



Guida per l'utente

AWS Servizio di iniezione dei guasti



AWS Servizio di iniezione dei guasti: Guida per l'utente

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Che cos'è la AWS FIS?	1
Concetti	1
Azioni	2
Destinazioni	2
Condizioni di arresto	2
Supportato Servizi AWS	3
AWS Accedere a FIS	3
Prezzi	3
Pianifica i tuoi esperimenti	4
Principi e linee guida di base	4
Linee guida sulla pianificazione degli esperimenti	5
Componenti del modello di esperimento	7
Sintassi del modello	7
Nozioni di base	8
Azioni	8
Sintassi dell'azione	9
Identificatori di azione	10
Parametri dell'operazione	10
Obiettivi d'azione	11
Durata operazione	12
Esempio di azioni	12
Target	15
Sintassi di Target	16
Tipi di risorse	17
Identifica le risorse target	18
Modalità di selezione	23
Obiettivi di esempio	23
Filtri di esempio	25
Condizioni di arresto	29
Sintassi della condizione di interruzione	30
Ulteriori informazioni	31
Ruolo dell'esperimento	31
Prerequisiti	32
Opzione 1: creare un ruolo sperimentale e allegare una policy AWS gestita	33

Opzione 2: creare un ruolo sperimentale e aggiungere un documento programmatico in linea	34
Configurazione del rapporto dell'esperimento	36
Sintassi di configurazione del report dell'esperimento	38
Autorizzazioni per il rapporto sull'esperimento	40
Procedure ottimali per i report sugli esperimenti	42
Opzioni di esperimento	42
Targeting dell'account	43
Modalità di risoluzione degli obiettivi vuota	45
modalità Azioni	45
Riferimento alle azioni	47
Azioni di iniezione dei guasti	48
aws:fis:inject-api-internal-error	48
aws:fis:inject-api-throttle-error	49
aws:fis:inject-api-unavailable-error	50
Azione di ripristino	50
aws:arc:start-zonal-autoshift	50
Attendi l'azione	52
aws:fis:wait	52
CloudWatch Azioni Amazon	52
aws:cloudwatch:assert-alarm-state	52
Azioni Amazon DynamoDB	53
aws:dynamodb:global-table-pause-replication	53
Azioni SQL di Amazon Aurora	57
aws:dsq:cluster-connection-failure	57
Azioni Amazon EBS	58
aws:ebs:pause-volume-io	58
aws:ebs:volume-io-latency	59
Azioni di Amazon EC2	60
aws:ec2:api-insufficient-instance-capacity-error	60
aws:ec2:asg-insufficient-instance-capacity-error	61
aws:ec2:reboot-instances	62
aws:ec2:send-spot-instance-interruptions	63
aws:ec2:stop-instances	64
aws:ec2:terminate-instances	65
Azioni Amazon ECS	65

aws:ecs:drain-container-instances	66
aws:ecs:stop-task	66
aws:ecs:task-cpu-stress	67
aws:ecs:task-io-stress	68
aws:ecs:task-kill-process	69
aws:ecs:task-network-blackhole-port	69
aws:ecs:task-network-latency	70
aws:ecs:task-network-packet-loss	72
Azioni Amazon EKS	73
aws:eks:inject-kubernetes-custom-resource	74
aws:eks:pod-cpu-stress	75
aws:eks:pod-delete	76
aws:eks:pod-io-stress	77
aws:eks:pod-memory-stress	78
aws:eks:pod-network-blackhole-port	80
aws:eks:pod-network-latency	81
aws:eks:pod-network-packet-loss	82
aws:eks:terminate-nodegroup-instances	84
ElastiCache Azioni Amazon	84
aws:elasticache:replicationgroup-interrupt-az-power	84
Azioni di Amazon Kinesis Data Streams	85
aws:kinesis:stream-provisioned-throughput-exception	86
aws:kinesis:stream-expired-iterator-exception	86
AWS Lambda azioni	87
aws:lambda:invocation-add-delay	87
aws:lambda:invocation-error	88
aws:lambda:invocation-http-integration-response	89
Azione Amazon MemoryDB	90
aws:memorydb:multi-region-cluster-pause-replication	90
Azioni di rete	91
aws:network:disrupt-connectivity	91
aws:network:route-table-disrupt-cross-region-connectivity	92
aws:network:transit-gateway-disrupt-cross-region-connectivity	94
aws:network:disrupt-vpc-endpoint	95
Azioni Amazon RDS	96
aws:rds:failover-db-cluster	96

aws:rds:reboot-db-instances	96
Operazioni di Amazon S3	97
aws:s3:bucket-pause-replication	97
Azioni di Systems Manager	98
aws:ssm:send-command	99
aws:ssm:start-automation-execution	100
AWS Direct Connect azioni	100
aws:directconnect:virtual-interface-disconnect	101
Azioni relative ai documenti SSM	101
Usa l'azione aws:ssm:send-command	102
Documenti FIS SSM preconfigurati AWS	103
Esempi	113
Limitazioni	114
script di rollback	114
Risoluzione dei problemi	115
azioni delle attività ECS	116
Azioni	117
Limitazioni	117
Requisiti	117
Versione di riferimento dello script	120
Esempio di modello di esperimento	123
Azioni EKS Pod	124
Azioni	125
Limitazioni	125
Requisiti	126
Crea un ruolo sperimentale	127
Configurazione dell'account di servizio Kubernetes	127
Concedi agli utenti e ai ruoli IAM l'accesso a Kubernetes APIs	128
Immagini del contenitore Pod	129
Esempio di modello di esperimento	132
AWS Lambda azioni	133
Azioni	133
Limitazioni	133
Prerequisiti	134
Configurazione di funzioni Lambda	135
Configura un esperimento AWS FIS	136

Registrazione dei log	136
Argomenti avanzati	138
AWS FIS Versioni di estensione Lambda	144
Gestione dei modelli di esperimento	148
Crea un modello di esperimento	148
Visualizza i modelli di esperimenti	151
Genera un'anteprima del bersaglio	152
Inizia un esperimento da un modello	152
Aggiorna un modello di esperimento	153
Modelli di esperimenti con tag	154
Eliminare un modello di esperimento	154
Modelli di esempio	155
Interrompi le EC2 istanze in base ai filtri	155
Interrompe un numero specificato di istanze EC2	157
Esegui un documento FIS SSM preconfigurato AWS	158
Esegui un runbook di automazione predefinito	159
Limita le azioni API sulle EC2 istanze con il ruolo IAM di destinazione	159
Stress test della CPU dei pod in un cluster Kubernetes	161
Eccezione di throughput assegnata per un numero specificato di Kinesis Data Streams	163
Sperimenta un esempio di autorizzazioni di ruolo	164
Gestione degli esperimenti	165
Avviate un esperimento	165
Visualizza i tuoi esperimenti	166
Stati dell'esperimento	166
Stati d'azione	167
Etichetta un esperimento	167
Interrompere un esperimento	168
Elenca gli obiettivi risolti	168
Esercitazioni	170
Arresto e avvio dell'istanza di test	170
Prerequisiti	170
Fase 1: Creare un modello di esperimento	170
Fase 2: Avviare l'esperimento	174
Fase 3: Tieni traccia dell'avanzamento dell'esperimento	174
Fase 4: Verifica il risultato dell'esperimento	174
Fase 5: rimozione	175

Esegui lo stress della CPU su un'istanza	176
Prerequisiti	176
Passaggio 1: creare un CloudWatch allarme per una condizione di arresto	177
Passaggio 2: crea un modello di esperimento	177
Fase 3: Avviare l'esperimento	180
Fase 4: Tieni traccia dei progressi dell'esperimento	180
Fase 5: Verifica i risultati dell'esperimento	181
Fase 6: pulizia	175
Interruzioni delle istanze Test Spot	183
Prerequisiti	183
Fase 1: Creare un modello di esperimento	185
Fase 2: Avviare l'esperimento	187
Fase 3: Tieni traccia dell'avanzamento dell'esperimento	188
Fase 4: Verifica il risultato dell'esperimento	188
Fase 5: rimozione	189
Simula un evento di connettività	189
Prerequisiti	190
Fase 1: Creare un modello di esperimento FIS AWS	191
Fase 2: Eseguire il ping di un endpoint Amazon S3	193
Fase 3: Iniziate l'esperimento AWS FIS	193
Fase 4: Tieni traccia dei progressi dell'esperimento AWS FIS	194
Fase 5: verifica l'interruzione della rete Amazon S3	194
Fase 5: rimozione	195
Pianifica un esperimento ricorrente	195
Prerequisiti	196
Fase 1: Creare un ruolo e una policy IAM	196
Fase 2: Creare uno Amazon EventBridge Scheduler	198
Passaggio 3: verifica l'esperimento	199
Fase 4: pulizia	199
Lavorare con la libreria di scenari	200
Visualizzazione di uno scenario	200
Utilizzando uno scenario	201
Esportazione di uno scenario	202
Riferimento agli scenari	202
AZ Availability: Power Interruption	206
AZ: Application Slowdown	221

Cross-AZ: Traffic Slowdown	228
Cross-Region: Connectivity	235
Lavorare con esperimenti con più account	250
Concetti	251
Best practice	251
Prerequisiti	251
Permissions	252
Condizioni di arresto (opzionale)	255
Leve di sicurezza per esperimenti con più account (opzionale)	255
Crea un modello di esperimento con più account	255
Aggiornare la configurazione di un account di destinazione	257
Eliminare la configurazione di un account di destinazione	257
Pianificazione degli esperimenti	259
Crea un ruolo di scheduler	259
Crea una pianificazione dell'esperimento	263
Per aggiornare la pianificazione utilizzando la console	264
Aggiorna la pianificazione di un esperimento	264
Disabilita o elimina la pianificazione di un esperimento	265
Leve di sicurezza	266
Concetti relativi alle leve di sicurezza	266
Risorsa della leva di sicurezza	267
Lavorare con le leve di sicurezza	267
Visualizzazione di una leva di sicurezza	267
Inserimento di una leva di sicurezza	268
Disinnestare una leva di sicurezza	268
Esperimenti di monitoraggio	270
Monitora utilizzando CloudWatch	271
Monitora gli AWS esperimenti FIS	271
AWS Metriche di utilizzo FIS	272
Monitora utilizzando EventBridge	273
Registrazione degli esperimenti	275
Autorizzazioni	275
Schema di registro	275
Destinazioni di registro	277
Esempi di record di registro	277
Abilita la registrazione degli esperimenti	282

Disabilita la registrazione degli esperimenti	283
Registra le chiamate API con AWS CloudTrail	283
Usa CloudTrail	283
Comprendi le AWS voci dei file di registro FIS	284
Risoluzione dei problemi	289
Codici di errore	289
Sicurezza	292
Protezione dei dati	292
Crittografia dei dati a riposo	293
Crittografia dei dati in transito	294
Gestione dell'identità e degli accessi	294
Destinatari	294
Autenticazione con identità	295
Gestione dell'accesso tramite policy	296
Come funziona AWS Fault Injection Service con IAM	298
Esempi di policy	303
Utilizzo dei ruoli collegati ai servizi	313
AWS politiche gestite	316
Sicurezza dell'infrastruttura	321
AWS PrivateLink	322
Considerazioni	322
Creazione di un endpoint VPC dell'interfaccia	322
Creazione di una policy di endpoint VPC	323
Tagging delle risorse	325
Restrizioni di tagging	325
Lavora con i tag	325
Quote e limiti	327
Cronologia dei documenti	342
.....	cccl

Cos'è il servizio di iniezione dei AWS guasti?

AWS AWS Fault Injection Service (FIS) è un servizio gestito che consente di eseguire esperimenti di iniezione dei guasti sui AWS carichi di lavoro. L'iniezione dei guasti si basa sui principi dell'ingegneria del caos. Questi esperimenti stressano un'applicazione creando eventi dirompenti in modo da poter osservare come risponde l'applicazione. È quindi possibile utilizzare queste informazioni per migliorare le prestazioni e la resilienza delle applicazioni in modo che si comportino come previsto.

Per utilizzare AWS FIS, è necessario configurare ed eseguire esperimenti che aiutano a creare le condizioni reali necessarie per scoprire problemi applicativi che potrebbero essere difficili da trovare in altro modo. AWS FIS fornisce modelli che generano interruzioni e i controlli e i guardrail necessari per eseguire gli esperimenti in produzione, ad esempio ripristinando automaticamente o interrompendo l'esperimento se vengono soddisfatte condizioni specifiche.

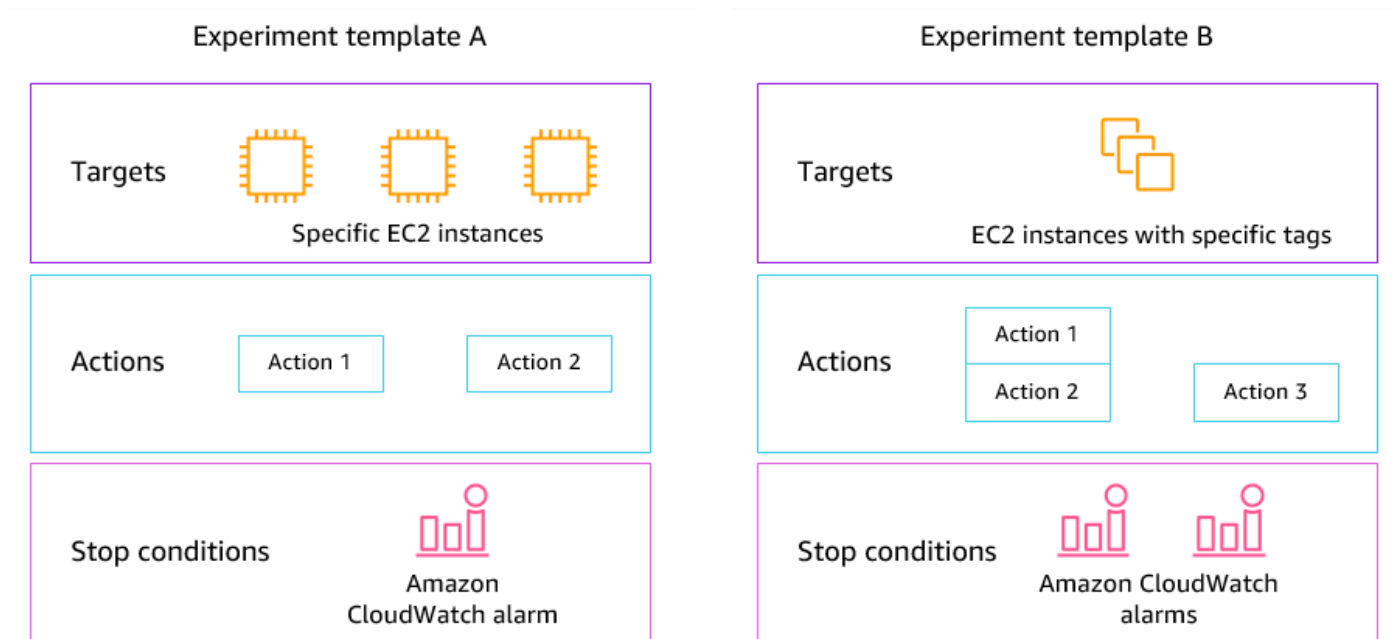
Important

AWS FIS esegue azioni reali su risorse reali del sistema. AWS Pertanto, prima di utilizzare AWS FIS per eseguire esperimenti in produzione, si consiglia vivamente di completare una fase di pianificazione ed eseguire gli esperimenti in un ambiente di preproduzione.

Per ulteriori informazioni sulla pianificazione dell'esperimento, vedere [Test Reliability e. Pianificazione degli AWS esperimenti FIS](#) Per ulteriori informazioni sulla AWS FIS, vedere [AWS Fault Injection Service](#).

AWS Concetti FIS

Per utilizzare AWS FIS, si eseguono esperimenti sulle AWS risorse per testare la teoria sulle prestazioni di un'applicazione o di un sistema in condizioni di guasto. Per eseguire esperimenti, è innanzitutto necessario creare un modello di esperimento. Un modello di esperimento è il modello del tuo esperimento. Contiene le azioni, gli obiettivi e le condizioni di interruzione dell'esperimento. Dopo aver creato un modello di esperimento, puoi utilizzarlo per eseguire un esperimento. Mentre l'esperimento è in esecuzione, puoi monitorarne l'avanzamento e visualizzarne lo stato. Un esperimento è completo quando tutte le azioni dell'esperimento sono state eseguite.



Azioni

Un'azione è un'attività che AWS FIS esegue su una AWS risorsa durante un esperimento. AWS FIS fornisce una serie di azioni preconfigurate in base al tipo di risorsa. AWS Ogni azione viene eseguita per una durata specificata durante un esperimento o finché l'esperimento non viene interrotto. Le azioni possono essere eseguite in sequenza o simultaneamente (in parallelo).

Destinazioni

Un obiettivo è una o più AWS risorse su cui la AWS FIS esegue un'azione durante un esperimento. È possibile scegliere risorse specifiche oppure selezionare un gruppo di risorse in base a criteri specifici, come tag o stato.

Condizioni di arresto

AWS FIS fornisce i controlli e i guardrail necessari per eseguire esperimenti in sicurezza sui carichi di lavoro. AWS Una condizione di arresto è un meccanismo per interrompere un esperimento se raggiunge una soglia definita come CloudWatch allarme Amazon. Se viene attivata una condizione di arresto mentre l'esperimento è in corso, AWS FIS interrompe l'esperimento.

Supportato Servizi AWS

AWS FIS fornisce azioni preconfigurate per tipi specifici di obiettivi tra diversi AWS servizi. Per l'elenco dei servizi supportati e delle relative azioni, vedere il riferimento alle azioni [AWS FIS](#).

Per gli esperimenti con un solo account, le risorse target devono essere le Account AWS stesse dell'esperimento. È possibile eseguire esperimenti AWS FIS che hanno come obiettivo le risorse di un Account AWS account diverso utilizzando esperimenti AWS FIS con più account.

Per ulteriori informazioni, consulta [Azioni per AWS FIS](#).

AWS Accedere a FIS

È possibile lavorare con AWS FIS in uno dei seguenti modi:

- Console di gestione AWS— Fornisce un'interfaccia web che è possibile utilizzare per accedere a AWS FIS. Per ulteriori informazioni, consulta l'argomento relativo all'[utilizzo di Console di gestione AWS](#).
- AWS Command Line Interface (AWS CLI) — Fornisce comandi per un'ampia gamma di AWS servizi, incluso AWS FIS, ed è supportato su Windows, macOS e Linux. Per ulteriori informazioni, consulta [AWS Command Line Interface](#). Per ulteriori informazioni sui comandi per AWS FIS, vedere [fis](#) nella Guida ai comandi.AWS CLI
- AWS CloudFormation— Crea modelli che descrivano le tue AWS risorse. I modelli vengono utilizzati per effettuare il provisioning e gestire queste risorse come unità singola. Per ulteriori informazioni, vedere il [riferimento al tipo di risorsa AWS Fault Injection Service](#).
- AWS SDKs— Fornisce informazioni specifiche per la lingua APIs e si occupa di molti dettagli di connessione, come il calcolo delle firme, la gestione dei tentativi di richiesta e la gestione degli errori. Per ulteriori informazioni, consulta [AWS SDKs](#).
- API HTTPS: fornisce azioni API di basso livello che è possibile chiamare utilizzando le richieste HTTPS. Per ulteriori informazioni, consulta il [AWS Fault Injection Service API Reference](#).

Prezzi per AWS FIS

Ti viene addebitato il costo per minuto di esecuzione di un'azione, dall'inizio alla fine, in base al numero di account target per l'esperimento. Per ulteriori informazioni, consulta la sezione [Prezzi AWS FIS](#).

Pianificazione degli AWS esperimenti FIS

L'iniezione di guasti è il processo di stress di un'applicazione in ambienti di test o di produzione creando eventi dirompenti, come interruzioni del server o limitazione delle API. Osservando la risposta del sistema, è possibile quindi implementare miglioramenti. L'esecuzione di esperimenti sul sistema può aiutarvi a identificare i punti deboli del sistema in modo controllato, prima che tali debolezze si ripercuotano sui clienti che dipendono dal sistema. È quindi possibile risolvere i problemi in modo proattivo per evitare esiti imprevedibili.

Prima di iniziare a eseguire esperimenti di iniezione dei guasti utilizzando AWS FIS, si consiglia di acquisire familiarità con i seguenti principi e linee guida.

Important

AWS FIS esegue azioni reali su AWS risorse reali del sistema. Pertanto, prima di iniziare a utilizzare AWS FIS per eseguire esperimenti, consigliamo vivamente di completare prima una fase di pianificazione e un test in un ambiente di preproduzione o di test.

Indice

- [Principi e linee guida di base](#)
- [Linee guida sulla pianificazione degli esperimenti](#)

Principi e linee guida di base

Prima di iniziare gli esperimenti con la AWS FIS, procedi nel seguente modo:

1. Identifica la distribuzione di destinazione per l'esperimento: inizia identificando la distribuzione di destinazione. Se questo è il tuo primo esperimento, ti consigliamo di iniziare in un ambiente di pre-produzione o di test.
2. Esamina l'architettura dell'applicazione: devi assicurarti di aver identificato tutti i componenti dell'applicazione, le dipendenze e le procedure di ripristino per ciascun componente. Iniziate con la revisione dell'architettura dell'applicazione. A seconda dell'applicazione, fare riferimento al [AWS Well-Architected](#) Framework.

3. Definisci il comportamento dello stato stazionario: definisci il comportamento dello stato stazionario del sistema in termini di importanti metriche tecniche e aziendali, come latenza, carico della CPU, accessi non riusciti al minuto, numero di tentativi o velocità di caricamento della pagina.
4. Formulare un'ipotesi: formulare un'ipotesi su come si prevede che il comportamento del sistema cambi durante l'esperimento. La definizione di un'ipotesi segue questo formato:

Se *fault injection action* viene eseguito, non *business or technical metric impact* deve superare *value*.

Ad esempio, un'ipotesi per un servizio di autenticazione potrebbe essere la seguente: «Se la latenza di rete aumenta del 10%, si verifica un aumento inferiore all'1% degli errori di accesso». Una volta completato l'esperimento, si valuta se la resilienza dell'applicazione è in linea con le aspettative aziendali e tecniche.

Consigliamo inoltre di seguire queste linee guida quando si lavora con AWS FIS:

- Inizia sempre a sperimentare con il AWS FIS in un ambiente di test. Non iniziare mai con un ambiente di produzione. Man mano che avanzate negli esperimenti di iniezione dei guasti, potete sperimentare in altri ambienti controllati oltre all'ambiente di test.
- Rafforza la fiducia del tuo team nella resilienza delle tue applicazioni iniziando con piccoli e semplici esperimenti, come l'esecuzione dell'azione `aws:ec2:stop-instances` su un obiettivo.
- L'iniezione dei guasti può causare problemi reali. Procedete con cautela e assicuratevi che le prime iniezioni avvengano su base sperimentale, in modo che nessun cliente ne risenta.
- Testate, testate e testate ancora. L'iniezione dei guasti è pensata per essere implementata in un ambiente controllato con esperimenti ben pianificati. Ciò consente di acquisire fiducia nelle capacità dell'applicazione e degli strumenti di resistere a condizioni turbolente.
- Ti consigliamo vivamente di disporre di un eccellente programma di monitoraggio e avviso prima di iniziare. Senza di esso, non sarete in grado di comprendere o misurare l'impatto dei vostri esperimenti, il che è fondamentale per pratiche sostenibili di iniezione dei guasti.

Linee guida sulla pianificazione degli esperimenti

Con AWS FIS, esegui esperimenti sulle tue AWS risorse per testare la tua teoria sulle prestazioni di un'applicazione o di un sistema in condizioni di guasto.

Di seguito sono riportate le linee guida consigliate per pianificare gli esperimenti AWS FIS.

- Rivedi la cronologia delle interruzioni: rivedi le interruzioni e gli eventi precedenti del tuo sistema. Questo può aiutarvi a costruire un quadro dello stato generale e della resilienza del sistema. Prima di iniziare a eseguire esperimenti sul sistema, è necessario risolvere i problemi e i punti deboli noti del sistema.
- Identifica i servizi con il maggiore impatto: esamina i tuoi servizi e identifica quelli che hanno il maggiore impatto sugli utenti finali o sui clienti in caso di guasto o non funzionano correttamente.
- Identifica il sistema di destinazione: il sistema di destinazione è il sistema su cui eseguirai gli esperimenti. Se non conoscete la AWS FIS o non avete mai eseguito esperimenti di fault injection prima d'ora, vi consigliamo di iniziare eseguendo esperimenti su un sistema di produzione o di test.
- Consultate il vostro team: chiedete che cosa li preoccupa. Puoi formulare un'ipotesi per dimostrare o confutare le loro preoccupazioni. Puoi anche chiedere al tuo team per cosa non è preoccupato. Questa domanda può rivelare due errori comuni: l'errore dei costi irrecuperabili e l'errore della propensione alla conferma. La formulazione di un'ipotesi basata sulle risposte del team può contribuire a fornire maggiori informazioni sulla realtà dello stato del sistema.
- Esamina l'architettura dell'applicazione: esegui un'analisi del sistema o dell'applicazione e assicurati di aver identificato tutti i componenti dell'applicazione, le dipendenze e le procedure di ripristino per ogni componente.

Ti consigliamo di rivedere il AWS Well-Architected Framework. Il framework può aiutarti a creare un'infrastruttura sicura, ad alte prestazioni, resiliente ed efficiente per le tue applicazioni e i tuoi carichi di lavoro. Per ulteriori informazioni, consulta [AWS Well-Architected](#).

- Identifica i parametri applicabili: puoi monitorare l'impatto di un esperimento sulle tue AWS risorse utilizzando i CloudWatch parametri di Amazon. Puoi utilizzare questi parametri per determinare la linea di base o lo «stato stazionario» quando l'applicazione funziona in modo ottimale. Quindi, puoi monitorare queste metriche durante o dopo l'esperimento per determinarne l'impatto. Per ulteriori informazioni, consulta [Monitora AWS i parametri di utilizzo FIS con Amazon CloudWatch](#).
- Definisci una soglia di prestazioni accettabile per il tuo sistema: identifica la metrica che rappresenta uno stato stabile e accettabile per il tuo sistema. Utilizzerai questa metrica per creare uno o più CloudWatch allarmi che rappresentano una condizione di interruzione dell'esperimento. Se l'allarme viene attivato, l'esperimento viene interrotto automaticamente. Per ulteriori informazioni, consulta [Condizioni di arresto per la AWS FIS](#).

AWS Componenti del modello di esperimento FIS

Utilizzate i seguenti componenti per creare modelli di esperimenti:

Azioni

Le [azioni AWS FIS](#) che si desidera eseguire. Le azioni possono essere eseguite in un ordine prestabilito specificato dall'utente oppure possono essere eseguite contemporaneamente. Per ulteriori informazioni, consulta [Azioni](#).

Target

Le AWS risorse su cui viene eseguita un'azione specifica. Per ulteriori informazioni, consulta [Target](#).

Condizioni di arresto

Gli CloudWatch allarmi che definiscono una soglia oltre la quale le prestazioni dell'applicazione non sono accettabili. Se viene attivata una condizione di arresto durante l'esecuzione di un esperimento, AWS FIS interrompe l'esperimento. Per ulteriori informazioni, consulta [Condizioni di arresto](#).

Ruolo dell'esperimento

Un ruolo IAM che concede a AWS FIS le autorizzazioni necessarie in modo che possa eseguire esperimenti per tuo conto. Per ulteriori informazioni, consulta [Ruolo dell'esperimento](#).

Configurazione del rapporto sull'esperimento

La configurazione per abilitare i report sugli esperimenti. Per ulteriori informazioni, consulta [Configurazioni dei report degli esperimenti per AWS FIS](#).

Opzioni dell'esperimento

Opzioni per il modello di esperimento. Per ulteriori informazioni, consulta [Opzioni di esperimento per AWS FIS](#).

Il tuo account ha delle quote relative alla AWS FIS. Ad esempio, esiste una quota sul numero di azioni per modello di esperimento. Per ulteriori informazioni, consulta [Quote e limiti](#).

Sintassi del modello

Di seguito è riportata la sintassi per un modello di esperimento.

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

Per alcuni esempi, consulta [Modelli di esempio](#).

Nozioni di base

Per creare un modello di esperimento utilizzando il Console di gestione AWS, vedere [Crea un modello di esperimento](#).

Per creare un modello di esperimento utilizzando il AWS CLI, vedere [Esempi di modelli di esperimenti AWS FIS](#).

Azioni per AWS FIS

Per creare un modello di esperimento, è necessario definire una o più azioni. Per un elenco delle azioni predefinite fornite da AWS FIS, vedere [Riferimento alle azioni](#)

È possibile eseguire un'azione solo una volta durante un esperimento. Per eseguire la stessa azione AWS FIS più di una volta nello stesso esperimento, aggiungila al modello più volte utilizzando nomi diversi.

Indice

- [Sintassi dell'azione](#)
- [Identificatori di azione](#)
- [Parametri dell'operazione](#)
- [Obiettivi d'azione](#)
- [Durata operazione](#)

- [Esempio di azioni](#)

Sintassi dell'azione

Di seguito è riportata la sintassi di un'azione.

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

Quando si definisce un'azione, si fornisce quanto segue:

action_name

Un nome per l'operazione.

actionId

L'[identificatore dell'azione](#).

description

Descrizione facoltativa

parameters

Qualsiasi [parametro di azione](#).

startAfter

Qualsiasi azione che deve essere completata prima che questa azione possa iniziare. Altrimenti, l'azione viene eseguita all'inizio dell'esperimento.

targets

Qualsiasi [obiettivo d'azione](#).

Per alcuni esempi, consulta [the section called “Esempio di azioni”](#).

Identificatori di azione

Ogni azione AWS FIS ha un identificatore con il seguente formato:

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

Ad esempio, la seguente azione arresta le istanze Amazon EC2 di destinazione:

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

Per un elenco completo delle azioni, consulta la [AWS FIS Riferimento alle azioni](#)

Parametri dell'operazione

Alcune azioni AWS FIS dispongono di parametri aggiuntivi specifici dell'azione. Questi parametri vengono utilizzati per passare informazioni al AWS FIS quando l'azione viene eseguita.

AWS FIS supporta tipi di errore personalizzati utilizzando l'`aws:ssm:send-command`, che utilizza l'agente SSM e un documento di comando SSM per creare la condizione di errore sulle istanze di destinazione. L'`aws:ssm:send-command` include un `documentArn` parametro che prende come valore l'Amazon Resource Name (ARN) di un documento SSM. Specifichi i valori per i parametri quando aggiungi l'azione al modello di esperimento.

Per ulteriori informazioni sulla specificazione dei parametri per l'`aws:ssm:send-command`, consultate [Usa l'azione aws:ssm:send-command](#).

Ove possibile, è possibile inserire una configurazione di rollback (nota anche come azione successiva) all'interno dei parametri dell'azione. Una post-operazione ripristina una destinazione allo stato in cui si trovava prima dell'esecuzione dell'operazione stessa. L'azione di post viene eseguita dopo il tempo specificato nella durata dell'azione. Non tutte le azioni possono supportare le azioni di pubblicazione. Ad esempio, se l'azione interrompe un'istanza Amazon EC2, non è possibile ripristinare l'istanza dopo che è stata terminata.

Obiettivi d'azione

Un'azione viene eseguita sulle risorse di destinazione specificate. Dopo aver definito un obiettivo, è possibile specificarne il nome quando si definisce un'azione.

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

AWS Le azioni FIS supportano i seguenti tipi di risorse per gli obiettivi d'azione:

- AutoScalingGroups— Gruppi Amazon EC2 Auto Scaling
- Bucket: bucket Amazon S3
- Cluster: cluster Amazon EKS
- Cluster: cluster Amazon ECS, Aurora DSQL o Amazon Aurora DB
- DBInstances— Istanze database Amazon RDS
- Funzioni: funzioni AWS Lambda
- Istanze: istanze Amazon EC2
- KinesisStreams— Flussi di dati Kinesis
- ManagedResources— Cluster Amazon EKS, Application and Network Load Balancer di Amazon EC2 e gruppi Amazon EC2 Auto Scaling abilitati per lo spostamento zonale ARC.
- MultiRegionClusters— Cluster multiregionali Amazon MemoryDB
- Nodegroups — gruppi di nodi Amazon EKS
- Pods: pod Kubernetes su Amazon EKS
- ReplicationGroups ElastiCache — Gruppi di replica
- Ruoli: ruoli IAM
- SpotInstances— Istanze Spot Amazon EC2
- Sottoreti: sottoreti VPC
- Tabelle: tabelle globali di Amazon DynamoDB multiregione fortemente e alla fine coerenti
- Attività: attività di Amazon ECS
- TransitGateways— Gateway di transito
- VirtualInterfaces— Interfacce Direct Connect virtuali

- Volumi: volumi Amazon EBS
- VPCEndpoints— Endpoint Amazon VPC

Per alcuni esempi, consulta [the section called “Esempio di azioni”](#).

Durata operazione

Se un'azione include un parametro che è possibile utilizzare per specificare la durata dell'azione, per impostazione predefinita, l'azione viene considerata completa solo dopo la scadenza della durata specificata. Se hai impostato l'opzione dell'emptyTargetResolutionModeesperimento suskip, l'azione verrà completata immediatamente con lo stato «ignorato» quando nessun obiettivo è stato risolto. Ad esempio, se si specifica una durata di 5 minuti, AWS FIS considera l'azione completa dopo 5 minuti. Quindi avvia l'azione successiva, fino al completamento di tutte le azioni.

La durata può essere il periodo di tempo in cui viene mantenuta una condizione di azione o il periodo di tempo per il quale vengono monitorate le metriche. Ad esempio, la latenza viene iniettata per il periodo di tempo specificato. Per i tipi di azioni quasi istantanee, come l'interruzione di un'istanza, le condizioni di arresto vengono monitorate per la durata specificata.

Se un'azione include un'azione di post all'interno dei parametri dell'azione, l'azione di post viene eseguita dopo il completamento dell'azione. Il tempo necessario per completare l'azione successiva potrebbe causare un ritardo tra la durata dell'azione specificata e l'inizio dell'azione successiva (o la fine dell'esperimento, se tutte le altre azioni sono state completate).

Esempio di azioni

Di seguito sono riportati alcuni esempi di azioni.

Esempi

- [Arresta le istanze EC2](#)
- [Interrompi le istanze Spot](#)
- [Interrompi il traffico di rete](#)
- [Licenziate i dipendenti EKS](#)
- [Avvia ARC zonal autoshift](#)

Esempio: fermare le istanze EC2

La seguente azione arresta le istanze EC2 identificate utilizzando la destinazione denominata *targetInstances*. Dopo due minuti, riavvia le istanze di destinazione.

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

Esempio: istanze Interrupt Spot

L'azione seguente interrompe le istanze Spot identificate utilizzando la destinazione denominata *targetSpotInstances*. Attende due minuti prima di interrompere l'istanza Spot.

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

Esempio: interrompere il traffico di rete

L'azione seguente nega il traffico tra le sottoreti di destinazione e le sottoreti in altre zone di disponibilità.

```
"actions": {
```

```

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

```

Esempio: licenziare i lavoratori EKS

La seguente azione interrompe il 50% delle istanze EC2 nel cluster EKS identificate utilizzando il target denominato. *targetNodeGroups*

```

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

```

Esempio: avvia ARC zonal autoshift

L'azione seguente avvia uno spostamento automatico zonale ARC che sposta le risorse gestite lontano da for. *az-in-parameters duration-in-parameteres* Il tipo di risorsa ManagedResources viene utilizzato come chiave per il nome della destinazione nel modello di esperimento FIS. AWS

```

{
  "description": "aaa",
  "targets": {
    "ManagedResources-Target-1": {
      "resourceType": "aws:arc:zonal-shift-managed-resource",

```

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

Obiettivi per la AWS FIS

Un obiettivo è una o più AWS risorse su cui viene eseguita un'azione dal AWS AWS Fault Injection Service (FIS) durante un esperimento. Gli obiettivi possono trovarsi nello stesso account AWS dell'esperimento o in un altro account utilizzando un esperimento con più account. Per ulteriori informazioni su come indirizzare le risorse in un account diverso, consulta [Lavorare con esperimenti con più account](#)

Gli obiettivi vengono definiti quando si [crea un modello di esperimento](#). È possibile utilizzare lo stesso obiettivo per più azioni nel modello di esperimento.

AWS FIS identifica tutti gli obiettivi all'inizio dell'esperimento, prima di iniziare qualsiasi azione nel set di azioni. AWS FIS utilizza le risorse target selezionate per l'intero esperimento. Se non viene trovato alcun obiettivo, l'esperimento fallisce.

Indice

- [Sintassi di Target](#)
- [Tipi di risorse](#)
- [Identifica le risorse target](#)
 - [Filtri di risorse](#)
 - [Parametri delle risorse](#)
- [Modalità di selezione](#)
- [Obiettivi di esempio](#)
- [Filtri di esempio](#)

Sintassi di Target

Di seguito è riportata la sintassi per un obiettivo.

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

```
}  
}
```

Quando si definisce un obiettivo, si fornisce quanto segue:

`target_name`

Un nome per la destinazione.

`resourceType`

Il [tipo di risorsa](#).

`resourceArns`

Gli Amazon Resource Names (ARN) di risorse specifiche.

`resourceTags`

I tag applicati a risorse specifiche.

`parameters`

I [parametri](#) che identificano gli obiettivi utilizzando attributi specifici.

`filters`

Il [filtro delle risorse](#) elabora le risorse target identificate utilizzando attributi specifici.

`selectionMode`

La [modalità di selezione](#) per le risorse identificate.

Per alcuni esempi, consulta [the section called “Obiettivi di esempio”](#).

Tipi di risorse

Ogni azione AWS FIS viene eseguita su un tipo di AWS risorsa specifico. Quando si definisce un obiettivo, è necessario specificare esattamente un tipo di risorsa. Quando si specifica un obiettivo per un'azione, l'obiettivo deve essere il tipo di risorsa supportato dall'azione.

I seguenti tipi di risorse sono supportati da AWS FIS:

- `aws:arc: zonal-shift-managed-resource` — Una AWS risorsa registrata con ARC zonal shift
- `aws:directconnect:virtual-interface` — Un'interfaccia virtuale Direct Connect

- `aws:dsql:cluster` — Un cluster DSQL Amazon Aurora
- `aws:dynamodb:global-table` — Una tabella globale multiregionale di Amazon DynamoDB
- `aws:ec2:autoscaling-group` — Un gruppo Amazon Auto Scaling EC2
- `aws:ec2:ebs-volume` — Un volume Amazon EBS
- `aws:ec2:instance` — Un'istanza Amazon EC2
- `aws:ec2:spot-instance` — Un'istanza Amazon Spot EC2
- `aws:ec2:subnet` — Una sottorete Amazon VPC
- `aws:ec2:transit-gateway` — Un gateway di transito
- `aws:ec2:vpc-endpoint` — Un endpoint Amazon VPC
- `aws:ecs:cluster` — Un cluster Amazon ECS
- `aws:ecs:task` — Un'attività di Amazon ECS
- `aws:eks:cluster` — Un cluster Amazon EKS
- `aws:eks:nodegroup` — Un gruppo di nodi Amazon EKS
- `aws:eks:pod` — Un pod Kubernetes
- `aws:elasticache:replicationgroup` — ElastiCache Un gruppo di replica
- `aws:iam:role` — Un ruolo IAM
- `aws:kinesis:stream` — Un flusso di dati di Amazon Kinesis
- `aws:lambda:function` — AWS Lambda Una funzione
- `aws:memorydb: — multi-region-cluster` Un cluster multiregionale Amazon MemoryDB
- `aws:rds:cluster` — Un cluster Amazon Aurora DB
- `aws:rds:db` — Un'istanza database Amazon RDS
- `aws:s3:bucket` — Un bucket Amazon S3

Identifica le risorse target

Quando si definisce un obiettivo nella console AWS FIS, è possibile scegliere AWS risorse specifiche (di un tipo di risorsa specifico) da destinare. In alternativa, puoi consentire a AWS FIS di identificare un gruppo di risorse in base ai criteri che fornisci.

Per identificare le risorse di destinazione, è possibile specificare quanto segue:

- **Risorsa IDs:** la risorsa IDs di AWS risorse specifiche. Tutte le risorse IDs devono rappresentare lo stesso tipo di risorsa.

- Tag delle risorse: i tag applicati a AWS risorse specifiche.
- Filtri di risorse: il percorso e i valori che rappresentano risorse con attributi specifici. Per ulteriori informazioni, consulta [Filtri di risorse](#).
- Parametri delle risorse: i parametri che rappresentano le risorse che soddisfano criteri specifici. Per ulteriori informazioni, consulta [Parametri delle risorse](#).

Considerazioni

- Non è possibile specificare sia un ID di risorsa che un tag di risorsa per lo stesso obiettivo.
- Non è possibile specificare sia un ID di risorsa che un filtro di risorse per lo stesso obiettivo.
- Se specificate un tag di risorsa con un valore di tag vuoto, non è equivalente a un carattere jolly. Corrisponde alle risorse che hanno un tag con la chiave di tag specificata e un valore di tag vuoto.
- Se specificate più di un tag, tutti i tag specificati devono essere presenti sulla risorsa di destinazione affinché possa essere selezionata (AND).

Filtri di risorse

I filtri delle risorse sono query che identificano le risorse di destinazione in base a attributi specifici. AWS FIS applica la query all'output di un'azione API che contiene la descrizione canonica della AWS risorsa, in base al tipo di risorsa specificato. Le risorse con attributi che corrispondono alla query sono incluse nella definizione della destinazione.

Ogni filtro è espresso come percorso di attributo e valori possibili. Un percorso è una sequenza di elementi, separati da punti, che descrivono il percorso per raggiungere un attributo nell'output dell'azione Descrivi per una risorsa. Ogni periodo rappresenta l'espansione di un elemento. Ogni elemento deve essere espresso in Pascal, anche se l'output dell'azione Descrivi per una risorsa è nel caso camel. Ad esempio, dovresti usare `AvailabilityZone`, non `availablityZone` come elemento di attributo.

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

La seguente logica si applica a tutti i filtri di risorse:

- Se vengono forniti più filtri, inclusi i filtri con lo stesso percorso, tutti i filtri devono essere abbinati per selezionare una risorsa: AND
- Se vengono forniti più valori per un singolo filtro, è necessario abbinare un valore qualsiasi per selezionare una risorsa: OR
- Se vengono trovati più valori nella posizione del percorso della chiamata API di descrizione, è necessario abbinare un valore qualsiasi per selezionare una risorsa: OR
- Per abbinare le key/value coppie di tag, dovresti invece selezionare le risorse di destinazione per tag (vedi sopra).

La tabella seguente include le azioni e AWS CLI i comandi API che è possibile utilizzare per ottenere le descrizioni canoniche per ogni tipo di risorsa. AWS FIS esegue queste azioni per conto dell'utente per applicare i filtri specificati. La documentazione corrispondente descrive le risorse incluse nei risultati per impostazione predefinita. Ad esempio, la documentazione relativa `DescribeInstances` agli stati in cui le istanze terminate di recente potrebbe apparire nei risultati.

