"
            }
        }
    },
    "actions": {
        "Pause-Instance-Launches": {
            "actionId": "aws:ec2:api-insufficient-instance-capacity-error",
            "parameters": {
                "availabilityZoneIdentifiers": "us-east-1a",
                "duration": "PT30M",
```

```
                "percentage": "100"
            },
            "targets": {
                "Roles": "IAM-role"
            }
        },
        "Pause-EBS-IO": {
            "actionId": "aws:ebs:pause-volume-io",
            "parameters": {
                "duration": "PT30M"
            },
            "targets": {
                "Volumes": "EBS-Volumes"
            },
            "startAfter": [
                "Stop-Instances",
                "Stop-ASG-Instances"
            ]
        },
        "Stop-Instances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "completeIfInstancesTerminated": "true",
                "startInstancesAfterDuration": "PT30M"
            },
            "targets": {
                "Instances": "EC2-Instances"
            }
        },
        "Pause-ASG-Scaling": {
            "actionId": "aws:ec2:asg-insufficient-instance-capacity-error",
            "parameters": {
                "availabilityZoneIdentifiers": "us-east-1a",
                "duration": "PT30M",
                "percentage": "100"
            },
            "targets": {
                "AutoScalingGroups": "ASG"
            }
        },
        "Stop-ASG-Instances": {
            "actionId": "aws:ec2:stop-instances",
            "parameters": {
                "completeIfInstancesTerminated": "true",
```

```
                "startInstancesAfterDuration": "PT30M"
            },
            "targets": {
                "Instances": "ASG-EC2-Instances"
            }
        },
        "Pause-network-connectivity": {
            "actionId": "aws:network:disrupt-connectivity",
            "parameters": {
                "duration": "PT2M",
                "scope": "all"
            },
            "targets": {
                "Subnets": "Subnet"
            }
        },
        "Failover-RDS": {
            "actionId": "aws:rds:failover-db-cluster",
            "parameters": {},
            "targets": {
                "Clusters": "RDS-Cluster"
            }
        },
        "Pause-ElastiCache": {
            "actionId": "aws:elasticache:interrupt-cluster-az-power",
            "parameters": {
                "duration": "PT30M"
            },
            "targets": {
                "ReplicationGroups": "ElastiCache-Cluster"
            }
        }
    },
    "stopConditions": [
        {
            "source": "aws:cloudwatch:alarm",
            "value": ""
        }
    ],
    "roleArn": "",
    "tags": {
        "Name": "AZ Impairment: Power Interruption"
    },
    "logConfiguration": {
```

```
        "logSchemaVersion": 2
    },
    "experimentOptions": {
        "accountTargeting": "single-account",
        "emptyTargetResolutionMode": "skip"
    },
    "description": "Affect multiple resource types in a single AZ, targeting by tags
 and explicit ARNs, to approximate power interruption in one AZ."
}
```

Cross-Region: Connectivity

Puede utilizar el escenario Cross-Region: Connectivity para bloquear el tráfico de red de las aplicaciones desde la región del experimento a la región de destino y pausar la replicación entre regiones para Amazon S3 y Amazon DynamoDB. Entre regiones: la conectividad afecta al tráfico saliente de las aplicaciones desde la región en la que se ejecuta el experimento (región del experimento). No se puede bloquear el tráfico entrante sin estado procedente de la región que desea aislar de la región del experimento (región de destino). No se puede bloquear el tráfico de los servicios gestionados de AWS.

Este escenario se puede utilizar para demostrar que las aplicaciones con varias regiones funcionan según lo esperado cuando no se puede acceder a los recursos de la región de destino desde la región del experimento. Esto incluye bloquear tráfico de red desde la región del experimento hasta la región de destino centrándose en las puertas de enlace y las tablas de enrutamiento. También pausar la replicación entre regiones para S3 y DynamoDB. De forma predeterminada, se omiten las acciones para las que no se encuentre ningún objetivo.

Acciones

En conjunto, las siguientes acciones bloquean la conectividad entre regiones para los servicios de AWS incluidos. Las acciones se ejecutan en paralelo. De forma predeterminada, el escenario bloquea el tráfico durante 3 horas, aunque puede ampliarlo hasta un máximo de 12 horas.

Interrumpir la conectividad de puerta de enlace de tránsito

Cross Region: Connectivityincluye aws:network: transit-gateway-disrupt-cross -region-connectivity para bloquear el tráfico de red entre regiones desde las VPC de la región del experimento hacia las VPC de la región de destino conectadas mediante una pasarela de tránsito. Esto no afecta al acceso

a los puntos de conexión de VPC en la región del experimento, pero bloqueará el tráfico de la región del experimento destinado a un punto de conexión de VPC en la región de destino.

El objetivo de esta acción son las puertas de enlace de tránsito que conectan la región del experimento y la región de destino. De forma predeterminada, su objetivo son las puertas de enlace con una etiqueta `DisruptTransitGateway`que tiene un valor de `Allowed`. Puede agregar esta etiqueta a sus puertas de enlace de tránsito o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran puertas de enlace de tránsito válidas, se omitirá esta acción.

Interrumpir la conectividad de subred

Cross Region: Connectivityincluye aws:network: route-table-disrupt-cross -region-connectivity para bloquear el tráfico de red entre regiones desde las VPC de la región del experimento hasta los bloques de IP públicos de AWS en la región de destino. Estos bloques de IP públicos incluyen puntos de conexión de servicios de AWS en la región de destino, por ejemplo, el punto de conexión regional S3, y los bloques de IP de AWS para servicios gestionados, por ejemplo, las direcciones IP utilizadas para equilibradores de carga y Amazon API Gateway. Esta acción también bloquea la conectividad de red a través de conexiones de emparejamiento de VPC entre regiones desde la región del experimento hasta la región de destino. Esto no afecta al acceso a los puntos de conexión de VPC en la región del experimento, pero bloqueará el tráfico de la región del experimento destinado a un punto de conexión de VPC en la región de destino.

El objetivo de esta acción son las subredes de la región del experimento. De forma predeterminada, su objetivo son las subredes con una etiqueta `DisruptSubnet` que tiene un valor de `Allowed`. Puede agregar esta etiqueta a sus subredes o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran subredes válidas, se omitirá esta acción.

Pausar replicación de S3

Cross Region: Connectivityincluye aws:s3: bucket-pause-replication para pausar la replicación de S3 desde la región del experimento hasta la región de destino para los segmentos de destino. La replicación desde la región de destino hasta la región del experimento no se verá afectada. Cuando finalice el escenario, la replicación del bucket se reanudará desde el punto en que se quedó en pausa. Tenga en cuenta que el tiempo que tarda la replicación en mantener todos los objetos sincronizados variará en función de la duración del experimento y de la velocidad a la que se carguen los objetos al bucket.

El objetivo de esta acción son los buckets de S3 de la región del experimento con la Replicación entre regiones (CRR) habilitada en un bucket de S3 de la región de destino. De forma predeterminada, su objetivo son los buckets con una etiqueta `DisruptS3` que tiene un valor de `Allowed`. Puede agregar esta etiqueta a sus buckets o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran buckets válidos, se omitirá esta acción.

Pausar la replicación de DynamoDB

Cross-Region: Connectivityincluye aws:dynamodb: encrypted-global-table-pause -replication para detener la replicación entre la región del experimento y todas las demás regiones, incluida la región de destino. Esto impide la replicación con origen y destino en la región del experimento, pero no afecta a la replicación entre otras regiones. Cuando finalice el escenario, la replicación de la tabla se reanudará desde el punto en que se quedó en pausa. Tenga en cuenta que el tiempo que tarda la replicación en mantener todos los datos sincronizados variará en función de la duración del experimento y de la velocidad de los cambios en la tabla.

El objetivo de esta acción son las tablas globales de DynamoDB cifradas en la región del experimento que están cifradas con claves administradas por el cliente. De forma predeterminada, su objetivo son las tablas con una etiqueta `DisruptDynamoDb` que tiene un valor de `Allowed`. Puede agregar esta etiqueta a sus tablas o reemplazar la etiqueta predeterminada por la suya propia en la plantilla de experimento. De forma predeterminada, si no se encuentran tablas globales válidas, se omitirá esta acción.

Limitaciones

- Este escenario no incluye condiciones de parada. Deben agregarse a la plantilla de experimento las condiciones de parada correctas para su aplicación.

Requisitos

- Agregue el permiso necesario al rol de experimento de AWS FIS.
- Se deben aplicar etiquetas de recursos a recursos que no son el objetivo del experimento. Pueden ser etiquetas que usen su propia convención de etiquetado o etiquetas predeterminadas definidas en el escenario.

Permisos

La siguiente política otorga a AWS FIS los permisos necesarios para ejecutar un experimento con el escenario Cross-Region: Connectivity. Esta política debe estar asociada al rol de experimento.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RouteTableDisruptConnectivity1",
            "Effect": "Allow",
            "Action": "ec2:CreateRouteTable",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity2",
            "Effect": "Allow",
            "Action": "ec2:CreateRouteTable",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity21",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateRouteTable",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity3",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
```

```
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity4",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateManagedPrefixList",
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity5",
            "Effect": "Allow",
            "Action": "ec2:DeleteRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*",
                "arn:aws:ec2:*:*:vpc/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity6",
            "Effect": "Allow",
            "Action": "ec2:CreateRoute",
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
```

```
            "Sid": "RouteTableDisruptConnectivity7",
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity8",
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity9",
            "Effect": "Allow",
            "Action": "ec2:DeleteNetworkInterface",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity10",
            "Effect": "Allow",
            "Action": "ec2:CreateManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity11",
            "Effect": "Allow",
```

```
            "Action": "ec2:DeleteManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity12",
            "Effect": "Allow",
            "Action": "ec2:ModifyManagedPrefixList",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity13",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeSubnets",
                "ec2:DescribeRouteTables",
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity14",
            "Effect": "Allow",
            "Action": "ec2:ReplaceRouteTableAssociation",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:route-table/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity15",
```

```
            "Effect": "Allow",
            "Action": "ec2:GetManagedPrefixListEntries",
            "Resource": "arn:aws:ec2:*:*:prefix-list/*"
        },
        {
            "Sid": "RouteTableDisruptConnectivity16",
            "Effect": "Allow",
            "Action": "ec2:AssociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:route-table/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity17",
            "Effect": "Allow",
            "Action": "ec2:DisassociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity18",
            "Effect": "Allow",
            "Action": "ec2:DisassociateRouteTable",
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*"
            ]
        },
        {
            "Sid": "RouteTableDisruptConnectivity19",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/managedByFIS": "true"
```

```
                }
            }
        },
        {
            "Sid": "RouteTableDisruptConnectivity20",
            "Effect": "Allow",
            "Action": "ec2:ModifyVpcEndpoint",
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity1",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateTransitGatewayRouteTable",
                "ec2:AssociateTransitGatewayRouteTable"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:transit-gateway-route-table/*",
                "arn:aws:ec2:*:*:transit-gateway-attachment/*"
            ]
        },
        {
            "Sid": "TransitGatewayDisruptConnectivity2",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTransitGatewayPeeringAttachments",
                "ec2:DescribeTransitGatewayAttachments",
                "ec2:DescribeTransitGateways"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion1",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion2",
            "Effect": "Allow",
```

```
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "S3CrossRegion3",
            "Effect": "Allow",
            "Action": [
                "s3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "StringLike": {
                    "s3:DestinationRegion": "*"
                }
            }
        },
        {
            "Sid": "S3CrossRegion4",
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:PutReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::*",
            "Condition": {
                "BoolIfExists": {
                    "s3:isReplicationPauseRequest": "true"
                }
            }
        },
        {
            "Sid": "DdbCrossRegion1",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DdbCrossRegion2",
            "Effect": "Allow",
            "Action": [
```

```
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:*:*:table/*",
                "arn:aws:dynamodb:*:*:global-table/*"
            ]
        },
        {
            "Sid": "DdbCrossRegion3",
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey",
                "kms:GetKeyPolicy",
                "kms:PutKeyPolicy"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        }
    ]
}
```

Contenido del escenario

El siguiente contenido define el escenario. Este JSON se puede guardar y utilizar para crear una plantilla de experimento mediante el create-experiment-templatecomando de la interfaz de línea de comandos de AWS (AWS CLI). Para ver la versión más reciente del escenario, visite la biblioteca de escenarios de la consola de FIS.

```
{
        "targets": {
                "Transit-Gateway": {
                        "resourceType": "aws:ec2:transit-gateway",
                        "resourceTags": {
                                "TgwTag": "TgwValue"
                        },
                        "selectionMode": "ALL"
                },
                "Subnet": {
                        "resourceType": "aws:ec2:subnet",
                        "resourceTags": {
                                "SubnetKey": "SubnetValue"
                        },
```

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

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

Experimentos para el FIS AWS

AWS El FIS le permite realizar experimentos de inyección de fallos en sus cargas de AWS trabajo. Para empezar, cree una plantilla de experimento. Después de crear una plantilla de experimento, puede utilizarla para iniciar un experimento.

Un experimento finaliza cuando se produce alguna de las siguientes situaciones:

- Todas las acciones de la plantilla se han completado correctamente.
- Se activa una condición de detención.
- No se puede completar una acción debido a un error. Por ejemplo, si no se encuentra el destino.
- El experimento se detiene manualmente.

No se puede reanudar un experimento detenido o fallido. Tampoco puede volver a ejecutar un experimento completado. Sin embargo, puede iniciar un experimento nuevo a partir de la misma plantilla de experimento. También puede actualizar la plantilla de experimento antes de volver a especificarla en un experimento nuevo.

Tareas

- Inicio de un experimento
- Visualización de sus experimentos
- Etiquetado de un experimento
- Detener un experimento
- Mostrar objetivos resueltos

Inicio de un experimento

El experimento se inicia a partir de una plantilla de experimento. Para obtener más información, consulte Inicio de un experimento a partir de una plantilla.

Puede programar sus experimentos como tarea única o como tareas recurrentes con Amazon EventBridge. Para obtener más información, consulte Tutorial: Programación de un experimento recurrente.

Puede monitorizar el experimento con alguna de las características siguientes:

- Vea sus experimentos en la consola AWS FIS. Para obtener más información, consulte Visualización de sus experimentos.
- Consulta CloudWatch las métricas de Amazon de los recursos objetivo en tus experimentos o consulta las métricas de uso del AWS FIS. Para obtener más información, consulte Monitoreo con CloudWatch.
- Habilite el registro de experimentos para capturar información detallada sobre su experimento a medida que se ejecuta. Para más información, consulte Registro de experimentos.

Visualización de sus experimentos

Puede comprobar el progreso de un experimento en ejecución y ver los experimentos que se han completado, se han detenido o han fallado.

Los experimentos detenidos, completados o fallidos se eliminan automáticamente de su cuenta después de 120 días.

Para consultar las métricas con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Experimentos.
3. Elija ID del experimento del experimento para abrir su página de detalles.
4. Realice una o más de las siguientes acciones:

    - Consulte Detalles, Estado para ver el estado del experimento.
    - Seleccione la pestaña Acciones para obtener información sobre las acciones del experimento.
    - Seleccione la pestaña Destinos para obtener información sobre los destinos del experimento.
    - Seleccione la pestaña Línea temporal para obtener una representación visual de las acciones en función de sus horas de inicio y finalización.

Para consultar las métricas con la CLI

Utilice el comando list-experiments para obtener una lista de experimentos, y el comando get-experiment para obtener información sobre un experimento específico.

Estados de experimento

Un experimento puede tener uno de los siguientes estados:

- pendiente: el experimento está pendiente.
- iniciando: el experimento se está preparando para comenzar.
- en ejecución: el experimento se está ejecutando.
- completado: todas las acciones del experimento se han completado correctamente.
- deteniéndose: la condición de detención se activó o el experimento se detuvo manualmente.
- detenido: se detienen todas las acciones pendientes o en ejecución del experimento.
- error: el experimento falló debido a un error, como permisos insuficientes o sintaxis incorrecta.

Estados de acción

Una acción puede tener uno de los siguientes estados:

- pendiente: la acción está pendiente, ya sea porque el experimento no se ha iniciado o porque la acción se iniciará más adelante en el experimento.
- iniciando: la acción se está preparando para comenzar.
- en ejecución: la acción se está ejecutando.
- completada: la acción se ha completado correctamente.
- cancelada: el experimento se detuvo antes de que comenzara la acción.
- omitida: se ha omitido la acción.
- deteniéndose: la acción se está deteniendo.
- detenida: se detienen todas las acciones pendientes o en ejecución del experimento.
- error: la acción falló debido a un error de cliente, como permisos insuficientes o sintaxis incorrecta.

Etiquetado de un experimento

Puede aplicar etiquetas a los experimentos para ayudarle a organizarlos. También puede implementar políticas de IAM basadas en etiquetas para controlar el acceso a los experimentos.

Para etiquetar un experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Experimentos.
3. Seleccione el experimento y elija Acciones, Administrar etiquetas.

4. Para agregar una nueva etiqueta, elija Agregar nueva etiqueta y especifique una clave y un valor.

   Para eliminar una etiqueta, elija Eliminar de la etiqueta.
5. Seleccione Guardar.

Para etiquetar un experimento con la CLI

Utilice el comando tag-resource.

Detener un experimento

Puede detener un experimento en ejecución en cualquier momento. Al detener un experimento, cualquier acción posterior que no se haya completado para una acción se completará antes de que se detenga el experimento. No se puede reanudar un experimento detenido.

Para detener un experimento con la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Experimentos.
3. Seleccione el experimento y elija Detener experimento.
4. En el cuadro de diálogo de confirmación, elija Detener experimento.

Para detener un experimento con la CLI

Utilice el comando stop-experiment.

Mostrar objetivos resueltos

Puede ver la información de los objetivos resueltos de un experimento una vez finalizada la resolución del objetivo.

Para ver objetivos resueltos mediante la consola

1. Abra la consola AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Experimentos.
3. Seleccione el experimento y elija Informe.

4. Consulte la información de objetivos resueltos en Recursos.

Para ver objetivos resueltos mediante la CLI

Utilice el comando list-experiment-resolved-targets.

Programador de experimentos

Con AWS Fault Injection Service (FIS), puede realizar experimentos de inyección de errores en sus cargas de trabajo de AWS. Estos experimentos se ejecutan en plantillas que contienen una o más acciones para ejecutarse en destinos específicos. Ahora puede programar sus experimentos como tarea única o como tareas recurrentes de forma nativa desde la consola de FIS. Además de las reglas programadas, FIS ahora ofrece una nueva capacidad de programación. FIS ahora se integra con EventBridge Scheduler y crea reglas en su nombre. EventBridge Scheduler es un programador sin servidor que le permite crear, ejecutar y gestionar tareas desde un servicio gestionado centralizado.

> **Important**
>
> El programador de experimentos con no AWS Fault Injection Service está disponible en AWS GovCloud (EE. UU. Este) ni en AWS GovCloud (EE. UU. Oeste).

Temas

Introducción

Una función de ejecución es una función de IAM que AWS Fault Injection Service asume para interactuar con el programador y para que el EventBridge programador de Event Bridge inicie el experimento FIS. Debe adjuntar políticas de permisos a esta función para que el EventBridge programador pueda acceder a FIS Experiment. Los siguientes pasos describen cómo crear un nuevo rol de ejecución y una política que permita EventBridge iniciar un experimento.

Creación de un rol con AWS CLI

Este es el rol de IAM necesario para que Event Bridge pueda programar un experimento en nombre del cliente.

1. Copie la siguiente política JSON de rol de asunción y guárdela localmente como `fis-execution-role.json`. Esta política de confianza permite a EventBridge Scheduler asumir la función en tu nombre.

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

2. Desde la Interfaz de la línea de comandos de AWS (AWS CLI), introduzca el siguiente comando para crear un rol nuevo. Sustituya `FisSchedulerExecutionRole` por el nombre que desee asignar a este rol.

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

   Si todo va bien, obtendrá el siguiente resultado:

```
{
    "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. Para crear una nueva política que permita a EventBridge Scheduler invocar el experimento, copia el siguiente JSON y guárdalo localmente como. `fis-start-experiment-permissions.json` La siguiente política permite a EventBridge Scheduler ejecutar la `fis:StartExperiment` acción en todas las plantillas de experimentos de tu cuenta. Sustituya el `*` que aparece al final de `"arn:aws:fis:*:*:experiment-template/*"` por el ID de la plantilla de experimento si quiere limitar el rol a una sola plantilla de experimento.

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

4. Ejecute el siguiente comando para crear la nueva política de permisos. Sustituya `FisSchedulerPolicy` por el nombre que desee asignar a esta política.

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

Si todo va bien, obtendrá el siguiente resultado. Anote el ARN de la política. Utilice este ARN en el siguiente paso para asociar la política a nuestro rol de ejecución.

```
{
    "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. Ejecute el siguiente comando para adjuntar la política a su rol de ejecución. Sustituya `your-policy-arn` por el ARN de la política que creó en el paso anterior. Sustituya `FisSchedulerExecutionRole` por el nombre de su rol de ejecución.

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

   La operación `attach-role-policy` no devuelve una respuesta en la línea de comandos.

6. Puede restringir el programador para que solo ejecute experimentos de AWS FIS que tengan un valor de etiqueta específico. Por ejemplo, la siguiente política concede el permiso `fis:StartExperiment` para todas las plantillas de experimento de AWS FIS, pero restringe al programador a ejecutar únicamente los experimentos que estén etiquetados como `Purpose=Schedule`.