Tipo di risorsa	Azione API	AWS CLI comando
aws:arc:zonal-shift-managed-resource	ListManagedResources	list-managed-resources
aws:directconnect:virtual-interface	DescribeVirtualInterfaces	describe-virtual-interfaces
aws:ec2:autoscaling-group	DescribeAutoScalingGroups	describe-auto-scaling-groups
aws:ec2:ebs-volume	DescribeVolumes	describe-volumes
aws:ec2:instance	DescribeInstances	descrivi le istanze
aws:ec2:subnet	DescribeSubnets	describe-subnets
aws:ec2:transit-gateway	DescribeTransitGateways	describe-transit-gateways
aws:ec2:vpc-endpoint	DescribeVpcEndpoints	describe-vpc-endpoints
aws:ecs:cluster	DescribeClusters	describe-clusters

Tipo di risorsa	Azione API	AWS CLI comando
aws:ecs:task	DescribeTasks	descrizione-attività
aws:eks:cluster	DescribeClusters	describe-clusters
aws:eks:nodegroup	DescribeNodegroup	describe-nodegroup
aws:elasticache:replication group	DescribeReplicationGroups	describe-replication-groups
aws:iam:role	ListRoles	elenca-ruoli
aws:kinesis:stream	DescribeStreamSummary	describe-stream-summary
aws:lambda:function	ListFunctions	funzioni di elenco
aws:memorydb:multi-region-cluster	DescribeMultiRegionClusters	describe-multi-region-clusters
aws:rds:cluster	Descrizione DBClusters	describe-db-clusters
aws:rds:db	Descriva DBInstances	describe-db-instances
aws:s3:bucket	ListBuckets	list-buckets
aws:dynamodb:global-table	DescribeTable	descrivi la tabella
aws:dsq:cluster	GetCluster	get-cluster

Per alcuni esempi, consulta [the section called “Filtri di esempio”](#).

Parametri delle risorse

I parametri delle risorse identificano le risorse target in base a criteri specifici.

Il seguente tipo di risorsa supporta i parametri.

aws:ec2:ebs-volume

- `availabilityZoneIdentifier`— Il codice (ad esempio, us-east-1a) della zona di disponibilità che contiene i volumi di destinazione.

aws:ec2:subnet

- `availabilityZoneIdentifier`— Il codice (ad esempio, us-east-1a) o l'ID AZ (ad esempio, use1-az1) della zona di disponibilità che contiene le sottoreti di destinazione.
- `vpc`— Il VPC che contiene le sottoreti di destinazione. Non supporta più di un VPC per account.

aws:ecs:task

- `cluster`— Il cluster che contiene le attività di destinazione.
- `service`— Il servizio che contiene le attività di destinazione.

aws:eks:pod

- `availabilityZoneIdentifier` : Opzionale. La zona di disponibilità che contiene i pod di destinazione. Ad esempio, us-east-1d. Determiniamo la zona di disponibilità di un pod confrontando il relativo hostIP e il CIDR della sottorete del cluster.
- `clusterIdentifier`: obbligatorio Il nome o l'ARN del cluster EKS di destinazione.
- `namespace`: obbligatorio Lo spazio dei nomi Kubernetes dei pod di destinazione.
- `selectorType`: obbligatorio Il tipo di selettore. I valori possibili sono `labelSelector`, `deploymentName` e `podName`.
- `selectorValue`: obbligatorio Il valore del selettore. Questo valore dipende dal valore di `selectorType`
- `targetContainerName` : Opzionale. Il nome del contenitore di destinazione come definito nelle specifiche del pod. L'impostazione predefinita è il primo contenitore definito in ogni specifica del pod di destinazione.

aws:lambda:function

- `functionQualifier` : Opzionale. La versione o l'alias della funzione da scegliere come target. Se non viene specificato alcun qualificatore, tutte le chiamate verranno prese in considerazione per il targeting. Se viene specificato un alias con più versioni, tutte le versioni incluse nell'alias verranno prese in considerazione per il targeting purché vengano richiamate utilizzando un ARN contenente l'alias. Se `$LATEST` viene utilizzato l'alias speciale, le invocazioni alla funzione di base ARN e le invocazioni incluse `$LATEST` nell'ARN verranno prese in considerazione per l'iniezione di errori. Per ulteriori informazioni sulle versioni Lambda, consulta Manage [Lambda function versions](#) nella guida per l'utente.AWS Lambda

aws:rds:cluster

- `writerAvailabilityZoneIdentifiers` : Opzionale. Le zone di disponibilità del writer del cluster DB. I valori possibili sono: un elenco separato da virgole di identificatori delle zone di disponibilità, `all`.

aws:rds:db

- `availabilityZoneIdentifiers` : Opzionale. Le zone di disponibilità dell'istanza DB interessate. I valori possibili sono: un elenco separato da virgole di identificatori della zona di disponibilità, `all`.

aws:elasticache:replicationgroup

- `availabilityZoneIdentifier`: obbligatorio Il codice (ad esempio, `us-east-1a`) o l'ID AZ (ad esempio, `use1-az1`) della zona di disponibilità che contiene i nodi di destinazione.

Modalità di selezione

È possibile definire l'ambito delle risorse identificate specificando una modalità di selezione. AWS FIS supporta le seguenti modalità di selezione:

- **ALL**— Esegue l'azione su tutti gli obiettivi.
- **COUNT(n)**— Esegui l'azione sul numero specificato di bersagli, scelti casualmente tra i bersagli identificati. Ad esempio, **COUNT(1)** seleziona uno degli obiettivi identificati.
- **PERCENT(n)**— Esegue l'azione sulla percentuale specificata di bersagli, scelti casualmente tra i bersagli identificati. Ad esempio, **PERCENT(25)** seleziona il 25% degli obiettivi identificati.

Se si dispone di un numero dispari di risorse e si specifica il 50%, AWS FIS arrotonda per difetto. Ad esempio, se aggiungi cinque EC2 istanze Amazon come obiettivi e ambito al 50%, AWS FIS arrotonda per difetto a due istanze. Non puoi specificare una percentuale inferiore a una risorsa. Ad esempio, se aggiungi quattro EC2 istanze Amazon e l'ambito AWS è al 5%, FIS non può selezionare un'istanza.

Se definisci più obiettivi utilizzando lo stesso tipo di risorsa di destinazione, AWS FIS può selezionare la stessa risorsa più volte.

Indipendentemente dalla modalità di selezione utilizzata, se l'ambito specificato non identifica alcuna risorsa, l'esperimento fallisce.

Obiettivi di esempio

Di seguito sono riportati alcuni esempi di obiettivi.

Esempi

- [Istanze nel VPC specificato con i tag specificati](#)

- [Attività con i parametri specificati](#)

Esempio: istanze nel VPC specificato con i tag specificati

Le possibili destinazioni per questo esempio sono le EC2 istanze Amazon nel VPC specificato con il tag. env=prod La modalità di selezione specifica che AWS FIS sceglie uno di questi obiettivi a caso.

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

Esempio: attività con i parametri specificati

I possibili obiettivi per questo esempio sono le attività di Amazon ECS con il cluster e il servizio specificati. La modalità di selezione specifica che AWS FIS scelga uno di questi obiettivi a caso.

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

```
    }  
  }  
}
```

Filtri di esempio

Di seguito sono riportati alcuni esempi di filtri.

Esempi

- [EC2 Istanze](#)
- [Cluster DB](#)

Esempio: istanze EC2

Quando specifichi un filtro per un'azione che supporta il tipo di risorsa `aws:ec2:instance`, AWS FIS utilizza il comando EC2 `describe-instances` Amazon e applica il filtro per identificare le destinazioni.

Il `describe-instances` comando restituisce un output JSON in cui ogni istanza è una struttura.

Instances Di seguito è riportato un output parziale che include i campi contrassegnati con *italics*. Forniremo esempi che utilizzano questi campi per specificare un percorso di attributo dalla struttura dell'output JSON.

```
{  
  "Reservations": [  
    {  
      "Groups": [],  
      "Instances": [  
        {  
          "ImageId": "ami-0011111111111111",  
          "InstanceId": "i-00aaaaaaaaaaaaaaaa",  
          "InstanceType": "t2.micro",  
          "KeyName": "virginia-kp",  
          "LaunchTime": "2020-09-30T11:38:17.000Z",  
          "Monitoring": {  
            "State": "disabled"  
          },  
          "Placement": {  
            "AvailabilityZone": "us-east-1a",  
            "GroupName": "",  
            "Tenancy": "default"  
          }  
        }  
      ]  
    }  
  ]  
}
```

```

    },
    "PrivateDnsName": "ip-10-0-1-240.ec2.internal",
    "PrivateIpAddress": "10.0.1.240",
    "ProductCodes": [],
    "PublicDnsName": "ec2-203-0-113-17.compute-1.amazonaws.com",
    "PublicIpAddress": "203.0.113.17",
    "State": {
        "Code": 16,
        "Name": "running"
    },
    },
    "StateTransitionReason": "",
    "SubnetId": "subnet-aabbcc112233445566",
    "VpcId": "vpc-00bbbbbbbbbbbbbbbb",
    ...
    "NetworkInterfaces": [
    {
        ...
        "Groups": [
            {
                "GroupName": "sec-group-1",
                "GroupId": "sg-a0011223344556677"
            },
            {
                "GroupName": "sec-group-1",
                "GroupId": "sg-b9988776655443322"
            }
        ],
        ...
    },
    ...
    ],
    ...
    {
        ...
    }
    ],
    "OwnerId": "123456789012",
    "ReservationId": "r-aaaaabbbbb111111"
    },
    ...
    ]
}

```

Per selezionare le istanze in una zona di disponibilità specifica utilizzando un filtro di risorse, specifica il percorso dell'attributo `AvailabilityZone` e il codice per la zona di disponibilità come valore.

Esempio:

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

Per selezionare le istanze in una sottorete specifica utilizzando un filtro di risorse, specificate il percorso dell'attributo `SubnetId` e l'ID della sottorete come valore. Esempio:

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

Per selezionare le istanze che si trovano in uno stato specifico dell'istanza, specificate il percorso dell'attributo `Name` e uno dei seguenti nomi di stato come valore: `pending` | `running` | `shutting-down`. `terminated` `stopping` `stopped` Esempio:

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

Per selezionare le istanze a cui è associato uno qualsiasi dei numerosi gruppi di sicurezza, specificate un singolo filtro con il percorso dell'attributo per `GroupId` un gruppo di sicurezza multiplo. IDs Esempio:

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

```

```

        "sg-f1100110011001100"
    ]
}
],

```

Per selezionare le istanze a cui sono collegati tutti i gruppi di sicurezza, specificate più filtri con il percorso dell'attributo `GroupId` e un singolo ID del gruppo di sicurezza per ogni filtro. Esempio:

```

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

```

Esempio: cluster Amazon RDS (cluster DB)

Quando specifichi un filtro per un'azione che supporta il tipo di risorsa `aws:rds:cluster`, FIS AWS esegue il `describe-db-clusters` comando Amazon RDS e applica il filtro per identificare le destinazioni.

Il `describe-db-clusters` comando restituisce un output JSON simile al seguente per ogni cluster DB. Di seguito è riportato un output parziale che include i campi contrassegnati con *italics*. Forniremo esempi che utilizzano questi campi per specificare un percorso di attributo dalla struttura dell'output 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",
    ...
  }
]

```

Per applicare un filtro di risorse che restituisca solo i cluster DB che utilizzano un motore DB specifico, specifica il percorso dell'attributo Engine e il valore aurora-postgresql come mostrato nell'esempio seguente.

```

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

```

Per applicare un filtro di risorse che restituisca solo i cluster DB in una zona di disponibilità specifica, specificate il percorso e il valore dell'attributo come illustrato nell'esempio seguente.

```

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

```

Condizioni di arresto per la AWS FIS

AWS AWS Fault Injection Service (FIS) fornisce controlli e barriere per eseguire esperimenti in sicurezza sui carichi di lavoro. AWS Una condizione di arresto è un meccanismo per interrompere

un esperimento se raggiunge una soglia definita come CloudWatch allarme Amazon. Se durante un esperimento viene attivata una condizione di arresto, la AWS FIS interrompe l'esperimento. Non è possibile riprendere un esperimento interrotto.

Per creare una condizione di arresto, definite innanzitutto lo stato stazionario dell'applicazione o del servizio. Lo stato stazionario si verifica quando l'applicazione funziona in modo ottimale, definito in termini di metriche aziendali o tecniche. Ad esempio, latenza, carico della CPU o numero di tentativi. È possibile utilizzare lo stato stazionario per creare un CloudWatch allarme da utilizzare per interrompere un esperimento se l'applicazione o il servizio raggiunge uno stato in cui le prestazioni non sono accettabili. Per ulteriori informazioni, consulta [Using Amazon CloudWatch alarms](#) nella Amazon CloudWatch User Guide.

Il tuo account dispone di una quota di condizioni di interruzione che puoi specificare in un modello di esperimento. Per ulteriori informazioni, consulta [Quote e limitazioni per il servizio AWS Fault Injection](#).

Sintassi della condizione di interruzione

Quando crei un modello di esperimento, specifichi una o più condizioni di interruzione specificando gli CloudWatch allarmi che hai creato.

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

L'esempio seguente indica che il modello di esperimento non specifica una condizione di arresto.

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

Ulteriori informazioni

Per un tutorial che dimostra come creare un CloudWatch allarme e aggiungere una condizione di arresto a un modello di esperimento, vedi [Esegui lo stress della CPU su un'istanza](#).

Per ulteriori informazioni sulle CloudWatch metriche disponibili per i tipi di risorse supportati da AWS FIS, consulta quanto segue:

- [Monitora le tue istanze utilizzando CloudWatch](#)
- [Metriche di Amazon ECS CloudWatch](#)
- [Monitoraggio dei parametri di Amazon RDS tramite CloudWatch](#)
- [Monitoraggio dei parametri di Run Command tramite CloudWatch](#)

Ruoli IAM per AWS esperimenti FIS

AWS Identity and Access Management (IAM) è un AWS servizio che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Per utilizzare AWS FIS, è necessario creare un ruolo IAM che conceda a AWS FIS le autorizzazioni necessarie affinché AWS FIS possa eseguire esperimenti per conto dell'utente. Questo ruolo dell'esperimento viene specificato quando si crea un modello di esperimento. Per un esperimento con account singolo, la policy IAM per il ruolo dell'esperimento deve concedere l'autorizzazione a modificare le risorse specificate come obiettivi nel modello di esperimento. Per un esperimento con più account, il ruolo dell'esperimento deve concedere al ruolo di orchestratore l'autorizzazione ad assumere il ruolo IAM per ogni account di destinazione. Per ulteriori informazioni, consulta [Autorizzazioni per esperimenti con più account](#).

Ti consigliamo di seguire la pratica di sicurezza standard che prevede la concessione del privilegio minimo. Puoi farlo specificando risorse ARNs o tag specifici nelle tue politiche.

Per aiutarti a iniziare rapidamente a usare AWS FIS, forniamo politiche AWS gestite che puoi specificare quando crei un ruolo sperimentale. In alternativa, puoi anche utilizzare queste politiche come modello mentre crei i tuoi documenti politici in linea.

Indice

- [Prerequisiti](#)
- [Opzione 1: creare un ruolo sperimentale e allegare una policy AWS gestita](#)
- [Opzione 2: creare un ruolo sperimentale e aggiungere un documento programmatico in linea](#)

Prerequisiti

Prima di iniziare, installa AWS CLI e crea la politica di attendibilità richiesta.

Installa AWS CLI

Prima di iniziare, installa e configura la AWS CLI. Quando si configura il AWS CLI, vengono richieste le AWS credenziali. Gli esempi in questa procedura presuppongono che tu abbia configurato una regione predefinita. In caso contrario, aggiungi l'opzione `--region` a ogni comando. Per ulteriori informazioni, consulta [Installazione o aggiornamento della AWS CLI](#) e [Configurazione della AWS CLI](#).

Creare una politica di relazione di fiducia

Un ruolo sperimentale deve avere una relazione di fiducia che consenta al servizio AWS FIS di assumere il ruolo. Creare un file di testo denominato `fis-role-trust-policy.json` e aggiungere la seguente politica di relazione di fiducia.

JSON

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

Si consiglia di utilizzare le chiavi di condizione `aws:SourceAccount` e `aws:SourceArn` per proteggersi dal [problema del "confused deputy"](#). L'account di origine è il proprietario dell'esperimento e l'ARN di origine è l'ARN dell'esperimento. Ad esempio, dovresti aggiungere il seguente blocco di condizioni alla tua politica di fiducia.

```
"Condition": {
```

```

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

```

Aggiungi le autorizzazioni per assumere i ruoli degli account di destinazione (solo esperimenti con più account)

Per gli esperimenti con più account, sono necessarie autorizzazioni che consentano all'account orchestrator di assumere i ruoli di account di destinazione. Puoi modificare il seguente esempio e aggiungerlo come documento di policy in linea per assumere i ruoli degli account di destinazione:

```

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

```

Opzione 1: creare un ruolo sperimentale e allegare una policy AWS gestita

Utilizza una delle politiche AWS gestite di AWS FIS per iniziare rapidamente.

Per creare un ruolo sperimentale e allegare una policy AWS gestita

1. Verifica che esista una policy gestita per le azioni AWS FIS del tuo esperimento. Altrimenti, dovrai invece creare il tuo documento di policy in linea. Per ulteriori informazioni, consulta [the section called “AWS politiche gestite”](#).
2. Utilizzate il seguente comando [create-role](#) per creare un ruolo e aggiungere la politica di fiducia che avete creato nei prerequisiti.

```

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

```

3. Utilizza il [attach-role-policy](#) comando seguente per allegare la policy gestita AWS .

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

fis-policy-arn Dov'è uno dei seguenti:

- `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`

Opzione 2: creare un ruolo sperimentale e aggiungere un documento programmatico in linea

Usa questa opzione per azioni che non prevedono una policy gestita o per includere solo le autorizzazioni necessarie per il tuo esperimento specifico.

Per creare un esperimento e aggiungere un documento di policy in linea

1. Usa il seguente comando [create-role](#) per creare un ruolo e aggiungere la politica di fiducia che hai creato nei prerequisiti.

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

2. Crea un file di testo denominato `fis-role-permissions-policy.json` e aggiungi una politica di autorizzazioni. Per un esempio da utilizzare come punto di partenza, consultate quanto segue.

- Azioni di iniezione dei guasti: iniziate dalla seguente politica.

JSON

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

```

        "Sid": "AllowFISExperimentRoleFaultInjectionActions",
        "Effect": "Allow",
        "Action": [
            "fis:InjectApiInternalError",
            "fis:InjectApiThrottleError",
            "fis:InjectApiUnavailableError"
        ],
        "Resource": "arn:*:fis:*:*:experiment/*"
    }
}

```

- Azioni di Amazon EBS: inizia dalla seguente policy.

JSON

```

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

```

- Azioni di Amazon EC2: inizia dalla policy di [AWSFaultInjectionSimulatorEC2accesso](#).
- Azioni di Amazon ECS: inizia dalla [AWSFaultInjectionSimulatorECSAccesspolicy](#).
- Azioni di Amazon EKS: inizia dalla [AWSFaultInjectionSimulatorEKSAccesspolicy](#).
- Azioni di rete: inizia dalla [AWSFaultInjectionSimulatorNetworkAccesspolitica](#).
- Azioni di Amazon RDS: inizia dalla [AWSFaultInjectionSimulatorRDSAccesspolicy](#).
- Azioni di Systems Manager: inizia dalla [AWSFaultInjectionSimulatorSSMAccesspolicy](#).

3. Utilizza il [put-role-policy](#) comando seguente per aggiungere la politica di autorizzazioni creata nel passaggio precedente.

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

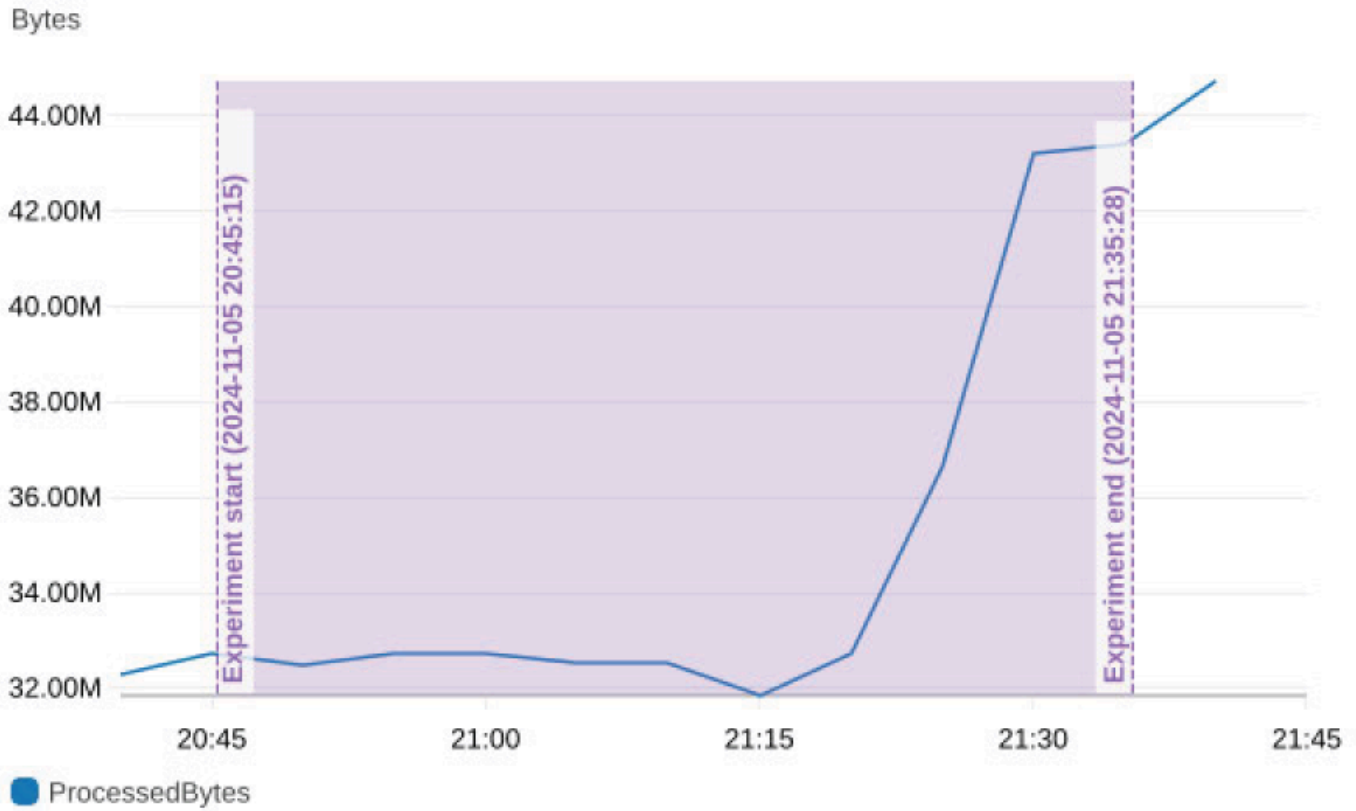
Configurazioni dei report degli esperimenti per AWS FIS

È possibile abilitare AWS Fault Injection Service (FIS) per generare report per gli esperimenti, semplificando la produzione di prove dei test di resilienza. Il rapporto sull'esperimento è un documento PDF che riassume le azioni dell'esperimento e, facoltativamente, acquisisce la risposta dell'applicazione da una CloudWatch dashboard specificata dall'utente. [Per vedere un esempio di rapporto sull'esperimento, scaricate il file zip qui.](#)

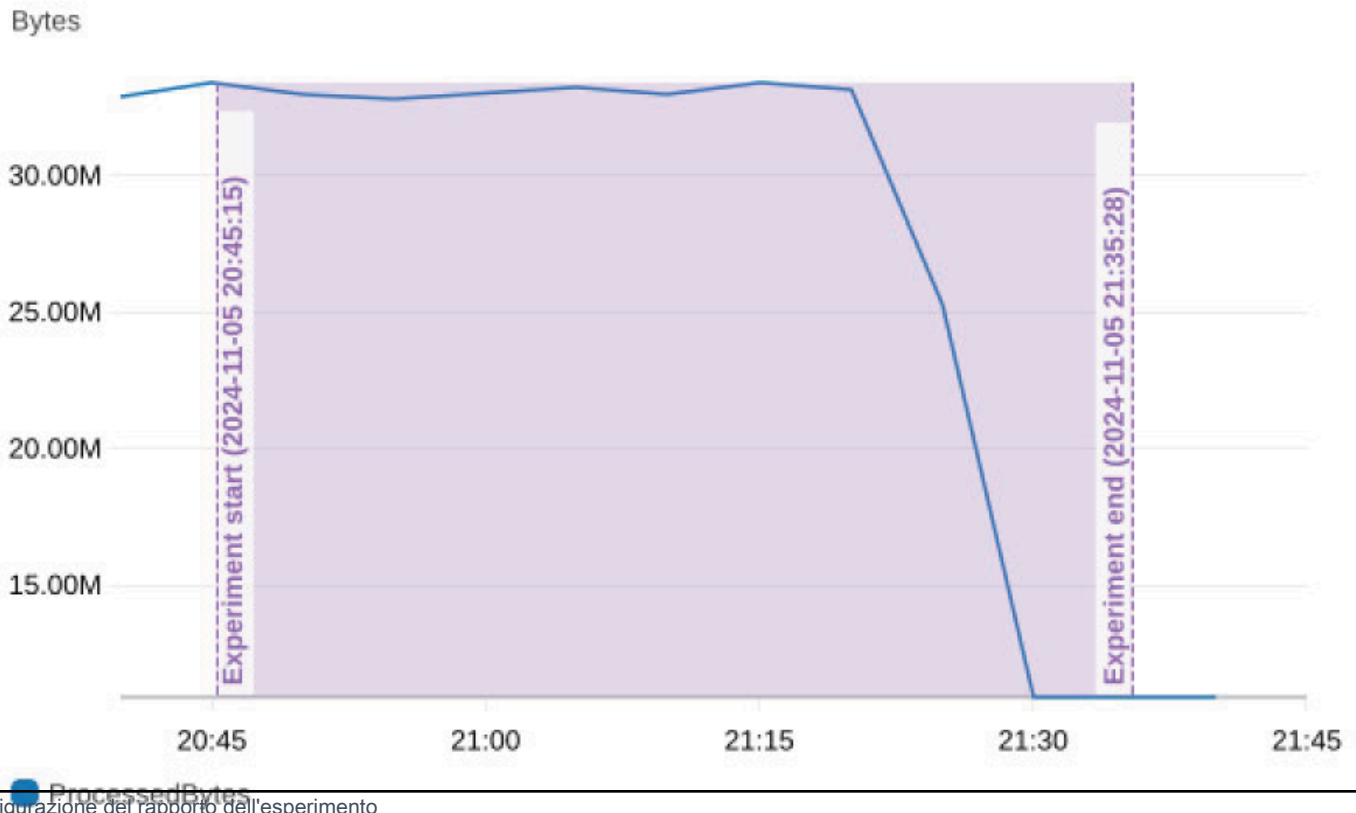
Per abilitare e configurare il contenuto del rapporto generato per l'esperimento, è necessario definire la configurazione del rapporto sull'esperimento per il modello dell'esperimento. Quando specificate un CloudWatch dashboard, AWS FIS include un grafico istantaneo di tutti i widget nella dashboard specificata, annotato con l'ora di inizio e di fine dell'esperimento per una durata specificata, come mostrato nell'esempio seguente.

Questo esempio dimostra l'impatto di un esperimento di perdita di pacchetti in una zona di disponibilità (AZ). Quando viene introdotta la perdita di pacchetti in AZ use1-az6, il traffico si sposta da use1-az6 a use1-az4, in modo che il numero di byte elaborati dal sistema di bilanciamento del carico in quella AZ diminuisca.

NLB ProcessedBytes use1-az4



NLB ProcessedBytes use1-az6



Al termine dell'esperimento, il report può essere scaricato dalla console AWS FIS ed è anche archiviato in un bucket Amazon S3. Se includi una CloudWatch dashboard nella configurazione del report, vengono fornite anche le immagini di ogni widget. I report non vengono generati per esperimenti che sono cancelled o vengono eseguiti come parte dell'anteprima di Target (con ActionsMode impostato su). skip-all Una volta che l'esperimento supera il limite di conservazione dei dati dell'esperimento, il report sarà disponibile solo dal bucket Amazon S3. AWS Le tariffe FIS si applicano per ogni report fornito, ad eccezione di quelli che non hanno esito positivo a causa di errori interni. Per ulteriori informazioni, consulta i [prezzi di AWS Fault Injection Service](#) e [Quote e limitazioni per il servizio AWS Fault Injection](#). Potrebbero essere applicati costi di importazione e archiviazione per Amazon S3 CloudWatch e costi GetMetricWidgetImageGetDashboard richieste API. [Per ulteriori informazioni, consulta i prezzi e CloudWatch i prezzi di Amazon S3.](#)

Indice

- [Sintassi di configurazione del report dell'esperimento](#)
- [Autorizzazioni per il rapporto sull'esperimento](#)
- [Procedure ottimali per i report sugli esperimenti](#)

Sintassi di configurazione del report dell'esperimento

Di seguito è riportata la sintassi per la configurazione del rapporto sull'esperimento, una sezione opzionale del modello dell'esperimento.

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

```
    "postExperimentDuration": "PT20M"  
  }  
}
```

Utilizzando `experimentReportConfiguration`, è possibile personalizzare la destinazione di output, i dati di input e le finestre temporali dei dati da includere nel rapporto dell'esperimento, in modo da comprendere meglio l'impatto e i risultati degli esperimenti AWS FIS. Quando si definisce la configurazione del rapporto sull'esperimento, si fornisce quanto segue:

outputs

Sezione `experimentReportConfiguration` che specifica dove verrà consegnato il rapporto sull'esperimento. In `outputs`, si specifica il `s3Configuration` fornendo quanto segue:

- `bucketName`- Il nome del bucket Amazon S3 in cui verrà archiviato il report. Il bucket deve trovarsi nella stessa regione dell'esperimento.
- `prefix`(Facoltativo): un prefisso all'interno del bucket Amazon S3 in cui verrà archiviato il report. Questo campo è fortemente consigliato in modo da poter limitare l'accesso solo al prefisso.

Fonti di dati

Sezione facoltativa del `experimentReportConfiguration` che specifica le fonti di dati aggiuntive che verranno incluse nel rapporto dell'esperimento.

- `cloudWatchDashboards`- Una serie di CloudWatch dashboard che verranno incluse nel rapporto. Limitato a una CloudWatch dashboard.
- `dashboardIdentifier`- L'ARN del cruscotto. CloudWatch I grafici istantanei di ogni widget con il tipo presente `metric` in questa dashboard verranno inclusi nel rapporto, ad eccezione delle metriche interregionali.

preExperimentDuration

Sezione opzionale `experimentReportConfiguration` che definisce la durata prima dell'esperimento per le metriche del CloudWatch dashboard da includere nel rapporto, fino a 30 minuti. Questo dovrebbe essere un periodo che rappresenta lo stato stazionario dell'applicazione. Ad esempio, una durata di 5 minuti prima dell'esperimento significa che i grafici delle istantanee includeranno le metriche 5 minuti prima dell'inizio dell'esperimento. Il formato per la durata è ISO 8601 e il formato predefinito è 20 minuti.

postExperimentDuration

Sezione opzionale `experimentReportConfiguration` che definisce la durata post-esperimento per le metriche del CloudWatch dashboard da includere nel rapporto, fino a 2 ore. Questa deve essere una durata che rappresenta lo stato stazionario o il periodo di ripristino dell'applicazione. Ad esempio, se specificate una durata post-esperimento di 5 minuti, i grafici delle istantanee includeranno le metriche fino a 5 minuti dopo la fine dell'esperimento. Il formato per la durata è ISO 8601 e il formato predefinito è 20 minuti.

Autorizzazioni per il rapporto sull'esperimento

Per consentire a AWS FIS di generare e archiviare il rapporto sull'esperimento, è necessario consentire le seguenti operazioni dal ruolo IAM dell'esperimento AWS FIS:

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

Ti consigliamo di seguire le migliori pratiche AWS di sicurezza e di limitare il ruolo dell'esperimento al bucket e al prefisso. Di seguito è riportato un esempio di dichiarazione politica che limita l'accesso al ruolo dell'esperimento.

JSON

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

```

    {
      "Action": [
        "cloudwatch:GetDashboard"
      ],
      "Resource": "arn:aws:cloudwatch::012345678912:dashboard/my-
experiment-report-dashboard",
      "Effect": "Allow"
    },
    {
      "Action": [
        "cloudwatch:GetMetricWidgetImage"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}

```

Autorizzazioni aggiuntive per i report distribuiti ai bucket Amazon S3 crittografati con chiavi gestite dal cliente (CMK)

Se il bucket Amazon S3 che specifichi `S3Configuration` è crittografato con CMK, devi concedere le seguenti autorizzazioni aggiuntive al ruolo dell'esperimento FIS sulla tua politica delle chiavi KMS:

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

Di seguito è riportato un esempio di dichiarazione politica chiave KMS che consente al ruolo dell'esperimento FIS di scrivere report su bucket crittografati:

```

{
  "Sid": "Allow FIS experiment report",
  "Effect": "Allow",
  "Principal":
  {
    "AWS": [
      "arn:aws:iam::012345678912:role/FISExperimentRole",
    ]
  },
  "Action": [

```

```
"kms:Decrypt",
"kms:GenerateDataKey"
],
"Resource": "*"
}
```

Procedure ottimali per i report sugli esperimenti

Di seguito sono riportate le migliori pratiche per l'utilizzo della configurazione del rapporto sugli esperimenti AWS FIS:

- Prima di iniziare un esperimento, genera un'anteprima del bersaglio per verificare che il modello dell'esperimento sia configurato come previsto. L'anteprima del target ti fornirà informazioni sugli obiettivi previsti del tuo esperimento. Per ulteriori informazioni, consulta [Genera un'anteprima del bersaglio da un modello di esperimento](#).
- Il rapporto non deve essere utilizzato per la risoluzione dei problemi relativi agli esperimenti falliti. Utilizzate invece i registri degli esperimenti per risolvere gli errori degli esperimenti. Ti consigliamo di fare affidamento sul rapporto solo per gli esperimenti che hai eseguito in precedenza e completati con successo.
- Limita l'inserimento del ruolo IAM dell'esperimento e ottieni l'accesso degli oggetti al bucket e al prefisso di destinazione S3. Ti consigliamo di dedicare il bucket/prefisso solo ai report degli esperimenti AWS FIS e di non concedere ad altri AWS servizi l'accesso a questo bucket e prefisso.
- Usa Amazon S3 Object Lock per evitare che il report venga eliminato o sovrascritto per un periodo di tempo fisso o indefinitamente. Per ulteriori informazioni, consulta [Bloccare gli oggetti](#) con Object Lock.
- Se la CloudWatch dashboard si trova in un account separato all'interno della stessa regione, puoi utilizzare l'osservabilità CloudWatch tra account per abilitare il tuo account AWS FIS orchestrator come account di monitoraggio e l'account separato come account di origine dalla CloudWatch console o dai comandi di Observability Access Manager nell'API and. AWS CLI [Per ulteriori informazioni, consulta osservabilità tra account. CloudWatch](#)

Opzioni di esperimento per AWS FIS

Le opzioni dell'esperimento sono impostazioni opzionali per un esperimento. È possibile definire determinate opzioni di esperimento nel modello di esperimento. All'inizio dell'esperimento vengono impostate opzioni aggiuntive per l'esperimento.

Di seguito è riportata la sintassi per le opzioni dell'esperimento definite nel modello di esperimento.

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

Se non specificate alcuna opzione di esperimento quando create il modello di esperimento, viene utilizzata l'impostazione predefinita per ciascuna opzione.

Di seguito è riportata la sintassi per le opzioni dell'esperimento che impostate all'inizio dell'esperimento.

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

Se non specificate alcuna opzione sperimentale all'inizio dell'esperimento, `run-all` viene utilizzata quella predefinita.

Indice

- [Targeting dell'account](#)
- [Modalità di risoluzione degli obiettivi vuota](#)
- [modalità Azioni](#)

Targeting dell'account

Se disponi di più AWS account con risorse che desideri utilizzare come target in un esperimento, puoi definire un esperimento con più account utilizzando l'opzione Account Targeting Experiment. Esegui esperimenti su più account da un account orchestrator che influiscono sulle risorse in più account di destinazione. L'account orchestrator possiede il modello e l'esperimento dell'esperimento. AWS FIS Un account target è un account AWS individuale con risorse che possono essere influenzate da un AWS FIS esperimento. Per ulteriori informazioni, consulta [Utilizzo di esperimenti con più account per AWS FIS](#).

Utilizzi il targeting per account per indicare la posizione delle risorse di destinazione. Puoi fornire due valori per il targeting dell'account:

- account singolo: impostazione predefinita. L'esperimento riguarderà solo le risorse dell' AWS account su cui viene eseguito l' AWS FIS esperimento.
- account multiplo: l'esperimento può indirizzare risorse in più account AWS.

configurazioni degli account Target

Per eseguire un esperimento con più account, devi definire una o più configurazioni di account di destinazione. Una configurazione di account di destinazione specifica l'AccountID, il roleArn e la descrizione per ogni account con risorse destinate all'esperimento. L'account IDs delle configurazioni dell'account di destinazione per un modello di esperimento deve essere univoco.

Quando crei un modello di esperimento con più account, il modello di esperimento restituirà un campo di sola lettura `targetAccountConfigurationsCount`, ovvero un conteggio di tutte le configurazioni dell'account di destinazione per il modello di esperimento.

Di seguito è riportata la sintassi per la configurazione di un account di destinazione.

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

Quando crei una configurazione di account di destinazione, fornisci quanto segue:

accountId

ID dell'account AWS a 12 cifre dell'account di destinazione.

roleArn

Un ruolo IAM che concede AWS FIS le autorizzazioni per intraprendere azioni nell'account di destinazione.

description

Descrizione facoltativa

Per saperne di più su come lavorare con le configurazioni degli account Target, consulta. [Utilizzo di esperimenti con più account per AWS FIS](#)

Modalità di risoluzione degli obiettivi vuota

Questa modalità offre la possibilità di consentire il completamento degli esperimenti anche quando una risorsa target non viene risolta.

- **fail**: impostazione predefinita. Se non viene risolta alcuna risorsa per l'obiettivo, l'esperimento viene interrotto immediatamente con uno stato di `failed`.
- **skip**: se non viene risolta alcuna risorsa per l'obiettivo, l'esperimento continuerà e tutte le azioni senza obiettivi risolti verranno ignorate. Le azioni con obiettivi definiti utilizzando identificatori univoci, ad esempio ARNs, non possono essere ignorate. Se non viene trovato un obiettivo definito utilizzando un identificatore univoco, l'esperimento viene interrotto immediatamente con uno stato di `failed`

modalità Azioni

La modalità Azioni è un parametro opzionale che è possibile specificare quando si avvia un esperimento. È possibile impostare la modalità azioni in modo `skip-all` da generare un'anteprima del bersaglio prima di iniettare errori nelle risorse di destinazione. L'anteprima dell'obiettivo consente di verificare quanto segue:

- Di aver configurato il modello di esperimento in modo da indirizzare le risorse previste. Le risorse effettive prese di mira all'avvio di questo esperimento potrebbero essere diverse dall'anteprima perché le risorse possono essere rimosse, aggiornate o campionate in modo casuale.
- Che le configurazioni di registrazione siano configurate correttamente.
- Che per gli esperimenti con più account hai impostato correttamente un ruolo IAM per ciascuna delle configurazioni dell'account di destinazione.

Note

La `skip-all` modalità non ti consente di verificare di disporre delle autorizzazioni necessarie per eseguire l' AWS FIS esperimento e intraprendere azioni sulle tue risorse.

Il parametro `actions mode` accetta i seguenti valori:

- `run-all`- (Impostazione predefinita) L'esperimento eseguirà azioni sulle risorse target.
- `skip-all`- L'esperimento ignorerà tutte le azioni sulle risorse target.

Per ulteriori informazioni su come impostare il parametro della modalità azioni all'avvio di un esperimento, consulta [Genera un'anteprima del bersaglio da un modello di esperimento](#).

AWS FIS Riferimento alle azioni

Un'azione è l'attività di iniezione dei guasti eseguita su un bersaglio utilizzando AWS Fault Injection Service (AWS FIS). AWS FIS fornisce azioni preconfigurate per tipi specifici di obiettivi tra AWS i servizi. Si aggiungono azioni a un modello di esperimento, che viene quindi utilizzato per eseguire esperimenti.

Questo riferimento descrive le azioni più comuni in AWS FIS, incluse le informazioni sui parametri dell'azione e sulle autorizzazioni IAM richieste. Puoi anche elencare AWS FIS le azioni supportate utilizzando la AWS FIS console o il comando [list-actions](#) di (). AWS Command Line Interface AWS CLI Una volta ottenuto il nome di un'azione specifica, è possibile visualizzare informazioni dettagliate sull'azione utilizzando il comando [get-action](#). Per ulteriori informazioni sull'uso dei AWS FIS comandi con AWS CLI, vedere la Guida per l'[AWS Command Line Interface utente e fis nella Guida](#) ai AWS CLI comandi.

Per ulteriori informazioni sul funzionamento AWS FIS delle azioni, vedere [Azioni per AWS FIS](#) e [Come funziona AWS Fault Injection Service con IAM](#).

Azioni

- [Azioni di iniezione dei guasti](#)
- [Azione di ripristino](#)
- [Attendi l'azione](#)
- [CloudWatch Azioni Amazon](#)
- [Azioni Amazon DynamoDB](#)
- [Azioni SQL di Amazon Aurora](#)
- [Azioni Amazon EBS](#)
- [Azioni di Amazon EC2](#)
- [Azioni Amazon ECS](#)
- [Azioni Amazon EKS](#)
- [ElastiCache Azioni Amazon](#)
- [Azioni di Amazon Kinesis Data Streams](#)
- [AWS Lambda azioni](#)
- [Azione Amazon MemoryDB](#)

- [Azioni di rete](#)
- [Azioni Amazon RDS](#)
- [Operazioni di Amazon S3](#)
- [Azioni di Systems Manager](#)
- [AWS Direct Connect azioni](#)
- [Usa i documenti SSM di Systems Manager con AWS FIS](#)
- [Usa le azioni AWS FIS `aws:ecs:task`](#)
- [Usa le azioni AWS FIS `aws:eks:pod`](#)
- [Usa le azioni AWS FIS `aws:lambda:function`](#)

Azioni di iniezione dei guasti

AWS FIS supporta le seguenti azioni di iniezione dei guasti.

Azioni

- [aws:fis:inject-api-internal-error](#)
- [aws:fis:inject-api-throttle-error](#)
- [aws:fis:inject-api-unavailable-error](#)

aws:fis:inject-api-internal-error

Inserisce errori interni nelle richieste effettuate dal ruolo IAM di destinazione. La risposta specifica dipende da ogni servizio e API. Per ulteriori informazioni, consulta la documentazione SDK e API del tuo servizio.