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

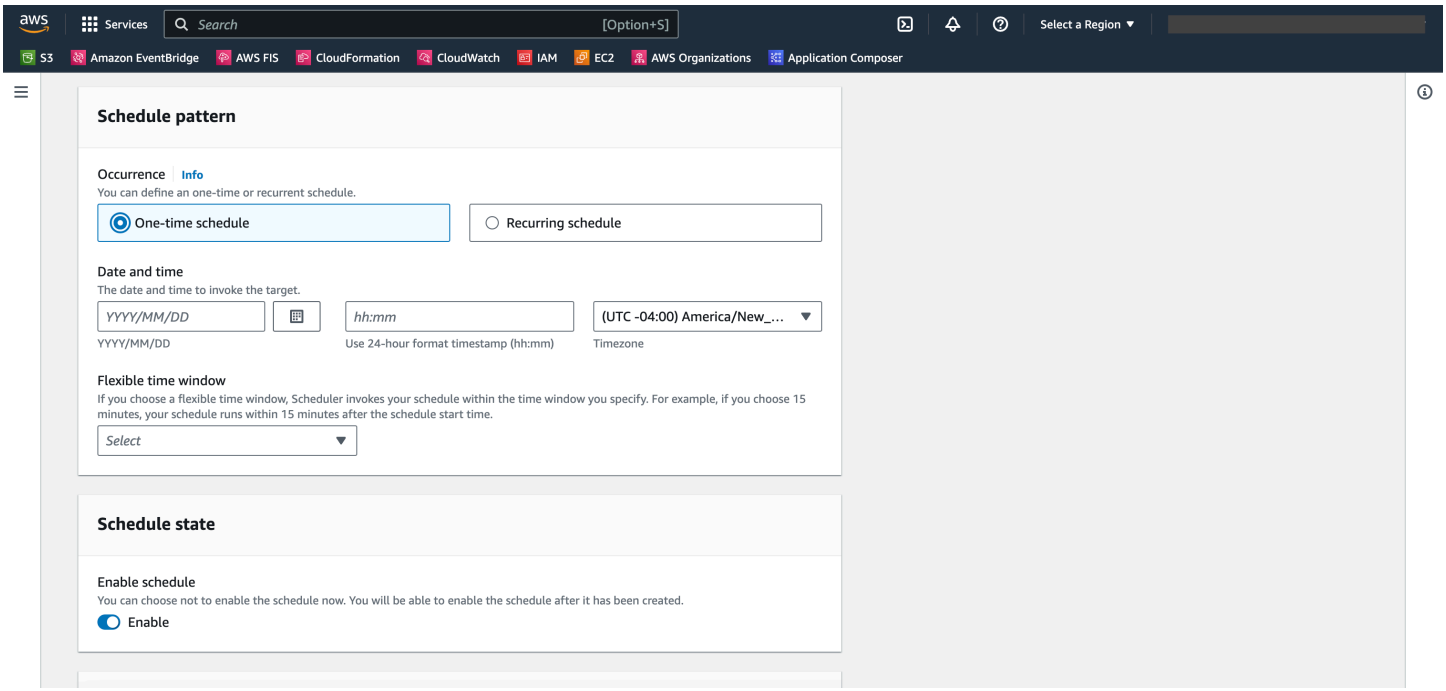
```
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Schedule"
                }
            }
        }
    ]
}
```

Programación de un experimento de FIS

Antes de programar un experimento, necesita uno o más Plantillas de experimento para poder invocarlos en su programación Puede usar un recurso de AWS existente o crear uno nuevo.

Una vez creada la plantilla de experimento, haga clic en Acciones y seleccione Programar experimento. Se le redirigirá a la página de programación del experimento. El nombre de la programación se rellenará automáticamente.

Siga la sección de patrones de programación y elija una programación única o periódica. Rellene los campos de entrada obligatorios y vaya a los permisos.



El estado de programación se encuentra habilitado de manera predeterminada. Nota: Si desactiva el estado de la programación, el experimento no se programará aunque cree una programación.

AWS FISEl Programador de experimentos se basa en Scheduler. EventBridge Puede consultar la documentación para ver los distintos tipos de programación compatibles.

Para actualizar la programación con la consola

1. Abra la consola de AWS FIS.
2. En el panel de navegación izquierdo, elija Plantillas de experimento.
3. Elija la plantilla de experimento para la que desea crear la programación.
4. Haga clic en Acciones y, en el menú desplegable, seleccione Programar experimento.
   a. En Nombre de la programación, el nombre se rellena automáticamente.
   b. En Patrón de programación, seleccione Programación periódica.
   c. En Tipo de programación, puede seleccionar una programación basada en frecuencia y ver los tipos de programación.
   d. En Expresión de frecuencia, elija una frecuencia que sea más lenta que el tiempo de ejecución del experimento, por ejemplo, 5 minutos.
   e. En Periodo, seleccione su zona horaria.
   f. En Fecha y hora de inicio, especifique una fecha y hora de inicio.
   g. En Fecha y hora de finalización, especifique una fecha y hora de finalización.
   h. En Estado de la programación, active la opción Habilitar programación.
   i. En Permisos, seleccione Usar rol existente y, a continuación, busque `FisSchedulerExecutionRole`.
   j. Elija Siguiente.
5. Seleccione Revisar y crear una programación, revise los detalles de su programador y, a continuación, elija Crear programación.

Actualización de la programación del experimento

Puede actualizar una programación del experimento para que se produzca a la fecha y hora específicas que mejor le convengan.

Para actualizar la ejecución de un experimento con la consola

1. Abra la consola de Amazon FIS.
2. En el panel de navegación, elija Plantillas de experimento.

3. Elija el Tipo de recurso: Plantilla de experimento para la que ya se ha creado una programación.
4. Haga clic en el ID del experimento de la plantilla. A continuación, diríjase a la pestaña de programaciones.
5. Compruebe si existe una programación asociada al experimento. Seleccione la programación asociada y haga clic en el botón Actualizar la programación.

Deshabilitación o eliminación de la ejecución de un experimento con la consola

Para impedir que un experimento se ejecute según una programación, puede eliminar o deshabilitar la regla. En los pasos siguientes, se explica cómo eliminar o deshabilitar un ejecución del experimento.

Para eliminar o deshabilitar una regla

1. Abra la consola de Amazon FIS.
2. En el panel de navegación, elija Plantillas de experimento.
3. Elija el Tipo de recurso: Plantilla de experimento para la que ya se ha creado una programación.
4. Haga clic en el ID del experimento de la plantilla. A continuación, diríjase a la pestaña de programaciones.
5. Compruebe si existe una programación asociada al experimento. Seleccione la programación asociada y haga clic en el botón Actualizar la programación.
6. Realice una de las acciones siguientes:
   a. Para eliminar la programación, seleccione el botón situado junto a la regla Eliminar programación. Escriba `delete` y haga clic en el botón Eliminar programación.
   b. Para deshabilitar la programación, seleccione el botón situado junto a la regla Deshabilitar programación. Escriba `disable` y haga clic en el botón Deshabilitar programación.

Monitorización de AWS FIS

Puede utilizar las siguientes herramientas para monitorizar el progreso y el impacto de sus experimentos de AWS Fault Injection Service (AWS FIS).

Consola de AWS FIS y la AWS CLI

Utilice la consola de AWS FIS o la AWS CLI para monitorizar el progreso de un experimento en ejecución. Puede ver el estado de cada acción del experimento y los resultados de cada acción. Para obtener más información, consulte the section called “Visualización de sus experimentos”.

CloudWatch métricas y alarmas de uso

Usa las métricas de CloudWatch uso para proporcionar visibilidad sobre el uso de los recursos de tu cuenta. AWS Las métricas de uso de AWS FIS se corresponden con AWS Service Quotas. Puede configurar alarmas que le avisen cuando su uso se acerque a una Service Quota. Para obtener más información, consulte Monitoreo con CloudWatch.

También puede crear condiciones de parada para sus experimentos de AWS FIS mediante la creación de CloudWatch alarmas que definan cuándo un experimento se sale de los límites. Cuando se activa la alarma, el experimento se detiene. Para obtener más información, consulte Condiciones de detención. Para obtener más información sobre la creación de CloudWatch alarmas, consulte Crear una CloudWatch alarma basada en un umbral estático y Crear una CloudWatch alarma basada en la detección de anomalías en la Guía del CloudWatch usuario de Amazon.

Registro de experimentos de AWS FIS

Habilite el registro de experimentos para capturar información detallada sobre su experimento a medida que se ejecuta. Para más información, consulte Registro de experimentos.

Eventos de cambio de estado de experimento

Amazon EventBridge le permite responder automáticamente a los eventos del sistema o a los cambios en los recursos. AWS FIS emite una notificación cuando el estado de un experimento cambia. Puede crear reglas para los eventos de su interés, que especifiquen la acción automática que se va a realizar cuando un evento cumple una de las reglas. Por ejemplo, enviar una notificación a un tema de Amazon SNS o invocar una función de Lambda. Para obtener más información, consulte Supervise con EventBridge.

CloudTrail registros

Se utiliza AWS CloudTrail para capturar información detallada sobre las llamadas realizadas a la API AWS FIS y almacenarlas como archivos de registro en Amazon S3. CloudTrail también registra las llamadas realizadas a las API de servicio de los recursos en los que está realizando los experimentos. Puede usar estos CloudTrail registros para determinar qué llamadas se realizaron, la dirección IP de origen de la llamada, quién realizó la llamada, cuándo se realizó la llamada, etc.

Notificaciones del panel de AWS Health

AWS Health proporciona una visibilidad continua del rendimiento de sus recursos y de la disponibilidad de sus cuentas y servicios de AWS. Cuando comienza un experimento, AWS FIS envía una notificación a su panel de AWS Health. La notificación estará presente durante todo el experimento en cada cuenta que contenga recursos a los que se destine un experimento, incluidos los experimentos con varias cuentas. Los experimentos con varias cuentas en los que solo se realicen acciones que no incluyan objetivos, como `aws:ssm:start-automation-execution` y `aws:fis:wait`, no emitirán ninguna notificación. La información sobre el rol utilizado para permitir el experimento aparecerá en Recursos afectados. Para obtener más información sobre el panel de AWS Health, consulte Panel de AWS Health en el Guía del usuario de AWS Health.

> **Note**
>
> AWS Health entrega eventos de la mejor forma posible.

Monitorización de métricas de uso de AWS FIS con Amazon CloudWatch

Puede utilizar Amazon CloudWatch para monitorizar el impacto de los experimentos de AWS FIS en los destinos. También puede monitorizar su uso del AWS FIS.

Para obtener más información sobre cómo visualizar el estado de un experimento, consulte Visualización de sus experimentos.

Monitorización de los experimentos de AWS FIS

Al planificar los experimentos de AWS FIS, identifique las métricas de CloudWatch que puede utilizar para identificar la referencia o el “estado estable” de los tipos de recursos de destino del experimento. Tras iniciar un experimento, puede monitorizar esas métricas de CloudWatch para los destinos seleccionados a través de la plantilla de experimento.

Para obtener más información acerca de las métricas de CloudWatch disponibles para un tipo de recurso de destino compatible con AWS FIS, consulte lo siguiente:

- Monitorización de las instancias con CloudWatch
- Métricas de Amazon ECS CloudWatch
- Monitorización de las métricas de Amazon RDS con CloudWatch
- Monitorización de métricas de Run Command con CloudWatch

Métricas de uso de AWS FIS

Puede utilizar las métricas de uso de CloudWatch para proporcionar visibilidad sobre el uso de los recursos de su cuenta. Utilice estas métricas para visualizar el uso actual del servicio en paneles y gráficos de CloudWatch.

Las métricas de uso de AWS FIS se corresponden con AWS Service Quotas. Puede configurar alarmas que le avisen cuando su uso se acerque a una cuota de servicio. Para obtener más información sobre las alarmas de CloudWatch, consulte la Guía del usuario de Amazon CloudWatch.

AWS FIS publica las siguientes métricas en el espacio de nombres AWS/Usage.

Métrica	Descripción
`ResourceCount`	El número total de los recursos especificados que se ejecutan en su cuenta. Los recursos se definen por las dimensiones asociadas a la métrica.

Las siguientes dimensiones se utilizan para ajustar las métricas de uso que publica AWS FIS.

Dimensión	Descripción
`Service`	El nombre del servicio de AWS que contiene el recurso. Para las métricas de uso de AWS FIS, el valor de esta dimensión es `FIS`.
`Type`	El tipo de entidad que se registra. Actualmente, el único valor válido para las métricas de uso de AWS FIS es `Resource`.
`Resource`	El tipo de recurso que se está ejecutando. Los valores posibles son `ExperimentTemplates` para las plantillas de experimento y `ActiveExperiments` para los experimentos activos.
`Class`	Esta dimensión se reserva para un uso ulterior.

AWS Supervise los experimentos del FIS con Amazon EventBridge

Cuando el estado de un experimento cambia, el AWS FIS emite una notificación. Estas notificaciones están disponibles como eventos a través de Amazon EventBridge (anteriormente denominado CloudWatch Events). AWS La FIS emite estos eventos haciendo todo lo posible. Los eventos se envían casi EventBridge en tiempo real.

Con EventBridge él, puede crear reglas que activen acciones programáticas en respuesta a un evento. Por ejemplo, puede configurar una regla que invoque un tema de SNS para enviar una notificación por correo electrónico o que active una función de Lambda para realizar alguna acción.

Para obtener más información EventBridge, consulta Cómo empezar a usar Amazon EventBridge en la Guía del EventBridge usuario de Amazon.

A continuación, la sintaxis de un evento de cambio de estado de un experimento:

```
{
    "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`

El ID del experimento cuyo estado ha cambiado.

`experiment-template-id`

El ID de la plantilla de experimento utilizada por el experimento.

`new_value`

El nuevo estado del experimento. Los valores posibles son:

- `completed`
- `failed`
- `initiating`
- `running`
- `stopped`
- `stopping`

`old_value`

El estado anterior del experimento. Los valores posibles son:

- `initiating`

- `pending`
- `running`
- `stopping`

Registro de experimentos de AWS FIS

Puede usar el registro de experimentos para capturar información detallada sobre su experimento a medida que se ejecuta.

Se le cobrará por el registro de los experimentos en función de los costes asociados a cada tipo de destino de registro. Para obtener más información, consulte CloudWatch los precios de Amazon (en Paid Tier, Logs, Vended Logs) y los precios de Amazon S3.

Permisos

Debe conceder permisos a AWS FIS para enviar registros a cada destino de registro que configure. Para obtener más información, consulte lo siguiente en la Guía del usuario de Amazon CloudWatch Logs:

- Registros enviados a CloudWatch Logs
- Registros enviados a Amazon S3

Esquema de registro

El siguiente esquema es el utilizado en el registro de experimentos. La versión actual es la 2. Los campos de `details` dependen del valor de `log_type`. Los campos de `resolved_targets` dependen del valor de `target_type`. Para obtener más información, consulte the section called "Ejemplos de entradas de registro".

```
{
    "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

    }
}
```

Notas de la versión

- La versión 2 presenta:
  - El campo `target_type` y cambia el campo `resolved_targets` de una lista de ARN a una lista de objetos. Los campos válidos del objeto `resolved_targets` dependen del valor de `target_type`, que es el tipo de recurso de los destinos.

  - Los tipos de eventos `action-error` y `target-resolution-detail` que agregan el campo `account_id`.
- La versión 1 es la inicial.

Registro de destinos

AWS FIS admite la entrega de registros a los siguientes destinos:

- Un bucket de Amazon S3.
- Un grupo de CloudWatch registros de Amazon Logs

Entrega de registros de S3

Los registros se entregan en la siguiente ubicación.

```
bucket-and-optional-prefix/AWSLogs/account-id/fis/region/experiment-
id/YYYY/MM/DD/account-id_awsfislogs_region_experiment-id_YYYYMMDDHHMMZ_hash.log
```

Los registros pueden tardar varios minutos en entregarse en el bucket.

CloudWatch Entrega de registros

Los registros se entregan a un flujo de registro denominado /aws/fis/*experiment-id*.

Los registros se entregan al grupo de registro en menos de un minuto.

Ejemplos de entradas de registro

A continuación, se muestran ejemplos de entradas de registro de un experimento que ejecuta la acción aws:ec2:reboot-instances en una instancia EC2 seleccionada al azar.

Registros

- experiment-start
- target-resolution-start
- target-resolution-detail
- target-resolution-end
- action-start

- action-end
- action-error
- experiment-end

experiment-start

A continuación, se ve un ejemplo de registro del evento `experiment-start`.

```
{
    "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

A continuación, se ve un ejemplo de registro del evento `target-resolution-start`.

```
{
    "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

A continuación, se ve un ejemplo de registro del evento `target-resolution-detail`. Si la resolución de destino falla, el registro también incluye el campo `error_information`.

```
{
    "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

Si la resolución de destino falla, el registro también incluye el campo `error_information`. Si `total_pages` es mayor que 1, el número de destinos resueltos ha superado el límite de tamaño de un registro. Hay registros de `target-resolution-end` adicionales que contienen el resto de los destinos resueltos.

A continuación, se ve un ejemplo de registro del evento `target-resolution-end` para una acción de EC2.

```
{
    "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
    }
}
```

A continuación, se ve un ejemplo de registro del evento `target-resolution-end` para una acción de EKS.

```
{
    "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

A continuación, se ve un ejemplo de registro del evento `action-start`. Si la plantilla de experimento especifica los parámetros de la acción, el registro también incluye el campo `parameters`.

```
{
    "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

A continuación, se ve un ejemplo de registro del evento `action-error`. Este evento solo se devuelve cuando se produce un error en una acción. Se devuelve para cada cuenta en la que se produce un error en la acción.

```
{
    "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

A continuación, se ve un ejemplo de registro del evento `action-end`.

```
{
    "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

A continuación, se ve un ejemplo de registro del evento `experiment-end`.

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

Habilitación del registro de experimentos

El registro de experimentos está deshabilitado de forma predeterminada. Para recibir registros de un experimento, debe crear el experimento a partir de una plantilla de experimento con el registro habilitado. La primera vez que ejecute un experimento que esté configurado para usar un destino que no se haya utilizado anteriormente para el registro, se retrasará el experimento para configurar la entrega de registros a este destino, lo que tarda unos 15 segundos.

Para habilitar el registro de experimentos con la consola

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Actualizar plantilla de experimento.

4. En Registros, configure las opciones de destino. Para enviar registros a un bucket de S3, seleccione Enviar a un bucket de Amazon S3 y escriba el nombre y el prefijo del bucket. Para enviar los registros a CloudWatch los registros, elija Enviar a CloudWatch los registros e introduzca el grupo de registros.
5. Elija Actualizar plantilla de experimento.

Para habilitar el registro de experimentos con la AWS CLI

Use el update-experiment-templatecomando y especifique una configuración de registro.

Deshabilitación del registro de experimentos

Si ya no quiere recibir registros para los experimentos, puede deshabilitar el registro de experimentos.

Para deshabilitar el registro de experimentos con la consola

1. Abra la consola de AWS FIS en https://console.aws.amazon.com/fis/.
2. En el panel de navegación, elija Plantillas de experimento.
3. Seleccione la plantilla de experimento y elija Acciones, Actualizar plantilla de experimento.
4. Para Logs, desactive Send to an Amazon S3 bucket y Send to CloudWatch Logs.
5. Elija Actualizar plantilla de experimento.

Para deshabilitar el registro de experimentos con la AWS CLI

Use el update-experiment-templatecomando y especifique una configuración de registro vacía.

Registre llamadas a la API con AWS CloudTrail

AWSEl servicio de inyección de fallos (AWSFIS) está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en el AWS FIS. CloudTrail captura todas las llamadas a la API del AWS FIS como eventos. Las llamadas capturadas incluyen las realizadas desde la consola de AWS FIS y las llamadas de código a las operaciones de la API de AWS FIS. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para AWS FIS. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a la AWS

FIS, la dirección IP desde la que se realizó la solicitud, quién la hizo, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía del AWS CloudTrail usuario.

Utilice CloudTrail

CloudTrail está activado en tu cuenta Cuenta de AWS al crear la cuenta. Cuando se produce una actividad en el AWS FIS, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar eventos recientes en su Cuenta de AWS. Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para mantener un registro continuo de eventos en la Cuenta de AWS, incluidos los eventos de AWS FIS, cree un registro de seguimiento. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte los siguientes temas:

- Crear un registro de seguimiento para su cuenta de AWS
- CloudTrail Integraciones y servicios compatibles
- Configuración de las notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas las acciones del AWS FIS se registran CloudTrail y se documentan en la referencia de la API del servicio de inyección de AWS fallos. Para ver las acciones experimentales que se llevan a cabo en un recurso de destino, consulte la documentación de referencia de la API del servicio propietario del recurso. Por ejemplo, para ver las acciones que se llevan a cabo en una instancia de Amazon EC2, consulte Referencia de la API de Amazon EC2.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario.

- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail .

Comprender las entradas de archivos de registro de AWS FIS

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

A continuación se muestra un ejemplo de entrada de CloudTrail registro para una llamada a la `StopExperiment` acción de la AWS FIS.


```
{
  "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"

}
```


A continuación se muestra un ejemplo de entrada de CloudTrail registro de una acción de la API que el AWS FIS invocó como parte de un experimento que incluye la acción del `aws:ssm:send-command` AWS FIS. El elemento `userIdentity` refleja una solicitud realizada con credenciales temporales obtenidas al asumir un rol. El nombre del rol asumido aparece en `userName`. El ID del

experimento, EXP21nT17WMzA6dnUgz, aparece en `principalId` y como parte del ARN del rol asumido.


```
{
    "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"
}
```

Seguridad en el servicio de inyección de AWS fallos

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento que se aplican al Servicio de Inyección de AWS Fallos, consulte AWSAWS Servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. Usted también es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a entender cómo aplicar el modelo de responsabilidad compartida al utilizar el AWS FIS. Los siguientes temas muestran cómo configurar el AWS FIS para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos del AWS FIS.