Tipo di risorsa

- `aws:iam:role`

Parameters

- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

- **service**— Lo spazio dei nomi dell' AWS API di destinazione. Il valore supportato è `ec2 and. kinesis`
- **percentage**— La percentuale (1-100) di chiamate in cui inserire il guasto.
- **operations**— Le operazioni in cui iniettare il guasto, separate da virgole. Per un elenco delle azioni API per lo spazio dei `ec2` nomi, consulta [Amazon EC2 API Reference](#) e [Amazon Kinesis Data Streams API Reference](#).

Permissions

- `fis:InjectApiInternalError`

`aws:fis:inject-api-throttle-error`

Inietta errori di limitazione nelle richieste effettuate dal ruolo IAM di destinazione. La risposta specifica dipende da ogni servizio e API. Per ulteriori informazioni, consulta la documentazione SDK e API del tuo servizio.

Tipo di risorsa

- `aws:iam:role`

Parameters

- **duration**— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **service**— Lo spazio dei nomi dell' AWS API di destinazione. Il valore supportato è `ec2 and. kinesis`
- **percentage**— La percentuale (1-100) di chiamate in cui inserire il guasto.
- **operations**— Le operazioni in cui iniettare il guasto, separate da virgole. Per un elenco delle azioni API per lo spazio dei `ec2` nomi, consulta [Amazon EC2 API Reference](#) e [Amazon Kinesis Data Streams API Reference](#).

Permissions

- `fis:InjectApiThrottleError`

aws:fis:inject-api-unavailable-error

Inietta errori non disponibili nelle richieste effettuate dal ruolo IAM di destinazione. La risposta specifica dipende da ogni servizio e API. Per ulteriori informazioni, consulta la documentazione SDK e API del tuo servizio.

Tipo di risorsa

- `aws:iam:role`

Parameters

- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- `service`— Lo spazio dei nomi dell' AWS API di destinazione. Il valore supportato è `ec2` and `kinesis`
- `percentage`— La percentuale (1-100) di chiamate in cui inserire il guasto.
- `operations`— Le operazioni in cui iniettare il guasto, separate da virgole. Per un elenco delle azioni API per lo spazio dei `ec2` nomi, consulta [Amazon EC2 API Reference](#) e [Amazon Kinesis Data Streams API Reference](#).

Permissions

- `fis:InjectApiUnavailableError`

Azione di ripristino

Le azioni di ripristino vengono eseguite per mitigare il rischio o proteggere le applicazioni in caso di danneggiamento.

AWS FIS supporta le seguenti azioni di ripristino.

aws:arc:start-zonal-autoshift

Sposta automaticamente il traffico per le risorse supportate lontano da una zona di disponibilità (AZ) potenzialmente compromessa e le reindirizza verso AZs l'integrità nella stessa regione AWS.

Ciò consente di sperimentare l'autoshift zonale tramite FIS. L'autoshift zonale è una funzionalità di Amazon Application Recovery Controller (ARC) che consente di AWS spostare il traffico di una risorsa lontano da una zona di disponibilità, per tuo conto, quando si verifica AWS una compromissione che potrebbe influire sui clienti della zona AZ.

Quando esegui l'`aws:arc:start-zonal-autoshift` azione, AWS FIS gestisce lo spostamento di zona utilizzando e `CancelZonalShift` APIs con il `StartZonalShift expiresIn` campo per queste richieste impostato su 1 minuto come meccanismo di sicurezza. `UpdateZonalShift` Ciò consente di AWS FIS ripristinare rapidamente lo spostamento zonale in caso di eventi imprevisti come interruzioni della rete o problemi di sistema. Nella console ARC, il campo del tempo di scadenza verrà visualizzato AWS FIS-managed e la scadenza effettiva prevista è determinata dalla durata specificata nell'azione di spostamento zonale.

Tipo di risorsa

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

Le risorse gestite con spostamento di zona sono tipi di risorse tra cui cluster Amazon EKS, Amazon EC2 Application and Network Load Balancers e gruppi Amazon EC2 Auto Scaling che possono essere abilitati per lo spostamento automatico zonale ARC. [Per ulteriori informazioni, consulta le risorse supportate e l'abilitazione delle risorse di trasferimento automatico zonale nella ARC Developer Guide.](#)

Parameters

- `duration`— Il periodo di tempo per il quale il traffico verrà spostato. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- `availabilityZoneIdentifier`— Il traffico si allontana da questa AZ. Può essere un nome AZ (`us-east-1a`) o un ID AZ (`use1-az1`).
- `managedResourceTypes`— I tipi di risorse da cui verrà spostato il traffico, separati da virgole. Le opzioni possibili sono ASG (Auto Scaling Group), ALB (Application Load Balancer)NLB, (Network Load Balancer) e EKS (Amazon EKS).
- `zonalAutoshiftStatus`— Lo `zonalAutoshiftStatus` stato delle risorse a cui desideri rivolgerti. Le opzioni possibili sono `ENABLEDDISABLED`, e`ANY`. Il valore predefinito è `ENABLED`.

Permissions

- `arc-zonal-shift:StartZonalShift`
- `arc-zonal-shift:GetManagedResource`
- `arc-zonal-shift:UpdateZonalShift`
- `arc-zonal-shift:CancelZonalShift`
- `arc-zonal-shift:ListManagedResources`
- `scalabilità automatica: DescribeTags`
- `etichetta: GetResources`

Attendi l'azione

AWS FIS supporta la seguente azione di attesa.

`aws:fis:wait`

Esegue l'azione di AWS FIS attesa.

Parameters

- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

- Nessuno

CloudWatch Azioni Amazon

AWS FIS supporta la seguente CloudWatch azione di Amazon.

`aws:cloudwatch:assert-alarm-state`

Verifica che gli allarmi specificati si trovino in uno degli stati di allarme specificati.

Tipo di risorsa

- Nessuno

Parameters

- `alarmArns`— Gli ARNs allarmi, separati da virgole. È possibile specificare fino a cinque allarmi.
- `alarmStates`— Gli stati di allarme, separati da virgole. I possibili stati di allarme sono OKALARM, eINSUFFICIENT_DATA.

Permissions

- `cloudwatch:DescribeAlarms`

Azioni Amazon DynamoDB

AWS FIS supporta la seguente azione di Amazon DynamoDB.

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

Sospende la replica delle tabelle globali multiregione di Amazon DynamoDB su qualsiasi tabella di replica. Le tabelle possono continuare a essere replicate fino a 5 minuti dopo l'inizio dell'azione.

Tabelle globali multiregionali fortemente coerenti (MRSC)

Le seguenti istruzioni verranno aggiunte dinamicamente alla policy per la tabella globale DynamoDB MRSC di destinazione:

```
{
  "Statement": [
    {
      "Sid": "DoNotModifyFisDynamoDbPauseReplicationEXPxxxxxxxxxxxxxxxx",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "dynamodb:UpdateTable"
      ],
    },
  ],
}
```

```

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

```

Se a una tabella di destinazione non è associata alcuna policy relativa alle risorse, viene creata una policy relativa alle risorse per la durata dell'esperimento ed eliminata automaticamente al termine dell'esperimento. In caso contrario, la dichiarazione di errore viene inserita in una politica esistente, senza ulteriori modifiche alle dichiarazioni politiche esistenti. L'indicazione di errore viene quindi rimossa dalla politica alla fine dell'esperimento.

Le tabelle globali Target di Amazon DynamoDB MRSC sono soggette a una quota aggiuntiva. Questa quota impone che nessuna singola tabella possa essere soggetta a più di 5.040 minuti di deterioramento in un periodo variabile di 7 giorni.

Table globali multiregionali, alla fine coerenti (MREC)

La seguente dichiarazione verrà aggiunta dinamicamente alla policy per la tabella globale DynamoDB MREC di destinazione:

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

L'istruzione seguente verrà aggiunta dinamicamente alla policy di flusso per la tabella globale DynamoDB MREC di destinazione:

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

Se una tabella o uno stream di destinazione non ha policy relative alle risorse allegare, viene creata una policy relativa alle risorse per la durata dell'esperimento ed eliminata automaticamente al termine dell'esperimento. In caso contrario, la dichiarazione di errore viene inserita in una politica esistente, senza ulteriori modifiche alle dichiarazioni politiche esistenti. L'indicazione di errore viene quindi rimossa dalla politica alla fine dell'esperimento.

Tipo di risorsa

- aws:dynamodb:global-table

Parameters

- **duration**— Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

* L'autorizzazione è richiesta solo se si utilizzano come target le tabelle globali MRSC

Azioni SQL di Amazon Aurora

AWS FIS supporta le seguenti azioni SQL di Amazon Aurora.

`aws:dsql:cluster-connection-failure`

Crea errori di connessione controllati in un cluster Aurora DSQL per una durata specificata per testare la resilienza dell'applicazione.

Tipo di risorsa

- `aws:dsql:cluster`

Parameters

- **duration**— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **percentage**— La percentuale (1-100) di chiamate in cui inserire il guasto.

Permissions

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

Per avviare l'esperimento con Aurora DSQL, [vedere Fault injection](#) testing nella Guida per l'utente di Aurora DSQL.

Azioni Amazon EBS

AWS FIS supporta la seguente azione di Amazon EBS.

Azioni

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

aws:ebs:pause-volume-io

Sospende I/O le operazioni sui volumi EBS di destinazione. I volumi di destinazione devono trovarsi nella stessa zona di disponibilità e devono essere collegati a istanze create sul sistema Nitro. I volumi non possono essere collegati alle istanze su un Outpost.

Per avviare l'esperimento utilizzando la console Amazon EC2, [consulta Fault testing on Amazon EBS nella Amazon EC2 User Guide](#).

Tipo di risorsa

- `aws:ec2:ebs-volume`

Parameters

- `duration`— La durata, da un secondo a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto, PT5 S rappresenta cinque secondi e PT6 H rappresenta sei ore. Nella AWS FIS console, si immette il numero di secondi, minuti o ore. Se la durata è piccola, ad esempio PT5 S, I/O viene messa in pausa per la durata specificata, ma il

completamento dell'esperimento potrebbe richiedere più tempo a causa del tempo necessario per inicializzarlo.

Permissions

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

aws:ebs:volume-io-latency

Inietta la latenza sulle I/O operazioni dei volumi EBS di destinazione. [I volumi di destinazione devono trovarsi nella stessa zona di disponibilità e devono essere collegati a istanze basate su Nitro](#). I volumi non possono essere collegati alle istanze in un Outpost.

Per avviare l'esperimento utilizzando la console Amazon EC2, [consulta Fault testing on Amazon EBS nella Amazon EBS User Guide](#).

Tipo di risorsa

- `aws:ec2:ebs-volume`

Parameters

- `readIOPercentage`— La percentuale di I/O operazioni di lettura su cui verrà iniettata la latenza, dallo 0,1% al 100%. Questa è la percentuale di tutte le I/O operazioni di lettura sul volume che verrà influenzata durante l'esperimento. Il valore di default è 100.
- `readIOLatencyMilliseconds`— La quantità di latenza iniettata nelle I/O operazioni di lettura in millisecondi, da 1 ms (volumi io2) o 10 ms (volumi non io2) a 60 secondi. Questo è il valore di latenza che verrà osservato sulla percentuale specificata di lettura durante l'esperimento. I/O Il valore di default è 100.
- `writelIOPercentage`— La percentuale di I/O operazioni di scrittura su cui verrà iniettata la latenza, dallo 0,1% al 100%. Questa è la percentuale di tutte le I/O operazioni di scrittura sul volume che verrà influenzata durante l'esperimento. Il valore di default è 100.
- `writelIOLatencyMilliseconds`— La quantità di latenza iniettata nelle I/O operazioni di scrittura in millisecondi, da 1 ms (volumi io2) o 10 ms (volumi non io2) a 60 secondi. Questo è il valore di

latenza che verrà osservato sulla percentuale specificata di lettura durante l'esperimento. I/O Il valore di default è 100.

- `duration`— La durata per la quale verrà iniettata la latenza, da 1 secondo a 12 ore.

Permissions

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

Azioni di Amazon EC2

AWS FIS supporta le seguenti azioni Amazon EC2.

Azioni

- [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 supporta anche azioni di iniezione degli errori tramite l'agente AWS Systems Manager SSM. Systems Manager utilizza un documento SSM che definisce le azioni da eseguire sulle istanze EC2. È possibile utilizzare il proprio documento per inserire errori personalizzati oppure utilizzare documenti SSM preconfigurati. Per ulteriori informazioni, consulta [the section called “Azioni relative ai documenti SSM”](#).

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

Inietta risposte `InsufficientInstanceCapacity` di errore alle richieste effettuate dai ruoli IAM di destinazione. Le operazioni supportate sono `RunInstances` `CreateCapacityReservation`, `StartInstances`, `CreateFleet` chiamate. Le richieste che includono richieste di capacità in più zone di disponibilità non sono supportate. Questa azione non supporta la definizione di obiettivi utilizzando tag di risorse, filtri o parametri.

Per l' `LaunchInstances` operazione `Auto Scaling`, `InsufficientInstanceCapacity` gli errori verranno restituiti nel `errors` campo della risposta, ma la capacità desiderata del gruppo `Auto Scaling` verrà comunque aggiornata, consentendo al processo di scalabilità asincrono di avviare potenzialmente istanze. Per un test più ampio dell'insufficiente gestione della capacità con, prendi in considerazione l'utilizzo di questa azione insieme `LaunchInstances` a [the section called "aws:ec2:asg-insufficient-instance-capacity-error"](#)

Tipo di risorsa

- `aws:iam:role`

Parameters

- `duration`— Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- `availabilityZoneIdentifiers`— L'elenco separato da virgole delle zone di disponibilità. Supporta i nomi delle zone IDs (ad esempio `"use1-az1, use1-az2"`) e delle zone (ad esempio `"us-east-1a"`).
- `percentage`— La percentuale (1-100) di chiamate in cui inserire il guasto.

Permissions

- `ec2:InjectApiError` con il `ec2:FisActionId` valore della chiave di condizione impostato su `aws:ec2:api-insufficient-instance-capacity-error` e la chiave di `ec2:FisTargetArns` condizione impostata per i ruoli IAM.

Per un esempio di policy, consulta [Esempio: utilizza i tasti condizionali per `ec2:InjectApiError`](#).

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

Inietta risposte `InsufficientInstanceCapacity` di errore alle richieste effettuate dai gruppi di `Auto Scaling` di destinazione. Questa azione supporta solo i gruppi di `Auto Scaling` che utilizzano modelli di avvio. Per ulteriori informazioni sugli errori di capacità insufficiente delle istanze, consulta la guida per l'[utente di Amazon EC2](#).

Tipo di risorsa

- `aws:ec2:autoscaling-group`

Parameters

- **duration**— Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **availabilityZoneIdentifiers**— L'elenco separato da virgole delle zone di disponibilità. Supporta i nomi delle zone IDs (ad esempio "use1-az1, use1-az2") e delle zone (ad esempio "us-east-1a").
- **percentage** : Opzionale. La percentuale (1-100) delle richieste di avvio del gruppo di Auto Scaling target per iniettare il guasto. Il valore di default è 100.

Permissions

- **ec2:InjectApiError** con chiave di condizione **ec2:FisActionId** valore impostato su **aws:ec2:asg-insufficient-instance-capacity-error** e chiave di **ec2:FisTargetArns** condizione impostata per gruppi di Auto Scaling.
- **autoscaling:DescribeAutoScalingGroups**

Per un esempio di policy, consulta [Esempio: utilizza i tasti condizionali per ec2:InjectApiError](#).

aws:ec2:reboot-instances

Esegue l'azione dell'API Amazon EC2 [RebootInstances](#) sulle istanze EC2 di destinazione.

Tipo di risorsa

- **aws:ec2:instance**

Parameters

- Nessuno

Permissions

- **ec2:RebootInstances**
- **ec2:DescribeInstances**

AWS politica gestita

- [AWSFaultInjectionSimulatorEC2Accesso](#)

aws:ec2:send-spot-instance-interruptions

Interrompe le istanze Spot di destinazione. Invia un [avviso di interruzione dell'istanza Spot](#) alle istanze Spot bersaglio due minuti prima di interromperle. Il tempo di interruzione è determinato dal parametro specificato. `durationBeforeInterruption` Due minuti dopo il periodo di interruzione, le istanze Spot vengono terminate o interrotte, a seconda del loro comportamento di interruzione. Un'istanza spot che arrestata tramite AWS FIS rimane arrestata fino a quando non viene riavviata.

[Immediatamente dopo l'avvio dell'azione, l'istanza di destinazione riceve una raccomandazione di ribilanciamento dell'istanza EC2.](#) Se lo hai specificato `durationBeforeInterruption`, potrebbe verificarsi un ritardo tra la raccomandazione di ribilanciamento e l'avviso di interruzione.

Per ulteriori informazioni, consulta [the section called “Interruzioni delle istanze Test Spot”](#). In alternativa, per avviare l'esperimento utilizzando la console Amazon EC2, [consulta Avviare un'interruzione di un'istanza Spot](#) nella Amazon EC2 User Guide.

Tipo di risorsa

- `aws:ec2:spot-instance`

Parameters

- `durationBeforeInterruption`— Il tempo di attesa prima di interrompere l'istanza, da 2 a 15 minuti. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT2 M` rappresenta due minuti. Nella AWS FIS console, si immette il numero di minuti.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEC2Accesso](#)

aws:ec2:stop-instances

Esegue l'azione dell'API Amazon EC2 [StopInstances](#) sulle istanze EC2 di destinazione.

Tipo di risorsa

- aws:ec2:instance

Parameters

- `startInstancesAfterDuration` : Opzionale. Il tempo di attesa prima di avviare l'istanza, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore. Se l'istanza ha un volume EBS crittografato, devi concedere l' AWS FIS autorizzazione alla chiave KMS utilizzata per crittografare il volume o aggiungere il ruolo di esperimento alla politica delle chiavi KMS.
- `completeIfInstancesTerminated` : Opzionale. Se vero, e se `startInstancesAfterDuration` è vero anche se è vero, questa azione non avrà esito negativo quando le istanze EC2 mirate vengono terminate da una richiesta separata esterna a FIS e non possono essere riavviate. Ad esempio, i gruppi di Auto Scaling possono terminare le istanze EC2 interrotte sotto il loro controllo prima del completamento di questa azione. Il valore predefinito è `false`.

Permissions

- `ec2:StopInstances`
- `ec2:StartInstances`
- `ec2:DescribeInstances` : Opzionale. Richiesto con `completeIfInstancesTerminated` per convalidare lo stato dell'istanza al termine dell'azione.
- `kms:CreateGrant` : Opzionale. Richiesto con `startInstancesAfterDuration` per riavviare le istanze con volumi crittografati.

AWS politica gestita

- [AWSFaultInjectionSimulatorEC2Accesso](#)

aws:ec2:terminate-instances

Esegue l'azione dell'API Amazon EC2 [TerminateInstances](#) sulle istanze EC2 di destinazione.

Tipo di risorsa

- `aws:ec2:instance`

Parameters

- Nessuno

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEC2Accesso](#)

Azioni Amazon ECS

AWS FIS supporta le seguenti azioni Amazon ECS.

Azioni

- [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

Esegue l'azione API Amazon ECS [UpdateContainerInstancesState](#) per drenare la percentuale specificata di istanze Amazon EC2 sottostanti sui cluster di destinazione.

Tipo di risorsa

- aws:ecs:cluster

Parameters

- drainagePercentage— La percentuale (1-100).
- duration— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorECSAccess](#)

aws:ecs:stop-task

Esegue l'azione API Amazon ECS [StopTask](#) per interrompere l'attività di destinazione.

Tipo di risorsa

- aws:ecs:task

Parameters

- Nessuno

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorECSAccess](#)

aws:ecs:task-cpu-stress

Esegue lo stress della CPU sulle attività di destinazione. Utilizza il documento [AWSFIS-RunSSM - CPU-stress](#). Le attività devono essere gestite da AWS Systems Manager. Per ulteriori informazioni, consulta [azioni delle attività ECS](#).

Tipo di risorsa

- `aws:ecs:task`

Parameters

- `duration`— La durata dello stress test, in formato ISO 8601.
- `percent` : Opzionale. La percentuale di carico target, da 0 (senza carico) a 100 (a pieno carico). Il valore di default è 100.
- `workers` : Opzionale. Il numero di fattori di stress da utilizzare. L'impostazione predefinita è 0, che utilizza tutti i fattori di stress.
- `installDependencies` : Opzionale. Se questo valore è `True`, Systems Manager installa le dipendenze richieste nel contenitore sidecar per l'agente SSM, se non sono già installate. Il valore predefinito è `True`. La dipendenza è `stress-ng`.

Permissions

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

aws:ecs:task-io-stress

I/O Esercita lo stress sulle attività target. Utilizza il documento [AWSFIS-RunSSM -IO-Stress](#). Le attività devono essere gestite da AWS Systems Manager Per ulteriori informazioni, consulta [azioni delle attività ECS](#).

Tipo di risorsa

- `aws:ecs:task`

Parameters

- `duration`— La durata dello stress test, in formato ISO 8601.
- `percent` : Opzionale. La percentuale di spazio libero sul file system da utilizzare durante lo stress test. L'impostazione predefinita è 80%.
- `workers` : Opzionale. Il numero di worker. I lavoratori eseguono una combinazione di read/write operazioni sequenziali, casuali e mappate in memoria, sincronizzazione forzata e eliminazione della cache. Più processi secondari eseguono operazioni diverse sullo stesso file. I/O Il valore di default è 1.
- `installDependencies` : Opzionale. Se questo valore è `True`, Systems Manager installa le dipendenze richieste nel contenitore sidecar per l'agente SSM, se non sono già installate. Il valore predefinito è `True`. La dipendenza è. `stress-ng`

Permissions

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

- `ssm:CancelCommand`

aws:ecs:task-kill-process

Arresta il processo specificato nelle attività, utilizzando il `killall` comando. Utilizza il documento [AWSFIS-RunSSM -Kill-Process](#). La definizione dell'attività deve essere impostata su. `pidMode task` Le attività devono essere gestite da AWS Systems Manager. Per ulteriori informazioni, consulta [azioni delle attività ECS](#).

Tipo di risorsa

- `aws:ecs:task`

Parameters

- `processName`— Il nome del processo da interrompere.
- `signal` : Opzionale. Il segnale da inviare insieme al comando. I valori possibili sono `SIGTERM` (che il ricevitore può scegliere di ignorare) e `SIGKILL` (che non possono essere ignorati). Il valore predefinito è `SIGTERM`.
- `installDependencies` Facoltativo. Se questo valore è `True`, Systems Manager installa le dipendenze richieste nel contenitore `sidecar` per l'agente SSM, se non sono già installate. Il valore predefinito è `True`. La dipendenza è. `killall`

Permissions

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

aws:ecs:task-network-blackhole-port

Elimina il traffico in entrata o in uscita per il protocollo e la porta specificati, utilizzando gli endpoint [Amazon ECS Fault Injection](#). [Utilizza il documento SSM AWSFIS-Run -Network-Blackhole-Port-ECS](#). La definizione dell'attività deve essere impostata su. `pidMode task` Le attività devono essere gestite

da AWS Systems Manager. Non è possibile `networkMode` impostarlo `bridge` nella definizione dell'attività. Per ulteriori informazioni, consulta [azioni delle attività ECS](#).

Quando `useEcsFaultInjectionEndpoints` è impostato su `false`, l'errore utilizza `iptables` lo strumento e utilizza il documento SSM [AWSFIS-Run-Network-Blackhole-Port](#).

Tipo di risorsa

- `aws:ecs:task`

Parameters

- `duration`— La durata del test, in formato ISO 8601.
- `port`— Il numero di porta.
- `trafficType`— Il tipo di traffico. I valori possibili sono `ingress` e `egress`.
- `protocol` : Opzionale. Il protocollo. I valori possibili sono `tcp` e `udp`. Il valore di default è `tcp`.
- `installDependencies` Facoltativo. Se questo valore è `True`, Systems Manager installa le dipendenze richieste nel contenitore `sidecar` per l'agente SSM, se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono, e. `atd curl-minimal dig jq`
- `useEcsFaultInjectionEndpoints` : Opzionale. Se impostato su `true`, APIs verrà utilizzata Amazon ECS Fault Injection. Il valore predefinito è `false`.

Permissions

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

aws:ecs:task-network-latency

Aggiunge latenza e jitter all'interfaccia di rete per il traffico in uscita verso fonti specifiche, utilizzando gli endpoint [Amazon ECS](#) Fault Injection. [Utilizza il documento SSM -Network-Latency-ECS](#). [AWSFIS-Run](#) La definizione dell'attività deve essere impostata su `task`. Le attività devono essere gestite da AWS Systems Manager. Non è possibile `networkMode` impostarlo `bridge` nella definizione dell'attività. Per ulteriori informazioni, consulta [azioni delle attività ECS](#).

Quando `useEcsFaultInjectionEndpoints` è impostato su `false`, l'errore utilizza `tc` lo strumento e utilizza il documento SSM [AWSFIS-Run-Network-Latency-Sources](#).

Utilizzate il `flowsPercent` parametro per aggiungere latenza su una percentuale delle connessioni. Per utilizzare il `flowsPercent` parametro, la versione dell'agente ECS deve essere uguale `1.100.0` o superiore.

Per utilizzare i nomi AZ o AZ IDs nel `sources` parametro, tutte le destinazioni dell'azione devono trovarsi sullo stesso VPC.

Tipo di risorsa

- `aws:ecs:task`

Parameters

- `duration`— La durata del test, in formato ISO 8601.
- `delayMilliseconds` : Opzionale. Il ritardo, in millisecondi. L'impostazione predefinita è 200.
- `jitterMilliseconds` : Opzionale. Il jitter, in millisecondi. Il valore predefinito è 10.
- `flowsPercent` : Opzionale. La percentuale di flussi di rete che saranno influenzati dall'azione. Il valore di default è 100%.
- `sources` : Opzionale. Le fonti, separate da virgole, senza spazi. I valori possibili sono: un IPv4 indirizzo, un blocco IPv4 CIDR, un nome di dominio, un nome AZ (`us-east-1a`), un ID AZ (`use1-az1`), `ALL` e `DYNAMODB S3`. Se si specifica `DYNAMODB S3`, ciò si applica solo all'endpoint regionale nella regione corrente. L'impostazione predefinita è `ALL`, che corrisponde a tutto IPv4 il traffico.
- `installDependencies` : Opzionale. Se questo valore è `True`, Systems Manager installa le dipendenze richieste nel contenitore sidecar per l'agente SSM, se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono `atd`, e. `curl-minimal` `dig` `jq` `ls` `sof`
- `useEcsFaultInjectionEndpoints` : Opzionale. Se impostato su `true`, APIs verrà utilizzata Amazon ECS Fault Injection. Il valore predefinito è `false`.

Permissions

- `ecs:DescribeTasks`
- `ecs:DescribeContainerInstances`
- `ec2:DescribeInstances`

- `ec2:DescribeSubnets`
- `ssm:SendCommand`
- `ssm:ListCommands`
- `ssm:CancelCommand`

aws:ecs:task-network-packet-loss

Aggiunge la perdita di pacchetti all'interfaccia di rete per il traffico in uscita verso fonti specifiche, utilizzando gli endpoint [Amazon ECS Fault Injection](#). [Utilizza il documento SSM AWSFIS-Run - Network-Packet-loss-ECS](#). La definizione dell'attività deve essere impostata su `pidMode task`. Le attività devono essere gestite da AWS Systems Manager. Non è possibile impostarlo su `networkMode bridge` nella definizione dell'attività. Per ulteriori informazioni, consulta [azioni delle attività ECS](#).

Quando `useEcsFaultInjectionEndpoints` è impostato su `false`, l'errore utilizza `tc` lo strumento e utilizza il documento SSM [AWSFIS-Run-Network-Packet-Loss-Sources](#).

Usa il `flowsPercent` parametro per iniettare la perdita di pacchetti su una percentuale delle connessioni. Per utilizzare il `flowsPercent` parametro, la versione dell'agente ECS deve essere `1.100.0` uguale o superiore.

Per utilizzare i nomi AZ o AZ IDs nel `sources` parametro, tutte le destinazioni dell'azione devono trovarsi sullo stesso VPC.

Tipo di risorsa

- `aws:ecs:task`

Parameters

- `duration`— La durata del test, in formato ISO 8601.
- `lossPercent` : Opzionale. La percentuale di perdita di pacchetti. L'impostazione predefinita è 7%.
- `flowsPercent` : Opzionale. La percentuale di flussi di rete che saranno interessati dall'azione. Il valore di default è 100%.
- `sources` : Opzionale. Le fonti, separate da virgole, senza spazi. I valori possibili sono: un IPv4 indirizzo, un blocco IPv4 CIDR, un nome di dominio, un nome AZ (us-east-1a), un ID AZ (use1-az1), ALL e DYNAMODB S3. Se si specifica DYNAMODB o S3, ciò si applica solo all'endpoint regionale nella regione corrente. L'impostazione predefinita è ALL, che corrisponde a tutto IPv4 il traffico.

- `installDependencies` : Opzionale. Se questo valore è `True`, Systems Manager installa le dipendenze richieste nel contenitore sidecar per l'agente SSM, se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono `atd`, e. `curl-minimal` `dig` `jq` `ls` `sof`
- `useEcsFaultInjectionEndpoints` : Opzionale. Se impostato su `true`, APIs verrà utilizzata Amazon ECS Fault Injection. Il valore predefinito è `false`.

Permissions

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

Azioni Amazon EKS

AWS FIS supporta le seguenti azioni Amazon EKS.

Azioni

- [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

Esegue un esperimento ChaosMesh or Litmus su un singolo cluster di destinazione. È necessario installare ChaosMesh o Litmus sul cluster di destinazione.

Quando crei un modello di esperimento e definisci un obiettivo di tipo `aws:eks:cluster`, devi indirizzare questa azione a un singolo Amazon Resource Name (ARN). Questa azione non supporta la definizione di obiettivi utilizzando tag di risorse, filtri o parametri.

Quando si installa ChaosMesh, è necessario specificare il runtime del contenitore appropriato. A partire dalla versione 1.23 di Amazon EKS, il runtime predefinito è cambiato da Docker a `containerd`. A partire dalla versione 1.24, Docker è stato rimosso.

Tipo di risorsa

- `aws:eks:cluster`

Parameters

- `kubernetesApiVersion`— La versione API della risorsa personalizzata [Kubernetes](#). I valori possibili sono `chaos-mesh.org/v1alpha1` `litmuschaos.io/v1alpha1`
- `kubernetesKind`— Il tipo di risorsa personalizzata Kubernetes. Il valore dipende dalla versione dell'API.
 - `chaos-mesh.org/v1alpha1`— I valori possibili sono `AWSChaos` `DNSChaos` `GCPChaos` | `HTTPChaos` | `IOChaos` | `JVMChaos` | `KernelChaos` | `NetworkChaos` | `PhysicalMachineChaos` | `PodChaos` | `PodHttpChaos` | `PodIOChaos` | `PodNetworkChaos` | `Schedule` | `StressChaos` | `TimeChaos` |
- `litmuschaos.io/v1alpha1`— Il valore possibile è `ChaosEngine`.
- `kubernetesNamespace`— Lo spazio dei nomi [Kubernetes](#).
- `kubernetesSpec`— La `spec` sezione della risorsa personalizzata Kubernetes, in formato JSON.
- `maxDuration`— Il tempo massimo consentito per il completamento dell'esecuzione dell'automazione, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

Per questa azione non sono richieste autorizzazioni AWS Identity and Access Management (IAM). Le autorizzazioni necessarie per utilizzare questa azione sono controllate da Kubernetes utilizzando l'autorizzazione RBAC. Per ulteriori informazioni, consulta [Using RBAC Authorization](#) nella documentazione ufficiale di Kubernetes. [Per ulteriori informazioni su Chaos Mesh, consulta la documentazione ufficiale di Chaos Mesh.](#) Per ulteriori informazioni su Litmus, consulta la documentazione [ufficiale di Litmus.](#)

aws:eks:pod-cpu-stress

Esercita lo stress della CPU sui pod bersaglio. Per ulteriori informazioni, consulta [Azioni EKS Pod.](#)

Tipo di risorsa

- aws:eks:pod

Parameters

- duration— La durata dello stress test, in formato ISO 8601.
- percent : Opzionale. La percentuale di carico target, da 0 (senza carico) a 100 (a pieno carico). Il valore di default è 100.
- workers : Opzionale. Il numero di fattori di stress da utilizzare. L'impostazione predefinita è 0, che utilizza tutti i fattori di stress.
- kubernetesServiceAccount— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called "Configurazione dell'account di servizio Kubernetes"](#).
- fisPodContainerImage : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called "Immagini del contenitore Pod"](#).
- maxErrorsPercent Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- fisPodLabels : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- fisPodAnnotations : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- fisPodSecurityPolicy : Opzionale. La politica degli [standard di sicurezza Kubernetes da](#) utilizzare per il pod di orchestrazione degli errori creato da FIS e i contenitori temporanei. I `privileged`

baseline valori restricted possibili sono, e. Questa azione è compatibile con tutti i livelli di policy.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:pod-delete

Elimina i pod di destinazione. Per ulteriori informazioni, consulta [Azioni EKS Pod](#).

Tipo di risorsa

- `aws:eks:pod`

Parameters

- `gracePeriodSeconds` : Opzionale. La durata, in secondi, di attesa che il pod termini correttamente. Se il valore è 0, eseguiamo l'azione immediatamente. Se il valore è nil, utilizziamo il periodo di grazia predefinito per il pod.
- `kubernetesServiceAccount`— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called “Configurazione dell'account di servizio Kubernetes”](#).
- `fisPodContainerImage` : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called “Immagini del contenitore Pod”](#).
- `maxErrorsPercent` Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- `fisPodLabels` : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.

- `fisPodAnnotations` : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodSecurityPolicy` : Opzionale. La politica degli [standard di sicurezza Kubernetes da](#) utilizzare per il pod di orchestrazione degli errori creato da FIS e i contenitori temporanei. I `privileged` `baseline` valori `restricted` possibili sono, e. Questa azione è compatibile con tutti i livelli di `policy`.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-io-stress`

Esercita I/O lo stress sui pod bersaglio. Per ulteriori informazioni, consulta [Azioni EKS Pod](#).

Tipo di risorsa

- `aws:eks:pod`

Parameters

- `duration`— La durata dello stress test, in formato ISO 8601.
- `workers` : Opzionale. Il numero di worker. I lavoratori eseguono una combinazione di read/write operazioni sequenziali, casuali e mappate in memoria, sincronizzazione forzata e eliminazione della cache. Più processi secondari eseguono operazioni diverse sullo stesso file. I/O Il valore di default è 1.
- `percent` : Opzionale. La percentuale di spazio libero sul file system da utilizzare durante lo stress test. L'impostazione predefinita è 80%.

- `kubernetesServiceAccount`— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called “Configurazione dell'account di servizio Kubernetes”](#).
- `fisPodContainerImage` : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called “Immagini del contenitore Pod”](#).
- `maxErrorsPercent` Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- `fisPodLabels` : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodAnnotations` : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodSecurityPolicy` : Opzionale. La politica degli [standard di sicurezza Kubernetes da](#) utilizzare per il pod di orchestrazione degli errori creato da FIS e i contenitori temporanei. I `privileged` `baseline` valori `restricted` possibili sono, e. Questa azione è compatibile con tutti i livelli di `policy`.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-memory-stress`

Allevia lo stress della memoria sui pod bersaglio. Per ulteriori informazioni, consulta [Azioni EKS Pod](#).

Tipo di risorsa

- `aws:eks:pod`

Parameters

- `duration`— La durata dello stress test, in formato ISO 8601.
- `workers` : Opzionale. Il numero di fattori di stress da utilizzare. Il valore di default è 1.
- `percent` : Opzionale. La percentuale di memoria virtuale da utilizzare durante lo stress test. L'impostazione predefinita è 80%.
- `kubernetesServiceAccount`— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called “Configurazione dell'account di servizio Kubernetes”](#).
- `fisPodContainerImage` : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called “Immagini del contenitore Pod”](#).
- `maxErrorsPercent` Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- `fisPodLabels` : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodAnnotations` : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodSecurityPolicy` : Opzionale. La politica degli [standard di sicurezza Kubernetes da](#) utilizzare per il pod di orchestrazione degli errori creato da FIS e i contenitori temporanei. I `privileged` `baseline` valori `restricted` possibili sono, e. Questa azione è compatibile con tutti i livelli di `policy`.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:pod-network-blackhole-port

Elimina il traffico in entrata o in uscita per il protocollo e la porta specificati. Compatibile solo con la politica degli standard di [sicurezza Kubernetes](#). `privileged` Per ulteriori informazioni, consulta [Azioni EKS Pod](#).

Tipo di risorsa

- `aws:eks:pod`

Parameters

- `duration`— La durata del test, in formato ISO 8601.
- `protocol`— Il protocollo. I valori possibili sono `tcp` e `udp`.
- `trafficType`— Il tipo di traffico. I valori possibili sono `ingress` e `egress`.
- `port`— Il numero di porta.
- `kubernetesServiceAccount`— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called "Configurazione dell'account di servizio Kubernetes"](#).
- `fisPodContainerImage` : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called "Immagini del contenitore Pod"](#).
- `maxErrorsPercent` Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- `fisPodLabels` : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodAnnotations` : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:pod-network-latency

Aggiunge latenza e jitter all'interfaccia di rete utilizzando lo strumento per il traffico da o verso fonti specifiche. Compatibile solo con la politica degli standard di [sicurezza Kubernetes](#). `privileged` Per ulteriori informazioni, consulta [Azioni EKS Pod](#).

Utilizza il `flowsPercent` parametro per aggiungere latenza su una percentuale delle connessioni.

Tipo di risorsa

- `aws:eks:pod`

Parameters

- `duration`— La durata del test, in formato ISO 8601.
- `interface` : Opzionale. Le interfacce di rete, separate da virgole. I valori ALL e DEFAULT sono supportati. L'impostazione predefinita è DEFAULT che avrà come target l'interfaccia di rete principale per il sistema operativo.
- `delayMilliseconds` : Opzionale. Il ritardo, in millisecondi. L'impostazione predefinita è 200.
- `jitterMilliseconds` : Opzionale. Il jitter, in millisecondi. Il valore predefinito è 10.
- `flowsPercent` : Opzionale. La percentuale di flussi di rete che saranno influenzati dall'azione. Il valore di default è 100%.
- `sources` : Opzionale. Le fonti, separate da virgole, senza spazi. I valori possibili sono: un IPv4 indirizzo, un blocco IPv4 CIDR, un nome di dominio, un nome AZ (us-east-1a), un ID AZ (use1-az1), ALL e DYNAMODB S3. Se si specifica DYNAMODB o S3, ciò si applica solo all'endpoint regionale nella regione corrente. Per i nomi di dominio, vengono effettuati 10 tentativi di risoluzione DNS per raccogliere gli indirizzi IP. A causa del bilanciamento e della rotazione del carico DNS, questa azione potrebbe non compromettere tutti i possibili indirizzi IP verso cui il dominio potrebbe risolversi. L'impostazione predefinita è ALL, che corrisponde a tutto il traffico. IPv4
- `kubernetesServiceAccount`— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called "Configurazione dell'account di servizio Kubernetes"](#).

- `fisPodContainerImage` : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called "Immagini del contenitore Pod"](#).
- `maxErrorsPercent` Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- `fisPodLabels` : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodAnnotations` : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

`aws:eks:pod-network-packet-loss`

Aggiunge la perdita di pacchetti all'interfaccia di rete utilizzando lo `tc` strumento. Compatibile solo con la politica [Kubernetes Security Standards. privileged](#) Per ulteriori informazioni, consulta [Azioni EKS Pod](#).

Usa il `flowsPercent` parametro per iniettare la perdita di pacchetti su una percentuale delle connessioni.

Tipo di risorsa

- `aws:eks:pod`

Parameters

- `duration`— La durata del test, in formato ISO 8601.

- `interface` : Opzionale. Le interfacce di rete, separate da virgole. I valori ALL e DEFAULT sono supportati. L'impostazione predefinita è DEFAULT che avrà come target l'interfaccia di rete principale per il sistema operativo.
- `lossPercent` : Opzionale. La percentuale di perdita di pacchetti. L'impostazione predefinita è 7%.
- `flowsPercent` : Opzionale. La percentuale di flussi di rete che saranno interessati dall'azione. Il valore di default è 100%.
- `sources` : Opzionale. Le fonti, separate da virgole, senza spazi. I valori possibili sono: un IPv4 indirizzo, un blocco IPv4 CIDR, un nome di dominio, un nome AZ (us-east-1a), un ID AZ (use1-az1), ALL e DYNAMODB S3. Se si specifica DYNAMODB o S3, ciò si applica solo all'endpoint regionale nella regione corrente. Per i nomi di dominio, vengono effettuati 10 tentativi di risoluzione DNS per raccogliere gli indirizzi IP. A causa del bilanciamento e della rotazione del carico DNS, questa azione potrebbe non compromettere tutti i possibili indirizzi IP verso cui il dominio potrebbe risolversi. L'impostazione predefinita è ALL, che corrisponde a tutto il traffico. IPv4
- `kubernetesServiceAccount`— L'account del servizio Kubernetes. Per informazioni sulle autorizzazioni richieste, consultare [the section called “Configurazione dell'account di servizio Kubernetes”](#).
- `fisPodContainerImage` : Opzionale. L'immagine del contenitore utilizzata per creare il Fault Injector Pod. L'impostazione predefinita prevede l'utilizzo delle immagini fornite da AWS FIS. Per ulteriori informazioni, consulta [the section called “Immagini del contenitore Pod”](#).
- `maxErrorsPercent` Facoltativo. La percentuale di bersagli che possono fallire prima che fallisca l'iniezione del guasto. Il valore predefinito è 0.
- `fisPodLabels` : Opzionale. Le etichette Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.
- `fisPodAnnotations` : Opzionale. Le annotazioni Kubernetes allegate al pod di orchestrazione degli errori creato da FIS.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

aws:eks:terminate-nodegroup-instances

Esegue l'azione dell'API Amazon EC2 [TerminateInstances](#) sul gruppo di nodi di destinazione. Compatibile solo con i gruppi di nodi gestiti di Amazon EKS. I gruppi di nodi autogestiti non sono supportati. Per ulteriori informazioni, consulta [EKS manage compute](#).

Tipo di risorsa

- aws:eks:nodegroup

Parameters

- instanceTerminationPercentage— La percentuale (1-100) di istanze da terminare.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEKSAccess](#)

ElastiCache Azioni Amazon

AWS FIS supporta la seguente ElastiCache azione.

aws:elasticache:replicationgroup-interrupt-az-power

Interrompe l'alimentazione ai nodi nella zona di disponibilità specificata per i gruppi di ElastiCache replica di destinazione con Multi-AZ abilitato. È possibile influire su una sola zona di disponibilità

per gruppo di replica alla volta. Quando viene preso di mira un nodo primario, la replica di lettura corrispondente con il minor ritardo di replica viene promossa a principale. Le repliche sostitutive in lettura nella zona di disponibilità specificata vengono bloccate per la durata di questa azione, il che significa che i gruppi di replica di destinazione operano con una capacità ridotta. L'obiettivo di questa azione supporta sia i motori Redis che Valkey. L'azione non supporta l'opzione di distribuzione «serverless».

Tipo di risorsa

- `aws:elasticache:replicationgroup`

Parameters

- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

Note

L' Elasticache interrupt AZ power action ora supporta tutti i tipi di gruppi di replica, inclusi Valkey e Redis. Per rappresentare meglio questa funzionalità, l'azione è stata rinominata. Se stai attualmente utilizzando `aws:elasticache:interrupt-cluster-az-power`, ti consigliamo di passare alla nuova azione `aws:elasticache:replicationgroup-interrupt-az-power` per sfruttare le funzionalità più recenti.

Azioni di Amazon Kinesis Data Streams

Amazon Kinesis Data Streams supporta le seguenti azioni Kinesis.

Azioni

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

aws:kinesis:stream-provisioned-throughput-exception

Inietta risposte `ProvisionedThroughputExceededException` di errore nelle richieste per i Kinesis Data Streams mirati. Le operazioni supportate includono: `GetRecords`, `GetShardIterator`, `PutRecord` e `PutRecords`.

Tipo di risorsa

- `aws:kinesis:stream`

Parameters

- `durata`: la durata, che varia da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- `percentuale`: la percentuale (1-100) di chiamate in cui inserire il guasto.

Permissions

- `kinesis:InjectApiError`

aws:kinesis:stream-expired-iterator-exception

Inietta risposte `ExpiredIteratorException` di errore per le `GetRecords` chiamate destinate a Kinesis Data Streams specificati.

Tipo di risorsa

- `aws:kinesis:stream`

Parameters

- **durata:** la durata, che varia da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **percentuale:** la percentuale (1-100) di chiamate in cui inserire il guasto.

Permissions

- `kinesis:InjectApiError`

AWS Lambda azioni

AWS Lambda supporta le seguenti azioni Lambda

Azioni

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

aws:lambda:invocation-add-delay

Ritarda l'avvio di una funzione per un numero di millisecondi specificato dall'utente. L'effetto di questa azione è simile agli avviamenti a freddo Lambda, ma il tempo aggiuntivo viene impiegato come parte della durata fatturata e viene applicato a tutti gli ambienti di esecuzione anziché interessare solo i nuovi ambienti di esecuzione. Ciò significa che potresti riscontrare sia un avviamento a freddo Lambda che questo ritardo. Impostando un valore di latenza superiore al timeout configurato nella funzione Lambda, questa azione fornirà anche l'accesso a un evento di timeout ad alta fedeltà.

Tipo di risorsa

- `aws:lambda:function`

Parameters

- **durata**: la durata dell'azione. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **InvocationPercentage**: facoltativo. La percentuale (1-100) di chiamate di funzioni in cui inserire l'errore. Il valore di default è 100.
- **startupDelayMilliseconds** : Opzionale. La quantità di tempo in millisecondi (0-900.000) di attesa tra l'invocazione e l'esecuzione del codice della funzione. Il valore predefinito è 1000.

Permissions

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

aws:lambda:invocation-error

Contrassegna le chiamate alla funzione Lambda come non riuscite. Questa azione è utile per testare i meccanismi di gestione degli errori, come allarmi e configurazioni di nuovi tentativi. Durante l'utilizzo di questa azione, è possibile selezionare se eseguire o meno il codice della funzione prima di restituire un errore.

Tipo di risorsa

- `aws:lambda:function`

Parameters

- **durata**: il periodo di tempo in cui dura l'azione. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **InvocationPercentage**: facoltativo. La percentuale (1-100) di chiamate di funzioni in cui inserire l'errore. Il valore di default è 100.

- `preventExecution`: se il valore è vero, l'azione restituirà l'errore senza eseguire la funzione.

Permissions

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

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

Modifica il comportamento della funzione. Seleziona un tipo di contenuto e un codice di risposta HTTP per supportare le integrazioni con ALB, API-GW e VPC Lattice. Per abilitare le integrazioni con impatto selettivo a monte o a valle, puoi scegliere se restituire direttamente la risposta modificata o se eseguire la funzione e sostituire la risposta al termine dell'esecuzione della funzione.

Tipo di risorsa

- `aws:lambda:function`

Parameters

- `contentTypeHeader`— Valore stringa dell'intestazione del tipo di contenuto HTTP da restituire dalla funzione Lambda.
- `durata`: il periodo di tempo in cui dura l'azione. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- `InvocationPercentage`: facoltativo. La percentuale (1-100) di chiamate di funzioni in cui inserire l'errore. Il valore di default è 100.
- `preventExecution`: se il valore è vero, l'azione restituirà la risposta senza eseguire la funzione.
- `statusCode` — Valore del codice di stato HTTP (000-999) da restituire dalla funzione Lambda.

Permissions

- `s3:PutObject`

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

Azione Amazon MemoryDB

AWS FIS supporta la seguente azione MemoryDB.

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

Sospende la replica tra un cluster regionale e tutti gli altri cluster regionali all'interno del cluster multiregionale. Il cluster regionale interessato è il cluster della regione in cui è in esecuzione l'esperimento FIS. Mentre la replica è in pausa, il cluster multiregionale non può essere aggiornato. Una volta completata l'azione, potrebbero essere necessari alcuni minuti prima che il cluster multiregionale ritorni allo stato disponibile. Per ulteriori informazioni su Amazon MemoryDB Multi-Region, consulta la Amazon MemoryDB Multi-Region [Developer Guide](#). [Per la disponibilità regionale, consulta Prerequisiti e limitazioni multiregionali di MemoryDB.](#)

Tipo di risorsa

- `aws:memorydb:multi-region-cluster`

Parameters

- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

Azioni di rete

AWS FIS supporta le seguenti azioni di rete.

Azioni

- [aws:network:disrupt-connectivity](#)
- [aws:network:route-table-disrupt-cross-region-connectivity](#)
- [aws:network:transit-gateway-disrupt-cross-region-connectivity](#)
- [aws:network:disrupt-vpc-endpoint](#)

aws:network:disrupt-connectivity

Nega il traffico specificato verso le sottoreti di destinazione clonando temporaneamente l'elenco di controllo dell'accesso alla rete originale (ACL di rete) associato alla sottorete di destinazione. FIS aggiunge regole di negazione all'ACL di rete clonato, che ha un tag `ManagedByFIS=true`, e lo associa alla sottorete per tutta la durata dell'azione. Al completamento dell'azione, FIS elimina l'ACL di rete clonato e ripristina l'associazione ACL di rete originale.

Tipo di risorsa

- `aws:ec2:subnet`

Parameters

- `scope`— Il tipo di traffico da negare. Se l'ambito non lo è `all`, il numero massimo di ingressi nella rete ACLs è 20. I valori possibili sono:
 - `all`— Impedisce a tutto il traffico in entrata e in uscita dalla sottorete. Si noti che questa opzione consente il traffico all'interno della sottorete, incluso il traffico da e verso le interfacce di rete nella sottorete.
 - `availability-zone`— Impedisce il traffico intra-VPC da e verso le sottoreti in altre zone di disponibilità. Il numero massimo di sottoreti che possono essere utilizzate come target in un VPC è 30.
 - `dynamodb`— Impedisce il traffico da e verso l'endpoint regionale per DynamoDB nella regione corrente.
 - `prefix-list`— Impedisce il traffico da e verso l'elenco di prefissi specificato.
 - `s3`— Impedisce il traffico da e verso l'endpoint regionale per Amazon S3 nella regione corrente.

- `s3express`— Impedisce il traffico da e verso l'endpoint zonale per Amazon S3 Express One Zone nella zona di destinazione delle sottoreti di destinazione. Le sottoreti di destinazione devono risiedere dove S3 Express One Zone è attualmente disponibile AZs . Per ulteriori informazioni, consulta Zone e regioni di disponibilità di [S3 Express One Zone](#).
- `vpc`— Impedisce al traffico di entrare e uscire dal VPC.
- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- `prefixListIdentifier`— Se l'ambito è `prefix-list`, questo è l'identificatore dell'elenco dei prefissi gestiti dal cliente. È possibile specificare un nome, un ID o un ARN. L'elenco dei prefissi può avere al massimo 10 voci.

Permissions

- `ec2:CreateNetworkAcl`— Crea l'ACL di rete con il tag `ManagedByFIS=true`.
- `ec2:CreateNetworkAclEntry`— L'ACL di rete deve avere il tag `ManagedByFIS=true`.
- `ec2:CreateTags`
- `ec2>DeleteNetworkAcl`— L'ACL di rete deve avere il tag `ManagedByFIS=true`.
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNetworkAcls`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `ec2:GetManagedPrefixListEntries`
- `ec2:ReplaceNetworkAclAssociation`

AWS politica gestita

- [AWSFaultInjectionSimulatorNetworkAccess](#)

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

Blocca il traffico che ha origine nelle sottoreti di destinazione ed è destinato alla regione specificata. Crea tabelle di rotte che includono tutte le rotte che la Regione deve isolare. Per consentire a FIS di

creare queste tabelle di routing, aumenta la quota `routes per route table` Amazon VPC a 250 (o 350 se `region` il parametro è `us-east-1`) più il numero di rotte nelle tabelle di routing esistenti.

Tipo di risorsa

- `aws:ec2:subnet`

Parameters

- `region`— Il codice della regione da isolare (ad esempio, `eu-west-1`).
- `duration`— La durata dell'azione. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

- `ec2:AssociateRouteTable`
- `ec2:CreateManagedPrefixList` †
- `ec2:CreateNetworkInterface` †
- `ec2:CreateRoute` †
- `ec2:CreateRouteTable` †
- `ec2:CreateTags` †
- `ec2>DeleteManagedPrefixList` †
- `ec2>DeleteNetworkInterface` †
- `ec2>DeleteRouteTable` †
- `ec2:DescribeManagedPrefixLists`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeRouteTables`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcPeeringConnections`
- `ec2:DescribeVpcs`
- `ec2:DisassociateRouteTable`
- `ec2:GetManagedPrefixListEntries`

- `ec2:ModifyManagedPrefixList †`
- `ec2:ModifyVpcEndpoint`
- `ec2:ReplaceRouteTableAssociation`

† Ambito utilizzando il tag `managedByFIS=true`. Non è necessario gestire questo tag. AWS FIS aggiunge e rimuove questo tag durante l'esperimento.

AWS politica gestita

- [AWSFaultInjectionSimulatorNetworkAccess](#)

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

Blocca il traffico proveniente dagli allegati di peering del gateway di transito di destinazione destinato alla regione specificata.

Tipo di risorsa

- `aws:ec2:transit-gateway`

Parameters

- `region`— Il codice della regione da isolare (ad esempio, `eu-west-1`).
- `duration`— La durata dell'azione. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, `PT1 M` rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorNetworkAccess](#)

aws:network:disrupt-vpc-endpoint

Blocca il traffico in entrata e in uscita degli endpoint VPC dell'interfaccia di destinazione. FIS crea un gruppo di sicurezza gestito con regole vuote e sostituisce temporaneamente i gruppi di sicurezza degli endpoint VPC di destinazione con questo gruppo di sicurezza gestito. Se vengono apportate modifiche alle risorse di destinazione durante l'esecuzione dell'azione, l'azione avrà esito negativo e le risorse non verranno ripristinate allo stato precedente all'esperimento. Inoltre, se un gruppo di sicurezza gestito da FIS viene modificato durante l'esecuzione dell'azione, non verrà eliminato da FIS.

Tipo di risorsa

- aws:ec2:vpc-endpoint

Parameters

- **duration**— La durata dell'azione. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

* L'autorizzazione è richiesta solo se si utilizzano come target endpoint VPC interregionali

Azioni Amazon RDS

AWS FIS supporta le seguenti azioni Amazon RDS.

Azioni

- [aws:rds:failover-db-cluster](#)
- [aws:rds:reboot-db-instances](#)

aws:rds:failover-db-cluster

Esegue l'action [failover](#) dell'API Amazon RDS DBCluster sul cluster Aurora DB di destinazione. Sono supportati i cluster RDS e i cluster DocumentDB.

Tipo di risorsa

- aws:rds:cluster

Parameters

- Nessuno

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorRDSAccess](#)

aws:rds:reboot-db-instances

Esegue l'azione [Reboot](#) dell'API Amazon RDS DBInstance sull'istanza DB di destinazione. Sono supportati i cluster RDS e i cluster DocumentDB.

Tipo di risorsa

- `aws:rds:db`

Parameters

- `forceFailover` : Opzionale. Se il valore è `true` e se le istanze sono Multi-AZ, forza il failover da una zona di disponibilità all'altra. Il valore predefinito è `false`.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorRDSAccess](#)

Operazioni di Amazon S3

AWS FIS supporta la seguente azione di Amazon S3.

Azioni

- [aws:s3:bucket-pause-replication](#)

aws:s3:bucket-pause-replication

Sospende la replica dai bucket di origine di destinazione ai bucket di destinazione. I bucket di destinazione possono trovarsi in regioni AWS diverse o all'interno della stessa regione del bucket di origine. Gli oggetti esistenti possono continuare a essere replicati fino a un'ora dopo l'inizio dell'azione. Questa azione supporta solo il targeting per tag. Per ulteriori informazioni su Amazon S3 Replication, consulta la guida per l'utente di [Amazon S3](#).

Tipo di risorsa

- `aws:s3:bucket`

Parameters

- **duration**— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.
- **region**— La regione AWS in cui si trovano i bucket di destinazione.
- **destinationBuckets** : Opzionale. Elenco separato da virgole dei bucket S3 di destinazione.
- **prefixes** : Opzionale. Elenco separato da virgole dei prefissi delle chiavi degli oggetti S3 dai filtri delle regole di replica. Le regole di replica dei bucket di destinazione con un filtro basato sui prefissi verranno messe in pausa.

Permissions

- **S3:PutReplicationConfiguration** con la chiave di condizione impostata su **S3:IsReplicationPauseRequest True**
- **S3:GetReplicationConfiguration** con chiave di condizione **S3:IsReplicationPauseRequest** impostata su **True**
- **S3:PauseReplication**
- **S3>ListAllMyBuckets**
- **tag:GetResources**

Per un esempio di policy, consulta [Esempio: utilizzare i tasti condizionali per aws:s3:bucket-pause-replication](#).

Azioni di Systems Manager

AWS FIS supporta le seguenti azioni di Systems Manager.

Azioni

- [aws:ssm:send-command](#)
- [aws:ssm:start-automation-execution](#)

aws:ssm:send-command

Esegue l'azione dell'API Systems Manager [SendCommand](#) sulle istanze EC2 di destinazione. Il documento Systems Manager (documento SSM) definisce le azioni che Systems Manager esegue sulle istanze. Per ulteriori informazioni, consulta [Usa l'azione aws:ssm:send-command](#).

Tipo di risorsa

- `aws:ec2:instance`

Parameters

- `documentArn`— L'Amazon Resource Name (ARN) del documento. Nella console, questo parametro viene completato automaticamente se scegli un valore da Action type che corrisponde a uno dei documenti [AWS FIS SSM preconfigurati](#).
- `documentVersion` : Opzionale. La versione del documento. Se vuoto, viene eseguita la versione predefinita.
- `documentParameters`— Condizionale. I parametri obbligatori e facoltativi accettati dal documento. Il formato è un oggetto JSON con chiavi che sono stringhe e valori che sono stringhe o matrici di stringhe.
- `duration`— La durata, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

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

AWS politica gestita

- [AWSFaultInjectionSimulatorEC2Accesso](#)

aws:ssm:start-automation-execution

Esegue l'azione dell'API Systems Manager [StartAutomationExecution](#).

Tipo di risorsa

- Nessuno

Parameters

- `documentArn`— L'Amazon Resource Name (ARN) del documento di automazione.
- `documentVersion` : Opzionale. La versione del documento. Se vuoto, viene eseguita la versione predefinita.
- `documentParameters`— Condizionale. I parametri obbligatori e facoltativi accettati dal documento. Il formato è un oggetto JSON con chiavi che sono stringhe e valori che sono stringhe o matrici di stringhe.
- `maxDuration`— Il tempo massimo consentito per il completamento dell'esecuzione dell'automazione, da un minuto a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT1 M rappresenta un minuto. Nella AWS FIS console, si immette il numero di secondi, minuti o ore.

Permissions

- `ssm:GetAutomationExecution`
- `ssm:StartAutomationExecution`
- `ssm:StopAutomationExecution`
- `iam:PassRole` : Opzionale. Obbligatorio se il documento di automazione assume un ruolo.

AWS politica gestita

- [AWSFaultInjectionSimulatorSSMAccess](#)

AWS Direct Connect azioni

AWS FIS supporta la seguente AWS Direct Connect azione.

Azioni

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

aws:directconnect:virtual-interface-disconnect

Verifica la resilienza della AWS Direct Connect connessione interrompendo temporaneamente le sessioni Border Gateway Protocol (BGP) tra le reti locali e i peer associati alle interfacce virtuali di destinazione (). VIFs Prima di iniziare l'esperimento, FIS verifica che tutti i VIF destinatari dell'esperimento siano in uno stato «disponibile» e che ogni VIF abbia tutti i peer BGP con stato «disponibile» e stato BGP «attivo». Durante l'esperimento, le sessioni di peering BGP per le interfacce virtuali mirate verranno disattivate. Per informazioni dettagliate sui test di failover di Direct Connect, consulta la [AWS Direct Connect documentazione](#).

Tipo di risorsa

- aws: directconnect: interfaccia virtuale

Parameters

- `duration`— La durata, da 10 minuti a 12 ore. Nell' AWS FIS API, il valore è una stringa in formato ISO 8601. Ad esempio, PT10 M rappresenta dieci minuti. Nella console AWS FIS, inserisci il numero di secondi, minuti o ore.

Permissions

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

Usa i documenti SSM di Systems Manager con AWS FIS

AWS FIS supporta tipi di errore personalizzati tramite l'agente AWS Systems Manager SSM e l' AWS azione FIS. [aws:ssm:send-command](#) I documenti SSM preconfigurati di Systems Manager

(documenti SSM) che possono essere utilizzati per creare azioni di errore comuni sono disponibili come AWS documenti pubblici che iniziano con il AWSFIS prefisso -.

SSM Agent è un software Amazon che può essere installato e configurato su istanze Amazon EC2, server locali o macchine virtuali (VMs). Ciò consente a Systems Manager di gestire queste risorse. L'agente elabora le richieste provenienti da Systems Manager e quindi le esegue come specificato nella richiesta. Puoi includere il tuo documento SSM per inserire errori personalizzati o fare riferimento a uno dei documenti pubblici di proprietà di Amazon.

Requisiti

Per le azioni che richiedono l'esecuzione dell'azione da parte dell'agente SSM sulla destinazione, devi assicurarti di quanto segue:

- L'agente è installato sulla destinazione. L'agente SSM è installato per impostazione predefinita su alcune Amazon Machine Images (AMIs). Altrimenti, puoi installare l'agente SSM sulle tue istanze. Per ulteriori informazioni, consulta [Installare manualmente l'agente SSM per le istanze EC2 nella Guida per l'utente](#).AWS Systems Manager
- Systems Manager è autorizzato a eseguire azioni sulle istanze. Concedi l'accesso utilizzando un profilo di istanza IAM. Per ulteriori informazioni, consulta [Creare un profilo di istanza IAM per Systems Manager](#) e [Collegare un profilo di istanza IAM a un'istanza EC2](#) nella Guida per l'AWS Systems Manager utente.

Usa l'azione aws:ssm:send-command

Un documento SSM definisce le operazioni eseguite da Systems Manager sulle istanze gestite. Systems Manager include una serie di documenti preconfigurati oppure è possibile crearne di personalizzati. Per ulteriori informazioni sulla creazione del proprio documento SSM, vedere [Creating Systems Manager documents](#) nella Guida per l'AWS Systems Manager utente. Per ulteriori informazioni sui documenti SSM in generale, consultate [AWS Systems Manager i documenti nella Guida](#) per l'AWS Systems Manager utente.

AWS FIS fornisce documenti SSM preconfigurati. [È possibile visualizzare i documenti SSM preconfigurati in Documenti nella console: documenti](#). AWS Systems Manager <https://console.aws.amazon.com/systems-manager/> Puoi anche scegliere tra una selezione di documenti preconfigurati nella console FIS. AWS Per ulteriori informazioni, consulta [Documenti FIS SSM preconfigurati AWS](#).

Per utilizzare un documento SSM nei vostri esperimenti AWS FIS, potete usare l'azione.

[aws:ssm:send-command](#) Questa azione recupera ed esegue il documento SSM specificato sulle istanze di destinazione.

Quando si utilizza l'`aws : ssm : send - command` nel modello di esperimento, è necessario specificare parametri aggiuntivi per l'azione, inclusi i seguenti:

- `documentArn`: obbligatorio L'Amazon Resource Name (ARN) del documento SSM.
- `documentParameters`— Condizionale. I parametri obbligatori e facoltativi accettati dal documento SSM. Il formato è un oggetto JSON con chiavi che sono stringhe e valori che sono stringhe o matrici di stringhe.
- `documentVersion` : Opzionale. La versione del documento SSM da eseguire.

È possibile visualizzare le informazioni per un documento SSM (inclusi i parametri del documento) utilizzando la console Systems Manager o la riga di comando.

Per visualizzare informazioni su un documento SSM utilizzando la console

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/>.
2. Nel pannello di navigazione, scegli Documenti.
3. Seleziona il documento e scegli la scheda Dettagli.

Per visualizzare informazioni su un documento SSM utilizzando la riga di comando

Usa il comando SSM [describe-document](#).

Scopri di più sullo stato dell'azione

Lo stato di azione SSM è determinato dagli stati dei [comandi SSM](#).

Documenti FIS SSM preconfigurati AWS

È possibile utilizzare documenti AWS FIS SSM preconfigurati con l'`aws : ssm : send - command` nei modelli di esperimento.

Requisiti

- I documenti SSM preconfigurati forniti da AWS FIS sono supportati solo sui seguenti sistemi operativi:
 - Amazon Linux 2023, Amazon Linux 2
 - Ubuntu
 - REGOLA 8, 9
 - CentOS 9
- I documenti SSM preconfigurati forniti da AWS FIS sono supportati solo sulle istanze EC2. Non sono supportati su altri tipi di nodi gestiti, come i server locali.

Per utilizzare questi documenti SSM in esperimenti sulle attività ECS, usa il corrispondente. [the section called “Azioni Amazon ECS”](#) Ad esempio, l'aws:ecs:task-cpu-stressazione utilizza il AWSFIS-Run-CPU-Stress documento.

Documenti

- [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](#)

Differenza tra la durata dell'azione e quella DurationSeconds nei AWS documenti FIS SSM

Alcuni documenti SSM limitano il proprio tempo di esecuzione, ad esempio il DurationSeconds parametro viene utilizzato da alcuni documenti FIS SSM preconfigurati AWS . Di conseguenza, è necessario specificare due durate indipendenti nella definizione dell'azione FIS: AWS

- **Action duration:** Per gli esperimenti con una singola azione, la durata dell'azione è equivalente alla durata dell'esperimento. Nel caso di azioni multiple, la durata dell'esperimento dipende dalla durata delle singole azioni e dall'ordine in cui vengono eseguite. AWS FIS monitora ogni azione fino al termine della sua durata.
- **Parametro del documento `DurationSeconds`:** la durata, specificata in secondi, per la quale verrà eseguito il documento SSM.

Puoi scegliere valori diversi per i due tipi di durata:

- **Action duration exceeds `DurationSeconds`:** L'esecuzione del documento SSM termina prima del completamento dell'azione. AWS FIS attende che sia trascorsa la durata dell'azione prima di avviare le azioni successive.
- **Action duration is shorter than `DurationSeconds`:** Il documento SSM continua l'esecuzione dopo il completamento dell'azione. Se l'esecuzione del documento SSM è ancora in corso e la durata dell'azione è scaduta, lo stato dell'azione viene impostato su `Completato`. AWS FIS monitora l'esecuzione solo fino alla scadenza della durata dell'azione.

Si noti che alcuni documenti SSM hanno durate variabili. Ad esempio, i documenti AWS FIS SSM hanno la possibilità di installare i prerequisiti, che possono estendere la durata complessiva dell'esecuzione oltre il parametro specificato. `DurationSeconds` Pertanto, se si imposta la durata dell'azione e `DurationSeconds` lo stesso valore, è possibile che lo script SSM venga eseguito più a lungo della durata dell'azione.

AWSFIS-Run-CPU-Stress

Esegui lo stress della CPU su un'istanza utilizzando lo `stress-ng` strumento. Utilizza il documento [AWSFIS-RunSSM -CPU-stress](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **`DurationSeconds`:** obbligatorio La durata dello stress test della CPU, in secondi.

- CPU : Opzionale. Il numero di fattori di stress della CPU da utilizzare. L'impostazione predefinita è 0, che utilizza tutti i fattori di stress della CPU.
- LoadPercent : Opzionale. La percentuale di carico della CPU di destinazione, da 0 (senza carico) a 100 (a pieno carico). Il valore di default è 100.
- InstallDependencies : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. La dipendenza è. `stress-ng`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Disk-Fill

Alloca lo spazio su disco sul volume principale di un'istanza per simulare un guasto completo del disco. Utilizza il documento SSM [AWSFIS-Run-Disk-Fill](#).

Se l'esperimento che inietta questo errore viene interrotto, manualmente o tramite una condizione di arresto, AWS FIS tenta di ripristinare il sistema annullando il documento SSM in esecuzione. Tuttavia, se il disco è pieno al 100%, a causa dell'errore o dell'attività dell'applicazione, Systems Manager potrebbe non essere in grado di completare l'operazione di annullamento. Pertanto, se è necessario interrompere l'esperimento, assicuratevi che il disco non si riempia al 100%.

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- DurationSeconds: obbligatorio La durata del test di riempimento del disco, in secondi.
- Percent : Opzionale. La percentuale del disco da allocare durante il test di riempimento del disco. L'impostazione predefinita è 95%.
- InstallDependencies : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono, e. `atd kmod fallocate`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-IO-Stress

Esegue lo stress IO su un'istanza utilizzando lo stress-ng strumento. Utilizza il documento [AWSFIS-RunSSM -IO-Stress](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **DurationSeconds**: obbligatorio La durata dello stress test IO, in secondi.
- **Workers** : Opzionale. Il numero di lavoratori che eseguono una combinazione di read/write operazioni sequenziali, casuali e mappate in memoria, sincronizzazione forzata e eliminazione della cache. Più processi secondari eseguono operazioni diverse sullo stesso file. I/O Il valore di default è 1.
- **Percent** : Opzionale. La percentuale di spazio libero sul file system da utilizzare durante lo stress test IO. L'impostazione predefinita è 80%.
- **InstallDependencies** : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. La dipendenza è. stress-ng

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Kill-Process

Arresta il processo specificato nell'istanza, utilizzando il killall comando. Utilizza il documento [AWSFIS-RunSSM -Kill-Process](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **ProcessName**: obbligatorio Il nome del processo da interrompere.
- **Signal** : Opzionale. Il segnale da inviare insieme al comando. I valori possibili sono SIGTERM (che il ricevitore può scegliere di ignorare) e SIGKILL (che non possono essere ignorati). Il valore predefinito è SIGTERM.
- **InstallDependencies** Facoltativo. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. La dipendenza è. `killall`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Memory-Stress

Esegue lo stress della memoria su un'istanza utilizzando lo `stress-ng` strumento. Utilizza il documento [AWSFIS-RunSSM -Memory-Stress](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **DurationSeconds**: obbligatorio La durata del test di stress della memoria, in secondi.
- **Workers** : Opzionale. Il numero di fattori di stress della memoria virtuale. Il valore di default è 1.

- **Percent**: obbligatorio La percentuale di memoria virtuale da utilizzare durante lo stress test della memoria.
- **InstallDependencies** : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. La dipendenza è. `stress-ng`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Network-Blackhole-Port

Elimina il traffico in entrata o in uscita per il protocollo e la porta utilizzando lo `iptables` strumento. Utilizza il documento SSM [AWSFIS-Run-Network-Blackhole-Port](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **Protocol**: obbligatorio Il protocollo. I valori possibili sono `tcp` e `udp`.
- **Port**: obbligatorio Il numero di porta.
- **TrafficType** : Opzionale. Il tipo di traffico. I valori possibili sono `ingress` e `egress`. Il valore di default è `ingress`.
- **DurationSeconds**: obbligatorio Durata del test del buco nero della rete, in secondi.
- **InstallDependencies** : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono `atd`, `edig`. `iptables`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Network-Latency

Aggiunge latenza all'interfaccia di rete utilizzando lo `tc` strumento. Utilizza il documento SSM [AWSFIS-Run-Network-Latency](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- `Interface` : Opzionale. L'interfaccia di rete. Il valore predefinito è `eth0`.
- `DelayMilliseconds` Facoltativo. Il ritardo, in millisecondi. L'impostazione predefinita è 200.
- `DurationSeconds`: obbligatorio La durata del test di latenza della rete, in secondi.
- `InstallDependencies` : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono `atd`, e. `dig tc`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Network-Latency-Sources

Aggiunge latenza e jitter all'interfaccia di rete utilizzando `tc` lo strumento per il traffico da o verso fonti specifiche. Utilizza il documento SSM [AWSFIS-Run-Network-Latency-Sources](#).

Utilizzate il `FlowsPercent` parametro per aggiungere latenza su una percentuale delle connessioni.

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **Interface** : Opzionale. Le interfacce di rete, separate da virgole. I valori ALL e DEFAULT sono supportati. L'impostazione predefinita è DEFAULT che avrà come target l'interfaccia di rete principale per il sistema operativo.
- **DelayMilliseconds** : Opzionale. Il ritardo, in millisecondi. L'impostazione predefinita è 200.
- **JitterMilliseconds** : Opzionale. Il jitter, in millisecondi. Il valore predefinito è 10.
- **FlowsPercent** : Opzionale. La percentuale di flussi di rete che saranno influenzati dall'azione. Il valore di default è 100%.
- **Sources**: obbligatorio Le fonti, separate da virgole, senza spazi. I valori possibili sono: un IPv4 indirizzo, un blocco IPv4 CIDR, un nome di dominio, un nome AZ (us-east-1a), un ID AZ (use1-az1), ALL e DYNAMODB S3. Se si specifica DYNAMODB o S3, ciò si applica solo all'endpoint regionale nella regione corrente.
- **TrafficType** : Opzionale. Il tipo di traffico. I valori possibili sono `ingress` e `egress`. Il valore di default è `ingress`.
- **DurationSeconds**: obbligatorio La durata del test di latenza della rete, in secondi.
- **InstallDependencies** : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione se non sono già installate. Il valore predefinito è `True`. Le dipendenze sono `atd`, `dig`, `jq` e `ls`.

Quando si utilizza questo documento, il ruolo dell'esperimento richiede le seguenti autorizzazioni:

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

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

```
{"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

Aggiunge la perdita di pacchetti all'interfaccia di rete utilizzando lo `tc` strumento. Utilizza il documento SSM [AWSFIS-Run-Network-Packet-Loss](#).

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **Interface** : Opzionale. L'interfaccia di rete. Il valore predefinito è `eth0`.
- **LossPercent** Facoltativo. La percentuale di perdita di pacchetti. L'impostazione predefinita è 7%.
- **DurationSeconds**: obbligatorio La durata del test di perdita di pacchetti di rete, in secondi.
- **InstallDependencies** : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione. Il valore predefinito è `True`. Le dipendenze sono `atd`, `elfos`, `dig` `tc`

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

AWSFIS-Run-Network-Packet-Loss-Sources

Aggiunge la perdita di pacchetti all'interfaccia di rete utilizzando lo `tc` strumento per il traffico da o verso fonti specifiche. Utilizza il documento SSM [AWSFIS-Run-Network-Packet-Loss-Sources](#).

Utilizzate il `FlowsPercent` parametro per iniettare la perdita di pacchetti su una percentuale delle connessioni.

Tipo di azione (solo console)

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

ARN

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

Parametri del documento

- **Interface** : Opzionale. Le interfacce di rete, separate da virgole. I valori ALL e DEFAULT sono supportati. L'impostazione predefinita è DEFAULT che avrà come target l'interfaccia di rete principale per il sistema operativo.
- **LossPercent** : Opzionale. La percentuale di perdita di pacchetti. L'impostazione predefinita è 7%.
- **FlowsPercent** : Opzionale. La percentuale di flussi di rete che saranno interessati dall'azione. Il valore di default è 100%.
- **Sources**: obbligatorio Le fonti, separate da virgole, senza spazi. I valori possibili sono: un IPv4 indirizzo, un blocco IPv4 CIDR, un nome di dominio, un nome AZ (us-east-1a), un ID AZ (use1-az1), ALL e. DYNAMODB S3 Se si specifica DYNAMODB oS3, ciò si applica solo all'endpoint regionale nella regione corrente.
- **TrafficType** : Opzionale. Il tipo di traffico. I valori possibili sono `ingress` e `egress`. Il valore di default è `ingress`.
- **DurationSeconds**: obbligatorio La durata del test di perdita di pacchetti di rete, in secondi.
- **InstallDependencies** : Opzionale. Se il valore è `True`, Systems Manager installa le dipendenze richieste sulle istanze di destinazione. Il valore predefinito è `True`. Le dipendenze sono `atd`, `dig`, `jq` e. `ls` `tc`

Quando si utilizza questo documento, il ruolo dell'esperimento richiede le seguenti autorizzazioni:

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

Di seguito è riportato un esempio della stringa che è possibile immettere nella console.

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

Esempi

Per un esempio di modello di esperimento, vedi [the section called “Esegui un documento FIS SSM preconfigurato AWS”](#).

Per un esempio di tutorial, consultare [Esegui lo stress della CPU su un'istanza](#).

Limitazioni

- I seguenti documenti non possono essere eseguiti in parallelo:
 - 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

script di rollback

AWS I documenti FIS SSM creano automaticamente script di rollback come meccanismo di sicurezza per ripristinare lo stato del sistema dopo esperimenti di failinjection. Questi script assicurano che gli errori iniettati vengano rimossi, anche se l'azione fallisce o viene interrotta in modo imprevisto.

Creazione di script di rollback

Gli script di rollback vengono creati automaticamente quando iniziano gli esperimenti di iniezione dei guasti.

Dettagli sulla creazione

- Posizione: gli script vengono creati nella `/var/lib/amazon/ssm/` directory.
- Schema di denominazione: `FAULT_NAME-FAULT_IDENTIFIER-Rollback.sh`
`FAULT_IDENTIFIER` dov'è una stringa di 32 caratteri generata casualmente
- Tempistica: creata all'inizio di ogni esperimento di iniezione del guasto, prima che inizi l'iniezione del guasto.
- Contenuto: contiene tutte le variabili di ambiente e i comandi necessari per invertire l'errore specifico.

Ad esempio, un esperimento di latenza di rete potrebbe creare uno script di rollback su `/var/lib/amazon/ssm/NetworkLatency-abc123-Rollback.sh`

Registrazione del rollback

Gli script di rollback implementano la doppia registrazione per acquisire tutte le attività di rollback a fini di risoluzione dei problemi e controllo.

Posizioni dei file di registro

Quando viene eseguito uno script di rollback, crea i log in due posizioni:

- File temporanei: `/tmp/aws-fis-rollback-TIMESTAMP-PID.log`
- Registri di sistema: inviati a syslog con funzionalità `local0.info`

Denominazione dei file di registro

I file di registro temporanei utilizzano la seguente convenzione di denominazione:

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

YYYY-MM-DDTHH:MM:SSZ Dov'è il timestamp UTC e *PID* l'ID di processo dello script di rollback.

Configurazione Syslog

I log di rollback vengono inviati a syslog con la seguente configurazione:

- Etichetta: `aws-fis-rollback`
- Priorità — `local0.info`
- Formato — `[YYYY-MM-DDTHH:MM:SSZ] log_message`

Per visualizzare i registri di rollback

Usa il seguente comando per visualizzare tutti i log di rollback dal journal systemd:

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

Risoluzione dei problemi

Utilizzare la procedura seguente per risolvere i problemi.

Per risolvere i problemi relativi ai documenti SSM

1. Apri la console all' AWS Systems Manager indirizzo. <https://console.aws.amazon.com/systems-manager/>
2. Nel riquadro di navigazione, scegli Gestione dei nodi, Esegui comando.

3. Nella scheda Cronologia dei comandi, utilizza i filtri per individuare l'esecuzione del documento.
4. Scegli l'ID del comando per aprirne la pagina dei dettagli.
5. Scegli l'ID dell'istanza. Controlla l'output e gli errori per ogni passaggio.

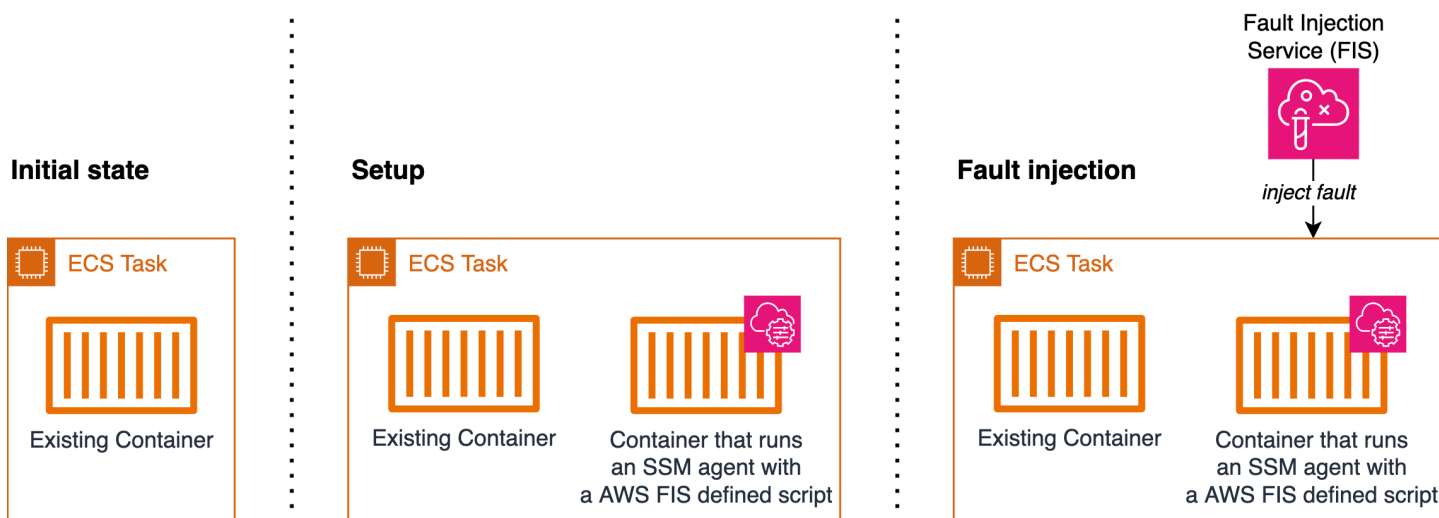
Usa le azioni AWS FIS `aws:ecs:task`

Puoi utilizzare le azioni `aws:ecs:task` per inserire errori nelle tue attività Amazon ECS. Sono supportati i tipi di capacità di Amazon EC2 e Fargate.

Queste azioni utilizzano [documenti AWS Systems Manager \(SSM\)](#) per inserire errori. Per utilizzare `aws:ecs:task` le azioni, dovrai aggiungere un contenitore con un agente SSM alla definizione del task Amazon Elastic Container Service (Amazon ECS). Il contenitore esegue uno [script definito dal AWS FIS](#) che registra l'attività Amazon ECS come istanza gestita nel servizio SSM. Inoltre, lo script recupera i metadati delle attività per aggiungere tag all'istanza gestita. La configurazione consentirà a AWS FIS di risolvere il task di destinazione. Questo paragrafo si riferisce alla configurazione riportata nello schema seguente.

Quando esegui un targeting di un esperimento AWS FIS `aws:ecs:task`, AWS FIS mappa le attività Amazon ECS di destinazione specificate in un modello di esperimento AWS FIS a un set di istanze gestite SSM utilizzando un tag di risorsa, `ECS_TASK_ARN`. Il valore del tag è l'ARN del task Amazon ECS associato in cui devono essere eseguiti i documenti SSM. Questo paragrafo si riferisce alla Fault Injection nel diagramma seguente.

Il diagramma seguente esemplifica la configurazione e l'iniezione dei guasti in un'operazione con un contenitore esistente.



Azioni

- [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”](#)

Limitazioni

- Le seguenti azioni non possono essere eseguite in parallelo:
 - aws:ecs:task-network-blackhole-port
 - aws:ecs:task-network-latency
 - aws:ecs:task-network-packet-loss
- Se hai abilitato Amazon ECS Exec, devi disabilitarlo prima di poter utilizzare queste azioni.
- L'esecuzione del documento SSM potrebbe avere lo stato annullato anche se l'esperimento ha lo stato completato. Quando si eseguono azioni Amazon ECS, la durata fornita dal cliente viene utilizzata sia per la durata dell'azione nell'esperimento che per la durata del documento Amazon EC2 Systems Manager (SSM). Dopo l'avvio dell'azione, è necessario del tempo prima che il documento SSM inizi a funzionare. Di conseguenza, quando viene raggiunta la durata dell'azione specificata, al documento SSM potrebbero rimanere ancora alcuni secondi per completare l'esecuzione. Quando viene raggiunta la durata dell'azione dell'esperimento, l'azione viene interrotta e l'esecuzione del documento SSM viene annullata. L'iniezione del guasto ha avuto successo.

Requisiti

- Aggiungere le seguenti autorizzazioni al ruolo dell'[esperimento AWS FIS](#):
 - ecs:DescribeTasks
 - ssm:SendCommand
 - ssm:ListCommands
 - ssm:CancelCommand

- Aggiungi le seguenti autorizzazioni al ruolo [IAM dell'attività](#) Amazon ECS:
 - `ssm:CreateActivation`
 - `ssm:AddTagsToResource`
 - `iam:PassRole`

Tieni presente che puoi specificare l'ARN del ruolo dell'istanza gestita come risorsa per.

`iam:PassRole`

- Crea un [ruolo IAM per l'esecuzione delle attività](#) di Amazon ECS e aggiungi la policy ECSTask ExecutionRolePolicy gestita di [Amazon](#).
- Nella definizione dell'attività, imposta la variabile `MANAGED_INSTANCE_ROLE_NAME` di ambiente sul nome del [ruolo dell'istanza gestita](#). Questo è il ruolo che verrà assegnato alle attività registrate come istanze gestite in SSM.
- Aggiungi le seguenti autorizzazioni al ruolo dell'istanza gestita:
 - `ssm>DeleteActivation`
 - `ssm:DeregisterManagedInstance`
- Aggiungi la policy SSManaged InstanceCore gestita di [Amazon](#) al ruolo dell'istanza gestita.
- Aggiungi un contenitore di agenti SSM alla definizione del task di Amazon ECS. Lo script di comando registra le attività di Amazon ECS come istanze gestite.

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

```

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

```

Per una versione più leggibile dello script, consulta [the section called “Versione di riferimento dello script”](#)

- Abilita Amazon ECS Fault Injection APIs impostando il `enableFaultInjection` campo nella definizione del task di Amazon ECS:

```

"enableFaultInjection": true,

```

- Quando si utilizzano le `aws:ecs:task-network-packet-loss` azioni `aws:ecs:task-network-blackhole-port` o `aws:ecs:task-network-latency`, o sulle attività di Fargate, l'azione deve avere il `useEcsFaultInjectionEndpoints` parametro impostato su `true`
- Quando si utilizzano le `aws:ecs:task-network-packet-loss` azioni `aws:ecs:task-kill-process` o `aws:ecs:task-network-blackhole-port`, `aws:ecs:task-network-latency`, o, la definizione del task di Amazon ECS deve essere `pidMode` impostata su `task`
- Quando si utilizzano le `aws:ecs:task-network-packet-loss` azioni `aws:ecs:task-network-blackhole-port` o `aws:ecs:task-network-latency`, o su attività con tipo di avvio EC2, l'[opzione di rete nella definizione dell'attività](#) deve essere impostata su `o. awsvpc host`

Versione di riferimento dello script

Di seguito è riportata una versione più leggibile dello script nella sezione Requisiti, come riferimento.