Contenido

Protección de datos en AWS Fault Injection Service

El modelo de se aplica a protección de datos en AWS Fault Injection Service. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta

infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS.

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con el AWS FIS o con otros dispositivos Servicios de AWS mediante la consola, la API o los SDK. AWS CLI AWS Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya la información de las credenciales en la URL para validar la solicitud para ese servidor.

Cifrado en reposo

AWS FIS siempre cifra los datos en reposo. Los datos del AWS FIS se cifran en reposo mediante un cifrado transparente del lado del servidor. Esto ayuda a reducir la carga y la complejidad operativas que conlleva la protección de información confidencial. Con el cifrado en reposo, puede crear aplicaciones sensibles a la seguridad que cumplen los requisitos de cifrado y normativos.

Cifrado en tránsito

AWS El FIS cifra los datos en tránsito entre el servicio y otros servicios integrados. AWS Todos los datos que pasan entre el AWS FIS y los servicios integrados se cifran mediante Transport Layer Security (TLS). Para obtener más información sobre otros AWS servicios integrados, consulte. Servicios de AWS admitidas

Administración de identidad y acceso para AWS Fault Injection Service

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos. AWS Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar AWS los recursos del FIS. La IAM es una opción Servicio de AWS que puede utilizar sin coste adicional.

Contenido

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que se realice en AWS FIS.

Usuario del servicio: si utiliza el servicio AWS FIS para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más funciones de la AWS FIS para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador.

Administrador de servicios: si está a cargo de los recursos del AWS FIS en su empresa, probablemente tenga pleno acceso al AWS FIS. Su trabajo consiste en determinar a qué funciones y recursos del AWS FIS deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM.

Administrador de IAM: si es administrador de IAM, tal vez desee obtener información detallada sobre cómo redactar políticas para administrar el acceso al FIS. AWS

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación

multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS Single Sign-On y Uso de la autenticación multifactor (MFA) en AWS en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS Single Sign-On. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran

credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. El IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS Single Sign-On.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.

- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
  - Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.
- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de

instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una

entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifique el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.

- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona el servicio de inyección de AWS fallos con IAM

Antes de utilizar IAM para gestionar el acceso al AWS FIS, averigüe qué funciones de IAM están disponibles para su uso con el FIS. AWS

Funciones de IAM que puede utilizar con el servicio de inyección de fallos AWS

Característica de IAM	AWS Soporte FIS
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACL	No
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	Sí

Para obtener una visión general de cómo funcionan el AWS FIS y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para el FIS AWS

Compatibilidad con las políticas basadas en identidades	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué

condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para la FIS AWS

Para ver ejemplos de políticas del AWS FIS basadas en la identidad, consulte. AWS Ejemplos de políticas del servicio de inyección de errores

Políticas basadas en recursos dentro del FIS AWS

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional.

Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones políticas para AWS el FIS

Admite acciones de política	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones del AWS FIS, consulte las acciones definidas por el servicio de AWS inyección de errores en la Referencia de autorización del servicio.

Las acciones políticas del AWS FIS utilizan el siguiente prefijo antes de la acción:

```
fis
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
    "fis:action1",
    "fis:action2"
]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones. Por ejemplo, para especificar todas las acciones que comiencen con la palabra `List`, incluya la siguiente acción:

```
"Action": "fis:List*"
```

Recursos políticos para el FIS AWS

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Algunas acciones AWS de la API FIS admiten varios recursos. Para especificar varios recursos en una única instrucción, separe los ARN con comas.

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

Para ver una lista de los tipos de recursos del AWS FIS y sus ARN, consulte los tipos de recursos definidos por el servicio de inyección de AWS errores en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por el servicio de inyección de AWS fallas.

Claves de condición de la política para el FIS AWS

Admite claves de condición de políticas específicas del servicio	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación lógica `AND`. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de las claves de condición de la AWS FIS, consulte las claves de condición del servicio de inyección AWS de errores en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por el servicio de inyección de AWS fallos.

Para ver ejemplos de políticas de la AWS FIS basadas en la identidad, consulte. AWS Ejemplos de políticas del servicio de inyección de errores

ACL en el FIS AWS

Admite las ACL	No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con FIS AWS

Admite ABAC (etiquetas en las políticas)	Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Para consultar un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte Ejemplo: Uso de etiquetas para controlar el uso de recursos.

Uso de credenciales temporales con FIS AWS

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para FIS AWS

Admite Forward access sessions (FAS)	Sí

Cuando utiliza un usuario o un rol de IAM para realizar acciones en él AWS, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio de AWS FIS

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener

más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

Funciones vinculadas al servicio para la FIS AWS

Compatible con roles vinculados al servicio	Sí

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información sobre la creación o administración de funciones vinculadas al servicio de la AWS FIS, consulte. Utilice funciones vinculadas al servicio para el servicio de inyección de errores AWS

AWS Ejemplos de políticas del servicio de inyección de errores

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar los recursos del AWS FIS. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles, y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por el AWS FIS, incluido el formato de los ARN de cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones del servicio de inyección de AWS errores en la Referencia de autorización de servicios.

Contenido

- Ejemplo: enumere las acciones del AWS FIS disponibles
- Ejemplo: Creación de una plantilla de experimento para una acción específica
- Ejemplo: Inicio de un experimento
- Ejemplo: Uso de etiquetas para controlar el uso de recursos
- Ejemplo: Eliminación de una plantilla de experimento con una etiqueta específica
- Ejemplo: Permitir que los usuarios vean sus propios permisos
- Ejemplo: utilice claves de condición para ec2:InjectApiError
- Ejemplo: utilizar claves de condición para aws:s3:bucket-pause-replication
- Ejemplo: experimentar un rol con permisos para ejecutar aws:dynamodb:encrypted-global-table-pause-replication

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear, acceder o eliminar los recursos del AWS FIS de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía del usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS

CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.

- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Ejemplo: utilice la consola FIS AWS

Para acceder a la consola del Servicio de Inyección de AWS Fallos, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos del AWS FIS que tiene en su Cuenta de AWS cuenta. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

El siguiente ejemplo de política otorga permiso para enumerar y ver todos los recursos de la AWS FIS mediante la consola de la AWS FIS, pero no para crearlos, actualizarlos ni eliminarlos. También concede permisos para ver los recursos disponibles que utilizan todas las acciones de la AWS FIS que se puedan especificar en una plantilla de experimento.

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

Ejemplo: enumere las acciones del AWS FIS disponibles

La siguiente política concede permiso para enumerar las acciones de la AWS FIS disponibles.

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

Ejemplo: Creación de una plantilla de experimento para una acción específica

La siguiente política concede permiso para crear una plantilla de experimento para la acción `aws:ec2:stop-instances`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
          "fis:CreateExperimentTemplate"
      ],
      "Resource": [
        "arn:aws:fis:*:*:action/aws:ec2:stop-instances",
        "arn:aws:fis:*:*:experiment-template/*"
      ]
    },
    {
      "Sid": "PolicyPassRoleExample",
      "Effect": "Allow",
      "Action": [
          "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::account-id:role/role-name"
      ]
    }
  ]
```

```
}
```

Ejemplo: Inicio de un experimento

La siguiente política concede permiso para iniciar un experimento con el rol de IAM y la plantilla de experimento especificados. También permite a la AWS FIS crear un rol vinculado al servicio en nombre del usuario. Para obtener más información, consulte Utilice funciones vinculadas al servicio para el servicio de inyección de errores AWS.

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

Ejemplo: Uso de etiquetas para controlar el uso de recursos

La siguiente política concede permiso para ejecutar experimentos a partir de plantillas de experimentos que tengan la etiqueta `Purpose=Test`. No concede permiso para crear o modificar

plantillas de experimentos, ni para ejecutar experimentos con plantillas que no tengan la etiqueta especificada.

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

Ejemplo: Eliminación de una plantilla de experimento con una etiqueta específica

La siguiente política concede permiso para eliminar una plantilla de experimento con la etiqueta `Purpose=Test`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "fis:DeleteExperimentTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

Ejemplo: Permitir que los usuarios vean sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. AWS CLI AWS

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

Ejemplo: utilice claves de condición para `ec2:InjectApiError`

La siguiente política de ejemplo utiliza la clave de condición `ec2:FisTargetArns` para determinar el alcance de los recursos de destino. Esta política permite las acciones de la AWS FIS y. `aws:ec2:api-insufficient-instance-capacity-error` `aws:ec2:asg-insufficient-instance-capacity-error`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:api-insufficient-instance-capacity-error",
                    ],
                    "ec2:FisTargetArns": [
                        "arn:aws:iam:*:*:role:role-name"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:InjectApiError",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "ec2:FisActionId": [
                        "aws:ec2:asg-insufficient-instance-capacity-error"
                    ],
                    "ec2:FisTargetArns": [

 "arn:aws:autoscaling:*:*:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": "autoscaling:DescribeAutoScalingGroups",
            "Resource": "*"
        }
    ]
}
```

Ejemplo: utilizar claves de condición para **aws:s3:bucket-pause-replication**

El siguiente ejemplo de política utiliza la clave de `S3:IsReplicationPauseRequest` condición para permitir `PutReplicationConfiguration` y `GetReplicationConfiguration` solo cuando AWS sea por parte de la FIS en el contexto de la acción de la AWS FIS. `aws:s3:bucket-pause-replication`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "S3:PauseReplication"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "StringEquals": {
                    "s3:DestinationRegion": "region"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "S3:PutReplicationConfiguration",
                "S3:GetReplicationConfiguration"
            ],
            "Resource": "arn:aws:s3:::mybucket",
            "Condition": {
                "BoolIfExists": {
                    "s3:IsReplicationPauseRequest": "true"
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "S3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
    }
```

Ejemplo: experimentar un rol con permisos para ejecutar **aws:dynamodb:encrypted-global-table-pause-replication**

El siguiente ejemplo de política otorga a la AWS FIS los permisos necesarios para ejecutar un experimento con una sola acción. `aws:dynamodb:encrypted-global-table-pause-replication`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DynamoDB",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeGlobalTable"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-2:123456789012:table/MyEncryptedGlobalTable"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/fis-enabled": "true"
                }
            }
        },
        {
            "Sid": "Tagging",
```