```

#!/usr/bin/env bash

# This is the activation script used to register ECS tasks as Managed Instances in SSM
# The script retrieves information from the ECS task metadata endpoint to add three
# tags to the Managed Instance
# - ECS_TASK_AVAILABILITY_ZONE: To allow customers to target Managed Instances / Tasks
# in a specific Availability Zone
# - ECS_TASK_ARN: To allow customers to target Managed Instances / Tasks by using the
# Task ARN
# - FAULT_INJECTION_SIDE CAR: To make it clear that the tasks were registered as
# managed instance for fault injection purposes. Value is always 'true'.
# The script will leave the SSM Agent running in the background

```

```

# When the container running this script receives a SIGTERM or SIGINT signal, it will
do the following cleanup:
# - Delete SSM activation
# - Deregister SSM managed instance

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

dnf upgrade -y
dnf install jq procs awscli -y

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

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

    kill -SIGTERM $SSM_AGENT_PID
}
trap term_handler SIGTERM SIGINT

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

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

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

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

```

```

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_SIDE CAR,Value=true \
    --region $ECS_TASK_REGION)

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

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

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

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

```

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

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

Esempio di modello di esperimento

Di seguito è riportato un esempio di modello di esperimento per [l'azione chiamata "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",
    }
  ]
}
```

```

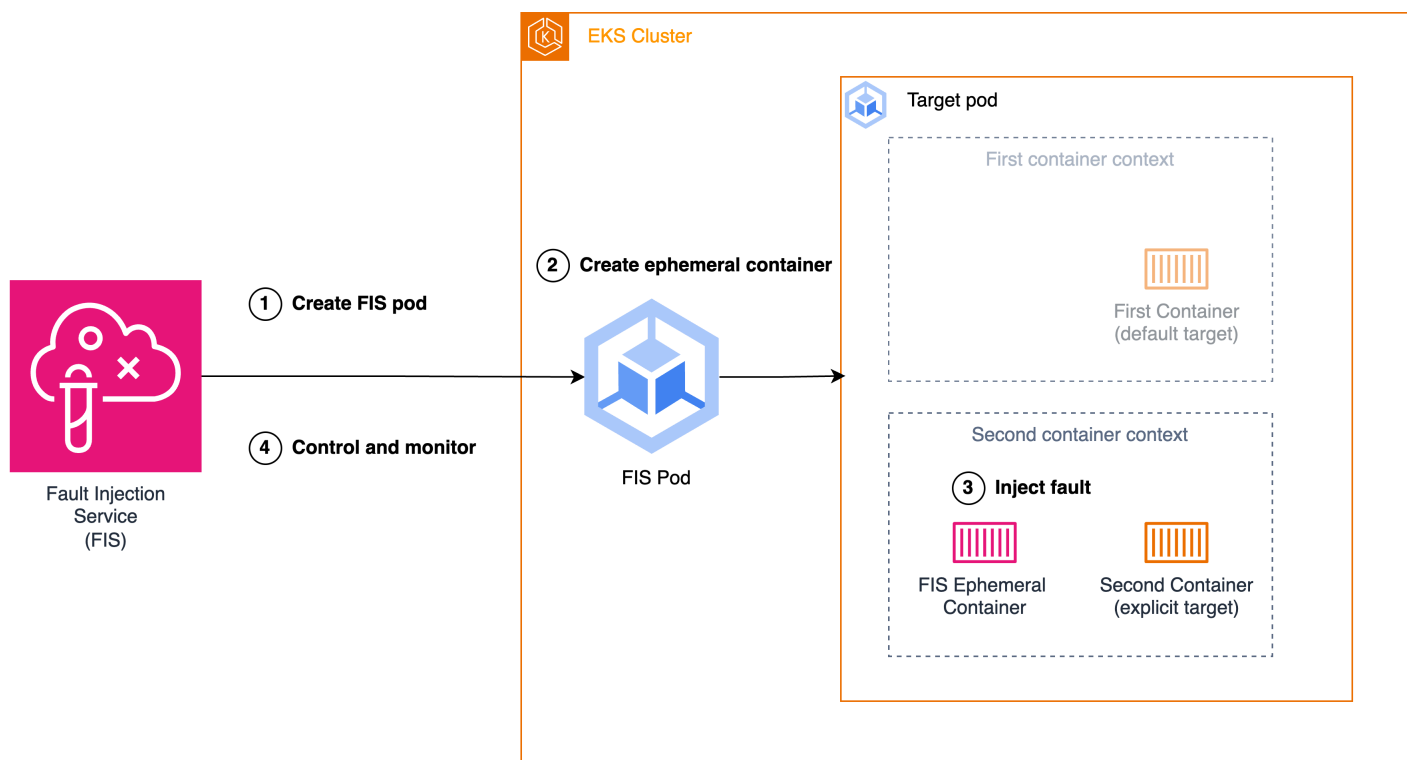
],
"roleArn": "arn:aws:iam::111122223333:role/fis-experiment-role",
"tags": {}
}

```

Usa le azioni AWS FIS aws:eks:pod

Puoi utilizzare le azioni `aws:eks:pod` per inserire errori nei Kubernetes Pod in esecuzione nei tuoi cluster EKS.

Quando viene avviata un'azione, FIS recupera l'immagine del contenitore FIS Pod. Questa immagine viene quindi utilizzata per creare un Pod nel cluster EKS di destinazione. Il Pod appena creato è responsabile dell'iniezione, del controllo e del monitoraggio del guasto. Per tutte le azioni FIS EKS, ad eccezione di `aws:eks:pod-delete`, l'iniezione dei guasti si ottiene mediante l'uso di contenitori effimeri, una funzionalità di Kubernetes che consente la creazione di contenitori temporanei all'interno di un Pod esistente. Il contenitore temporaneo viene avviato nello stesso namespace del contenitore di destinazione ed esegue le attività di fault injection desiderate. Se non viene specificato alcun contenitore di destinazione, il primo contenitore nella specifica Pod viene selezionato come destinazione.



1. FIS crea il pod FIS nel cluster di destinazione specificato nel modello di esperimento.

2. Il pod FIS crea un contenitore temporaneo nel Target Pod nello stesso namespace del contenitore di destinazione.
3. Il contenitore temporaneo inserisce errori nel namespace del contenitore di destinazione.
4. Il FIS Pod controlla e monitora l'iniezione dei guasti del contenitore temporaneo e il FIS controlla e monitora il FIS Pod.

Al termine dell'esperimento o in caso di errore, il contenitore temporaneo e il FIS Pod vengono rimossi.

Azioni

- [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"](#)

Limitazioni

- Le seguenti azioni non funzionano con: AWS Fargate
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency
 - aws:eks:pod-network-packet-loss
- Le seguenti azioni non supportano la [modalità bridge di rete](#):
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency
 - aws:eks:pod-network-packet-loss
- Le seguenti azioni richiedono i permessi di root all'interno del contenitore temporaneo.
 - aws:eks:pod-network-blackhole-port
 - aws:eks:pod-network-latency

- `aws:eks:pod-network-packet-loss`

Il contenitore temporaneo erediterà le sue autorizzazioni dal contesto di sicurezza del Pod di destinazione. Se devi eseguire i contenitori nel Pod come utente non root, puoi impostare contesti di sicurezza separati per i contenitori nel Pod di destinazione.

- Non puoi identificare obiettivi di tipo `aws:eks:pod` nel tuo modello di esperimento utilizzando tag di risorse o risorse. ARNs È necessario identificare gli obiettivi utilizzando i parametri di risorsa richiesti.
- Le azioni `aws:eks: pod-network-latency` e `aws:eks: non pod-network-packet-loss` devono essere eseguite in parallelo e indirizzate allo stesso Pod. A seconda del valore del `maxErrors` parametro specificato, l'azione può terminare con lo stato completato o fallito:
 - Se `maxErrorsPercent` è 0 (impostazione predefinita), l'azione terminerà in stato di errore.
 - In caso contrario, l'errore inciderà sul `maxErrorsPercent` budget. Se il numero di iniezioni fallite non raggiunge quello fornito `maxErrors`, l'azione finirà per essere completata.
 - È possibile identificare questi errori dai registri del contenitore effimero iniettato nel Pod di destinazione. `Exit Code: 16` Fallirà con.
- L'azione `aws:eks: non pod-network-blackhole-port` deve essere eseguita in parallelo con altre azioni che hanno come target lo stesso Pod e lo utilizzano. `trafficType` Sono supportate azioni parallele che utilizzano diversi tipi di traffico.
- Il FIS può monitorare lo stato dell'iniezione dei guasti solo quando il Pod `securityContext` di destinazione è impostato su. `readOnlyRootFilesystem: false` Senza questa configurazione, tutte le azioni EKS Pod falliranno.

Requisiti

- Installalo AWS CLI sul tuo computer. Questo è necessario solo se lo utilizzerai AWS CLI per creare ruoli IAM. Per ulteriori informazioni, consulta [Installazione o aggiornamento di AWS CLI](#).
- Installare kubectl sul computer. Ciò è necessario solo per interagire con il cluster EKS per configurare o monitorare l'applicazione di destinazione. Per ulteriori informazioni, consulta <https://kubernetes.io/docs/tasks/tools/>.
- La versione minima supportata di EKS è la 1.23.

Crea un ruolo sperimentale

Per eseguire un esperimento, devi configurare un ruolo IAM per l'esperimento. Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#). Le autorizzazioni richieste per questo ruolo dipendono dall'azione che stai utilizzando. Fai riferimento alle [azioni AWS FIS mirate `aws:eks:pod`](#) per trovare le autorizzazioni necessarie per la tua azione.

Configurazione dell'account di servizio Kubernetes

Configura un account di servizio Kubernetes per eseguire esperimenti con obiettivi nello spazio dei nomi Kubernetes specificato. Nell'esempio seguente, l'account del servizio è e lo spazio dei nomi è `myserviceaccount default`. Tieni presente che `default` è uno dei namespace Kubernetes standard.

Per configurare il tuo account di servizio Kubernetes

1. Crea un file denominato `rbac.yaml` e aggiungi quanto segue.

```
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. Eseguire il seguente comando seguente.

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

Concedi agli utenti e ai ruoli IAM l'accesso a Kubernetes APIs

Segui i passaggi spiegati in [Associare le identità IAM alle autorizzazioni Kubernetes](#) nella documentazione. EKS

Opzione 1: creare voci di accesso

Consigliamo l'uso di Access Entries. Puoi utilizzare il seguente comando per creare un Access Entry che associ il ruolo IAM all'utente Kubernetes. *fis-experiment* Per ulteriori informazioni, consulta [Concedere agli utenti IAM l'accesso a Kubernetes con le voci di accesso](#) EKS.

```

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

```

```
--cluster-name my-cluster
```

⚠ Important

Per sfruttare le voci di accesso, la modalità di autenticazione del cluster EKS deve essere configurata in modalità `or. API_AND_CONFIG_MAP API`

Opzione 2: aggiungi voci a aws-auth ConfigMap

Puoi anche usare il seguente comando per creare una mappatura dell'identità. Per ulteriori informazioni, consulta [Manage IAM users and roles](#) nella documentazione di eksctl.

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

⚠ Important

L'utilizzo del toolkit eksctl per configurare le mappature delle identità comporterà la creazione di voci all'interno di `aws-auth ConfigMap`. È importante notare che queste voci generate non supportano l'inclusione di un componente del percorso. Di conseguenza, l'ARN fornito come input non deve contenere un segmento di percorso (ad es.).

```
arn:aws:iam::123456789012:role/service-role/fis-experiment-role
```

Immagini del contenitore Pod

Le immagini dei contenitori Pod fornite da AWS FIS sono ospitate in Amazon ECR. Quando fai riferimento a un'immagine da Amazon ECR, devi utilizzare l'URI completo dell'immagine.

L'immagine del contenitore Pod è disponibile anche nella [galleria pubblica AWS ECR](#).

Regione AWS	URI immagine
Stati Uniti orientali (Ohio)	<code>051821878176.dkr.ecr.us-east-2.amazonaws.com/aws-fis-pod:0.1</code>

Regione AWS	URI immagine
Stati Uniti orientali (Virginia settentrionale)	731367659002.dkr.ecr.us-east-1.amazonaws.com/aws-fis-pod:0.1
Stati Uniti occidentali (California settentrionale)	080694859247.dkr.ecr.us-west-1.amazonaws.com/aws-fis-pod:0.1
Stati Uniti occidentali (Oregon)	864386544765.dkr.ecr.us-west-2.amazonaws.com/aws-fis-pod:0.1
Africa (Città del Capo)	056821267933.dkr.ecr.af-south-1.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Hong Kong)	246405402639.dkr.ecr.ap-east-1.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Mumbai)	524781661239.dkr.ecr.ap-south-1.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Osaka)	148336246925.dkr.ecr.ap-northeast-3.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Seoul)	526524659354.dkr.ecr.ap-northeast-2.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Singapore)	316401638346.dkr.ecr.ap-southeast-1.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Sydney)	488104106298.dkr.ecr.ap-southeast-2.amazonaws.com/aws-fis-pod:0.1
Asia Pacifico (Tokyo)	635234321696.dkr.ecr.ap-northeast-1.amazonaws.com/aws-fis-pod:0.1
Canada (Centrale)	490658072207.dkr.ecr.ca-central-1.amazonaws.com/aws-fis-pod:0.1

Regione AWS	URI immagine
Europa (Francoforte)	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 (Londra)	327424803546.dkr.ecr.eu-west-2.amazonaws.com/ aws-fis-pod:0.1
Europa (Milano)	478809367036.dkr.ecr.eu-south-1.amazonaws.com/ aws-fis-pod:0.1
Europa (Parigi)	154605889247.dkr.ecr.eu-west-3.amazonaws.com/ aws-fis-pod:0.1
Europa (Spagna)	395402409451.dkr.ecr.eu-south-2.amazonaws.com/ aws-fis-pod:0.1
Europa (Stoccolma)	263175118295.dkr.ecr.eu-north-1.amazonaws.com/ aws-fis-pod:0.1
Europa (Zurigo)	604225987275.dkr.ecr.eu-central-2.amazonaws.com/ aws-fis-pod:0.1
Medio Oriente (Bahrein)	065825543785.dkr.ecr.me-south-1.amazonaws.com/ aws-fis-pod:0.1
Medio Oriente (Emirati Arabi Uniti)	438374459301.dkr.ecr.me-central-1.amazonaws.com/ aws-fis-pod:0.1
Sud America (San Paolo)	767113787785.dkr.ecr.sa-east-1.amazonaws.com/ aws-fis-pod:0.1
AWS GovCloud (Stati Uniti orientali)	246533647532.dkr.ecr.us-gov-east-1.amazonaws.com/ aws-fis-pod:0.1
AWS GovCloud (Stati Uniti occidentali)	246529956514.dkr.ecr.us-gov-west-1.amazonaws.com/ aws-fis-pod:0.1

Esempio di modello di esperimento

Di seguito è riportato un esempio di modello di esperimento per l'[the section called “aws:eks:pod-network-latency”](#) azione.

```
{
  "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": {
```

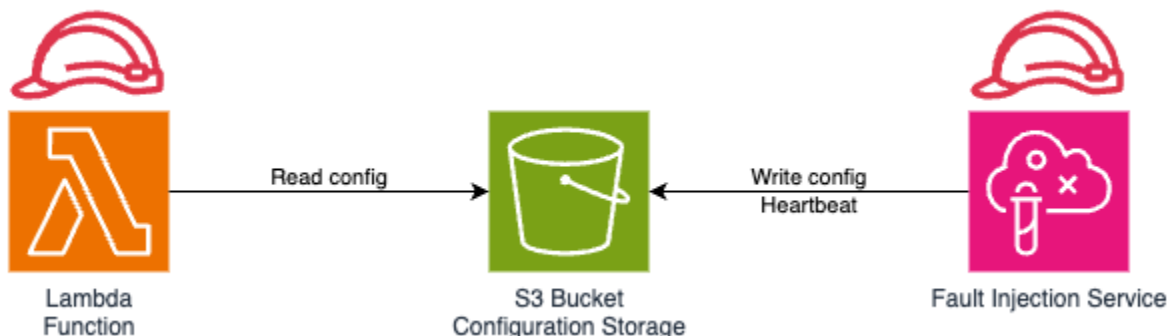
```
    "Name": "EksPodNetworkLatency"  
  }  
}
```

Usa le azioni AWS FIS `aws:lambda:function`

Puoi usare le azioni `aws:lambda:function` per inserire errori nelle chiamate delle tue funzioni. AWS Lambda

Queste azioni utilizzano un'estensione gestita per iniettare errori. AWS FIS Per utilizzare le azioni `aws:lambda:function`, dovrai collegare l'estensione come livello alle tue funzioni Lambda e configurare un bucket Amazon S3 per comunicare tra e l'estensione. AWS FIS

Quando esegui un AWS FIS esperimento che ha come target `aws:lambda:function`, legge la configurazione di AWS FIS Amazon S3 dalla tua funzione Lambda e scrive le informazioni di fault injection nella posizione Amazon S3 specificata, come mostrato nel diagramma seguente.



Azioni

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

Limitazioni

- L'estensione AWS FIS Lambda non può essere utilizzata con funzioni che utilizzano lo streaming di risposte. Anche quando non vengono applicati errori, l'estensione AWS FIS Lambda sopprimerà le configurazioni di streaming. Per ulteriori informazioni, consulta [lo streaming di risposte per le funzioni Lambda nella guida](#) per l'AWS Lambda utente.

Prerequisiti

Prima di utilizzare le azioni AWS FIS Lambda, assicurati di aver completato queste attività una tantum:

- Crea un bucket Amazon S3 nella regione da cui intendi iniziare un esperimento - Puoi usare un singolo bucket Amazon S3 per più esperimenti e condividere il bucket tra più account. AWS Tuttavia, è necessario disporre di un bucket separato per ciascuno. Regione AWS
- Crea una policy IAM per concedere l'accesso in lettura per l'estensione Lambda al bucket Amazon S3 - Nel modello seguente, `my-config-distribution-bucket` sostituisilo con il nome del bucket Amazon S3 che hai creato sopra `FisConfigs` e con il nome di una cartella nel bucket Amazon S3 che desideri utilizzare.

JSON

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

- Crea una policy IAM per concedere l'accesso in scrittura per l' AWS FIS esperimento al bucket Amazon S3 - Nel modello seguente, sostituisilo `my-config-distribution-bucket` con il

nome del bucket Amazon S3 che hai creato sopra e FisConfigs con il nome di una cartella nel bucket Amazon S3 che desideri utilizzare.

JSON

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

Configurazione di funzioni Lambda

Segui i passaggi seguenti per ogni funzione Lambda su cui desideri influire:

1. Allega la policy di accesso alla lettura di Amazon S3 creata sopra alla funzione Lambda.

2. Associa l' AWS FIS estensione come livello alla funzione. Per ulteriori informazioni sul layer ARNs, vedere [Versioni disponibili dell' AWS FIS estensione per Lambda](#).
3. Imposta la `AWS_FIS_CONFIGURATION_LOCATION` variabile sull'ARN della cartella di configurazione di Amazon S3, ad esempio. `arn:aws:s3:::my-config-distribution-bucket/FisConfigs/`
4. Impostare la variabile `AWS_LAMBDA_EXEC_WRAPPER` su `/opt/aws-fis/bootstrap`.

Configura un esperimento AWS FIS

Prima di eseguire l'esperimento, assicurati di aver collegato la policy di accesso alla scrittura di Amazon S3 che hai creato nei prerequisiti ai ruoli dell'esperimento che utilizzeranno le azioni AWS FIS Lambda. Per ulteriori informazioni su come configurare un AWS FIS esperimento, consulta.

[Gestione dei AWS modelli di esperimento FIS](#)

Registrazione dei log

L'estensione AWS FIS Lambda scrive i log nella console e i log. CloudWatch La registrazione può essere configurata utilizzando la variabile. `AWS_FIS_LOG_LEVEL` I valori supportati sono INFO, WARN e ERROR. I log verranno scritti nel formato di registro configurato per la funzione Lambda.

Di seguito è riportato un esempio di registro in formato testo:

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

Di seguito è riportato un esempio di registro in formato JSON:

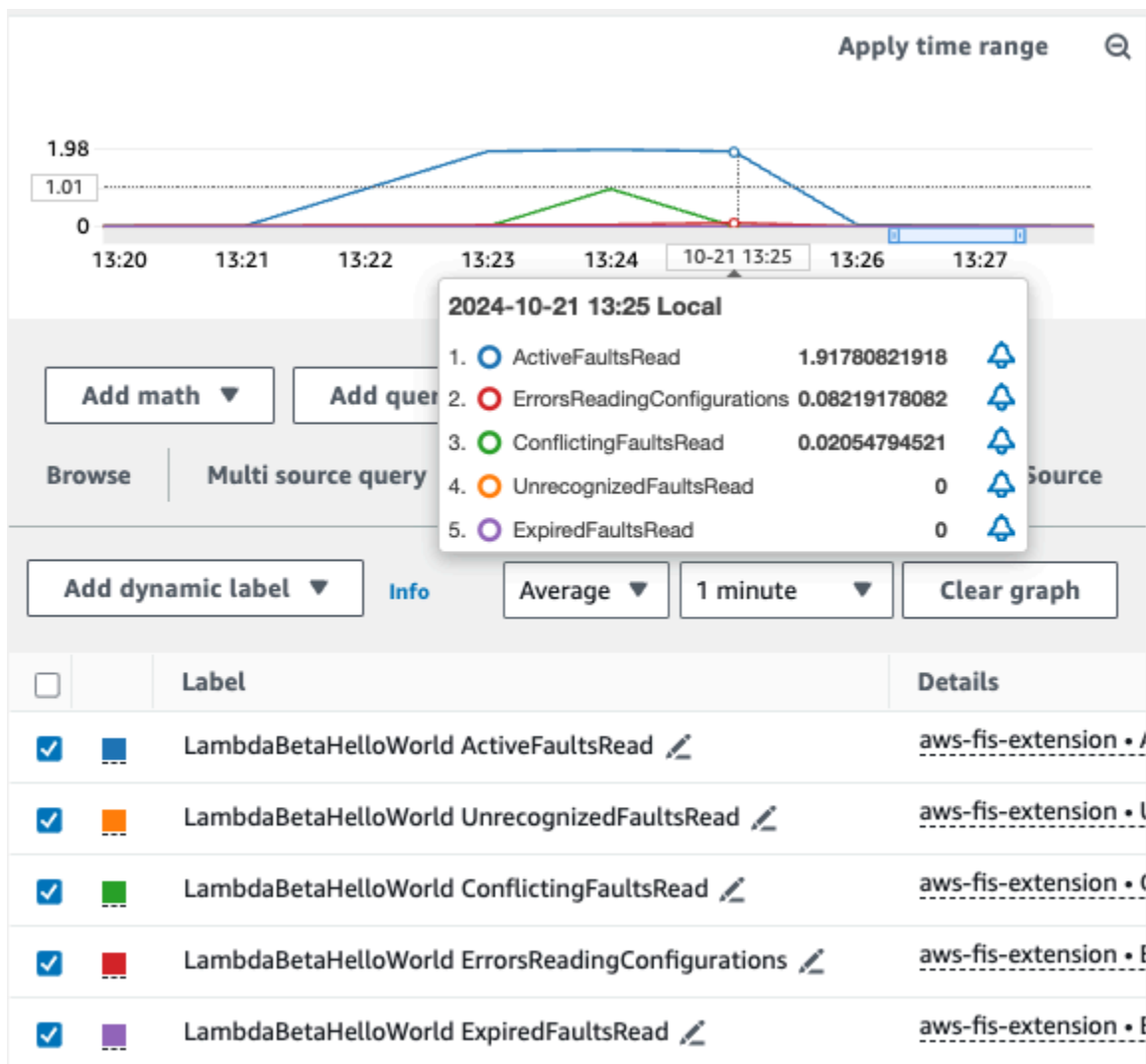
```
{
  "timestamp": "2024-10-08T17:15:36.953905Z",
  "level": "INFO",
  "fields": {
    "message": "AWS FIS EXTENSION - adding 5000 milliseconds of latency to function invocation",
    "requestId": "0608bf70-908f-4a17-bbfe-3782cd783d8b"
  }
}
```

I log emessi possono essere utilizzati con i filtri CloudWatch metrici di Amazon per generare parametri personalizzati. Per ulteriori informazioni sui filtri metrici, consulta [Creazione di metriche da eventi di log utilizzando filtri](#) nella guida per l'utente di Amazon CloudWatch Logs.

Utilizzo del formato CloudWatch Embedded Metric Format (EMF)

È possibile configurare l'estensione AWS FIS Lambda per l'emissione di log EMF impostando la variabile su `AWS_FIS_EXTENSION_METRICS all`. Per impostazione predefinita, l'estensione non emette registri EMF e l'impostazione predefinita è `AWS_FIS_EXTENSION_METRICS none`. I registri EMF vengono pubblicati nella console `aws-fis-extension` namespace CloudWatch.

All'interno del `aws-fis-extension` namespace, è possibile selezionare determinate metriche da visualizzare in un grafico. L'esempio seguente mostra alcune delle metriche disponibili nel namespace `aws-fis-extension`.



Argomenti avanzati

Questa sezione fornisce informazioni aggiuntive su come AWS FIS funziona l'estensione Lambda e su casi d'uso speciali.

Argomenti

- [Comprendere i sondaggi](#)
- [Comprendere la concorrenza](#)
- [Comprensione della percentuale di invocazione](#)
- [Considerazioni speciali per SnapStart](#)
- [Considerazioni speciali per le funzioni veloci e poco frequenti](#)
- [Configurazione di più estensioni utilizzando il proxy API Lambda Runtime](#)
- [Utilizzo AWS FIS con i runtime dei container](#)
- [AWS FIS Variabili di ambiente Lambda](#)

Comprendere i sondaggi

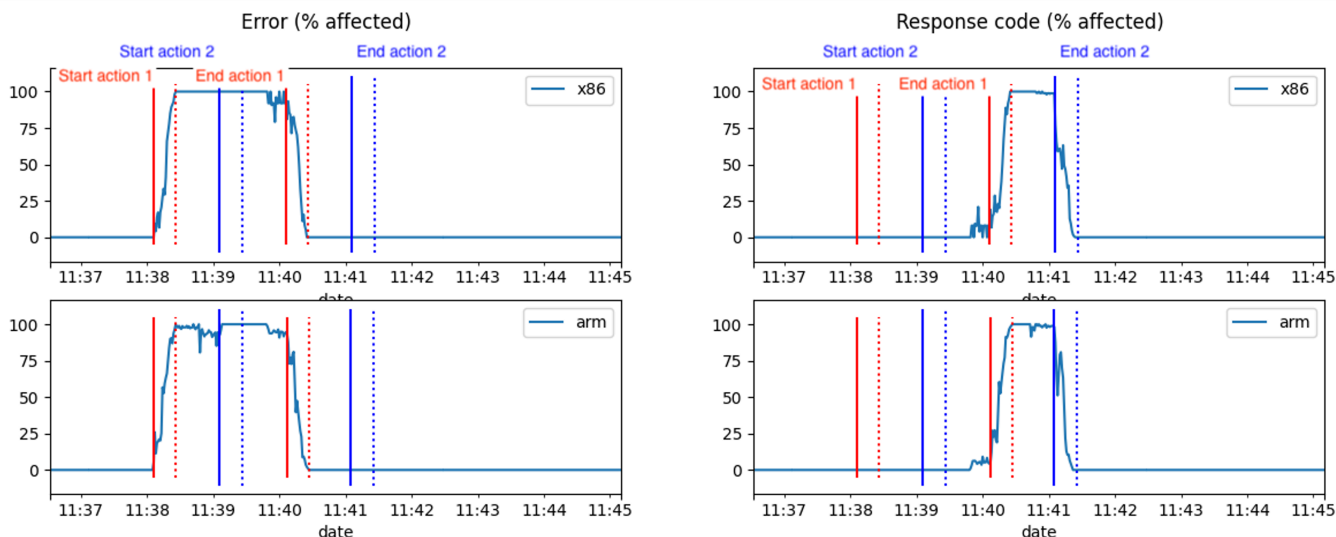
È possibile che si verifichi un periodo di accelerazione fino a 60 minuti prima che i guasti inizino a influire su tutte le chiamate. Questo perché l'estensione Lambda richiede raramente informazioni di configurazione in attesa dell'inizio di un esperimento. È possibile regolare l'intervallo di polling impostando la variabile di `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` ambiente (impostazione predefinita 60s). Un valore più basso verrà sottoposto a sondaggi più frequenti, ma comporterà un maggiore impatto sulle prestazioni e sui costi. È inoltre possibile che si verifichi un periodo di rallentamento fino a 20 secondi dopo l'iniezione del guasto. Questo perché l'estensione esegue i sondaggi più frequentemente durante l'esecuzione degli esperimenti.

Comprendere la concorrenza

Puoi indirizzare le stesse funzioni Lambda con più azioni contemporaneamente. Se le azioni sono tutte diverse l'una dall'altra, verranno applicate tutte le azioni. Ad esempio, è possibile aggiungere un ritardo iniziale prima di restituire un errore. Se due azioni identiche o in conflitto vengono applicate alla stessa funzione, verrà applicata solo l'azione con la data di inizio più vicina.

La figura seguente mostra due azioni in conflitto, `aws:lambda:invocation-error` e `aws:lambda:`, che si sovrappongono. `invocation-http-integration-response` Inizialmente, `aws:lambda:invocation-error`

aumenta alle 11:38 e dura 2 minuti. Quindi, `aws:lambda: invocation-http-integration-response` tenta di iniziare alle 11:39, ma non entra in vigore fino alle 11:40 dopo la conclusione della prima azione. Per mantenere la tempistica dell'esperimento, `aws:lambda: termina` comunque all'ora originariamente prevista di 11:41. `invocation-http-integration-response`



Comprensione della percentuale di invocazione

Le azioni AWS Fault Injection Service Lambda utilizzano un target `aws:lambda:function` che consente di selezionare una o più funzioni. AWS Lambda ARNs Usando queste ARNs, le azioni AWS Fault Injection Service Lambda possono iniettare errori in ogni chiamata della funzione Lambda selezionata. Per consentire di inserire errori solo in una frazione delle invocazioni, ogni azione consente di specificare un parametro con valori compresi tra 0 e 100. `invocationPercentage` Utilizzando il `invocationPercentage` parametro, è possibile garantire che le azioni siano simultanee anche per percentuali di invocazione inferiori al 100%.

Considerazioni speciali per SnapStart

AWS Lambda le funzioni SnapStart abilitate avranno una maggiore probabilità di attendere l'intera durata `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS` prima di rilevare la configurazione del primo errore, anche se un esperimento è già in corso. Questo perché Lambda SnapStart utilizza una singola istantanea come stato iniziale per più ambienti di esecuzione e lo storage temporaneo persiste. Per l'estensione AWS Fault Injection Service Lambda, manterrà la frequenza di polling e salterà il controllo di configurazione iniziale all'inizializzazione dell'ambiente di esecuzione. Per ulteriori informazioni su Lambda SnapStart, consulta [Migliorare le prestazioni di avvio con Lambda SnapStart](#) nella guida per l'utente AWS Lambda

Considerazioni speciali per le funzioni veloci e poco frequenti

Se la funzione Lambda viene eseguita per meno della durata media del sondaggio di 70 millisecondi, il thread di polling potrebbe aver bisogno di più invocazioni per ottenere configurazioni di errore.

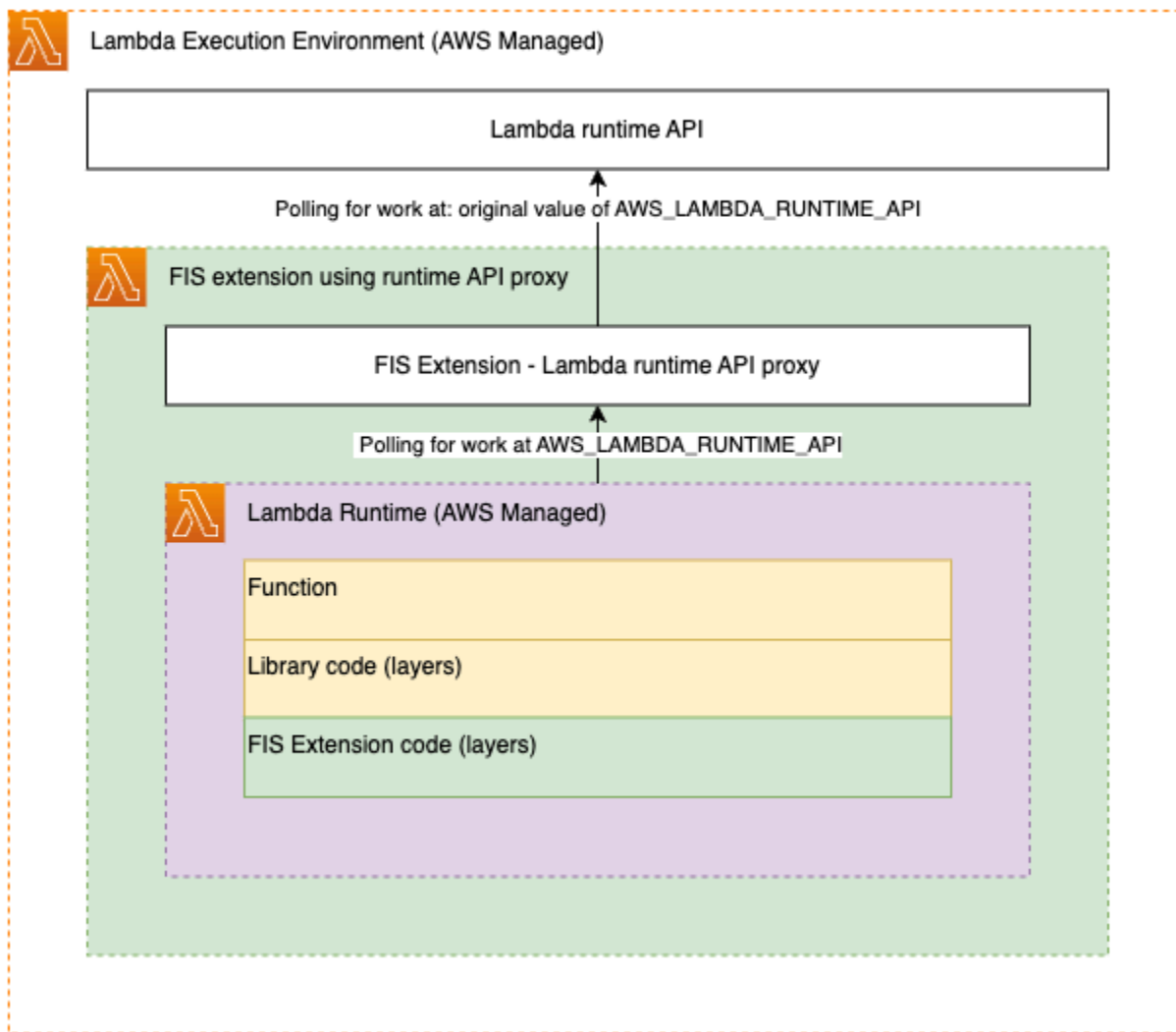
Se la funzione viene eseguita raramente, ad esempio una volta ogni 15 minuti, il sondaggio non verrà mai completato. Per assicurarti che il thread di polling possa terminare, imposta il parametro.

`AWS_FIS_POLL_MAX_WAIT_MILLISECONDS` L'estensione aspetterà il completamento di un sondaggio in volo fino alla durata impostata prima di avviare la funzione. Tieni presente che ciò aumenterà la durata della funzione fatturata e comporterà un ulteriore ritardo su alcune chiamate.

Configurazione di più estensioni utilizzando il proxy API Lambda Runtime

L'estensione Lambda utilizza il proxy dell'API AWS Lambda Runtime per intercettare le chiamate di funzione prima che raggiungano il runtime. Lo fa esponendo un proxy per l'API AWS Lambda Runtime al runtime e pubblicizzando la sua posizione nella variabile `AWS_LAMBDA_RUNTIME_API`

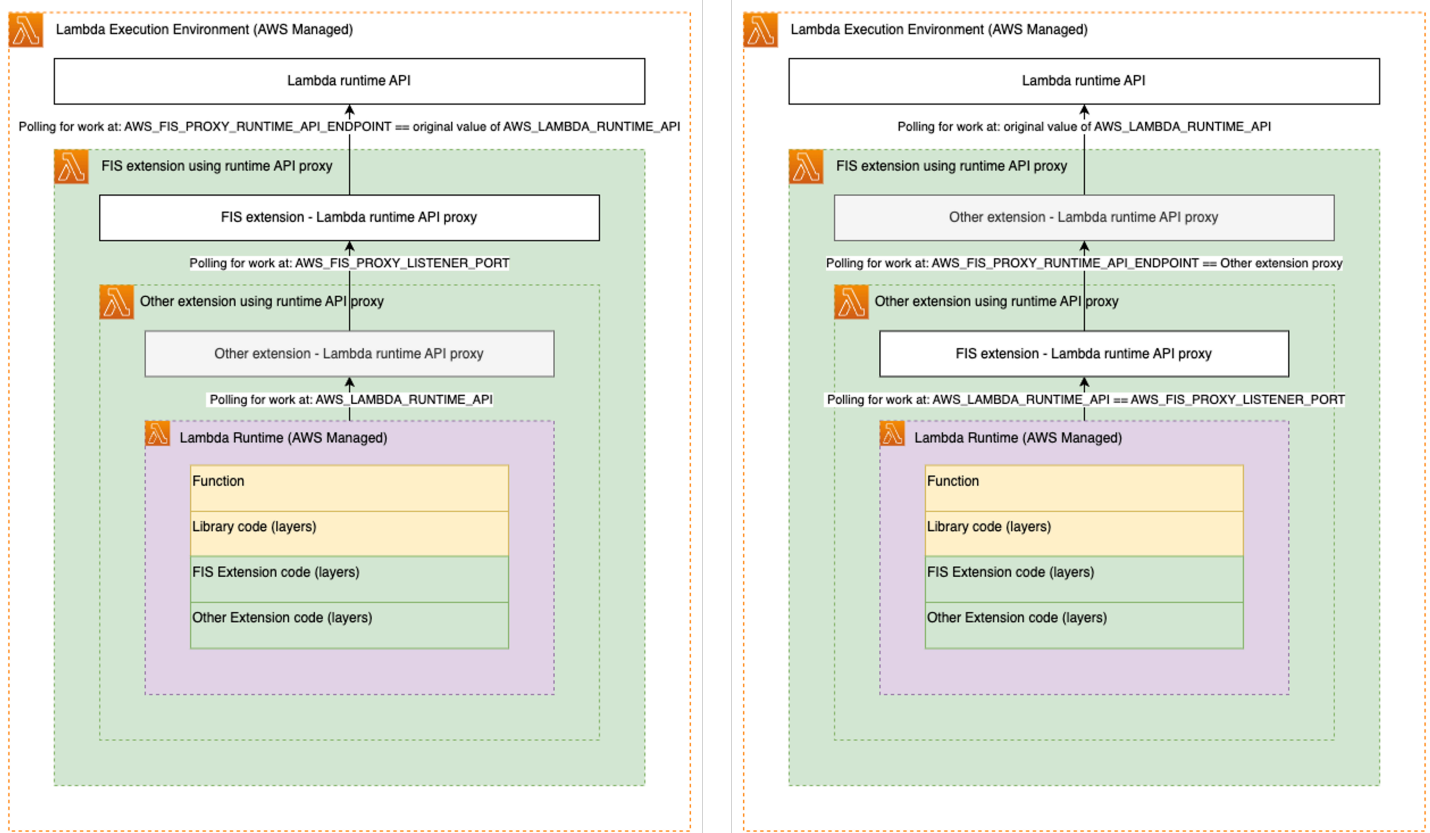
Il diagramma seguente mostra la configurazione per una singola estensione utilizzando il proxy API Lambda Runtime:



Per utilizzare l'estensione AWS FIS Lambda con un'altra estensione utilizzando il pattern proxy dell'API AWS Lambda Runtime, è necessario concatenare i proxy utilizzando uno script di bootstrap personalizzato. L'estensione AWS FIS Lambda accetta le seguenti variabili di ambiente:

- **AWS_FIS_PROXY_RUNTIME_API_ENDPOINT**- Prende una stringa nel formato `127.0.0.1:9876` che rappresenta l'IP locale e la porta del listener per l'API AWS Lambda Runtime. Potrebbe essere il valore originale `AWS_LAMBDA_RUNTIME_API` o la posizione di un altro proxy.
- **AWS_FIS_PROXY_LISTENER_PORT**- Per impostazione predefinita, richiede un numero di porta su cui l'AWS FIS estensione deve avviare il proprio proxy `9100`.

Con queste impostazioni puoi concatenare l'AWS FIS estensione con un'altra estensione utilizzando il proxy API Lambda Runtime in due ordini diversi.



Per ulteriori informazioni sul proxy dell'API AWS Lambda Runtime, consulta [Migliorare la sicurezza e la governance del AWS Lambda runtime con l'estensione proxy Runtime API](#) e [Uso dell'API di runtime Lambda per i runtime personalizzati](#) nella guida per l'AWS Lambda utente.

Utilizzo AWS FIS con i runtime dei container

Per AWS Lambda le funzioni che utilizzano immagini del contenitore che accettano la variabile di ambiente `AWS_LAMBDA_RUNTIME_API`, puoi inserire l'estensione AWS FIS Lambda nell'immagine del contenitore seguendo i passaggi seguenti:

1. Determina l'ARN del livello da cui estrarre l'estensione. Per ulteriori informazioni su come trovare l'ARN, vedere. [Configurazione di funzioni Lambda](#)
2. Utilizza AWS Command Line Interface (CLI) per richiedere dettagli sull'estensione. `aws lambda get-layer-version-by-arn --arn fis-extension-arn` La risposta conterrà un `Location` campo contenente un URL prefirmato da cui è possibile scaricare l'estensione FIS come file ZIP.
3. Decomprimi il contenuto dell'estensione nel tuo filesystem `/opt` Docker. Di seguito è riportato un esempio di Dockerfile basato sul runtime NodeJS Lambda:

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

Per ulteriori informazioni sulle immagini dei contenitori, consulta [Creare una funzione Lambda utilizzando un'immagine del contenitore nella guida](#) per l'AWS Lambda utente.

AWS FIS Variabili di ambiente Lambda

Di seguito è riportato un elenco di variabili di ambiente per l'estensione AWS FIS Lambda

- **AWS_FIS_CONFIGURATION_LOCATION**- Obbligatorio. Posizione in cui AWS FIS verranno scritte le configurazioni di errore attive e l'estensione leggerà le configurazioni di errore. Le posizioni devono essere in formato Amazon S3 ARN, inclusi un bucket e un percorso. Ad esempio, `arn:aws:s3:::my-fis-config-bucket/FisConfigs/`.
- **AWS_LAMBDA_EXEC_WRAPPER**- Richiesto. Posizione dello [script AWS Lambda wrapper utilizzato per](#) configurare l'estensione Lambda AWS FIS . Dovrebbe essere impostato sullo `/opt/aws-fis/bootstrap` script incluso nell'estensione.
- **AWS_FIS_LOG_LEVEL**- Facoltativo. Livello di registro per i messaggi emessi dall'estensione AWS FIS Lambda. I valori supportati sono INFO, WARN e ERROR. Se non è impostata, AWS FIS l'estensione verrà impostata per impostazione predefinita su. INFO
- **AWS_FIS_EXTENSION_METRICS**- Facoltativo. I valori possibili sono `all` e `none`. Se impostato `all` sull'estensione, emetterà le metriche EMF sotto il `aws-fis-extension` namespace

- `AWS_FIS_SLOW_POLL_INTERVAL_SECONDS`- Facoltativo. Se impostato, annullerà l'intervallo di polling (in secondi) mentre l'estensione non inietta errori e attende che venga aggiunta una configurazione di errore alla posizione di configurazione. L'impostazione predefinita è 60.
- `AWS_FIS_PROXY_RUNTIME_API_ENDPOINT`- Opzionale. If set sovrascriverà il valore di `AWS_LAMBDA_RUNTIME_API` per definire dove l' AWS FIS estensione interagisce con l'API di AWS Lambda runtime per controllare l'invocazione della funzione. Si aspetta IP:PORT, ad esempio. `127.0.0.1:9000` Per ulteriori informazioni `AWS_LAMBDA_RUNTIME_API`, consulta [Using the Lambda runtime API for custom runtime](#) nella guida per l'AWS Lambda utente.
- `AWS_FIS_PROXY_LISTENER_PORT`- Facoltativo. Definisce la porta su cui l'estensione AWS FIS Lambda espone un proxy API AWS Lambda di runtime che può essere utilizzato da un'altra estensione o dal runtime. L'impostazione predefinita è 9100.
- `AWS_FIS_POLL_MAX_WAIT_MILLISECONDS`- Facoltativo. Se impostata su un valore diverso da zero, questa variabile definisce il numero di millisecondi in cui l'estensione attenderà il completamento di un sondaggio asincrono in volo prima di valutare le configurazioni di errore e avviare l'invocazione del runtime. L'impostazione predefinita è 0.

Versioni disponibili dell' AWS FIS estensione per Lambda

Questa sezione include informazioni sulle versioni dell'estensione AWS FIS Lambda. L'estensione supporta le funzioni Lambda sviluppate per le piattaforme x86-64 e ARM64 (Graviton2). La tua funzione Lambda deve essere configurata per utilizzare lo specifico Amazon Resource Name (ARN) per il Regione AWS luogo in cui è attualmente ospitata. Di seguito è possibile visualizzare Regione AWS i dettagli dell'ARN.

Argomenti

- [AWS FIS Note sulla versione dell'estensione Lambda](#)
- [Guida di accesso per Lambda Extension ARNs](#)
- [Individuazione del numero di versione dell'estensione Lambda](#)

AWS FIS Note sulla versione dell'estensione Lambda

La tabella seguente descrive le modifiche apportate alle versioni recenti dell'estensione AWS FIS Lambda.

Versione	Data di lancio	Note
1.0.0	2024-10-29	Rilascio iniziale

Guida di accesso per Lambda Extension ARNs

È necessario disporre di almeno un parametro nel proprio Account AWS e Regione AWS prima di poter cercare parametri pubblici utilizzando la console. Per scoprire i parametri pubblici, vedi [Discovering public parameters in Parameter Store](#).

Accesso alla console:

1. Apri la AWS Systems Manager console all'indirizzo <https://console.aws.amazon.com/systems-manager/>.
2. Nel riquadro di navigazione, selezionare Parameter Store (Archivio parametri).
3. Seleziona la scheda Public parameters (Parametri pubblici).
4. Seleziona il menu a discesa Select a service (Seleziona un servizio). Dalle opzioni del menu a discesa, scegli `fis`.
5. (Facoltativo) Filtra i parametri selezionati inserendo ulteriori informazioni nella barra di ricerca. Per le architetture `arm64`, filtra i parametri inserendo «`arm64`». Per le architetture `x86_64`, filtra i parametri inserendo «`x86_64`».
6. Scegliere il parametro pubblico che si desidera utilizzare.
7. Dai dettagli del parametro, individua il valore ARN. Copia l'ARN da utilizzare per configurare le estensioni di livello sulle funzioni Lambda di destinazione.

AWS CLI Accesso:

Nomi dei parametri SSM

I seguenti nomi di parametri SSM sono disponibili per diverse architetture:

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

AWS CLI Formato del comando

Per recuperare l'estensione ARNs, utilizzate il seguente formato di AWS CLI comando in cui `ParameterName` è il nome dell'architettura e la regione è la destinazione: Regione AWS

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

Esempio di utilizzo

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

Formato della risposta

Il comando restituisce un oggetto JSON contenente i dettagli dei parametri come i seguenti. L'ARN del livello Lambda è incluso nel campo `Value` dell'oggetto `Parameter`. Copia l'ARN da utilizzare per configurare le estensioni di livello sulle funzioni Lambda di destinazione.

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

Accesso programmatico:

Recupera questi parametri pubblici a livello di codice durante la creazione o la configurazione delle funzioni Lambda utilizzando Infrastructure as Code (IaC). Questo approccio aiuta a mantenere le funzioni Lambda con l'ultima versione di layer ARN senza richiedere aggiornamenti manuali del

codice che sarebbero necessari se l'ARN del AWS FIS livello di estensione fosse codificato. Le seguenti risorse mostrano come recuperare i parametri pubblici utilizzando piattaforme IaC comuni:

- [Ottieni parametri pubblici utilizzando l'SDK AWS](#)
- [Ottieni i parametri pubblici da AWS Systems Manager Parameter Store utilizzando il CDK AWS](#)
- [Ottieni parametri pubblici usando Terraform](#)

Individuazione del numero di versione dell'estensione Lambda

Usa la seguente procedura per individuare il numero di versione dell'estensione AWS FIS Lambda attualmente configurata.

1. Apri la AWS Lambda console all'indirizzo <https://console.aws.amazon.com/lambda/>.
2. Scegli la funzione Lambda in cui desideri aggiungere il AWS-FIS-Extension layer.
3. Nella sezione Livelli, seleziona Modifica.
4. Nella sezione Modifica livelli, scegli Aggiungi un livello.
5. Nella sezione Scegli un livello, scegli Specificare un ARN.
6. Inserisci l'ARN per il livello di AWS FIS estensione che corrisponde alla tua architettura Regione AWS . È possibile trovare l'ARN utilizzando la console o i AWS CLI metodi di accesso programmatico descritti nelle sezioni precedenti.
7. Scegli Verifica per confermare che l'ARN del layer è valido, quindi scegli Aggiungi.
8. Utilizzate la scheda Test per testare la funzione.
9. Al termine del test, visualizza l'output del registro. Individua la versione dell'estensione AWS FIS Lambda nella sezione Dettagli dell'esecuzione.

Gestione dei AWS modelli di esperimento FIS

È possibile creare e gestire modelli di esperimenti utilizzando la console AWS FIS o la riga di comando. Un modello di esperimento contiene una o più azioni da eseguire su obiettivi specifici durante un esperimento. Contiene anche le condizioni di interruzione che impediscono all'esperimento di superare i limiti. Per ulteriori informazioni sui componenti di un modello di esperimento, vedere [Componenti del modello di esperimento](#). Dopo aver creato un modello di esperimento, puoi utilizzarlo per eseguire un esperimento.

Processi

- [Crea un modello di esperimento](#)
- [Visualizza i modelli di esperimenti](#)
- [Genera un'anteprima del bersaglio da un modello di esperimento](#)
- [Inizia un esperimento da un modello](#)
- [Aggiorna un modello di esperimento](#)
- [Modelli di esperimenti con tag](#)
- [Eliminare un modello di esperimento](#)
- [Esempi di modelli di esperimenti AWS FIS](#)

Crea un modello di esperimento

Prima di iniziare, completa le seguenti attività:

- [Pianifica il tuo esperimento](#).
- Crea un ruolo IAM che conceda al servizio AWS FIS l'autorizzazione a eseguire azioni per tuo conto. Per ulteriori informazioni, consulta [Ruoli IAM per AWS esperimenti FIS](#).
- Assicurati di avere accesso al AWS FIS. Per ulteriori informazioni, consulta [Esempi di policy AWS FIS](#).

Per creare un modello di esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Scegli Crea modello di esperimento.

4. Per la Fase 1, Specificate i dettagli del modello, effettuate le seguenti operazioni:
 - a. In Descrizione e nome, inserisci una descrizione per il modello, ad esempio Amazon S3 Network Disrupt Connectivity.
 - b. (Facoltativo) Per il targeting per account, scegli Account multipli per configurare un modello di esperimento con più account.
 - c. Scegli Avanti e vai al passaggio 2, Specificare azioni e obiettivi.
5. Per Azioni, specificate il set di azioni per il modello. Per ogni azione, scegli Aggiungi azione e completa quanto segue:

- In Nome, inserisci un nome per l'azione.

I caratteri consentiti sono caratteri alfanumerici, trattini (-) e caratteri di sottolineatura (_). Il nome deve iniziare con una lettera. Gli spazi non sono consentiti. Ogni nome di azione deve essere univoco in questo modello.

- (Facoltativo) In Descrizione, inserisci una descrizione dell'azione. La lunghezza massima è 512 caratteri.
- (Facoltativo) In Inizia dopo, selezionate un'altra azione definita in questo modello che deve essere completata prima dell'inizio dell'azione corrente. Altrimenti, l'azione viene eseguita all'inizio dell'esperimento.
- Per Tipo di azione, scegliete l'azione AWS FIS.
- Per Target, scegli un obiettivo definito nella sezione Obiettivi. Se non hai ancora definito un obiettivo per questa azione, AWS FIS crea un nuovo obiettivo per te.
- Per Parametri di azione, specifica i parametri per l'azione. Questa sezione viene visualizzata solo se l'azione AWS FIS ha dei parametri.
- Scegli Save (Salva).

6. Per Target, definisci le risorse target su cui eseguire le azioni. È necessario specificare almeno un ID di risorsa o un tag di risorsa come destinazione. Scegliete Modifica per modificare la destinazione che AWS FIS ha creato per voi nel passaggio precedente oppure scegliete Aggiungi destinazione. Per ogni obiettivo, procedi come segue:

- In Nome, inserisci un nome per l'obiettivo.

I caratteri consentiti sono caratteri alfanumerici, trattini (-) e caratteri di sottolineatura (_). Il nome deve iniziare con una lettera. Gli spazi non sono consentiti. Ogni nome di destinazione deve essere univoco in questo modello.

- Per Tipo di risorsa, scegli un tipo di risorsa supportato per l'azione.
 - Per il metodo Target, esegui una delle seguenti operazioni:
 - Scegli Risorsa IDs, quindi scegli o aggiungi la risorsa IDs.
 - Scegli i tag, i filtri e i parametri delle risorse, quindi aggiungi i tag e i filtri di cui hai bisogno. Per ulteriori informazioni, consulta [the section called "Identifica le risorse target"](#).
 - Per la modalità Selezione, scegli Conteggio per eseguire l'azione sul numero specificato di obiettivi identificati o scegli Percentuale per eseguire l'azione sulla percentuale specificata di obiettivi identificati. Per impostazione predefinita, l'azione viene eseguita su tutti gli obiettivi identificati.
 - Scegli Save (Salva).
7. Per aggiornare un'azione con l'obiettivo che hai creato, trova l'azione in Azioni, scegli Modifica, quindi aggiorna Target. Puoi utilizzare lo stesso obiettivo per più azioni.
 8. (Facoltativo) Per le opzioni sperimentali, selezionate il comportamento della modalità di risoluzione del bersaglio vuoto.
 9. Scegliete Avanti per passare alla Fase 3, Configurazione dell'accesso al servizio.
 10. Per Service Access, scegli Usa un ruolo IAM esistente, quindi scegli il ruolo IAM che hai creato come descritto nei prerequisiti di questo tutorial. Se il tuo ruolo non viene visualizzato, verifica che abbia la relazione di fiducia richiesta. Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).
 11. (Solo esperimenti con più account) Per le configurazioni degli account Target, aggiungi un Role ARN e una descrizione opzionale per ogni account di destinazione. Per caricare il ruolo dell'account di destinazione ARNs con un file CSV, scegli Carica ruolo ARNs per tutti gli account di destinazione, quindi scegli Scegli il file.CSV
 12. Scegli Avanti per passare al Passaggio 4, Configura le impostazioni opzionali.
 13. (Facoltativo) Per le condizioni di arresto, seleziona gli CloudWatch allarmi Amazon per le condizioni di arresto. Per ulteriori informazioni, consulta [Condizioni di arresto per la AWS FIS](#).
 14. (Facoltativo) Per i log, configura l'opzione di destinazione. Per inviare i log a un bucket S3, scegli Invia a un bucket Amazon S3 e inserisci il nome e il prefisso del bucket. Per inviare i log a Logs, scegli Send to CloudWatch Logs e inserisci il gruppo di log. CloudWatch
 15. (Facoltativo) Per i tag, scegliete Aggiungi nuovo tag e specificate una chiave e un valore per il tag. I tag che aggiungi vengono applicati al modello dell'esperimento, non agli esperimenti eseguiti utilizzando il modello.
 16. Scegli Avanti per passare alla Fase 5, Rivedi e crea.

17. Rivedi il modello e scegli Crea modello di esperimento. Quando viene richiesta la conferma, inserisci `create`, quindi scegli Crea modello di esperimento.

Per creare un modello di esperimento utilizzando la CLI

Utilizza il comando [create-experiment-template](#).

È possibile caricare un modello di esperimento da un file JSON.

Utilizzo del parametro `--cli-input-json`.

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

Per ulteriori informazioni, consulta [Generazione di un modello di scheletro CLI](#) nella Guida per l'AWS Command Line Interface utente. Per esempi di modelli, vedere [Esempi di modelli di esperimenti AWS FIS](#)

Visualizza i modelli di esperimenti

Puoi visualizzare i modelli di esperimento che hai creato.

Per visualizzare un modello di esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Per visualizzare le informazioni su un modello specifico, seleziona l'ID del modello di esperimento.
4. Nella sezione Dettagli, puoi visualizzare la descrizione e le condizioni di interruzione del modello.
5. Per visualizzare le azioni per il modello di esperimento, scegli Azioni.
6. Per visualizzare gli obiettivi per il modello di esperimento, scegli Obiettivi.
7. Per visualizzare i tag per il modello di esperimento, scegli Tag.

Per visualizzare un modello di esperimento utilizzando la CLI

Utilizzate il [list-experiment-templates](#) comando per ottenere un elenco di modelli di esperimento e utilizzate il [get-experiment-template](#) comando per ottenere informazioni su un modello di esperimento specifico.

Genera un'anteprima del bersaglio da un modello di esperimento

Prima di iniziare un esperimento, puoi generare un'anteprima del target per verificare che il modello dell'esperimento sia configurato per indirizzare le risorse previste. Le risorse scelte come target all'inizio dell'esperimento vero e proprio possono essere diverse da quelle dell'anteprima, poiché le risorse possono essere rimosse, aggiornate o campionate in modo casuale. Quando si genera un'anteprima dell'obiettivo, si avvia un esperimento che salta tutte le azioni.

Note

La generazione di un'anteprima del target non consente di verificare di disporre delle autorizzazioni necessarie per eseguire azioni sulle risorse.

Per avviare un'anteprima dell'obiettivo utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Per visualizzare gli obiettivi per il modello di esperimento, scegli Obiettivi.
4. Per verificare le risorse target per il modello di esperimento, scegli Genera anteprima. Quando esegui un esperimento, l'anteprima dell'obiettivo verrà aggiornata automaticamente con gli obiettivi dell'esperimento più recente.

Per avviare un'anteprima del target utilizzando la CLI

- Esegui il seguente comando [start-experiment](#). Sostituisci i valori in corsivo con i tuoi valori.

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

Inizia un esperimento da un modello

Dopo aver creato un modello di esperimento, puoi iniziare gli esperimenti utilizzando quel modello.

Quando inizi un esperimento, creiamo un'istantanea del modello specificato e la utilizziamo per eseguire l'esperimento. Pertanto, se il modello dell'esperimento viene aggiornato o eliminato mentre l'esperimento è in esecuzione, tali modifiche non hanno alcun impatto sull'esperimento in esecuzione.

Quando si avvia un esperimento, AWS FIS crea un ruolo collegato al servizio per conto dell'utente. Per ulteriori informazioni, consulta [Utilizza ruoli collegati ai servizi per Fault Injection Service AWS](#).

Dopo aver iniziato l'esperimento, puoi interromperlo in qualsiasi momento. Per ulteriori informazioni, consulta [Interrompere un esperimento](#).

Per iniziare un esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Avvia esperimento.
4. (Facoltativo) Per aggiungere un tag al tuo esperimento, scegli Aggiungi nuovo tag e inserisci una chiave per il tag e un valore per il tag.
5. Scegli Inizia un esperimento. Quando viene richiesta la conferma, inserisci **start** e scegli Avvia esperimento.

Per avviare un esperimento utilizzando la CLI

Utilizzate il comando [start-experiment](#).

Aggiorna un modello di esperimento

È possibile aggiornare un modello di esperimento esistente. Quando si aggiorna un modello di esperimento, le modifiche non influiscono sugli esperimenti in esecuzione che utilizzano il modello.

Per aggiornare un modello di esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Aggiorna modello di esperimento.
4. Modifica i dettagli del modello secondo necessità e scegli Aggiorna modello di esperimento.

Per aggiornare un modello di esperimento utilizzando la CLI

Utilizza il comando [update-experiment-template](#).

Modelli di esperimenti con tag

Puoi applicare i tuoi tag ai modelli sperimentali per aiutarti a organizzarli. Puoi anche implementare [policy IAM basate su tag](#) per controllare l'accesso ai modelli di esperimento.

Per etichettare un modello di esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Gestisci tag.
4. Per aggiungere un nuovo tag, scegli Aggiungi nuovo tag, quindi specifica una chiave e un valore.

Per rimuovere un tag, scegli Rimuovi per il tag.

5. Scegli Save (Salva).

Per etichettare un modello di esperimento utilizzando la CLI

Usa il comando [tag-resource](#).

Eliminare un modello di esperimento

Se non hai più bisogno di un modello di esperimento, puoi eliminarlo. Quando si elimina un modello di esperimento, gli esperimenti in corso che utilizzano il modello non vengono modificati. L'esperimento continua a funzionare fino al completamento o all'arresto. Tuttavia, i modelli di esperimento che vengono eliminati non sono disponibili per la visualizzazione dalla pagina Esperimenti della console.

Per eliminare un modello di esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Elimina modello di esperimento.
4. Quando viene richiesta la conferma, inserisci **delete** e scegli Elimina modello di esperimento.

Per eliminare un modello di esperimento utilizzando la CLI

Utilizza il comando [delete-experiment-template](#).

Esempi di modelli di esperimenti AWS FIS

Se utilizzi l'API AWS FIS o uno strumento da riga di comando per creare un modello di esperimento, puoi creare il modello in JavaScript Object Notation (JSON). Per ulteriori informazioni sui componenti di un modello di esperimento, consulta. [AWS Componenti del modello di esperimento FIS](#)

Per creare un esperimento utilizzando uno dei modelli di esempio, salvalo in un file JSON (ad esempio, `my-template.json`), sostituisci i valori segnaposto *italics* con i tuoi valori e quindi esegui il comando seguente. [create-experiment-template](#)

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

Modelli di esempio

- [Interrompi le EC2 istanze in base ai filtri](#)
- [Interrompe un numero specificato di istanze EC2](#)
- [Esegui un documento FIS SSM preconfigurato AWS](#)
- [Esegui un runbook di automazione predefinito](#)
- [Limita le azioni API sulle EC2 istanze con il ruolo IAM di destinazione](#)
- [Stress test della CPU dei pod in un cluster Kubernetes](#)
- [Eccezione di throughput assegnata per un numero specificato di Kinesis Data Streams](#)
- [Sperimenta un esempio di autorizzazioni di ruolo](#)

Interrompi le EC2 istanze in base ai filtri

L'esempio seguente interrompe tutte le EC2 istanze Amazon in esecuzione nella regione specificata con il tag specificato nel VPC specificato. Le riavvia dopo due minuti.