```
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "KMS",
            "Effect": "Allow",
            "Action": [
                "kms:PutKeyPolicy",
                "kms:DescribeKey",
                "kms:GetKeyPolicy"
            ],
            "Resource": "arn:aws:kms:us-east-2:123456789012:key/
MyGlobalTableEncryptionKey"
        },
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/fis-enabled": "true"
            }
        }
    ]
}
```

> **Note**
>
> AWS FIS lo utiliza `kms:PutKeyPolicy` para denegar el acceso a DynamoDB, a la clave AWS KMS gestionada por el cliente, lo que detiene la replicación. Se recomienda utilizar el rol solo cuando se esté realizando activamente un experimento con esta acción; de lo contrario, se recomienda eliminarlo. Al eliminar el rol, se eliminan los permisos de FIS para `kms:PutKeyPolicy`. Una vez finalizado el experimento, busque el rol en los detalles de la plantilla de experimento. Elija el enlace al rol de IAM en la consola de IAM y elija eliminar. Tras eliminar el rol, vaya a la AWS KMS consola y busque la AWS KMS clave utilizada para proteger los datos en la tabla de DynamoDB de destino. Compruebe que la política AWS KMS clave coincide con sus expectativas. Ya no debería ver una declaración de la AWS FIS (por ejemplo,`FIS_DDB_PAUSE_REPLICATION-EXP123456789012345_DO_NOT_MODIFY`).

Las acciones de FIS `aws:dynamodb:encrypted-global-table-pause-replication` agregan dinámicamente los siguientes permisos a la política para la clave KMS utilizada para proteger los datos de las tablas globales de DynamoDB de destino:

```
{
    "Sid": "DO_NOT_MODIFY_FIS_DDB_PAUSE_REPLICATION-EXP123456789012345",
    "Effect": "Deny",
    "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/aws-service-role/
replication.dynamodb.amazonaws.com/AWSServiceRoleForDynamoDBReplication"
        },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:dynamodb:tableName": [
                "transactions-global-table",
                "inventory-global-table"
            ]
        }
    }
}
```

Estos permisos se adjuntarán al final del documento de política AWS KMS clave existente. La instrucción de la política anterior elimina los permisos para que el rol vinculado al servicio de DynamoDB replique datos con origen y destino en las tablas que se muestran en la clave de contexto `kms:EncryptionContext:aws:dynamodb:tableName`. En el ejemplo anterior, la replicación se detendría para las tablas globales de DynamoDB con los nombres `transaction-global-table`, `inventory-global-table`.

Utilice funciones vinculadas al servicio para el servicio de inyección de errores AWS

AWS El servicio de inyección de errores utiliza funciones vinculadas al servicio AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está

vinculado directamente a la FIS. AWS Los roles vinculados a servicios están predefinidos por AWS FIS e incluyen todos los permisos que el servicio requiere para llamar a otros servicios de AWS en su nombre.

Un rol vinculado a un servicio facilita la configuración del AWS FIS, ya que no es necesario añadir manualmente los permisos necesarios para gestionar la supervisión y la selección de recursos para los experimentos. AWS El FIS define los permisos de sus funciones vinculadas al servicio y, a menos que se defina lo contrario, solo AWS el FIS puede asumir sus funciones. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Además del rol vinculado a servicios, también debe especificar un rol de IAM que conceda permiso para modificar los recursos que especifique como destinos en una plantilla de experimento. Para obtener más información, consulte Roles de IAM para los experimentos de AWS FIS.

Solo puede eliminar un rol vinculado a servicios después de eliminar los recursos relacionados. Esto protege sus recursos de la AWS FIS porque no puede eliminar inadvertidamente el permiso de acceso a los recursos.

Permisos de rol vinculados al servicio para FIS AWS

AWS El FIS utiliza el rol vinculado al servicio denominado AWSServiceRoleForFISpara poder gestionar la supervisión y la selección de recursos para los experimentos.

El rol AWSServiceRoleForFISvinculado al servicio confía en los siguientes servicios para asumir el rol:

- `fis.amazonaws.com`

La función AWSServiceRoleForFISvinculada al servicio utiliza la política gestionada AmazonFIS. ServiceRolePolicy Esta política permite al AWS FIS gestionar la supervisión y la selección de recursos para los experimentos. Para obtener más información, consulte AmazonFIS ServiceRolePolicy en la Referencia de políticas AWS gestionadas.

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para que el rol AWSServiceRoleForFISvinculado al servicio se cree correctamente, la identidad de IAM con la que utilice el AWS FIS debe tener los permisos necesarios. Para conceder los permisos necesarios, asocie la siguiente política a la identidad de IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "fis.amazonaws.com"
                }
            }
        }
    ]
}
```

Para obtener más información, consulte Permisos de roles vinculados a servicios en la Guía del usuario de IAM.

Cree un rol vinculado a un servicio para FIS AWS

No necesita crear manualmente un rol vinculado a servicios. Al iniciar un experimento del AWS FIS en la AWS Management Console, la o la AWS API AWS CLI, el AWS FIS crea automáticamente el rol vinculado al servicio.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Al iniciar un experimento de FIS, AWS FIS vuelve a crear el AWS rol vinculado al servicio para usted.

Edite un rol vinculado a un servicio para FIS AWS

AWS El FIS no permite editar el rol vinculado al servicio. AWSServiceRoleForFIS Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte Editar un rol vinculado a servicios en la Guía del usuario de IAM.

Elimine un rol vinculado a un servicio para FIS AWS

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se monitorice

ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

> **Note**
>
> Si el servicio AWS FIS utiliza el rol al intentar limpiar los recursos, es posible que la limpieza no se realice correctamente. En tal caso, espere unos minutos e intente de nuevo la operación.

Para limpiar los recursos del AWS FIS utilizados por el AWSServiceRoleForFIS

Asegúrese de que ninguno de sus experimentos se esté ejecutando actualmente. Si es necesario, detenga los experimentos. Para obtener más información, consulte Detener un experimento.

Cómo eliminar manualmente el rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar la función vinculada al AWSServiceRoleForFISservicio. Para obtener más información, consulte Eliminación de un rol vinculado a servicios en la Guía del usuario de IAM.

Regiones compatibles para AWS las funciones vinculadas al servicio de la FIS

AWS El FIS admite el uso de funciones vinculadas al servicio en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte AWS Puntos de conexión y cuotas de Fault Injection Service.

AWS políticas gestionadas para el servicio de inyección de fallos AWS

Una política AWS gestionada es una política independiente creada y administrada por. AWS AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades

principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

AWS política gestionada: AmazonFIS ServiceRolePolicy

Esta política se adjunta a la función vinculada al servicio denominada para permitir que el AWS FIS gestione AWSServiceRoleForFISla supervisión y la selección de recursos para los experimentos. Para obtener más información, consulte Utilice funciones vinculadas al servicio para el servicio de inyección de errores AWS.

AWS política gestionada: AWSFaultInjectionSimulatorEC2Access

Utilice esta política en una función de experimento para conceder permiso al AWS FIS para ejecutar experimentos que utilicen las acciones del AWS FIS para Amazon EC2. Para obtener más información, consulte the section called “Rol de experimento”.

Para ver los permisos de esta política, consulte la Referencia de políticas AWSFaultInjectionSimulatorEC2Access AWSadministradas.

AWS política gestionada: AWSFaultInjectionSimulatorECSAccess

Utilice esta política en una función de experimento para conceder permiso al AWS FIS para ejecutar experimentos que utilicen las acciones del AWS FIS para Amazon ECS. Para obtener más información, consulte the section called “Rol de experimento”.

Para ver los permisos de esta política, consulte la Referencia AWSFaultInjectionSimulatorECSAccessde políticas AWS administradas.

AWS política gestionada: AWSFaultInjectionSimulatorEKSAccess

Utilice esta política en una función de experimento para conceder permiso al AWS FIS para ejecutar experimentos que utilicen las acciones del AWS FIS para Amazon EKS. Para obtener más información, consulte the section called “Rol de experimento”.

Para ver los permisos de esta política, consulte la Referencia AWSFaultInjectionSimulatorEKSAccessde políticas AWS administradas.

AWS política gestionada: AWSFaultInjectionSimulatorNetworkAccess

Utilice esta política en una función de experimento para conceder permiso a la AWS FIS para ejecutar experimentos que utilicen las acciones de red de la AWS FIS. Para obtener más información, consulte the section called “Rol de experimento”.

Para ver los permisos de esta política, consulte la Referencia AWSFaultInjectionSimulatorNetworkAccessde políticas AWS gestionadas.

AWS política gestionada: AWSFaultInjectionSimulatorRDSAccess

Utilice esta política en una función de experimento para conceder permiso al AWS FIS para ejecutar experimentos que utilicen las acciones del AWS FIS para Amazon RDS. Para obtener más información, consulte the section called “Rol de experimento”.

Para ver los permisos de esta política, consulte la Referencia de políticas AWSFaultInjectionSimulatorRDSAccess AWSadministradas.

AWS política gestionada: AWSFaultInjectionSimulatorSSMAccess

Utilice esta política en una función de experimento para conceder permiso a la AWS FIS para ejecutar experimentos que utilicen las acciones de la AWS FIS para Systems Manager. Para obtener más información, consulte the section called “Rol de experimento”.

Para ver los permisos de esta política, consulte la Referencia AWSFaultInjectionSimulatorSSMAccessde políticas AWS gestionadas.

AWS El FIS actualiza las políticas AWS gestionadas

Consulte los detalles sobre las actualizaciones de las políticas AWS gestionadas para el AWS FIS desde que este servicio comenzó a rastrear estos cambios.

Cambio	Descripción	Fecha
AWSFaultInjectionSimulatorECSAccess: actualización de una política actual	Se agregaron permisos para permitir que el AWS FIS resuelva los objetivos del ECS.	25 de enero de 2024
AWSFaultInjectionSimulatorNetworkAccess: actualización de una política actual	Se han añadido permisos para que el AWS FIS pueda ejecutar experimentos con las acciones aws:netwo	25 de enero de 2024

Cambio	Descripción	Fecha
	rk:route-table-disrupt-cross-region-connectivity yaws:network:transit-gateway-disrupt-cross-region-connectivity.	
AWSFaultInjectionSimulatorEC2Access: actualización de una política actual	Se han añadido permisos para permitir que el AWS FIS resuelva las instancias de EC2.	13 de noviembre de 2023
AWSFaultInjectionSimulatorEKSAccess: actualización de una política actual	Se agregaron permisos para permitir que el AWS FIS resuelva los objetivos de EKS.	13 de noviembre de 2023
AWSFaultInjectionSimulatorRDSAccess: actualización de una política actual	Se agregaron permisos para permitir que el AWS FIS resuelva los objetivos de RDS.	13 de noviembre de 2023
AWSFaultInjectionSimulatorEC2Access: actualización de una política actual	Se agregaron permisos para permitir a la AWS FIS ejecutar documentos SSM en instancias de EC2 y terminar las instancias de EC2.	2 de junio de 2023
AWSFaultInjectionSimulatorSSMAccess: actualización de una política actual	Se agregaron permisos para permitir que el AWS FIS ejecute documentos SSM en instancias EC2.	2 de junio de 2023
AWSFaultInjectionSimulatorECSAccess: actualización de una política actual	Se han añadido permisos para que el AWS FIS pueda realizar experimentos con las nuevas acciones. aws:ecs:task	1 de junio de 2023
AWSFaultInjectionSimulatorEKSAccess: actualización de una política actual	Se han añadido permisos para que el AWS FIS pueda ejecutar experimentos con las nuevas aws:eks:pod acciones.	1 de junio de 2023

Cambio	Descripción	Fecha
AWSFaultInjectionSimulatorEC2Access: política nueva	Se ha añadido una política que permite a la AWS FIS ejecutar un experimento que utilice acciones de la AWS FIS para Amazon EC2.	26 de octubre de 2022
AWSFaultInjectionSimulatorECSAccess: política nueva	Se agregó una política que permite a la AWS FIS ejecutar un experimento que utilice acciones de la AWS FIS para Amazon ECS.	26 de octubre de 2022
AWSFaultInjectionSimulatorEKSAccess: política nueva	Se agregó una política que permite a la AWS FIS ejecutar un experimento que utilice acciones de la AWS FIS para Amazon EKS.	26 de octubre de 2022
AWSFaultInjectionSimulatorNetworkAccess: política nueva	Se agregó una política que permite al AWS FIS realizar un experimento que utilice las acciones de red del AWS FIS.	26 de octubre de 2022
AWSFaultInjectionSimulatorRDSAccess: política nueva	Se ha añadido una política que permite a la AWS FIS ejecutar un experimento que utilice acciones de la AWS FIS para Amazon RDS.	26 de octubre de 2022
AWSFaultInjectionSimulatorSSMAccess: política nueva	Se ha añadido una política que permite a la AWS FIS ejecutar un experimento que utilice las acciones de la AWS FIS para Systems Manager.	26 de octubre de 2022
AmazonFISServiceRolePolicy: actualización de una política existente	Se agregaron permisos para permitir que el AWS FIS describa las subredes.	26 de octubre de 2022

Cambio	Descripción	Fecha
AmazonFISServiceRolePolicy: actualización de una política existente	Se han añadido permisos para que el AWS FIS pueda describir los clústeres de EKS.	7 de julio de 2022
AmazonFIS ServiceRolePolicy: actualización de una política existente	Se han añadido permisos para que el AWS FIS pueda enumerar y describir las tareas de sus clústeres.	7 de febrero de 2022
AmazonFIS ServiceRolePolicy: actualización de una política existente	Se ha eliminado la condición `events:ManagedBy` de la acción `events:DescribeRule` .	6 de enero de 2022
AmazonFISServiceRolePolicy: actualización de una política existente	Se agregaron permisos para permitir que el AWS FIS recupere el historial de las CloudWatch alarmas utilizadas en condiciones de parada.	30 de junio de 2021
AWS El FIS comenzó a rastrear los cambios	AWS La FIS comenzó a rastrear los cambios en sus políticas gestionadas AWS	1 de marzo de 2021

La seguridad de la infraestructura en el servicio de inyección de AWS fallas

Como servicio gestionado, el servicio de inyección de AWS fallas está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder al AWS FIS a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.

- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Acceda al AWS FIS mediante un punto final de VPC de interfaz ()AWS PrivateLink

Puede establecer una conexión privada entre su VPC y el servicio de inyección de AWS errores mediante la creación de un punto final de VPC de interfaz. Los puntos finales de VPC funcionan con una tecnología que le permite acceder de forma privada a las API de la AWS FIS sin una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión Direct AWS Connect. AWS PrivateLink Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con las API de la AWS FIS.

Cada punto de conexión de la interfaz está representado por una o más interfaces de redes elásticas en las subredes.

Para obtener más información, consulte Acceso directo AWS PrivateLink en la Servicios de AWSAWS PrivateLink guía.

Consideraciones sobre los puntos finales AWS de VPC de FIS

Antes de configurar un punto final de VPC de interfaz para AWS FIS, consulte Acceder y Servicio de AWS usar un punto final de VPC de interfaz en la Guía.AWS PrivateLink

AWS FIS admite realizar llamadas a todas sus acciones de API desde su VPC.

Cree un punto final de VPC de interfaz para FIS AWS

Puede crear un punto de enlace de VPC para el servicio AWS FIS mediante la consola Amazon VPC o el (). AWS Command Line Interface AWS CLI Para obtener más información, consulte Create a VPC endpoint (Creación de un punto de conexión de VPC) en la Guía de AWS PrivateLink .

Cree un punto final de VPC para AWS FIS con el siguiente nombre de servicio:. `com.amazonaws.region.fis`

Si habilita el DNS privado para el punto final, puede realizar solicitudes de API al AWS FIS utilizando su nombre de DNS predeterminado para la región, por ejemplo,. `fis.us-east-1.amazonaws.com`

Cree una política de puntos finales de VPC para FIS AWS

Puede adjuntar una política de punto final a su punto final de VPC que controle el acceso a AWS la FIS. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulte Uso de políticas de punto de conexión para controlar el acceso a puntos de conexión de VPC en la Guía de AWS PrivateLink .

Ejemplo: política de puntos finales de VPC para acciones de FIS específicas AWS

La siguiente política de puntos finales de la VPC otorga acceso a las acciones de la AWS FIS enumeradas en todos los recursos a todos los principales.

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "fis:ListExperimentTemplates",
            "fis:StartExperiment",
            "fis:StopExperiment",
            "fis:GetExperiment"
         ],
         "Resource":"*",
         "Principal":"*"
      }
   ]
}
```

Ejemplo: política de punto final de VPC que deniega el acceso desde un punto de conexión específico Cuenta de AWS

La siguiente política de puntos finales de VPC deniega el Cuenta de AWS acceso especificado a todas las acciones y recursos, pero concede a todos los demás el Cuentas de AWS acceso a todas las acciones y recursos.