```
{
  "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"
}
```

Interrompe un numero specificato di istanze EC2

L'esempio seguente arresta tre istanze con il tag specificato. AWS FIS seleziona le istanze specifiche da interrompere a caso. Riavvia queste istanze dopo due minuti.

```
{
  "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"
}
```

Esegui un documento FIS SSM preconfigurato AWS

[L'esempio seguente esegue un'iniezione di errore della CPU per 60 secondi sull' EC2 istanza specificata utilizzando un documento AWS FIS SSM preconfigurato, -CPU-stress. AWSFIS-Run AWS](#)

La FIS monitora l'esperimento per due minuti.

```
{
  "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"
}
```

Esegui un runbook di automazione predefinito

L'esempio seguente pubblica una notifica su Amazon SNS utilizzando un runbook fornito da [Systems Manager, AWS-Publish. SNSNotification](#) Il ruolo deve disporre delle autorizzazioni per pubblicare notifiche sull'argomento SNS specificato.

```
{
  "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"
}
```

Limita le azioni API sulle EC2 istanze con il ruolo IAM di destinazione

L'esempio seguente limita il 100% delle chiamate API specificate nella definizione dell'azione per le chiamate API effettuate dai ruoli IAM specificati nella definizione di destinazione.

Note

Se desideri scegliere come target EC2 le istanze che fanno parte di un gruppo Auto Scaling, usa l'azione `aws:ec2asg-insufficient-instance-capacity: -error` e scegli invece come target il

gruppo Auto Scaling. Per ulteriori informazioni, consulta [aws:ec2:asg-insufficient-instance-capacity-error](#).

```
{
  "tags": {
    "Name": "ThrottleEC2APIActions"
  },
  "description": "Throttle the specified EC2 API actions on the specified IAM role",
  "targets": {
    "myRole": {
      "resourceType": "aws:iam:role",
      "resourceArns": ["arn:aws:iam::111122223333:role/role-name"],
      "selectionMode": "ALL"
    }
  },
  "actions": {
    "ThrottleAPI": {
      "actionId": "aws:fis:inject-api-throttle-error",
      "description": "Throttle APIs for 5 minutes",
      "parameters": {
        "service": "ec2",
        "operations": "DescribeInstances,DescribeVolumes",
        "percentage": "100",
        "duration": "PT2M"
      },
      "targets": {
        "Roles": "myRole"
      }
    }
  },
  "stopConditions": [
    {
      "source": "aws:cloudwatch:alarm",
      "value": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
    }
  ],
  "roleArn": "arn:aws:iam::111122223333:role/role-name"
}
```

Stress test della CPU dei pod in un cluster Kubernetes

L'esempio seguente utilizza Chaos Mesh per sottoporre a stress test la CPU dei pod in un cluster Amazon EKS Kubernetes per 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\"]},\nlabelSelectors\":{\"run\":{\"nginx\"}},\nmode\":{\"all\"},\nstressors\":{\"cpu\":{\"workers\":1,\nload\":50}},\nduration\":{\"1m\"}}",
        "targets": {
          "Cluster": "Cluster-Target-1"
        }
      }
    }
  },
  "stopConditions": [{
    "source": "none"
  }],
  "roleArn": "arn:aws:iam::111122223333:role/role-name",
  "tags": {}
}
```

L'esempio seguente utilizza Litmus per sottoporre a stress test la CPU dei pod in un cluster Amazon EKS Kubernetes per 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": "{\n  \"engineState\": \"active\",\n  \"appinfo\":\n  {\n    \"appns\": \"default\",\n    \"applabel\": \"run=nginx\",\n    \"appkind\": \"deployment\",\n    \"chaosServiceAccount\": \"litmus-admin\",\n    \"experiments\": [\n      {\n        \"name\": \"pod-cpu-hog\",\n        \"spec\": {\n          \"components\": [\n            {\n              \"env\": [\n                {\n                  \"name\": \"TOTAL_CHAOS_DURATION\",\n                  \"value\": \"60\",\n                },\n                {\n                  \"name\": \"CPU_CORES\",\n                  \"value\": \"1\",\n                },\n                {\n                  \"name\": \"PODS_AFFECTED_PERC\",\n                  \"value\": \"100\",\n                },\n                {\n                  \"name\": \"CONTAINER_RUNTIME\",\n                  \"value\": \"docker\",\n                },\n                {\n                  \"name\": \"SOCKET_PATH\",\n                  \"value\": \"/var/run/docker.sock\",\n                }\n              ],\n              \"probe\": []\n            }\n          ],\n          \"annotationCheck\": \"false\"\n        }\n      }\n    ]\n  }\n}"
      },
      "targets": {
        "Cluster": "MyCluster"
      }
    }
  },
  "stopConditions": [
    {
      "source": "none"
    }
  ],
  "roleArn": "arn:aws:iam::111122223333:role/role-name",
  "tags": {}
}

```

Eccezione di throughput assegnata per un numero specificato di Kinesis Data Streams

L'esempio seguente genera un'eccezione di throughput assegnato per il 100% delle richieste, fino a cinque Kinesis Data Streams con il tag specificato. AWS FIS seleziona casualmente gli stream da influenzare. Dopo 5 minuti l'errore viene rimosso.

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

```
}  
}
```

Sperimenta un esempio di autorizzazioni di ruolo

La seguente autorizzazione consente di eseguire le `aws:kinesis:stream-expired-iterator-exception` azioni `aws:kinesis:stream-provisioned-throughput-exception` and su un flusso specifico che ha un impatto sul 50% delle richieste.

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

Gestione degli esperimenti AWS FIS

AWS FIS consente di eseguire esperimenti di iniezione dei guasti sui AWS carichi di lavoro. Per iniziare, crea un modello di [esperimento](#). Dopo aver creato un modello di esperimento, puoi utilizzarlo per iniziare un esperimento.

Un esperimento è terminato quando si verifica una delle seguenti condizioni:

- Tutte le [azioni](#) nel modello sono state completate correttamente.
- Viene attivata una [condizione di arresto](#).
- Un'azione non può essere completata a causa di un errore. Ad esempio, se l'[obiettivo](#) non può essere trovato.
- L'esperimento viene [interrotto manualmente](#).

Non è possibile riprendere un esperimento interrotto o fallito. Inoltre, non è possibile eseguire nuovamente un esperimento completato. Tuttavia, puoi iniziare un nuovo esperimento dallo stesso modello di esperimento. Facoltativamente, è possibile aggiornare il modello di esperimento prima di specificarlo nuovamente in un nuovo esperimento.

Processi

- [Avviate un esperimento](#)
- [Visualizza i tuoi esperimenti](#)
- [Etichetta un esperimento](#)
- [Interrompere un esperimento](#)
- [Elenca gli obiettivi risolti](#)

Avviate un esperimento

Si avvia un esperimento da un modello di esperimento. Per ulteriori informazioni, consulta [Inizia un esperimento da un modello](#).

Puoi pianificare i tuoi esperimenti come attività singola o come attività ricorrenti utilizzando Amazon EventBridge. Per ulteriori informazioni, consulta [Tutorial: Pianifica un esperimento ricorrente](#).

È possibile monitorare l'esperimento utilizzando una delle seguenti funzionalità:

- Visualizza i tuoi esperimenti nella console AWS FIS. Per ulteriori informazioni, consulta [Visualizza i tuoi esperimenti](#).
- Visualizza i CloudWatch parametri di Amazon per le risorse target nei tuoi esperimenti o visualizza i parametri di utilizzo AWS FIS. Per ulteriori informazioni, consulta [Monitora utilizzando CloudWatch](#).
- Abilita la registrazione degli esperimenti per acquisire informazioni dettagliate sull'esperimento mentre viene eseguito. Per ulteriori informazioni, consulta [Registrazione degli esperimenti](#).

Visualizza i tuoi esperimenti

È possibile visualizzare lo stato di avanzamento di un esperimento in corso e visualizzare gli esperimenti completati, interrotti o non riusciti.

Gli esperimenti interrotti, completati e non riusciti vengono rimossi automaticamente dal tuo account dopo 120 giorni.

Per visualizzare gli esperimenti utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Scegli l'ID dell'esperimento per aprirne la pagina dei dettagli.
4. Effettuare una o più delle seguenti operazioni:
 - Seleziona Dettagli, Stato per [lo stato dell'esperimento](#).
 - Scegli la scheda Azioni per informazioni sulle azioni dell'esperimento.
 - Scegli la scheda Obiettivi per informazioni sugli obiettivi dell'esperimento.
 - Scegli la scheda Cronologia per una rappresentazione visiva delle azioni in base all'ora di inizio e di fine.

Per visualizzare gli esperimenti utilizzando la CLI

Usa il comando [list-experimentals](#) per ottenere un elenco di esperimenti e usa il comando [get-experiment per ottenere informazioni su un esperimento](#) specifico.

Stati dell'esperimento

Un esperimento può trovarsi in uno dei seguenti stati:

- in sospeso: l'esperimento è in sospeso.
- avvio: l'esperimento si sta preparando a iniziare.
- in esecuzione: l'esperimento è in corso.
- completato: tutte le azioni dell'esperimento sono state completate con successo.
- arresto: la condizione di arresto è stata attivata o l'esperimento è stato interrotto manualmente.
- interrotto: tutte le azioni in corso o in sospeso nell'esperimento vengono interrotte.
- fallito: l'esperimento non è riuscito a causa di un errore, ad esempio autorizzazioni insufficienti o sintassi errata.
- annullato: l'esperimento è stato interrotto o ne è stato impedito l'avvio a causa di una leva di sicurezza innestata.

Stati d'azione

Un'azione può trovarsi in uno dei seguenti stati:

- in sospeso: l'azione è in sospeso, o perché l'esperimento non è iniziato o l'azione deve iniziare più avanti nell'esperimento.
- in corso: l'azione si sta preparando per iniziare.
- in esecuzione: l'azione è in esecuzione.
- completata: l'azione è stata completata con successo.
- annullato: l'esperimento si è interrotto prima dell'inizio dell'azione.
- ignorata: l'azione è stata ignorata.
- arresto: l'azione si sta interrompendo.
- interrotto: tutte le azioni in esecuzione o in sospeso nell'esperimento vengono interrotte.
- fallita: l'azione non è riuscita a causa di un errore del client, ad esempio autorizzazioni insufficienti o sintassi errata.

Etichetta un esperimento

Puoi applicare tag agli esperimenti per organizzarli meglio. Puoi anche implementare [policy IAM basate su tag](#) per controllare l'accesso agli esperimenti.

Per etichettare un esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Seleziona l'esperimento e scegli Azioni, Gestisci tag.
4. Per aggiungere un nuovo tag, scegli Aggiungi nuovo tag e specifica una chiave e un valore.

Per rimuovere un tag, scegli Rimuovi per il tag.

5. Scegli Save (Salva).

Per etichettare un esperimento utilizzando la CLI

Usa il comando [tag-resource](#).

Interrompere un esperimento

Puoi interrompere un esperimento in corso in qualsiasi momento. Quando interrompi un esperimento, tutte le azioni successive che non sono state completate per un'azione vengono completate prima che l'esperimento si interrompa. Non è possibile riprendere un esperimento interrotto.

Per interrompere un esperimento utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Seleziona l'esperimento e scegli Interrompi esperimento.
4. Nella finestra di dialogo di conferma, scegliete Interrompi esperimento.

Per interrompere un esperimento utilizzando la CLI

Usa il comando [stop-experiment](#).

Elenca gli obiettivi risolti

È possibile visualizzare le informazioni relative agli obiettivi risolti per un esperimento al termine della risoluzione degli obiettivi.

Per visualizzare gli obiettivi risolti utilizzando la console

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Seleziona l'esperimento e scegli Segnala.
4. Visualizza le informazioni sugli obiettivi risolti in Risorse.

Per visualizzare gli obiettivi risolti utilizzando la CLI

Utilizza il comando [list-experiment-resolved-targets](#).

Tutorial per il servizio di iniezione dei AWS guasti

I seguenti tutorial mostrano come creare ed eseguire esperimenti utilizzando AWS Fault Injection Service (FIS).AWS

Esercitazioni

- [Tutorial: interruzione e avvio dell'istanza di test con AWS FIS](#)
- [Tutorial: Esegui lo stress della CPU su un'istanza utilizzando AWS FIS](#)
- [Tutorial: Test delle interruzioni delle istanze Spot con AWS FIS](#)
- [Tutorial: simula un evento di connettività](#)
- [Tutorial: Pianifica un esperimento ricorrente](#)

Tutorial: interruzione e avvio dell'istanza di test con AWS FIS

È possibile utilizzare AWS AWS Fault Injection Service (FIS) per testare il modo in cui le applicazioni gestiscono l'arresto e l'avvio delle istanze. Utilizzate questo tutorial per creare un modello di esperimento che utilizzi l'`aws:ec2:stop-instances` azione AWS FIS per arrestare un'istanza e poi una seconda istanza.

Prerequisiti

Per completare questo tutorial, assicuratevi di fare quanto segue:

- Avvia due EC2 istanze di test nel tuo account. Dopo aver avviato le istanze, annota quelle IDs di entrambe.
- Crea un ruolo IAM che consenta al servizio AWS FIS di eseguire l'`aws:ec2:stop-instances` azione per tuo conto. Per ulteriori informazioni, consulta [Ruoli IAM per AWS esperimenti FIS](#).
- Assicurati di avere accesso al AWS FIS. Per ulteriori informazioni, consulta Esempi di [policy AWS FIS](#).

Fase 1: Creare un modello di esperimento

Crea il modello di esperimento utilizzando la console AWS FIS. Nel modello, specificate due azioni che verranno eseguite in sequenza per tre minuti ciascuna. La prima azione interrompe una delle

istanze di test, scelta casualmente da AWS FIS. La seconda azione interrompe entrambe le istanze di test.

Per creare un modello di esperimento

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Scegli Crea modello di esperimento.
4. Per la Fase 1, Specificate i dettagli del modello, effettuate le seguenti operazioni:
 - a. In Descrizione e nome, inserisci una descrizione per il modello, ad esempio Amazon S3 Network Disrupt Connectivity.
 - b. Scegliete Avanti e passate al Passaggio 2, Specificate azioni e obiettivi.
5. Alla voce Actions (Operazioni), procedere nel seguente modo:
 - a. Selezionare Add action (Aggiungi operazione).
 - b. Immettete un nome per l'azione. Ad esempio, specifica **stopOneInstance**.
 - c. Per Tipo di azione, scegli aws:ec2:stop-instances.
 - d. Per Target, mantieni l'obiettivo che FIS crea per te. AWS
 - e. Per i parametri di azione, Avvia istanze dopo la durata, specifica 3 minuti (PT3M).
 - f. Seleziona Salva.
6. Per Targets (Target) esegui queste operazioni:
 - a. Scegliete Modifica per la destinazione che AWS FIS ha creato automaticamente per voi nel passaggio precedente.
 - b. Sostituite il nome predefinito con un nome più descrittivo. Ad esempio, specifica **oneRandomInstance**.
 - c. Verifica che il tipo di risorsa sia aws:ec2:instance.
 - d. Per il metodo Target, scegli Resource IDs, quindi scegli una delle due istanze IDs di test.
 - e. Per la modalità Selezione, scegli Count. Per Numero di risorse, immettere **1**.
 - f. Seleziona Salva.
7. Scegliete Aggiungi obiettivo ed effettuate le seguenti operazioni:
 - a. Inserisci un nome per l'obiettivo. Ad esempio, specifica **bothInstances**.
 - b. Per Tipo di risorsa, scegli aws:ec2:instance.

- c. Per il metodo Target, scegli Resource IDs, quindi scegli una delle due istanze IDs di test.
 - d. Per la modalità Selezione, scegliete Tutto.
 - e. Seleziona Salva.
8. Nella sezione Azioni, scegli Aggiungi azione. Esegui questa operazione:
- a. In Nome, inserisci un nome per l'azione. Ad esempio, specifica **stopBothInstances**.
 - b. Per Tipo di azione, scegli `aws:ec2:stop-instances`.
 - c. Per Inizia dopo, scegli la prima azione che hai aggiunto (). **stopOneInstance**
 - d. Per Target, scegli il secondo obiettivo che hai aggiunto (**bothInstances**).
 - e. Per i parametri di azione, avvia le istanze dopo la durata, specifica 3 minuti (PT3M).
 - f. Seleziona Salva.
9. Scegli Avanti per passare alla Fase 3, Configurazione dell'accesso al servizio.
10. Per Service Access, scegli Usa un ruolo IAM esistente, quindi scegli il ruolo IAM che hai creato come descritto nei prerequisiti di questo tutorial. Se il tuo ruolo non viene visualizzato, verifica che abbia la relazione di fiducia richiesta. Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).
11. Scegli Avanti per passare al Passaggio 4, Configura le impostazioni opzionali.
12. (Facoltativo) Per i tag, scegliete Aggiungi nuovo tag e specificate una chiave e un valore per il tag. I tag che aggiungi vengono applicati al modello dell'esperimento, non agli esperimenti eseguiti utilizzando il modello.
13. Scegli Avanti per passare alla Fase 5, Rivedi e crea.
14. Rivedi il modello e scegli Crea modello di esperimento. Quando viene richiesta la conferma, inserisci `create`, quindi scegli Crea modello di esperimento.

(Facoltativo) Per visualizzare il modello di esperimento JSON

Scegliete la scheda Esporta. Di seguito è riportato un esempio di JSON creato dalla precedente procedura della console.

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

}

Fase 2: Avviare l'esperimento

Quando hai finito di creare il modello di esperimento, puoi usarlo per iniziare un esperimento.

Per iniziare un esperimento

1. Dovresti trovarti nella pagina dei dettagli del modello di esperimento che hai appena creato. Altrimenti, scegli Modelli di esperimento, quindi seleziona l'ID del modello di esperimento per aprire la pagina dei dettagli.
2. Scegli Inizia un esperimento.
3. (Facoltativo) Per aggiungere un tag al tuo esperimento, scegli Aggiungi nuovo tag e inserisci una chiave per il tag e un valore per il tag.
4. Scegli Inizia un esperimento. Quando viene richiesta la conferma, inserisci **start** e scegli Avvia esperimento.

Fase 3: Tieni traccia dell'avanzamento dell'esperimento

È possibile tenere traccia dell'avanzamento di un esperimento in corso fino al completamento, all'interruzione o al fallimento dell'esperimento.

Per tenere traccia dello stato di avanzamento di un esperimento

1. Dovresti essere nella pagina dei dettagli dell'esperimento che hai appena iniziato. Altrimenti, scegli Esperimenti, quindi seleziona l'ID dell'esperimento per aprire la pagina dei dettagli.
2. Per visualizzare lo stato dell'esperimento, seleziona Stato nel riquadro Dettagli. Per ulteriori informazioni, consulta [gli stati dell'esperimento](#).
3. Quando lo stato dell'esperimento è In esecuzione, vai al passaggio successivo.

Fase 4: Verifica il risultato dell'esperimento

È possibile verificare che le istanze siano state interrotte e avviate dall'esperimento come previsto.

Per verificare il risultato dell'esperimento

1. Apri la EC2 console Amazon <https://console.aws.amazon.com/ec2/> in una nuova scheda o finestra del browser. Ciò consente di continuare a monitorare l'avanzamento dell'esperimento nella console AWS FIS mentre si visualizza il risultato dell'esperimento nella EC2 console Amazon.
2. Nel pannello di navigazione, seleziona Instances (Istanze).
3. Quando lo stato della prima azione passa da In sospeso a In esecuzione (console AWS FIS), lo stato di una delle istanze di destinazione cambia da Running a Stopped (console Amazon EC2).
4. Dopo tre minuti, lo stato della prima azione diventa Completato, lo stato della seconda azione diventa In esecuzione e lo stato dell'altra istanza di destinazione diventa Stoppato.
5. Dopo tre minuti, lo stato della seconda azione diventa Completato, lo stato delle istanze di destinazione diventa In esecuzione e lo stato dell'esperimento diventa Completato.

Fase 5: rimozione

Se non hai più bisogno delle EC2 istanze di test che hai creato per questo esperimento, puoi terminarle.

Per terminare le istanze

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel pannello di navigazione, seleziona Instances (Istanze).
3. Seleziona l'istanza e scegli Instance state (Stato istanza), Terminate instance (Termina istanza).
4. Quando viene richiesta la conferma, seleziona Terminate (Termina).

Se non ti serve più il modello di esperimento, puoi eliminarlo.

Per eliminare un modello di esperimento utilizzando la console AWS FIS

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Elimina modello di esperimento.
4. Quando viene richiesta la conferma, inserisci **delete** e scegli Elimina modello di esperimento.

Tutorial: Esegui lo stress della CPU su un'istanza utilizzando AWS FIS

È possibile utilizzare AWS AWS Fault Injection Service (FIS) per verificare in che modo le applicazioni gestiscono lo stress della CPU. Usa questo tutorial per creare un modello di esperimento che utilizzi AWS FIS per eseguire un documento SSM preconfigurato che esegue lo stress della CPU su un'istanza. Il tutorial utilizza una condizione di arresto per interrompere l'esperimento quando l'utilizzo della CPU dell'istanza supera una soglia configurata.

Per ulteriori informazioni, consulta [the section called “Documenti FIS SSM preconfigurati AWS”](#).

Prerequisiti

Prima di utilizzare AWS FIS per eseguire lo stress della CPU, completare i seguenti prerequisiti.

Creazione di un ruolo IAM

Create un ruolo e allegare una politica che consenta a AWS FIS di utilizzare l'`aws:ssm:send-command` per il vostro conto. Per ulteriori informazioni, consulta [Ruoli IAM per AWS esperimenti FIS](#).

Verifica l'accesso a AWS FIS

Assicurati di avere accesso al AWS FIS. Per ulteriori informazioni, consulta Esempi di [policy AWS FIS](#).

Preparare un'istanza EC2 di prova

- Avvia un'istanza EC2 utilizzando Amazon Linux 2 o Ubuntu, come richiesto dai documenti SSM preconfigurati.
- L'istanza deve essere gestita da SSM. Per verificare che l'istanza sia gestita da SSM, apri la console [Fleet Manager](#). Se l'istanza non è gestita da SSM, verifica che l'agente SSM sia installato e che all'istanza sia associato un ruolo IAM con la policy di Amazon SSMManaged InstanceCore. Per verificare l'agente SSM installato, connettiti all'istanza ed esegui il comando seguente.

Amazon Linux 2

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

Ubuntu

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

- Abilita il monitoraggio dettagliato dell'istanza. Ciò fornisce i dati in periodi di 1 minuto, a un costo aggiuntivo. Seleziona l'istanza e scegli Azioni, Monitoraggio e risoluzione dei problemi, Gestisci il monitoraggio dettagliato.

Passaggio 1: creare un CloudWatch allarme per una condizione di arresto

Configura un CloudWatch allarme in modo da poter interrompere l'esperimento se l'utilizzo della CPU supera la soglia specificata. La procedura seguente imposta la soglia al 50% di utilizzo della CPU per l'istanza di destinazione. Per ulteriori informazioni, consulta [Condizioni di arresto](#).

Per creare un allarme che indichi quando l'utilizzo della CPU supera una soglia

1. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, scegliere Instances (Istanze).
3. Seleziona l'istanza di destinazione e scegli Azioni, Monitoraggio e risoluzione dei problemi, Gestione allarmi. CloudWatch
4. Per la notifica di allarme, usa l'interruttore per disattivare le notifiche di Amazon SNS.
5. Per le soglie di allarme, utilizza le seguenti impostazioni:
 - Raggruppa i campioni per: Massimo
 - Tipo di dati da campionare: utilizzo della CPU
 - Percentuale: **50**
 - Periodo: **1 Minute**
6. Quando hai finito di configurare l'allarme, scegli Crea.

Passaggio 2: crea un modello di esperimento

Crea il modello di esperimento utilizzando la console AWS FIS. Nel modello, si specifica la seguente azione da eseguire: [AWSFIS-Runaws:ssm:send-command/](#) -CPU-stress.

Come creare un modello di esperimento

1. AWS Apri <https://console.aws.amazon.com/fis/> la console FIS all'indirizzo.

2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Scegli Crea modello di esperimento.
4. Per la Fase 1, Specificate i dettagli del modello, effettuate le seguenti operazioni:
 - a. In Descrizione e nome, inserisci una descrizione per il modello.
 - b. Scegliete Avanti e passate alla Fase 2, Specificate azioni e obiettivi.
5. Alla voce Actions (Operazioni), procedere nel seguente modo:
 - a. Selezionare Add action (Aggiungi operazione).
 - b. Immettete un nome per l'azione. Ad esempio, specifica **runCpuStress**.
 - c. Per Tipo di azione, scegli AWSFIS-Runaws:ssm:send-command/ -CPU-stress. Questo aggiunge automaticamente l'ARN del documento SSM all'ARN del documento.
 - d. Per Target, mantieni la destinazione che AWS FIS crea per te.
 - e. Per Parametri di azione, Parametri del documento, inserisci quanto segue:

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

- f. Per Parametri di azione, Durata, specificare 5 minuti (PT5M).
 - g. Scegli Save (Salva).
6. Per Destinazioni, esegui queste operazioni:
 - a. Scegliete Modifica per la destinazione che AWS FIS ha creato automaticamente per voi nel passaggio precedente.
 - b. Sostituite il nome predefinito con un nome più descrittivo. Ad esempio, specifica **testInstance**.
 - c. Verifica che il tipo di risorsa sia aws:ec2:instance.
 - d. Per il metodo Target, scegli Resource IDs, quindi scegli l'ID dell'istanza di test.
 - e. Per la modalità Selezione, scegli Tutto.
 - f. Scegli Save (Salva).
7. Scegli Avanti per passare alla Fase 3, Configurazione dell'accesso al servizio.
8. Per Service Access, scegli Usa un ruolo IAM esistente, quindi scegli il ruolo IAM che hai creato come descritto nei prerequisiti di questo tutorial. Se il tuo ruolo non viene visualizzato, verifica che abbia la relazione di fiducia richiesta. Per ulteriori informazioni, consulta [the section called](#)

[“Ruolo dell'esperimento”](#).

9. Scegli Avanti per passare al Passaggio 4, Configura le impostazioni opzionali.
10. Per le condizioni di interruzione, seleziona l' CloudWatch allarme creato nel passaggio 1.
11. (Facoltativo) Per i tag, scegli Aggiungi nuovo tag e specifica una chiave e un valore per il tag. I tag che aggiungi vengono applicati al modello dell'esperimento, non agli esperimenti eseguiti utilizzando il modello.
12. Scegli Avanti per passare alla Fase 5, Rivedi e crea.
13. Rivedi il modello e scegli Crea modello di esperimento. Quando viene richiesta la conferma, inseriscicreate, quindi scegli Crea modello di esperimento.

(Facoltativo) Per visualizzare il modello di esperimento JSON

Scegli la scheda Esporta. Di seguito è riportato un esempio di JSON creato dalla precedente procedura della console.

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

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

Fase 3: Avviare l'esperimento

Quando hai finito di creare il modello di esperimento, puoi usarlo per iniziare un esperimento.

Come avviare un esperimento

1. Dovresti trovarti nella pagina dei dettagli del modello di esperimento che hai appena creato. Altrimenti, scegli Modelli di esperimento, quindi seleziona l'ID del modello di esperimento per aprire la pagina dei dettagli.
2. Scegli Inizia un esperimento.
3. (Facoltativo) Per aggiungere un tag al tuo esperimento, scegli Aggiungi nuovo tag e inserisci una chiave per il tag e un valore per il tag.
4. Scegli Inizia un esperimento. Quando viene richiesta la conferma, immetti **start**. Scegli Inizia un esperimento.

Fase 4: Tieni traccia dei progressi dell'esperimento

È possibile tenere traccia dell'avanzamento di un esperimento in corso fino al completamento, all'arresto o al fallimento dell'esperimento.

Per tenere traccia dell'avanzamento di un esperimento

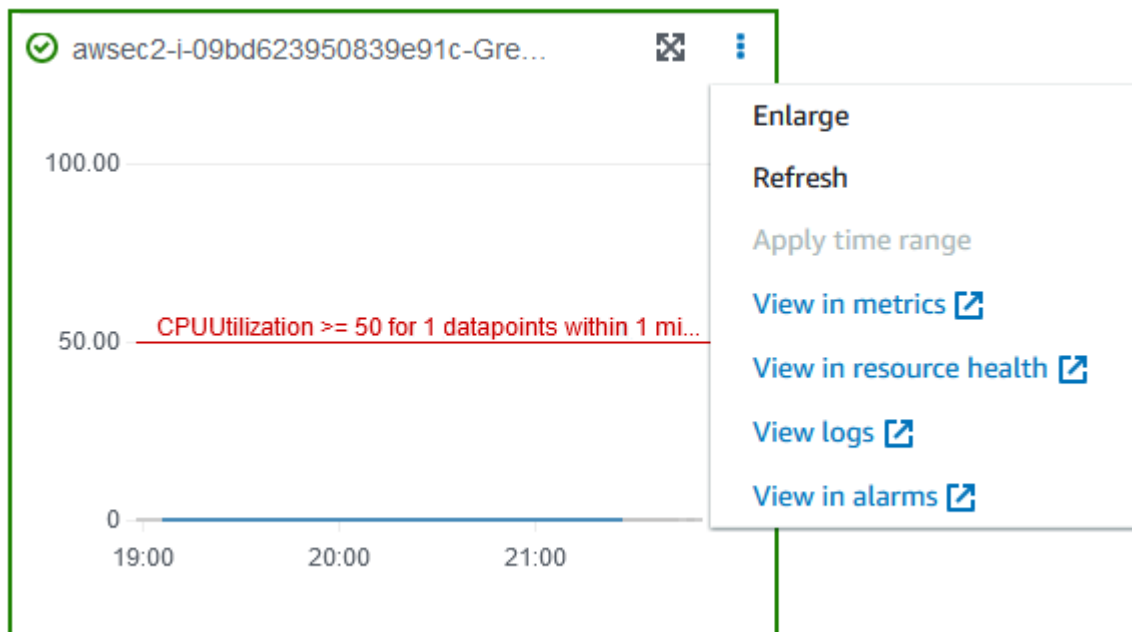
1. Dovresti essere nella pagina dei dettagli dell'esperimento che hai appena iniziato. Altrimenti, scegli Esperimenti, quindi seleziona l'ID dell'esperimento per aprire la pagina dei dettagli dell'esperimento.
2. Per visualizzare lo stato dell'esperimento, seleziona Stato nel riquadro Dettagli. Per ulteriori informazioni, consulta [gli stati dell'esperimento](#).
3. Quando lo stato dell'esperimento è In esecuzione, vai al passaggio successivo.

Fase 5: Verifica i risultati dell'esperimento

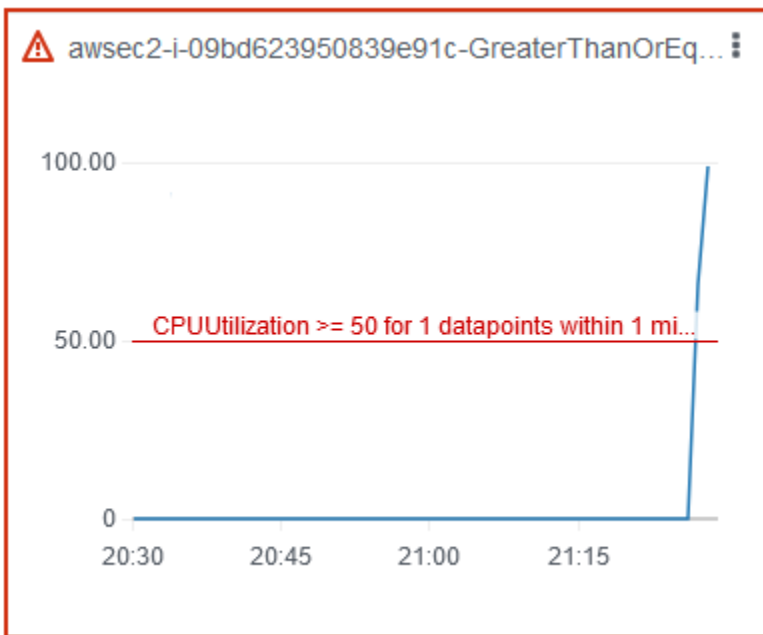
È possibile monitorare l'utilizzo della CPU dell'istanza mentre l'esperimento è in esecuzione. Quando l'utilizzo della CPU raggiunge la soglia, viene attivato l'allarme e l'esperimento viene interrotto dalla condizione di arresto.

Per verificare i risultati dell'esperimento

1. Scegli la scheda Condizioni di arresto. Il bordo verde e l'icona con il segno di spunta verde indicano che lo stato iniziale dell'allarme è OK. La linea rossa indica la soglia di allarme. Se preferisci un grafico più dettagliato, scegli Ingrandisci dal menu del widget.



2. Quando l'utilizzo della CPU supera la soglia, il bordo rosso e l'icona del punto esclamativo rosso nella scheda Condizioni di arresto indicano che lo stato di allarme è cambiato in ALARM. Nel riquadro Dettagli, lo stato dell'esperimento è Interrotto. Se si seleziona lo stato, il messaggio visualizzato è «Esperimento interrotto dalla condizione di interruzione».



3. Quando l'utilizzo della CPU scende al di sotto della soglia, il bordo verde e l'icona con il segno di spunta verde indicano che lo stato di allarme è cambiato in OK
4. (Facoltativo) Scegliete Visualizza negli allarmi dal menu del widget. Si apre la pagina dei dettagli dell'allarme nella CloudWatch console, dove puoi ottenere maggiori dettagli sull'allarme o modificare le impostazioni dell'allarme.

Fase 6: pulizia

Se non hai più bisogno dell'istanza EC2 di test che hai creato per questo esperimento, puoi terminarla.

Come terminare l'istanza

1. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, scegliere Instances (Istanze).
3. Seleziona le istanze di test e scegli Instance state, Termina istanza.
4. Quando viene richiesta la conferma, seleziona Terminate (Termina).

Se non hai più bisogno del modello di esperimento, puoi eliminarlo.

Per eliminare un modello di esperimento utilizzando la console AWS FIS

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Elimina modello di esperimento.
4. Quando viene richiesta la conferma, inserisci **delete** e scegli Elimina modello di esperimento.

Tutorial: Test delle interruzioni delle istanze Spot con AWS FIS

Le istanze Spot utilizzano la EC2 capacità inutilizzata disponibile, con uno sconto fino al 90% rispetto ai prezzi On-Demand. Tuttavia, Amazon EC2 può interrompere le tue istanze Spot quando ha bisogno di recuperare la capacità. Quando utilizzi le istanze Spot, devi essere preparato a potenziali interruzioni. Per ulteriori informazioni, consulta le [interruzioni delle istanze Spot](#) nella Amazon EC2 User Guide.

Puoi utilizzare AWS AWS Fault Injection Service (FIS) per testare in che modo le tue applicazioni gestiscono un'interruzione di un'istanza Spot. Usa questo tutorial per creare un modello di esperimento che utilizzi l'`aws:ec2:send-spot-instance-interruptions` azione AWS FIS per interrompere una delle tue istanze Spot.

In alternativa, per avviare l'esperimento utilizzando la EC2 console Amazon, consulta [Avviare un'interruzione di un'istanza Spot](#) nella Amazon EC2 User Guide.

Prerequisiti

Prima di poter utilizzare AWS FIS per interrompere un'istanza Spot, completa i seguenti prerequisiti.

1. Creazione di un ruolo IAM

Crea un ruolo e allega una politica che consenta a AWS FIS di eseguire l'`aws:ec2:send-spot-instance-interruptions` azione per tuo conto. Per ulteriori informazioni, consulta [Ruoli IAM per AWS esperimenti FIS](#).

2. Verifica l'accesso al AWS FIS

Assicurati di avere accesso al AWS FIS. Per ulteriori informazioni, consulta Esempi di [policy AWS FIS](#).

3. (Facoltativo) Crea una richiesta di istanza Spot

Se desideri utilizzare una nuova istanza Spot per questo esperimento, usa il comando [run-instances](#) per richiedere un'istanza Spot. L'impostazione predefinita prevede la chiusura delle istanze Spot interrotte. Se si imposta il comportamento di interruzione su `stop`, è necessario impostare anche il tipo su `persistent`. In questo tutorial, non impostate il comportamento di interruzione su `hibernate`, poiché il processo di ibernazione inizia immediatamente.

```
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
```

Di seguito è riportato un esempio del file `spot-options.json`.

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

L'opzione `--query` del comando di esempio fa in modo che il comando restituisca solo l'ID dell'istanza Spot. Di seguito è riportato un output di esempio.

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

4. Aggiunge un tag in modo che AWS FIS possa identificare l'istanza Spot di destinazione

Utilizzate il comando [create-tags](#) per aggiungere il tag `Name=interruptMe` all'istanza Spot di destinazione.

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

Fase 1: Creare un modello di esperimento

Crea il modello di esperimento utilizzando la console AWS FIS. Nel modello, si specifica l'azione che verrà eseguita. L'azione interrompe l'istanza Spot con il tag specificato. Se esiste più di un'istanza Spot con il tag, AWS FIS ne sceglie una a caso.

Per creare un modello di esperimento

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Scegli Crea modello di esperimento.
4. Per la Fase 1, Specificate i dettagli del modello, effettuate le seguenti operazioni:
 - a. In Descrizione e nome, inserisci una descrizione e un nome per il modello.
 - b. Scegliete Avanti e passate al Passaggio 2, Specificate azioni e obiettivi.
5. Alla voce Actions (Operazioni), procedere nel seguente modo:
 - a. Selezionare Add action (Aggiungi operazione).
 - b. Immettete un nome per l'azione. Ad esempio, specifica **interruptSpotInstance**.
 - c. Per Tipo di azione, scegli aws:ec2:.. send-spot-instance-interruptions
 - d. Per Target, mantieni l'obiettivo che AWS FIS crea per te.
 - e. Per i parametri di azione, Durata prima dell'interruzione, specifica 2 minuti (PT2M).
 - f. Seleziona Salva.
6. Per Targets (Target) esegui queste operazioni:
 - a. Scegliete Modifica per la destinazione che AWS FIS ha creato automaticamente per voi nel passaggio precedente.
 - b. Sostituite il nome predefinito con un nome più descrittivo. Ad esempio, specifica **oneSpotInstance**.
 - c. Verifica che il tipo di risorsa sia aws:ec2:spot-instance.
 - d. Per il metodo Target, scegli i tag, i filtri e i parametri delle risorse.
 - e. Per i tag delle risorse, scegli Aggiungi nuovo tag e inserisci la chiave e il valore del tag. Utilizza il tag che hai aggiunto all'istanza Spot per interrompere, come descritto nei Prerequisiti di questo tutorial.

- f. Per i filtri delle risorse, scegli Aggiungi nuovo filtro e inserisci **State.Name** come percorso e **running** come valore.
 - g. Per la modalità Selezione, scegliete Conteggio. Per Numero di risorse, immettere **1**.
 - h. Seleziona Salva.
7. Scegli Avanti per passare alla Fase 3, Configurazione dell'accesso al servizio.
 8. Per Service Access, scegli Usa un ruolo IAM esistente, quindi scegli il ruolo IAM che hai creato come descritto nei prerequisiti di questo tutorial. Se il tuo ruolo non viene visualizzato, verifica che abbia la relazione di fiducia richiesta. Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).
 9. Scegli Avanti per passare al Passaggio 4, Configura le impostazioni opzionali.
 10. (Facoltativo) Per i tag, scegliete Aggiungi nuovo tag e specificate una chiave e un valore per il tag. I tag che aggiungi vengono applicati al modello dell'esperimento, non agli esperimenti eseguiti utilizzando il modello.
 11. Scegli Avanti per passare alla Fase 5, Rivedi e crea.
 12. Rivedi il modello e scegli Crea modello di esperimento. Quando viene richiesta la conferma, inserisci `create`, quindi scegli Crea modello di esperimento.

(Facoltativo) Per visualizzare il modello di esperimento JSON

Scegli la scheda Esporta. Di seguito è riportato un esempio di JSON creato dalla precedente procedura della console.

```
{
  "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"
}
}
```

Fase 2: Avviare l'esperimento

Quando hai finito di creare il modello di esperimento, puoi usarlo per iniziare un esperimento.

Per iniziare un esperimento

1. Dovresti trovarti nella pagina dei dettagli del modello di esperimento che hai appena creato. Altrimenti, scegli Modelli di esperimento, quindi seleziona l'ID del modello di esperimento per aprire la pagina dei dettagli.
2. Scegli Inizia un esperimento.
3. (Facoltativo) Per aggiungere un tag al tuo esperimento, scegli Aggiungi nuovo tag e inserisci una chiave per il tag e un valore per il tag.
4. Scegli Inizia un esperimento. Quando viene richiesta la conferma, inserisci **start** e scegli Avvia esperimento.

Fase 3: Tieni traccia dell'avanzamento dell'esperimento

È possibile tenere traccia dell'avanzamento di un esperimento in corso fino al completamento, all'interruzione o al fallimento dell'esperimento.

Per tenere traccia dell'avanzamento di un esperimento

1. Dovresti essere nella pagina dei dettagli dell'esperimento che hai appena iniziato. Altrimenti, scegli Esperimenti, quindi seleziona l'ID dell'esperimento per aprire la pagina dei dettagli.
2. Per visualizzare lo stato dell'esperimento, seleziona Stato nel riquadro Dettagli. Per ulteriori informazioni, consulta [gli stati dell'esperimento](#).
3. Quando lo stato dell'esperimento è In esecuzione, vai al passaggio successivo.

Fase 4: Verifica il risultato dell'esperimento

Quando l'azione per questo esperimento è completata, si verifica quanto segue:

- L'istanza Spot di destinazione riceve una raccomandazione di [ribilanciamento dell'istanza](#).
- Un [avviso di interruzione dell'istanza Spot](#) viene emesso due minuti prima che Amazon chiuda o EC2 interrompa l'istanza.
- Dopo due minuti, l'istanza Spot viene interrotta o interrotta.
- Un'istanza Spot interrotta da AWS FIS rimane ferma finché non viene riavviata.

Per verificare che l'istanza sia stata interrotta dall'esperimento

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, apri Spot Requests (Richieste spot) e Instances (Istanze) in schede o finestre separate del browser.
3. Per Spot Requests (Richieste spot) seleziona la richiesta dell'istanza spot. Lo stato iniziale è fulfilled. Al termine dell'esperimento, lo stato cambia come segue:
 - terminate- Lo stato cambia in instance-terminated-by-experiment
 - stop- Lo stato cambia in marked-for-stop-by-experiment e poi instance-stopped-by-experiment.
4. Per Istanze, seleziona l'istanza spot. Lo stato iniziale è Running. Due minuti dopo aver ricevuto l'avviso di interruzione dell'istanza Spot, lo stato cambia come segue:

- `stop`- Lo stato cambia in `Stopping` e poi `Stopped`.
- `terminate`- Lo stato cambia in `Shutting-down` e poi `Terminated`.

Fase 5: rimozione

Se hai creato l'istanza Spot di prova per questo esperimento con un comportamento di interruzione `stop` e non ti serve più, puoi annullare la richiesta dell'istanza Spot e terminare l'istanza Spot.

Per annullare la richiesta e terminare l'istanza, utilizzare il AWS CLI

1. Utilizza il [cancel-spot-instance-requests](#) comando per annullare la richiesta dell'istanza Spot.

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

2. Utilizzate il comando [terminate-instances](#) per terminare l'istanza.

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

Se non hai più bisogno del modello di esperimento, puoi eliminarlo.

Per eliminare un modello di esperimento utilizzando la console AWS FIS

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Elimina modello di esperimento.
4. Quando viene richiesta la conferma, inserisci **delete** e scegli Elimina modello di esperimento.

Tutorial: simula un evento di connettività

È possibile utilizzare AWS Fault Injection Service (AWS FIS) per simulare una serie di eventi di connettività. AWS FIS simula gli eventi di connettività bloccando le connessioni di rete in uno dei seguenti modi:

- `all`— Impedisce a tutto il traffico in entrata e in uscita dalla sottorete. Si noti che questa opzione consente il traffico all'interno della sottorete, incluso il traffico da e verso le interfacce di rete nella sottorete.

- `availability-zone`— Impedisce il traffico intra-VPC da e verso le sottoreti in altre zone di disponibilità.
- `dynamodb`— Impedisce il traffico da e verso l'endpoint regionale per DynamoDB nella regione corrente.
- `prefix-list`— Impedisce il traffico da e verso l'elenco di prefissi specificato.
- `s3`— Impedisce il traffico da e verso l'endpoint regionale per Amazon S3 nella regione corrente.
- `s3express`— Impedisce il traffico da e verso l'endpoint zonale per Amazon S3 Express One Zone nella zona di destinazione delle sottoreti di destinazione. Le sottoreti di destinazione devono risiedere dove S3 Express One Zone è attualmente disponibile AZs . Per ulteriori informazioni, consulta Zone e regioni di disponibilità di [S3 Express One Zone](#) .
- `vpc`— Impedisce al traffico di entrare e uscire dal VPC.

Usa questo tutorial per creare un modello di esperimento che utilizzi l'`aws:network:disrupt-connectivity` azione AWS FIS per introdurre la perdita di connettività con Amazon S3 in una sottorete di destinazione.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare un modello di esperimento FIS AWS](#)
- [Fase 2: Eseguire il ping di un endpoint Amazon S3](#)
- [Fase 3: Iniziate l'esperimento AWS FIS](#)
- [Fase 4: Tieni traccia dei progressi dell'esperimento AWS FIS](#)
- [Fase 5: verifica l'interruzione della rete Amazon S3](#)
- [Fase 5: rimozione](#)

Prerequisiti

Prima di iniziare questo tutorial, devi avere un ruolo con le autorizzazioni appropriate nella tua Account AWS istanza Amazon EC2 e testarla:

Un ruolo con autorizzazioni nel tuo Account AWS

Crea un ruolo e allega una politica che consenta alla AWS FIS di eseguire l'`aws:network:disrupt-connectivity` azione per tuo conto.

Il tuo ruolo IAM richiede la seguente policy:

- [AWSFaultInjectionSimulatorNetworkAccess](#)— Concede l'autorizzazione al servizio AWS FIS nella rete di Amazon EC2 e in altri servizi necessari per AWS eseguire azioni FIS relative all'infrastruttura di rete.

Note

Per semplicità, questo tutorial utilizza una politica gestita. AWS Per l'uso in produzione, ti consigliamo invece di concedere solo le autorizzazioni minime necessarie per il tuo caso d'uso.

Per ulteriori informazioni su come creare un ruolo IAM, consulta [IAM roles for AWS FIS experimentals \(AWS CLI\)](#) o [Creating an IAM role \(console\)](#) nella IAM User Guide.

Un'istanza Amazon EC2 di prova

Avvia e connettiti a un'istanza Amazon EC2 di prova. Puoi utilizzare il seguente tutorial per avviare e connetterti a un'istanza Amazon EC2: [Tutorial: Guida introduttiva alle istanze Amazon EC2 Linux nella Amazon EC2 User Guide](#).

Fase 1: Creare un modello di esperimento FIS AWS

Creare il modello di esperimento utilizzando il AWS FIS Console di gestione AWS. Un modello AWS FIS è composto da azioni, obiettivi, condizioni di arresto e un ruolo sperimentale. Per ulteriori informazioni sul funzionamento dei modelli, consulta Modelli di [esperimenti per AWS FIS](#).

Prima di iniziare, assicurati di avere a portata di mano quanto segue:

- Un ruolo IAM con le autorizzazioni corrette.
- Un'istanza di Amazon EC2.
- L'ID di sottorete dell'istanza Amazon EC2.

Come creare un modello di esperimento

1. Apri la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione a sinistra, scegli Modelli di esperimenti.

3. Scegli Crea modello di esperimento.
4. Per la Fase 1, Specificate i dettagli del modello, effettuate le seguenti operazioni:
 - a. In Descrizione e nome, inserisci una descrizione per il modello, ad esempio `Amazon S3 Network Disrupt Connectivity`.
 - b. Scegliete Avanti e passate al Passaggio 2, Specificate azioni e obiettivi.
5. In Azioni, scegli Aggiungi azione.
 - a. Per il nome, inserisci `disruptConnectivity`.
 - b. Per Tipo di azione, seleziona `aws:network:disrupt-connectivity`.
 - c. In Parametri di azione, imposta la Durata su `2 minutes`
 - d. In Ambito, seleziona `s3`.
 - e. In alto, scegli Salva.
6. In Target, dovresti vedere l'obiettivo che è stato creato automaticamente. Scegli Modifica.
 - a. Verifica che il tipo di risorsa sia `aws:ec2:subnet`.
 - b. In Target method, seleziona Resource IDs, quindi scegli la sottorete utilizzata per creare l'istanza Amazon EC2 nei passaggi relativi ai [prerequisiti](#).
 - c. Verifica che la modalità di selezione sia impostata su Tutto.
 - d. Scegli Save (Salva).
7. Scegli Avanti per passare alla Fase 3, Configurazione dell'accesso al servizio.
8. In Service Access, seleziona il ruolo IAM che hai creato come descritto nei [Prerequisiti](#) per questo tutorial. Se il tuo ruolo non viene visualizzato, verifica che abbia la relazione di fiducia richiesta. Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).
9. Scegli Avanti per passare al Passaggio 4, Configura le impostazioni opzionali.
10. (Facoltativo) In Condizioni di interruzione, è possibile selezionare un CloudWatch allarme per interrompere l'esperimento se si verifica la condizione. Per ulteriori informazioni, vedere [Condizioni di arresto per AWS FIS](#).
11. (Facoltativo) In Logs, puoi selezionare un bucket Amazon S3 o inviare i log CloudWatch per il tuo esperimento.
12. Scegli Avanti per passare alla Fase 5, Rivedi e crea.
13. Rivedi il modello e scegli Crea modello di esperimento. Quando viene richiesta la conferma, inserisci `create`, quindi scegli Crea modello di esperimento.

Fase 2: Eseguire il ping di un endpoint Amazon S3

Verifica che la tua istanza Amazon EC2 sia in grado di raggiungere un endpoint Amazon S3.

1. Connect all'istanza Amazon EC2 creata nei passaggi relativi ai [prerequisiti](#).

Per la risoluzione dei problemi, consulta [Risoluzione dei problemi di connessione alla tua istanza nella Guida](#) per l'utente di Amazon EC2.

2. Verifica Regione AWS dove si trova la tua istanza. Puoi farlo nella console Amazon EC2 o eseguendo il seguente comando.

```
hostname
```

Ad esempio, se hai lanciato un'istanza Amazon EC2 in us-west-2, vedrai il seguente output.

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

3. Esegui il ping di un endpoint Amazon S3 nel tuo. Regione AWS Sostituisci *Regione AWS* con la tua regione.

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

Per quanto riguarda l'output, dovresti vedere un ping riuscito con una perdita di pacchetti pari allo 0%, come mostrato nell'esempio seguente.

```
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
```

Fase 3: Iniziate l'esperimento AWS FIS

Inizia un esperimento con il modello di esperimento che hai appena creato.

1. Aprire la console AWS FIS all'indirizzo <https://console.aws.amazon.com/fis/>.

2. Nel riquadro di navigazione a sinistra, scegli Modelli di esperimenti.
3. Seleziona l'ID del modello di esperimento che hai creato per aprirne la pagina dei dettagli.
4. Scegli Inizia un esperimento.
5. (Facoltativo) Nella pagina di conferma, aggiungi i tag per l'esperimento.
6. Nella pagina di conferma, scegli Inizia esperimento.

Fase 4: Tieni traccia dei progressi dell'esperimento AWS FIS

È possibile tenere traccia dell'avanzamento di un esperimento in corso fino al completamento, all'interruzione o al fallimento dell'esperimento.

1. Dovresti essere nella pagina dei dettagli dell'esperimento che hai appena iniziato. Se non lo sei, scegli Esperimenti, quindi seleziona l'ID dell'esperimento per aprirne la pagina dei dettagli.
2. Per visualizzare lo stato dell'esperimento, seleziona Stato nel riquadro dei dettagli. Per ulteriori informazioni, consulta [Stati dell'esperimento](#).
3. Quando lo stato dell'esperimento è In esecuzione, vai al passaggio successivo.

Fase 5: verifica l'interruzione della rete Amazon S3

Puoi convalidare l'avanzamento dell'esperimento eseguendo il ping dell'endpoint Amazon S3.

- Dalla tua istanza Amazon EC2, esegui il ping dell'endpoint Amazon S3 nel tuo. Regione AWS Sostituisci *Regione AWS* con la tua regione.

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

Per quanto riguarda l'output, dovresti vedere un ping non riuscito con perdita di pacchetti del 100%, come mostrato nell'esempio seguente.

```
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
```

Fase 5: rimozione

Se non hai più bisogno dell'istanza Amazon EC2 che hai creato per questo esperimento o del modello AWS FIS, puoi rimuoverli.

Per rimuovere l'istanza Amazon EC2

1. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, scegliere Instances (Istanze).
3. Seleziona l'istanza di test, scegli Instance state, quindi scegli Terminate instance.
4. Quando viene richiesta la conferma, seleziona Terminate (Termina).

Per eliminare il modello di esperimento utilizzando la console AWS FIS

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento, quindi scegli Azioni, Elimina modello di esperimento.
4. Quando viene richiesta la conferma, immettete `delete`, quindi scegliete Elimina modello di esperimento.

Tutorial: Pianifica un esperimento ricorrente

Con AWS AWS Fault Injection Service (FIS), puoi eseguire esperimenti di iniezione dei guasti sui tuoi AWS carichi di lavoro. Questi esperimenti vengono eseguiti su modelli che contengono una o più azioni da eseguire su obiettivi specifici. Se li utilizzi anche Amazon EventBridge, puoi pianificare gli esperimenti come attività singola o come attività ricorrenti.

Usa questo tutorial per creare una EventBridge pianificazione che esegua un modello di esperimento AWS FIS ogni 5 minuti.

Processi

- [Prerequisiti](#)
- [Fase 1: Creare un ruolo e una policy IAM](#)
- [Fase 2: Creare uno Amazon EventBridge Scheduler](#)
- [Passaggio 3: verifica l'esperimento](#)

- [Fase 4: pulizia](#)

Prerequisiti

Prima di iniziare questo tutorial, AWS è necessario disporre di un modello di esperimento FIS da eseguire secondo una pianificazione. Se hai già un modello di esperimento funzionante, prendi nota dell'ID del modello e Regione AWS. Altrimenti, puoi creare un modello seguendo le istruzioni riportate in [the section called "Arresto e avvio dell'istanza di test"](#), e poi tornare a questo tutorial.

Fase 1: Creare un ruolo e una policy IAM

Creare un ruolo e una policy IAM

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, scegli Ruoli, quindi Crea ruolo.
3. Scegli Criteri di fiducia personalizzati, quindi inserisci il frammento seguente per consentire a Amazon EventBridge Scheduler di assumere il ruolo per tuo conto.

JSON

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

Scegli Next (Successivo).

4. In Aggiungi autorizzazioni, scegli Crea politica.
5. Scegli JSON, quindi inserisci la seguente politica. Sostituisci il *your-experiment-template-id* valore con l'ID del modello del tuo esperimento indicato nei passaggi Prerequisiti.

JSON

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

È possibile limitare lo scheduler all'esecuzione solo di modelli di esperimenti AWS FIS con un valore di tag specifico. Ad esempio, la seguente politica concede l'`StartExperiment` autorizzazione per tutti gli esperimenti AWS FIS, ma limita lo scheduler all'esecuzione solo di modelli di esperimenti con tag `Purpose=Schedule`

JSON

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