```
{
   "Statement":[
      {
         "Effect": "Allow",
         "Action": "*",
         "Resource": "*",
         "Principal": "*"
      },
      {
         "Effect":"Deny",
         "Action": "*",
         "Resource": "*",
         "Principal": {
           "AWS": [ "123456789012" ]
         }
      }
   ]
}
```

Etiquetado de los recursos de AWS FIS

Una tag (etiqueta) es una etiqueta de metadatos que usted o AWS asignan a un recurso de AWS. Cada etiqueta consta de una clave y un valor. En el caso de etiquetas que usted asigna, debe definir la clave y el valor. Por ejemplo, podría definir la clave como `purpose` y el valor como `test` para un recurso.

Las etiquetas le ayudan a hacer lo siguiente:

- Identificar y organizar sus recursos de AWS. Muchos servicios de AWS admiten el etiquetado, por lo que puede asignar la misma etiqueta a los recursos de diferentes servicios para indicar que los recursos están relacionados.
- Controle el acceso a los recursos de AWS. Para obtener más información, consulte Control del acceso mediante etiquetas en la Guía del usuario de IAM de .

Restricciones de etiquetado

Las siguientes restricciones básicas se aplican a las etiquetas en recursos de AWS FIS:

- Número máximo de etiquetas que puede asignar a un recurso: 50
- Longitud máxima de la clave: 128 caracteres Unicode
- Longitud máxima del valor: 256 caracteres Unicode
- Caracteres válidos para claves y valores: a-z, A-Z, 0-9, espacio y los siguientes caracteres: _ . : / = + - y @
- Las claves y los valores distinguen entre mayúsculas y minúsculas
- No puede utilizar `aws:` como prefijo para claves, ya que su uso está reservado a AWS.

Trabajo con etiquetas

Los siguientes recursos de AWS Fault Injection Service (AWS FIS) admiten el etiquetado:

- Acciones
- Experimentos
- Plantillas de experimento

Puede utilizar la consola para trabajar con etiquetas para experimentos y plantillas de experimento. Para obtener más información, consulte los siguientes temas:

- Etiquetado de un experimento
- Etiquetado de plantillas de experimento

Puede utilizar los siguientes comandos de la AWS CLI para trabajar con etiquetas para acciones, experimentos y plantillas de experimento:

- tag-resource: adición de etiquetas a un recurso.
- untag-resource: eliminación de las etiquetas de un recurso.
- list-tags-for-resource— Listar las etiquetas de un recurso específico.

Cuotas y limitaciones del servicio de inyección de AWS averías

Cuenta de AWS Tiene cuotas predeterminadas, anteriormente denominadas límites, para cada AWS servicio. A menos que se indique lo contrario, cada cuota es específica de la región de . Puede solicitar el aumento de algunas cuotas, pero no de todas.

Para ver las cuotas de AWS FIS, abra la consola Service Quotas. En el panel de navegación, elija Servicios de AWS y seleccione AWS Fault Injection Service.

Para solicitar un aumento de cuota, consulte Solicitud de un aumento de cuota en la Guía de usuario de Service Quotas.

Cuenta de AWS Tiene las siguientes cuotas relacionadas con la AWS FIS.

Nombre	Valor predeterminado	Ajusta	Descripción
Duración de la acción en horas	Cada región admitida: 12	No	El número máximo de horas permitidas para ejecutar una única acción en esta cuenta en la región actual.
Acciones por plantilla de experimento	Cada región admitida: 20	No	El número máximo de acciones que puede crear en una plantilla de experimento en esta cuenta en la región actual.
Experimentos activos	Cada región admitida: 5	No	El número máximo de experimentos activos que puede ejecutar simultáneamente en esta cuenta en la región actual.

Nombre	Valor predeterminado	Ajusta	Descripción
Retención de datos de experimentos completados en días	Cada región admitida: 120	No	El número máximo de días permitido al AWS FIS para conservar los datos sobre los experimentos finalizados en esta cuenta en la región actual.
Duración del experimento en horas	Cada región admitida: 12	No	El número máximo de horas permitidas para ejecutar un experimento en esta cuenta en la región actual.
Plantillas de experimento	Cada región admitida: 500	No	El número máximo de plantillas de experimento que puede crear en esta cuenta en la región actual.
Número máximo de listas de prefijos gestionadas en aws:network: -region-connectivity route-table-disrupt-cross	Cada región admitida: 15	No	El número máximo de listas de prefijos administradas que aws:network: -region-connectivity permitirá por acción. route-table-disrupt-cross
Número máximo de tablas de rutas en aws:network: -region-connectivity route-table-disrupt-cross	Cada región admitida: 10	No	El número máximo de tablas de enrutamiento que aws:network: route-table-disrupt-cross -region-connectivity permitirá por acción.

Nombre	Valor predeterminado	Ajusta	Descripción
Número máximo de rutas en aws:network: -region-connectivity route-table-disrupt-cross	Cada región admitida: 200	No	El número máximo de rutas que aws:network: route-table-disrupt-cross -region-connectivity permitirá por acción.
Acciones paralelas por experimento	Cada región admitida: 10	No	El número máximo de acciones que puede ejecutar en paralelo en un experimento en esta cuenta en la región actual.
Condiciones de parada por plantilla de experimento	Cada región admitida: 5	No	El número máximo de condiciones de parada que puede añadir a una plantilla de experimento en esta cuenta en la región actual.
Grupos de Auto Scaling de destino para aws:ec2: -error asg-insufficient-instance-capacity	Cada región admitida: 5	Sí	El número máximo de grupos de Auto Scaling a los que aws:ec2: asg-insufficient-instance-capacity -error puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Target Buckets para aws:s3: bucket-pause-replication	Cada región admitida: 20	Sí	El número máximo de cubos S3 a los que aws:s3: bucket-pause-replication puede apuntar al identificar los objetivos mediante etiquetas, por experimento.
Clústeres objetivo para aws:ecs: drain-container-instances	Cada región admitida: 5	Sí	El número máximo de clústeres a los que aws:ecs: drain-container-instances puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.
Clústeres objetivo para aws:rds: failover-db-cluster	Cada región admitida: 5	Sí	El número máximo de clústeres a los que aws:rds: failover-db-cluster puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.
Instancias de base de datos de destino para aws:rds: reboot-db-instances	Cada región admitida: 5	Sí	El número máximo de instancias de base de datos a las que aws:rds: reboot-db-instances puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Instancias de destino para aws:ec2:reboot-instances	Cada región admitida: 5	Sí	El número máximo de instancias a las que aws:ec2:reboot-instances puede dirigirse al identificar objetivos mediante etiquetas, por experimento.
Instancias de destino para aws:ec2:stop-instances	Cada región admitida: 5	Sí	El número máximo de instancias a las que aws:ec2:stop-instances puede dirigirse al identificar objetivos mediante etiquetas, por experimento.
Instancias de destino para aws:ec2:terminate-instances	Cada región admitida: 5	Sí	El número máximo de instancias a las que aws:ec2:terminate-instances puede dirigirse al identificar objetivos mediante etiquetas, por experimento.
Instancias de destino para aws:ssm:send-command	Cada región admitida: 5	Sí	El número máximo de instancias a las que aws:ssm:send-command puede dirigirse al identificar objetivos mediante etiquetas, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Grupos de nodos objetivo para aws:eks:terminate-nodegroup-instances	Cada región admitida: 5	Sí	El número máximo de grupos de nodos a los que aws:eks: terminate -nodegroup-instances puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.
Target Pods para aws:eks: pod-cpu-stress	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks: pod-cpu-stress puede apuntar cuando identificas los objetivos mediante parámetros, por experimento.
Pods de destino para aws:eks:pod-delete	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks:pod-delete puede dirigirse al identificar objetivos utilizando parámetros, por experimento.
Target Pods para aws:eks: pod-io-stress	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks: pod-io-stress puede apuntar cuando identificas los objetivos mediante parámetros, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Target Pods para aws:eks: pod-memory-stress	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks: pod-memory-stress puede apuntar cuando identificas los objetivos mediante parámetros, por experimento.
Target Pods para aws:eks: pod-network-blackhole-port	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks: pod-network-blackhole-port puede apuntar cuando identificas los objetivos mediante parámetros, por experimento.
Target Pods para aws:eks: pod-network-latency	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks: pod-network-latency puede apuntar cuando identificas los objetivos mediante parámetros, por experimento.
Target Pods para aws:eks: pod-network-packet-loss	Cada región admitida: 50	Sí	El número máximo de pods a los que aws:eks: pod-network-packet-loss puede apuntar cuando identificas los objetivos mediante parámetros, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Objetivo ReplicationGroups para aws:elasticache: interrupt-cluster-az-power	Cada región admitida: 5	Sí	El número máximo al ReplicationGroups que aws:elasticache: interrupt-cluster-az-power puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.
Objetivo para aws:ec2: SpotInstances send-spot-instance-interruptions	Cada región admitida: 5	Sí	El número máximo al SpotInstances que aws:ec2: send-spot-instance-interruptions puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.
Subredes de destino para aws:network:disrupt-connectivity	Cada región admitida: 5	Sí	El número máximo de subredes a las que aws:network:disrupt-connectivity puede dirigirse al identificar objetivos mediante etiquetas, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Subredes de destino para aws:network: -region-connectivity route-table-disrupt-cross	Cada región admitida: 6	Sí	El número máximo de subredes a las que aws:network: route-table-disrupt-cross -region-connectivity puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.
Tareas de destino para aws:ecs:stop-task	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs:stop-task puede dirigirse al identificar objetivos mediante etiquetas, por experimento.
Tareas objetivo para aws:ecs: task-cpu-stress	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs: task-cpu-stress puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.
Tareas objetivo para aws:ecs: task-io-stress	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs: task-io-stress puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Tareas objetivo para aws:ecs: task-kill-process	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs: task-kill-process puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.
Tareas objetivo para aws:ecs: task-network-blackhole-port	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs: task-network-blackhole-port puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.
Tareas objetivo para aws:ecs: task-network-latency	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs: task-network-latency puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.
Tareas objetivo para aws:ecs: task-network-packet-loss	Cada región admitida: 5	Sí	El número máximo de tareas a las que aws:ecs: task-network-packet-loss puede dirigirse al identificar los objetivos mediante etiquetas o parámetros, por experimento.

Nombre	Valor predeterminado	Ajusta	Descripción
Objetivo para aws:network TransitGateways : -region-connectivity transit-gateway-disrupt-cross	Cada región admitida: 5	Sí	El número máximo de pasarelas de tránsito a las que aws:network: transit-gateway-disrupt-cross -region-connectivity puede dirigirse al identificar los objetivos mediante etiquetas, por experimento.
Configuraciones de cuenta de destino por plantillas de experimento	Cada región admitida: 10	Sí	Número máximo de configuraciones de cuenta de destino que puede crear para una plantilla de experimento en esta cuenta en la región actual.

El uso del AWS FIS está sujeto a las siguientes limitaciones adicionales:

Nombre	Limitación
Objetivos de acción `aws:elasticache:interrupt-cluster-az-power`	Limitado a 10 `aws:elasticache:redis-replicationgroup` clústeres dañados por cuenta, región y día. Para solicitar un aumento, cree un caso de soporte en la consola de AWS Support Center.

Historial del documento

En la siguiente tabla se describen las actualizaciones importantes de la documentación de la Guía del usuario de AWS Fault Injection Service.

Cambio	Descripción	Fecha
Nuevo modo de acciones (opción de experimento)	Puedes configurar el modo de acciones `skip-all` para generar una vista previa del objetivo antes de ejecutar un experimento.	13 de marzo de 2024
AWS actualizaciones de políticas gestionadas	AWS La FIS actualizó las políticas gestionadas existente s.	25 de enero de 2024
Nuevos escenarios y acciones	Ahora puede utilizar los escenarios AWS FIS entre regiones: conectividad y disponibilidad en zonas de disponibilidad: interrupción de energía.	30 de noviembre de 2023
Nueva acción	Ahora también puede usar la acción aws:ec2:asg-insuff icient-instance-capacity-error.	30 de noviembre de 2023
Nueva acción	Ahora también puede usar la acción aws:ec2:api-insuff icient-instance-capacity-error.	30 de noviembre de 2023
Nueva acción	Ahora también puede usar la acción aws:network:route-table-disrupt-cross-region-connectivity.	30 de noviembre de 2023

Nueva acción	Ahora también puede usar la acción aws:network:transit-gateway-disrupt-cross-region-connectivity.	30 de noviembre de 2023
Nueva acción	Ahora también puede usar la acción aws:dynamodb:encrypted-global-table-pause-replication.	30 de noviembre de 2023
Nueva acción	Ahora también puede usar la acción aws:s3:bucket-pause-replication.	30 de noviembre de 2023
Nueva acción	Ahora también puede usar la acción aws:elasticache:interrupt-cluster-az-power.	30 de noviembre de 2023
Nuevas opciones de experimento	Ahora puede utilizar las opciones de experimento AWS del FIS para segmentar cuentas y resolver objetivos vacíos.	27 de noviembre de 2023
Cambio de nombre de FIS AWS	Se actualizó el nombre del servicio a AWS Fault Injection Service.	15 de noviembre de 2023
AWS actualizaciones de políticas gestionadas	AWS La FIS actualizó las políticas gestionadas existentes.	13 de noviembre de 2023
Nueva biblioteca de escenarios	Ahora puede utilizar la función de biblioteca de escenarios del AWS FIS.	7 de noviembre de 2023

Nuevo programador de experimentos	Ahora puede utilizar la función de programación de experimentos del AWS FIS.	7 de noviembre de 2023
AWS actualizaciones de políticas gestionadas	AWS La FIS actualizó las políticas gestionadas existente s.	2 de junio de 2023
Nuevas acciones	Puede utilizar las nuevas acciones aws:ecs:task y aws:eks:pod.	1 de junio de 2023
AWS actualizaciones de políticas gestionadas	AWS La FIS actualizó las políticas gestionadas existente s.	1 de junio de 2023
Nuevo documento de SSM preconfigurado	Puede utilizar el siguiente documento SSM preconfig urado: -Run-Disk-Fill. AWSFIS	28 de abril de 2023
Nueva acción	Puede utilizar la acción aws:ebs:pause-volume-io para pausar la E/S entre los volúmenes de destino y las instancias a las que se asocian.	27 de enero de 2023
Nueva acción	Puede utilizar la acción aws:network:disrupt-connect ivity para denegar tipos específicos de tráfico a las subredes de destino.	26 de octubre de 2022

Nueva acción	Puedes usar la aws:eks:inject-kubernetes-custom-resource acción para ejecutar un experimento ChaosMesh o un experimento de Litmus en un único clúster objetivo.	7 de julio de 2022
Registro de experimentos	Puedes configurar tus plantillas de experimentos para enviar los registros de actividad del experimento a CloudWatch Logs o a un bucket de S3.	28 de febrero de 2022
Nuevas notificaciones	Cuando el estado de un experimento cambia, el AWS FIS emite una notificación. Estas notificaciones están disponibles como eventos a través de Amazon EventBridge.	24 de febrero de 2022
Nueva acción	Puede utilizar la acción aws:ecs:stop-task para detener la tarea especificada.	9 de febrero de 2022
Nueva acción	Puede utilizar la acción aws:cloudwatch:assert-alarm-state para comprobar que las alarmas especificadas se encuentran en uno de los estados de alarma especificados.	5 de noviembre de 2021

Nuevos documentos de SSM preconfigurados	Puede utilizar los siguientes documentos SSM preconfig urados: AWSFIS -Run-IO-S tress, -Run-Network-Black hold-Port, -Run-Network-Latency-Sources, -Run-Netw ork-Packet-Loss y AWSFIS -Run-Network-Packet-Loss-Sources. AWSFIS AWSFIS AWSFIS	4 de noviembre de 2021
Nueva acción	Puede utilizar la acción aws:ec2:send-spot-instance-interruptions para enviar un aviso de interrupción de la instancia de spot a las instancias spot de destino y, a continuación, interrumpirlas.	20 de octubre de 2021
Nueva acción	Puede utilizar la acción aws:ssm:start-automation-execution para iniciar la ejecución de un manual de procedimientos de Automatio n.	17 de septiembre de 2021
Versión inicial	La versión AWS inicial de la guía del usuario del servicio de inyección de fallos.	15 de marzo de 2021

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.