```
}  
  }  
} }  
} }
```

Scegliere Next: Tags (Successivo: Tag).

6. Scegliere Next:Review (Successivo: Rivedi).
7. In Revisione della politica, assegna un nome alla politica **FIS_RecurringExperiment**, quindi scegli Crea politica.
8. In Aggiungi autorizzazioni, aggiungi la nuova FIS_RecurringExperiment politica al tuo ruolo, quindi scegli Avanti.
9. In Nome, rivedi e crea, assegna un nome al ruolo **FIS_RecurringExperiment_role**, quindi scegli Crea ruolo.

Fase 2: Creare uno Amazon EventBridge Scheduler

Per creare uno Scheduler Amazon EventBridge

1. Apri la EventBridge console Amazon all'indirizzo <https://console.aws.amazon.com/events/>.
2. Nel riquadro di navigazione a sinistra, scegli Pianificazioni.
3. Verifica di utilizzare lo stesso modello Regione AWS di esperimento AWS FIS.
4. Scegli Crea pianificazione e compila quanto segue:
 - In Nome della pianificazione, inserisci **FIS_recurring_experiment_tutorial**.
 - In Schema di pianificazione, seleziona Pianificazione ricorrente.
 - In Tipo di pianificazione, seleziona Pianificazione basata sulla tariffa.
 - In Espressione della frequenza, scegli 5 minuti.
 - In Finestra temporale flessibile, seleziona Disattivato.
 - (Facoltativo) In Intervallo di tempo, seleziona il tuo fuso orario.
 - Scegli Next (Successivo).
5. In Seleziona destinazione, scegli Tutto APIs, quindi cerca AWS FIS.
6. Scegli AWS FIS, quindi seleziona. StartExperiment

7. In Input, inserisci il seguente payload JSON. Sostituisci il *your-experiment-template-id* valore con l'ID del modello del tuo esperimento. ClientToken è un identificatore univoco per lo scheduler. In questo tutorial, stiamo usando una parola chiave contestuale consentita da Amazon EventBridge Scheduler. Per ulteriori informazioni, consulta [Aggiungere attributi di contesto](#) nella Amazon EventBridge User Guide.

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

Scegli Next (Successivo).

8. (Facoltativo) In Impostazioni, puoi configurare la politica Retry, le impostazioni DLQ (Dead-letter Queue) e Encryption. In alternativa, è possibile mantenere i valori predefiniti.
9. In Autorizzazioni, seleziona Usa ruolo esistente, quindi cerca FIS_RecurringExperiment_role.
10. Scegli Next (Successivo).
11. In Rivedi e crea pianificazione, esamina i dettagli dello scheduler, quindi scegli Crea pianificazione.

Passaggio 3: verifica l'esperimento

Per verificare che l'esperimento AWS FIS sia stato eseguito nei tempi previsti

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione a sinistra, scegli Esperimenti.
3. Cinque minuti dopo aver creato la pianificazione, dovresti vedere l'esperimento in corso.

Fase 4: pulizia

Per disattivare il tuo Amazon EventBridge Scheduler

1. Apri la EventBridge console Amazon all'indirizzo <https://console.aws.amazon.com/events/>.
2. Nel riquadro di navigazione a sinistra, scegli Pianificazioni.
3. Seleziona lo scheduler appena creato, quindi scegli Disabilita.

Utilizzo della libreria di AWS FIS scenari

Gli scenari definiscono eventi o condizioni che i clienti possono applicare per testare la resilienza delle loro applicazioni, come l'interruzione delle risorse di elaborazione su cui l'applicazione è in esecuzione. Gli scenari sono creati e di proprietà di AWS e riducono al minimo il carico di lavoro indifferenziato fornendo un gruppo di obiettivi e azioni di errore predefiniti (ad esempio, arrestando il 30% delle istanze in un gruppo di scalabilità automatica) per i problemi più comuni delle applicazioni.

Gli scenari vengono forniti tramite una libreria di scenari disponibile solo per console ed eseguiti utilizzando un modello di esperimento. AWS FIS Per eseguire un esperimento utilizzando uno scenario, selezionerai lo scenario dalla libreria, specificherai i parametri corrispondenti ai dettagli del tuo carico di lavoro e lo salverai come modello di esperimento nel tuo account.

Argomenti

- [Visualizzazione di uno scenario](#)
- [Utilizzo di uno scenario](#)
- [Esportazione di uno scenario](#)
- [Riferimento agli scenari](#)

Visualizzazione di uno scenario

Per visualizzare uno scenario utilizzando la console:

1. Apri la AWS FIS console all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Libreria Scenari.
3. Per visualizzare informazioni su uno scenario specifico, seleziona la scheda dello scenario per visualizzare un pannello diviso.
 - Nella scheda Descrizione del pannello diviso nella parte inferiore della pagina, puoi visualizzare una breve descrizione dello scenario. È inoltre disponibile un breve riepilogo dei prerequisiti contenente un riepilogo delle risorse target richieste e delle azioni da intraprendere per preparare le risorse da utilizzare con lo scenario. Infine, puoi anche visualizzare informazioni aggiuntive sugli obiettivi e sulle azioni nello scenario, nonché sulla durata prevista quando l'esperimento viene eseguito correttamente con le impostazioni predefinite.

- Nella scheda Contenuto del pannello diviso nella parte inferiore della pagina, è possibile visualizzare in anteprima una versione parzialmente popolata del modello di esperimento che verrà creato a partire dallo scenario.
- Nella scheda Dettagli del pannello diviso nella parte inferiore della pagina, puoi trovare una spiegazione dettagliata di come viene implementato lo scenario. Questo può contenere informazioni dettagliate su come vengono approssimati i singoli aspetti dello scenario. Laddove applicabile, puoi anche leggere quali metriche utilizzare come condizioni di arresto e per fornire osservabilità per imparare dall'esperimento. Infine troverai consigli su come espandere il modello di esperimento risultante.

Utilizzo di uno scenario

Per utilizzare uno scenario utilizzando la console:

1. Apri la AWS FIS console all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Libreria Scenari.
3. Per visualizzare informazioni su uno scenario specifico, seleziona la scheda dello scenario per visualizzare un pannello diviso
4. Per utilizzare lo scenario, seleziona la scheda dello scenario e scegli Crea modello con scenario.
5. Nella vista Crea modello di esperimento, inserisci gli elementi mancanti.
 - a. Alcuni scenari consentono di modificare i parametri condivisi tra più azioni o obiettivi. Questa funzionalità verrà disattivata una volta apportate modifiche allo scenario, incluse le modifiche mediante la modifica dei parametri condivisi. Per utilizzare questa funzione, seleziona il pulsante Modifica parametri condivisi. Modifica i parametri nella modalità modale e seleziona il pulsante Salva.
 - b. Alcuni modelli di esperimento possono presentare parametri di azione o bersaglio mancanti, evidenziati su ogni azione e scheda bersaglio. Seleziona il pulsante Modifica per ogni scheda, aggiungi le informazioni mancanti e seleziona il pulsante Salva sulla scheda.
 - c. Tutti i modelli richiedono un ruolo di esecuzione dell'accesso al servizio. Puoi scegliere un ruolo esistente o creare un nuovo ruolo per questo modello di esperimento.
 - d. Consigliamo di definire una o più condizioni di stop opzionali selezionando un CloudWatch allarme AWS esistente. Ulteriori informazioni su [Condizioni di arresto per la AWS FIS](#). Se non hai ancora configurato un allarme, puoi seguire le istruzioni in [Uso di Amazon CloudWatch Alarms](#) e aggiornare il modello di esperimento in un secondo momento.

- e. Ti consigliamo di abilitare i log degli esperimenti opzionali su Amazon CloudWatch Logs o su un bucket Amazon S3. Ulteriori informazioni su [Registrazione degli esperimenti per AWS FIS](#). Se non hai ancora configurato le risorse appropriate, puoi aggiornare il modello di esperimento in un secondo momento.
6. In Crea modello di esperimento, seleziona Crea modello di esperimento.
7. Dalla vista Modelli di esperimento della AWS FIS console, seleziona Avvia esperimento. Ulteriori informazioni su [Gestione dei AWS modelli di esperimento FIS](#).

Esportazione di uno scenario

Gli scenari sono un'esperienza disponibile solo su console. Sebbene simili ai modelli di esperimento, gli scenari non sono modelli di esperimento completi e non possono essere importati direttamente in essi. AWS FIS Se desideri utilizzare gli scenari come parte della tua automazione, puoi utilizzare uno dei due percorsi seguenti:

1. Segui i passaggi indicati [Utilizzo di uno scenario](#) per creare un modello di AWS FIS esperimento valido ed esportarlo.
2. Segui i passaggi descritti nel [Visualizzazione di uno scenario](#) passaggio 3, dalla scheda Contenuto, copia e salva il contenuto dello scenario, quindi aggiungi manualmente i parametri mancanti per creare un modello di esperimento valido.

Riferimento agli scenari

Gli scenari inclusi nella libreria di scenari sono progettati per utilizzare i [tag](#) laddove possibile e ogni scenario descrive i tag richiesti nelle sezioni Prerequisiti e Come funziona della descrizione dello scenario. Puoi etichettare le tue risorse con questi tag predefiniti oppure puoi impostare tag personalizzati utilizzando l'esperienza di modifica dei parametri condivisa (vedi [Utilizzo di uno scenario](#)).

Questo riferimento descrive gli scenari comuni nella libreria di scenari AWS FIS. Puoi anche elencare gli scenari supportati utilizzando la console AWS FIS.

Per ulteriori informazioni, consulta [Utilizzo della libreria di AWS FIS scenari](#).

AWS FIS supporta i seguenti scenari di Amazon EC2. [Questi scenari prendono di mira le istanze utilizzando tag](#). È possibile utilizzare tag personalizzati o utilizzare i tag predefiniti inclusi nello scenario. Alcuni di questi scenari [utilizzano documenti SSM](#).

- **Stress EC2: errore dell'istanza** - Esplora l'effetto del fallimento dell'istanza arrestando una o più istanze EC2.

Scegli come target le istanze nella regione corrente a cui è associato un tag specifico. In questo scenario interromperemo tali istanze e le riavvieremo al termine della durata dell'azione, per impostazione predefinita 5 minuti.

- **Stress EC2: disco** - Esplora l'impatto di un maggiore utilizzo del disco sulla tua applicazione basata su EC2.

In questo scenario prenderemo di mira le istanze EC2 nella regione corrente a cui è associato un tag specifico. In questo scenario è possibile personalizzare una quantità crescente di utilizzo del disco iniettato su istanze EC2 mirate per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di stress del disco.

- **Stress EC2: CPU** - Esplora l'impatto dell'aumento della CPU sulla tua applicazione basata su EC2.

In questo scenario prenderemo di mira le istanze EC2 nella regione corrente a cui è associato un tag specifico. In questo scenario è possibile personalizzare una quantità crescente di stress della CPU iniettato su istanze EC2 mirate per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di stress della CPU.

- **Stress EC2: memoria** - Esplora l'impatto di un maggiore utilizzo della memoria sulla tua applicazione basata su EC2.

In questo scenario prenderemo di mira le istanze EC2 nella regione corrente a cui è associato un tag specifico. In questo scenario è possibile personalizzare una quantità crescente di stress di memoria iniettato su istanze EC2 mirate per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di stress da memoria.

- **Stress EC2: latenza di rete** - Esplora l'impatto dell'aumento della latenza di rete sulla tua applicazione basata su EC2.

In questo scenario, prenderemo di mira le istanze EC2 nella regione corrente a cui è associato un tag specifico. In questo scenario puoi personalizzare una quantità crescente di latenza di rete iniettata su istanze EC2 mirate per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di latenza.

AWS FIS supporta i seguenti scenari Amazon EKS. Questi scenari riguardano i pod EKS che utilizzano le etichette di un'applicazione Kubernetes. Puoi utilizzare le tue etichette o utilizzare le

etichette predefinite incluse nello scenario. Per ulteriori informazioni su EKS con FIS, vedere [Azioni EKS Pod](#).

- Stress EKS: Pod Delete - Esplora l'effetto del fallimento del pod EKS eliminando uno o più pod.

In questo scenario prenderemo di mira i pod della regione corrente associati all'etichetta di un'applicazione. In questo scenario elimineremo tutti i pod corrispondenti. La ricreazione dei pod sarà controllata dalla configurazione di Kubernetes.

- Stress EKS: CPU - Esplora l'impatto dell'aumento della CPU sulla tua applicazione basata su EKS.

In questo scenario prenderemo di mira i pod della regione corrente associati a un'etichetta di applicazione. In questo scenario è possibile personalizzare una quantità crescente di stress della CPU iniettato sui pod EKS mirati per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di stress della CPU.

- Stress EKS: disco: esplorate l'impatto di un maggiore utilizzo del disco sulla vostra applicazione basata su EKS.

In questo scenario prenderemo di mira i pod della regione corrente associati a un'etichetta di applicazione. In questo scenario è possibile personalizzare una quantità crescente di stress su disco iniettato su pod EKS mirati per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di stress della CPU.

- Stress EKS: memoria - Esplora l'impatto di un maggiore utilizzo della memoria sulla tua applicazione basata su EKS.

In questo scenario prenderemo di mira i pod della regione corrente associati a un'etichetta di applicazione. In questo scenario è possibile personalizzare una quantità crescente di stress di memoria iniettato su pod EKS mirati per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di stress da memoria.

- Stress EKS: latenza di rete - Esplora l'impatto dell'aumento della latenza di rete sulla tua applicazione basata su EKS.

In questo scenario prenderemo di mira i pod della regione corrente associati a un'etichetta di applicazione. In questo scenario è possibile personalizzare una quantità crescente di latenza di rete iniettata su pod EKS mirati per la durata dell'azione, per impostazione predefinita 5 minuti per ogni azione di latenza.

AWS FIS supporta i seguenti scenari per applicazioni Single-AZ, Multi-AZ e multiregione. Questi scenari si rivolgono a più tipi di risorse.

- **AZ Availability: Power Interruption-** Inietta i sintomi attesi di un'interruzione completa dell'alimentazione in una zona di disponibilità (AZ). Ulteriori informazioni su [AZ Availability: Power Interruption](#).
- **AZ: Application Slowdown-** Aggiungi latenza tra le risorse all'interno di una singola zona di disponibilità (AZ) per rallentare un'applicazione. Ulteriori informazioni su [AZ: Application Slowdown](#).
- **Cross-AZ: Traffic Slowdown-** Inietta la perdita di pacchetti per interrompere e rallentare il traffico tra le zone di disponibilità (). AZs Ulteriori informazioni su [Cross-AZ: Traffic Slowdown](#).
- **Cross-Region: Connectivity-** Blocca il traffico di rete delle applicazioni dalla regione sperimentale alla regione di destinazione e sospendi la replica dei dati tra regioni. Scopri di più sull'utilizzo. [Cross-Region: Connectivity](#)

AWS FIS supporta i seguenti scenari per i volumi Amazon EBS. Questi scenari hanno come target i volumi utilizzando tag. È possibile utilizzare tag personalizzati o utilizzare i tag predefiniti inclusi nello scenario. I volumi di destinazione devono trovarsi nella stessa zona di disponibilità. Per ulteriori informazioni, [consulta Fault testing su Amazon EBS](#).

- **EBS: Sustained Latency—** Esplora l'impatto della I/O latenza persistente sulla tua applicazione.

In questo scenario, prenderemo di mira i volumi nell'attuale zona di disponibilità a cui è associato un tag specifico. Questo scenario inietta una latenza costante di 500 ms sul 50 per cento delle operazioni di lettura e il 100 per cento delle operazioni di scrittura per un volume, utilizzando una singola azione di latenza per un periodo di 15 minuti. In questo scenario, è possibile personalizzare la quantità di latenza iniettata, la percentuale di I/O iniezioni e la durata dell'azione.

- **EBS: Increasing Latency—** Esplora l'impatto dell'aumento della I/O latenza sulla tua applicazione.

In questo scenario, utilizzeremo come target i volumi nell'attuale zona di disponibilità a cui è associato un tag specifico. Questo scenario prevede una latenza crescente di 50 ms, 200 ms, 700 ms, 1 secondo e 15 secondi sul 10% delle operazioni di lettura e il 25% delle operazioni di scrittura per un volume utilizzando cinque azioni di latenza in un periodo di 15 minuti. In questo scenario, è possibile personalizzare la quantità di latenza iniettata, la percentuale di iniettata e la durata dell' I/O azione per ogni azione di latenza.

- **EBS: Intermittent Latency—** Esplora l'impatto dei picchi di latenza intermittenti I/O sulla tua applicazione.

In questo scenario, prenderemo di mira i volumi nell'attuale zona di disponibilità a cui è associato un tag specifico. Questo scenario prevede tre forti picchi di latenza intermittenti di 30 secondi, 10 secondi e 20 secondi sullo 0,1% delle I/O operazioni di lettura e scrittura per un volume, utilizzando tre azioni di latenza, con intervalli di ripristino tra ogni picco nell'arco di 15 minuti. In questo scenario, è possibile personalizzare la quantità di latenza iniettata, la percentuale di iniettata e la durata dell'azione per ogni azione di I/O latenza.

- EBS: Decreasing Latency— Esplora l'impatto della riduzione I/O della latenza sulla tua applicazione.

In questo scenario, utilizzeremo come target i volumi nell'attuale zona di disponibilità a cui è associato un tag specifico. Questo scenario inietta una latenza decrescente di 20 secondi, 5 secondi, 900 ms, 300 ms e 40 ms sul 10% delle operazioni di lettura e scrittura per un volume, utilizzando cinque azioni di latenza in un periodo di 15 minuti. In questo scenario, è possibile personalizzare la quantità di latenza iniettata, la percentuale di iniettata e la durata dell'azione per I/O ogni azione di latenza.

AZ Availability: Power Interruption

È possibile utilizzare lo AZ Availability: Power Interruption scenario per indurre i sintomi previsti di un'interruzione completa dell'alimentazione in una zona di disponibilità (AZ).

Questo scenario può essere utilizzato per dimostrare che le applicazioni Multi-AZ funzionano come previsto durante una singola interruzione di alimentazione AZ completa. Include la perdita di elaborazione zonale (Amazon EC2, EKS ed ECS), nessun ridimensionamento del calcolo in AZ, perdita di connettività di sottorete, failover RDS, failover ElastiCache , accesso compromesso ai bucket di directory S3 Express One Zone e volumi EBS che non rispondono. Per impostazione predefinita, le azioni per le quali non viene trovato alcun obiettivo verranno ignorate.

Azioni

Nel loro insieme, le seguenti azioni creano molti dei sintomi previsti di un'interruzione completa dell'alimentazione in una singola AZ. Disponibilità AZ: L'interruzione dell'alimentazione influisce solo sui servizi che dovrebbero subire un impatto durante una singola interruzione dell'alimentazione in AZ. Per impostazione predefinita, lo scenario inietta i sintomi di interruzione dell'alimentazione per 30 minuti e quindi, per altri 30 minuti, inietta i sintomi che possono verificarsi durante il ripristino.

Stop-Instances

Durante un'interruzione dell'alimentazione AZ, le istanze EC2 nella zona di zona interessata si spegneranno. Una volta ripristinata l'alimentazione, le istanze si riavvieranno. AZ Availability: Power Interruption include [aws:ec2:stop-instances per arrestare tutte le istanze nella AZ](#) interessata per la durata dell'interruzione. Dopo la durata, le istanze vengono riavviate. L'arresto delle istanze EC2 gestite da Amazon EKS causa l'eliminazione dei pod EKS dipendenti. L'arresto delle istanze EC2 gestite da Amazon ECS causa l'interruzione delle attività ECS dipendenti.

Questa azione si rivolge alle istanze EC2 in esecuzione nella zona di emergenza interessata. Per impostazione predefinita, si rivolge alle istanze con un tag denominato `AzImpairmentPower` con un valore di `StopInstances`. Puoi aggiungere questo tag alle tue istanze o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovate istanze valide, questa azione verrà ignorata.

Stop-ASG-Instances

Durante un'interruzione dell'alimentazione in zona di emergenza, le istanze EC2 gestite da un gruppo di Auto Scaling nella zona interessata si spegneranno. Una volta ripristinata l'alimentazione, le istanze si riavvieranno. AZ Availability: Power Interruption include [aws:ec2:stop-instances per arrestare tutte le istanze](#), incluse quelle gestite da Auto Scaling, nella zona di emergenza interessata per tutta la durata dell'interruzione. Dopo la durata, le istanze vengono riavviate.

Questa azione si rivolge alle istanze EC2 in esecuzione nella zona di zona interessata. Per impostazione predefinita, si rivolge alle istanze con un tag denominato `AzImpairmentPower` con un valore di `IceAsg`. Puoi aggiungere questo tag alle tue istanze o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovate istanze valide, questa azione verrà ignorata.

Sospendi l'avvio delle istanze

Durante un'interruzione dell'alimentazione in AZ, le chiamate API EC2 per fornire capacità nell'AZ falliranno. In particolare, APIs verranno influenzati: `ec2:StartInstances`, `ec2:CreateFleet` e `ec2:RunInstances`. AZ Availability: Power Interruption include [aws:ec2: api-insufficient-instance-capacity -error per impedire il provisioning](#) di nuove istanze nella zona di disponibilità interessata.

Questa azione si rivolge ai ruoli IAM utilizzati per il provisioning delle istanze. Questi devono essere mirati utilizzando un ARN. Per impostazione predefinita, se non vengono trovati ruoli IAM validi, questa azione verrà ignorata.

Mettili in pausa ASG Scaling

Durante un'interruzione dell'alimentazione in AZ, le chiamate API EC2 effettuate dal piano di controllo Auto Scaling per recuperare la capacità persa nella zona di disponibilità falliranno. In particolare, APIs verranno influenzati i seguenti fattori: `ec2:StartInstances`, e. `ec2:CreateFleet` `ec2:RunInstances` AZ Availability: Power Interruption include [aws:ec2: asg-insufficient-instance-capacity -error per impedire il provisioning](#) di nuove istanze nella zona di disponibilità interessata. Ciò impedisce inoltre la scalabilità di Amazon EKS e Amazon ECS nella zona interessata.

Questa azione si rivolge ai gruppi di Auto Scaling. Per impostazione predefinita, si rivolge ai gruppi di Auto Scaling con un tag denominato `AzImpairmentPower` con un valore di. `IceAsg` Puoi aggiungere questo tag ai tuoi gruppi di Auto Scaling o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati gruppi di Auto Scaling validi, questa azione verrà ignorata.

Mettili in pausa la connettività di rete

Durante un'interruzione dell'alimentazione AZ, la rete nella AZ non sarà disponibile. Quando ciò accade, alcuni servizi AWS potrebbero impiegare fino a qualche minuto per aggiornare il DNS in modo da indicare che gli endpoint privati nella zona di zona interessata non sono disponibili. Durante questo periodo, le ricerche DNS possono restituire indirizzi IP inaccessibili. AZ Availability: Power Interruption include [aws:network:disrupt-connectivity per bloccare per 2 minuti tutta la connettività](#) di rete per tutte le sottoreti nella zona di disponibilità interessata. Ciò forzerà i timeout e gli aggiornamenti del DNS per la maggior parte delle applicazioni. La fine dell'azione dopo 2 minuti consente il successivo ripristino del DNS del servizio regionale mentre l'AZ continua a non essere disponibile.

Questa azione si rivolge alle sottoreti. Per impostazione predefinita, si rivolge ai cluster con un tag denominato `AzImpairmentPower` con un valore di. `DisruptSubnet` Puoi aggiungere questo tag alle tue sottoreti o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovate sottoreti valide, questa azione verrà ignorata.

RDS di failover

Durante un'interruzione dell'alimentazione AZ, i nodi RDS nell'area AZ interessata si spegneranno. I singoli nodi RDS AZ nella zona di residenza interessata non saranno completamente disponibili. Per i cluster Multi-AZ, il nodo writer eseguirà il failover in una zona AZ non interessata e i nodi di lettura nella zona AZ interessata non saranno disponibili. Per i cluster Multi-AZ, AZ Availability: Power Interruption include [aws:rds:](#) per il failover se il writer si trova nella zona failover-db-cluster di disponibilità interessata.

Questa azione si rivolge ai cluster RDS. Per impostazione predefinita, si rivolge ai cluster con un tag denominato `AzImpairmentPower` con il valore di `DisruptRds`. Puoi aggiungere questo tag ai tuoi cluster o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati cluster validi, questa azione verrà ignorata.

Metti in pausa ElastiCache il gruppo di replica

Durante un'interruzione dell'alimentazione AZ, ElastiCache i nodi della zona non sono disponibili. **AZ Availability: Power Interruption** include [aws:elasticache: replicationgroup-interrupt-az-power](#) per terminare ElastiCache i nodi nella AZ interessata. Per tutta la durata dell'interruzione, non verranno fornite nuove istanze nella zona di disponibilità interessata, quindi il gruppo di replica rimarrà a capacità ridotta.

Questa azione è destinata ai gruppi di replica. ElastiCache Per impostazione predefinita, si rivolge ai gruppi di replica con un tag denominato `AzImpairmentPower` con il valore di `ElasticacheImpact`. È possibile aggiungere questo tag ai gruppi di replica o sostituire il tag predefinito con il proprio tag nel modello di esperimento. Per impostazione predefinita, se non vengono trovati gruppi di replica validi, questa azione verrà ignorata. Si noti che solo i gruppi di replica con nodi nella zona di disponibilità interessata verranno considerati obiettivi validi.

Avvia ARC Zonal Autoshift

Cinque minuti dopo l'inizio dell'interruzione dell'alimentazione AZ, l'azione di ripristino sposta `aws:arc:start-zonal-autoshift` automaticamente il traffico di risorse dalla zona di disponibilità specificata per i restanti 25 minuti dell'interruzione dell'alimentazione. Dopo tale durata, il traffico torna alla zona AZ originale. Tieni presente che durante una fase di emergenza reale, l'interruzione dell'alimentazione AWS rileverà la compromissione e sposterà il traffico delle risorse se è abilitato lo spostamento automatico. Sebbene la tempistica di questo turno vari, si stima che si verifichi cinque minuti dopo l'inizio della menomazione.

Questa azione è destinata alle risorse abilitate all'autoshift di Amazon Application Recovery Controller (ARC). Per impostazione predefinita, si rivolge alle risorse con il tag `key` e `value`. `AzImpairmentPower RecoverAutoshiftResources` Puoi aggiungere questo tag alle tue risorse o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Ad esempio, potresti voler usare un tag specifico per l'applicazione. Per impostazione predefinita, se non vengono trovate risorse valide, questa azione verrà ignorata.

Metti in pausa I/O EBS

Dopo un'interruzione dell'alimentazione AZ, una volta ripristinata l'alimentazione, è possibile che in una percentuale molto piccola di istanze i volumi EBS non rispondano. AZ Availability: Power Interruption include [aws:ebs:pause-io per lasciare 1 volume EBS](#) in uno stato di non risposta.

Per impostazione predefinita, vengono presi di mira solo i volumi impostati per persistere dopo la chiusura dell'istanza. Questa azione si rivolge ai volumi con un tag denominato `AzImpairmentPower` con un valore di `APIPauseVolume`. Puoi aggiungere questo tag ai tuoi volumi o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati volumi validi, questa azione verrà ignorata.

Interrompi la connettività ai bucket di directory S3 Express One Zone

Durante un'interruzione dell'alimentazione AZ, i dati memorizzati nei bucket di directory S3 Express One Zone nella zona AZ non sono accessibili. AZ Availability: Power Interruption include [aws:network:disrupt-connectivity](#) per interrompere la connettività tra le sottoreti e i bucket di directory One Zone nella zona interessata per tutta la durata dell'esperimento, con conseguenti timeout per le operazioni dell'API Zonal Endpoint Data Plane. Usa questa azione per verificare l'interruzione delle attività quando l'elaborazione è collocata insieme allo storage in una zona di zona.

Questa azione si rivolge alle sottoreti. Per impostazione predefinita, si rivolge alle sottoreti con un tag denominato `AzImpairmentPower` con il valore di `DisruptSubnet`. Puoi aggiungere questo tag alle tue sottoreti o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovate sottoreti valide, questa azione verrà ignorata.

Limitazioni

- [Questo scenario non include le condizioni di arresto](#). Le condizioni di arresto corrette per l'applicazione devono essere aggiunte al modello dell'esperimento.
- Nella zona di destinazione, i pod Amazon EKS in esecuzione su EC2 verranno terminati con nodi di lavoro EC2 e l'avvio di nuovi nodi EC2 verrà bloccato. Tuttavia, i pod Amazon EKS in esecuzione su AWS Fargate non sono supportati.
- Nella zona di distribuzione interessata, le attività di Amazon ECS in esecuzione su EC2 verranno terminate con nodi di lavoro EC2 e l'avvio di nuovi nodi EC2 verrà bloccato. Tuttavia, le attività di Amazon ECS in esecuzione su AWS Fargate non sono supportate.
- [Amazon RDS Multi-AZ](#) con due istanze DB in standby leggibili non è supportato. In questo caso, le istanze verranno chiuse, RDS eseguirà il failover e la capacità verrà immediatamente ripristinata.

nella zona di disponibilità interessata. La modalità di standby leggibile nella zona interessata rimarrà disponibile.

Requisiti

- Aggiungi l'autorizzazione richiesta al ruolo dell' AWS FIS [esperimento](#).
- I tag delle risorse devono essere applicati alle risorse che devono essere prese di mira dall'esperimento. Questi possono utilizzare la propria convenzione di etichettatura o i tag predefiniti nello scenario.

Permissions

ARC zonal autoshift utilizza un ruolo collegato al servizio IAM per eseguire lo spostamento zonale `AWSServiceRoleForZonalAutoshiftPracticeRun` per tuo conto. Questo ruolo utilizza la policy gestita da IAM. [AWSZonalAutoshiftPracticeRunSLRPolicy](#) Non è necessario creare il ruolo manualmente. Quando create un modello di esperimento dallo scenario AZ Power Interruption in Console di gestione AWS, il o un AWS SDK AWS CLI, ARC crea automaticamente il ruolo collegato al servizio. Per ulteriori informazioni, consulta [Using the service-linked role for zonal autoshift in ARC](#).

La seguente policy concede ad AWS FIS le autorizzazioni necessarie per eseguire un esperimento con lo scenario. AZ Availability: Power Interruption [Questa policy deve essere associata al ruolo dell'esperimento](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowFISExperimentLoggingActionsCloudwatch",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ],
}
```

```

    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:*:network-acl/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkAcl",
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:*:network-acl/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/managedByFIS": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkAclEntry",
        "ec2>DeleteNetworkAcl"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-acl/*",
        "arn:aws:ec2:*:*:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/managedByFIS": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateNetworkAcl",
      "Resource": "arn:aws:ec2:*:*:vpc/*"
    },
    {
      "Effect": "Allow",

```

```

    "Action": [
      "ec2:DescribeVpcs",
      "ec2:DescribeManagedPrefixLists",
      "ec2:DescribeSubnets",
      "ec2:DescribeNetworkAcls"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:ReplaceNetworkAclAssociation",
    "Resource": [
      "arn:aws:ec2:*:*:subnet/*",
      "arn:aws:ec2:*:*:network-acl/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "rds:FailoverDBCluster"
    ],
    "Resource": [
      "arn:aws:rds:*:*:cluster:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "rds:RebootDBInstance"
    ],
    "Resource": [
      "arn:aws:rds:*:*:db:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:DescribeReplicationGroups",
      "elasticache:InterruptClusterAzPower"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ]
  },
},

```

```
{
  "Sid": "TargetResolutionByTags",
  "Effect": "Allow",
  "Action": [
    "tag:GetResources"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:StartInstances",
    "ec2:StopInstances"
  ],
  "Resource": "arn:aws:ec2:*:*:instance/*"
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeInstances"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant"
  ],
  "Resource": [
    "arn:aws:kms:*:*:key/*"
  ],
  "Condition": {
    "StringLike": {
      "kms:ViaService": "ec2.*.amazonaws.com"
    },
    "Bool": {
      "kms:GrantIsForAWSResource": "true"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeVolumes"
  ]
}
```

```

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

```

Contenuto dello scenario

Il seguente contenuto definisce lo scenario. Questo codice JSON può essere salvato e utilizzato per creare un [modello di esperimento](#) utilizzando il [create-experiment-template](#) comando dell'AWS Command Line Interface (AWS CLI). Per la versione più recente dello scenario, visita la libreria di scenari nella console 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:replicationgroup",
    "resourceTags": {
        "AzImpairmentPower": "DisruptElasticache"
    },
    "selectionMode": "ALL",
    "parameters": {
        "availabilityZoneIdentifier": "us-east-1a"
    }
}
},
"actions": {
    "Pause-Instance-Launches": {
        "actionId": "aws:ec2:api-insufficient-instance-capacity-error",
        "parameters": {
            "availabilityZoneIdentifiers": "us-east-1a",
            "duration": "PT30M",
            "percentage": "100"
        },
        "targets": {
            "Roles": "IAM-role"
        }
    },
    "Pause-EBS-IO": {

```

```
    "actionId": "aws:ebs:pause-volume-io",
    "parameters": {
      "duration": "PT30M"
    },
    "targets": {
      "Volumes": "EBS-Volumes"
    },
    "startAfter": [
      "Stop-Instances",
      "Stop-ASG-Instances"
    ]
  },
  "Stop-Instances": {
    "actionId": "aws:ec2:stop-instances",
    "parameters": {
      "completeIfInstancesTerminated": "true",
      "startInstancesAfterDuration": "PT30M"
    },
    "targets": {
      "Instances": "EC2-Instances"
    }
  },
  "Pause-ASG-Scaling": {
    "actionId": "aws:ec2:asg-insufficient-instance-capacity-error",
    "parameters": {
      "availabilityZoneIdentifiers": "us-east-1a",
      "duration": "PT30M",
      "percentage": "100"
    },
    "targets": {
      "AutoScalingGroups": "ASG"
    }
  },
  "Stop-ASG-Instances": {
    "actionId": "aws:ec2:stop-instances",
    "parameters": {
      "completeIfInstancesTerminated": "true",
      "startInstancesAfterDuration": "PT30M"
    },
    "targets": {
      "Instances": "ASG-EC2-Instances"
    }
  },
  "Pause-network-connectivity": {
```

```
    "actionId": "aws:network:disrupt-connectivity",
    "parameters": {
      "duration": "PT2M",
      "scope": "all"
    },
    "targets": {
      "Subnets": "Subnet"
    }
  },
  "Failover-RDS": {
    "actionId": "aws:rds:failover-db-cluster",
    "parameters": {},
    "targets": {
      "Clusters": "RDS-Cluster"
    }
  },
  "Pause-ElastiCache": {
    "actionId": "aws:elasticache:replicationgroup-interrupt-az-power",
    "parameters": {
      "duration": "PT30M"
    },
    "targets": {
      "ReplicationGroups": "ElastiCache-Cluster"
    }
  }
},
"stopConditions": [
  {
    "source": "aws:cloudwatch:alarm",
    "value": ""
  }
],
"roleArn": "",
"tags": {
  "Name": "AZ Impairment: Power Interruption"
},
"logConfiguration": {
  "logSchemaVersion": 2
},
"experimentOptions": {
  "accountTargeting": "single-account",
  "emptyTargetResolutionMode": "skip"
},
```

```
"description": "Affect multiple resource types in a single AZ, targeting by tags and explicit ARNs, to approximate power interruption in one AZ."  
}
```

AZ: Application Slowdown

È possibile utilizzare lo scenario AZ: Application Slowdown per introdurre una latenza aggiuntiva tra le risorse all'interno di una singola zona di disponibilità (AZ). Questa latenza crea molti dei sintomi di un rallentamento dell'applicazione, un'interruzione parziale, a volte nota come errore grigio. Aggiunge latenza ai flussi di rete tra le risorse di destinazione. I flussi di rete rappresentano il traffico tra le risorse di elaborazione, ovvero i pacchetti di dati che trasportano richieste, risposte e altre comunicazioni tra server, contenitori e servizi. Lo scenario può aiutare a convalidare le configurazioni di osservabilità, regolare le soglie di allarme, scoprire la sensibilità delle applicazioni ai rallentamenti e prendere decisioni operative critiche come l'evacuazione dalla zona di emergenza.

Per impostazione predefinita, lo scenario aggiunge 200 ms di latenza al 100% dei flussi di rete tra le risorse di destinazione all'interno della AZ selezionata per una durata di 30 minuti. Puoi utilizzare la finestra di dialogo Modifica parametri condivisi nella console AWS FIS per regolare i seguenti parametri a livello di scenario, che poi si applicano alle azioni sottostanti:

- Zona di disponibilità: puoi selezionare l'AZ da compromettere nello scenario.
- Millisecondi (ms) di latenza: regolala in base alla sensibilità e alle esigenze dell'applicazione. È possibile impostare una latenza inferiore per le applicazioni più sensibili o superiore per testare la gestione del timeout, ad esempio. Prendi in considerazione l'utilizzo di multipli della latenza attuale delle applicazioni come riferimento di base.
- Percentuale dei flussi: riduci per compromettere un sottoinsieme di traffico. Ad esempio, puoi aggiungere una latenza di 200 ms che influisce sul 25% dei flussi di rete per test ancora più sottili.
- Durata: imposta la durata dell'esperimento. Puoi abbreviarlo per test più rapidi o eseguire test prolungati più lunghi. Ad esempio, imposta la durata su 2 ore per testare i meccanismi di recupero in condizioni compromesse.
- Targeting delle risorse: puoi definire le risorse target per lo scenario generale utilizzando tag (per istanze EC2 o attività ECS su EC2 o Fargate) o etichette (per i pod EKS su EC2). Puoi specificare tag ed etichette personalizzati o utilizzare i valori predefiniti forniti nello scenario. Se non desideri utilizzare tag o etichette, puoi modificare l'azione per indirizzare le risorse specificando altri parametri.

- **Personalizzazione:** se non desideri utilizzare come target le risorse EC2 o ECS, puoi lasciare le azioni con i tag predefiniti. L'esperimento non troverà alcuna risorsa da indirizzare e l'azione verrà ignorata. Tuttavia, se non si desidera utilizzare come target le risorse EKS, è necessario rimuovere completamente l'azione e il target EKS dallo scenario, poiché è necessario fornire un identificatore del cluster EKS. Per una personalizzazione ancora più granulare, puoi modificare direttamente le singole azioni nel modello di esperimento.

Azioni

Insieme, le azioni seguenti aiutano a creare molti dei sintomi di rallentamento di un'applicazione in una singola AZ introducendo una latenza aggiuntiva sui flussi di rete, che poi si propaga attraverso l'applicazione. Queste azioni vengono eseguite in parallelo, ognuna delle quali aggiunge una latenza di 200 ms per 30 minuti per impostazione predefinita. Dopo questo periodo, la latenza torna ai livelli normali. Lo scenario richiede almeno uno dei seguenti tipi di risorse per l'esecuzione: istanza EC2, task ECS o pod EKS.

Latenza di rete ECS

AZ: Application Slowdown include [aws:ecs: task-network-latency per introdurre la latenza per le attività ECS](#). L'azione si rivolge alle attività nella AZ selezionata. Per impostazione predefinita, si rivolge alle attività con un [tag](#) denominato AZApplicationSlowdown con un valore di `LatencyForECS`. Puoi sostituire il tag predefinito con il tuo o aggiungere il tag scenario alle tue attività. Se non viene trovata alcuna attività valida, questa azione verrà ignorata. Prima di eseguire un esperimento su ECS, è necessario seguire i [passaggi di configurazione per le azioni delle attività ECS](#).

Latenza di rete EKS

AZ: Application Slowdown include [aws:eks: pod-network-latency](#) per introdurre la latenza per i pod EKS. L'azione si rivolge ai pod nella AZ selezionata. Per impostazione predefinita, si rivolge ai pod all'interno di un cluster che hanno etichette con il formato `key=value`. L'etichetta predefinita fornita è `AZApplicationSlowdown=LatencyForEKS`. Puoi sostituire l'etichetta predefinita con la tua o aggiungere questa etichetta ai tuoi pod. Se non vengono trovati pod validi, questa azione verrà ignorata. Prima di eseguire un esperimento su EKS, è necessario seguire i [passaggi di configurazione per le azioni dei pod EKS](#).

Latenza di rete EC2

[AZ: Application Slowdown](#) utilizza l'azione `aws:ssm:send-command` per eseguire il documento `Network-Latency-Sources` per introdurre la latenza per le istanze EC2. `AWSFIS-Run` L'azione si rivolge alle istanze nella AZ selezionata. Per impostazione predefinita, si rivolge alle istanze con un `tag` denominato `AZApplicationSlowdown` con un valore di `LatencyForEC2`. Puoi sostituire il tag predefinito con il tuo o aggiungere questo tag alle tue istanze. Se non vengono trovate istanze valide, questa azione verrà ignorata. Prima di eseguire un esperimento su EC2 utilizzando SSM, devi [configurare l'agente AWS Systems Manager](#).

Limitazioni

- Questo scenario non include le condizioni di [arresto](#). Le condizioni di arresto corrette per l'applicazione devono essere aggiunte al modello dell'esperimento.

Requisiti

- Aggiungi le autorizzazioni richieste al ruolo [sperimentale](#) AWS FIS.
- È necessario indirizzare una o più risorse tra i seguenti 3 tipi all'interno della AZ selezionata: istanze EC2, attività ECS o pod EKS.
- Tutte le destinazioni dello scenario devono trovarsi nello stesso VPC.

Permissions

Per eseguire questo scenario è necessario un ruolo IAM con una policy di fiducia che consenta a FIS di assumere il ruolo e le policy gestite per i tipi di risorse scelti nell'esperimento: EC2, ECS ed EKS. Quando crei un modello di esperimento dallo scenario `AZ: Application Slowdown`, FIS crea il ruolo per te con la policy di fiducia e le seguenti policy gestite da AWS:

- [AWSFaultInjectionSimulatorEC2Accesso](#)
- [AWSFaultInjectionSimulatorECSAccess](#)
- [AWSFaultInjectionSimulatorEKSAccess](#)

Se utilizzi un [ruolo IAM](#) esistente per eseguire lo scenario `AZ: Application Slowdown`, puoi allegare la seguente policy per concedere ad AWS FIS le autorizzazioni necessarie:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "DescribeTasks",
    "Effect": "Allow",
    "Action": "ecs:DescribeTasks",
    "Resource": "*"
  },
  {
    "Sid": "DescribeContainerInstances",
    "Effect": "Allow",
    "Action": "ecs:DescribeContainerInstances",
    "Resource": "arn:aws:ecs:*:*:container-instance/*/*"
  },
  {
    "Sid": "DescribeInstances",
    "Effect": "Allow",
    "Action": "ec2:DescribeInstances",
    "Resource": "*"
  },
  {
    "Sid": "DescribeSubnets",
    "Effect": "Allow",
    "Action": "ec2:DescribeSubnets",
    "Resource": "*"
  },
  {
    "Sid": "DescribeCluster",
    "Effect": "Allow",
    "Action": "eks:DescribeCluster",
    "Resource": "arn:aws:eks:*:*:cluster/*"
  },
  {
    "Sid": "TargetResolutionByTags",
    "Effect": "Allow",
    "Action": "tag:GetResources",
    "Resource": "*"
  },
  {
    "Sid": "SendCommand",
    "Effect": "Allow",
    "Action": [
      "ssm:SendCommand"
    ]
  }
],
```

```

    "Resource": [
      "arn:aws:ec2:*:*:instance/*",
      "arn:aws:ssm:*:*:managed-instance/*",
      "arn:aws:ssm:*:*:document/*"
    ]
  },
  {
    "Sid": "ListCommands",
    "Effect": "Allow",
    "Action": [
      "ssm:ListCommands"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CancelCommand",
    "Effect": "Allow",
    "Action": [
      "ssm:CancelCommand"
    ],
    "Resource": "*"
  }
]
}

```

Contenuto dello scenario

Il seguente contenuto definisce lo scenario. Questo codice JSON può essere salvato e utilizzato per creare un [modello di esperimento](#) utilizzando il [create-experiment-template](#) comando dell'AWS Command Line Interface (AWS CLI). Per la versione più recente dello scenario, visita la libreria di scenari nella console FIS e vai alla scheda Contenuto.

```

{
  "tags": {
    "Name": "AZ: Application Slowdown"
  },
  "description": "Add latency between resources within a single AZ.",
  "actions": {
    "LatencyForEKS": {
      "actionId": "aws:eks:pod-network-latency",
      "parameters": {
        "delayMilliseconds": "200",
        "duration": "PT30M",

```

```

        "flowsPercent": "100",
        "interface": "DEFAULT",
        "kubernetesServiceAccount": "fis-service-account",
        "sources": "us-east-1a"
    },
    "targets": {
        "Pods": "TargetsForEKS"
    }
},
"LatencyForEC2": {
    "actionId": "aws:ssm:send-command",
    "parameters": {
        "duration": "PT30M",
        "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Latency-Sources",
        "documentParameters": "{\"DelayMilliseconds\": \"200\", \"Sources\": \"us-
east-1a\", \"Interface\": \"DEFAULT\", \"TrafficType\": \"egress\", \"DurationSeconds\":
\"1800\", \"FlowsPercent\": \"100\", \"InstallDependencies\": \"True\"}"
    },
    "targets": {
        "Instances": "TargetsForEC2"
    }
},
"LatencyForECS": {
    "actionId": "aws:ecs:task-network-latency",
    "parameters": {
        "delayMilliseconds": "200",
        "duration": "PT30M",
        "flowsPercent": "100",
        "installDependencies": "true",
        "sources": "us-east-1a",
        "useEcsFaultInjectionEndpoints": "true"
    },
    "targets": {
        "Tasks": "TargetsForECS"
    },
    "startAfter": []
}
},
"targets": {
    "TargetsForEKS": {
        "parameters": {
            "availabilityZoneIdentifier": "us-east-1a",
            "clusterIdentifier": "",

```

```
        "namespace": "default",
        "selectorType": "labelSelector",
        "selectorValue": "AZApplicationSlowdown=LatencyForEKS"
    },
    "resourceType": "aws:eks:pod",
    "selectionMode": "ALL"
},
"TargetsForEC2": {
    "filters": [
        {
            "path": "Placement.AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "resourceTags": {
        "AZApplicationSlowdown": "LatencyForEC2"
    },
    "resourceType": "aws:ec2:instance",
    "selectionMode": "ALL"
},
"TargetsForECS": {
    "filters": [
        {
            "path": "AvailabilityZone",
            "values": [
                "us-east-1a"
            ]
        }
    ],
    "resourceTags": {
        "AZApplicationSlowdown": "LatencyForECS"
    },
    "resourceType": "aws:ecs:task",
    "selectionMode": "ALL"
}
},
"experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "skip"
},
"stopConditions": [
    {
```

```
        "source": "none"
    }
  ]
}
```

Cross-AZ: Traffic Slowdown

È possibile utilizzare lo scenario Cross-AZ: Traffic Slowdown per iniettare la perdita di pacchetti per interrompere e rallentare il traffico tra le zone di disponibilità (). AZs La perdita di pacchetti compromette la comunicazione Cross-AZ, un'interruzione parziale, a volte nota come errore grigio. Inietta la perdita di pacchetti nei flussi di rete tra le risorse di destinazione. I flussi di rete rappresentano il traffico tra le risorse di elaborazione, ovvero i pacchetti di dati che trasportano richieste, risposte e altre comunicazioni tra server, contenitori e servizi. Lo scenario può aiutare a convalidare le configurazioni di osservabilità, regolare le soglie di allarme, scoprire la sensibilità e le dipendenze delle applicazioni nella comunicazione tra le AZ e prendere decisioni operative critiche come l'evacuazione dalla zona di emergenza.

Per impostazione predefinita, lo scenario inietta una perdita di pacchetti del 15% sul 100% dei flussi di rete in uscita per le risorse di destinazione della zona di attività selezionata per una durata di 30 minuti. Puoi utilizzare la finestra di dialogo Modifica parametri condivisi nella console AWS FIS per regolare i seguenti parametri a livello di scenario, che poi si applicano alle azioni sottostanti:

- **Zona di disponibilità:** puoi selezionare l'AZ da compromettere e la perdita di pacchetti verrà iniettata da quella AZ all'altra AZs all'interno della regione.
- **Perdita di pacchetti:** riduci la perdita di pacchetti per verificare eventuali interruzioni minime, ad esempio il 5%, o una percentuale superiore per testare gravi meccanismi di degradazione e ripristino della comunicazione, ad esempio il 50% o addirittura il 100% per ottenere un impatto totale sulla connettività.
- **Percentuale dei flussi:** riduci per compromettere un sottoinsieme di traffico. Ad esempio, puoi iniettare una perdita di pacchetti del 15% che incide sul 25% dei flussi di rete per test ancora più accurati.
- **Durata:** imposta la durata dell'esperimento. Puoi abbreviarlo per test più rapidi o eseguire test prolungati più lunghi. Ad esempio, imposta la durata su 2 ore per testare i meccanismi di recupero in condizioni compromesse.
- **Targeting delle risorse:** puoi definire le risorse target per lo scenario generale utilizzando tag (per istanze EC2 o attività ECS su EC2 o Fargate) o etichette (per i pod EKS su EC2). Puoi specificare tag ed etichette personalizzati o utilizzare i valori predefiniti forniti nello scenario. Se non desideri

utilizzare tag o etichette, puoi modificare l'azione per indirizzare le risorse specificando altri parametri.

- Personalizzazione: se non desideri utilizzare come target le risorse EC2 o ECS, puoi lasciare le azioni con i tag predefiniti. L'esperimento non troverà alcuna risorsa da indirizzare e l'azione verrà ignorata. Tuttavia, se non si desidera utilizzare come target le risorse EKS, è necessario rimuovere completamente l'azione e il target EKS dallo scenario, poiché è necessario fornire un identificatore del cluster EKS. Per una personalizzazione ancora più granulare, puoi modificare direttamente le singole azioni nel modello di esperimento.

Azioni

Insieme, le azioni seguenti aiutano a creare i sintomi di un rallentamento del traffico tra AZ introducendo la perdita di pacchetti nelle comunicazioni in uscita dalla AZ di destinazione ad altre AZ della regione a livello di rete. Queste azioni vengono eseguite in parallelo, ognuna delle quali comporta una perdita di pacchetti del 15% per 30 minuti per impostazione predefinita. Dopo questo periodo, la comunicazione torna alla normalità. Lo scenario richiede almeno uno dei seguenti tipi di risorse nella zona di disponibilità selezionata per essere eseguito: istanza EC2, task ECS o pod EKS.

Perdita di pacchetti di rete ECS

Cross-AZ: Traffic Slowdown include [aws:ecs: task-network-packet-loss per iniettare la perdita di pacchetti per le attività ECS](#). L'azione si rivolge alle attività nella AZ selezionata e ne compromette la comunicazione in uscita con tutte le altre nella Regione. AZs È possibile personalizzare ulteriormente l'ambito dell'impatto modificando l'azione e aggiungendo o rimuovendo AZs dal Sources campo. Per impostazione predefinita, assegna alle attività un [tag](#) denominato `CrossAZTrafficSlowdown` con un valore di `PacketLossForECS`. Puoi sostituire il tag predefinito con il tuo o aggiungere il tag scenario alle tue attività. Se non viene trovata alcuna attività valida, questa azione verrà ignorata. Prima di eseguire un esperimento su ECS, è necessario seguire i [passaggi di configurazione per le azioni delle attività ECS](#).

Perdita di pacchetti di rete EKS

Cross-AZ: Traffic Slowdown include [aws:eks: pod-network-packet-loss per iniettare la perdita di pacchetti](#) per i pod EKS. L'azione si rivolge ai pod nella AZ selezionata e ne compromette la comunicazione in uscita con tutti gli altri nella regione. AZs Puoi personalizzare ulteriormente l'ambito dell'impatto modificando l'azione e aggiungendola o rimuovendola AZs dal campo. Sources Per impostazione predefinita, ha come target i pod all'interno di un cluster che hanno etichette con il formato `key=value`. L'etichetta predefinita fornita è. `CrossAZTraffic=PacketLossForEKS`

Puoi sostituire l'etichetta predefinita con la tua o aggiungere questa etichetta ai tuoi pod. Se non vengono trovati pod validi, questa azione verrà ignorata. Prima di eseguire un esperimento su EKS, è necessario seguire i [passaggi di configurazione per le azioni dei pod EKS](#).

Perdita di pacchetti di rete EC2

Cross-AZ: Traffic Slowdown utilizza l'azione [aws:ssm:send-command per eseguire il documento AWSFIS-Run -Network-Packet-Loss-Sources](#) per iniettare la perdita di pacchetti per le istanze EC2 e compromettere la comunicazione in uscita con tutte le altre istanze della regione. AZs Puoi personalizzare ulteriormente l'ambito dell'impatto modificando l'azione e aggiungendo AZs o Sources rimuovendo dal campo. L'azione si rivolge alle istanze nella AZ selezionata. Per impostazione predefinita, si rivolge alle istanze con un [tag](#) denominato CrossAZTrafficSlowdown con un valore di PacketLossForEC2. Puoi sostituire il tag predefinito con il tuo o aggiungere questo tag alle tue istanze. Se non vengono trovate istanze valide, questa azione verrà ignorata. Prima di eseguire un esperimento su EC2 utilizzando SSM, devi [configurare l'agente AWS Systems Manager](#).

Limitazioni

- Questo scenario non include le condizioni di [arresto](#). Le condizioni di arresto corrette per l'applicazione devono essere aggiunte al modello dell'esperimento.

Requisiti

- Aggiungi le autorizzazioni richieste al ruolo [sperimentale](#) AWS FIS.
- È necessario indirizzare una o più risorse tra i seguenti 3 tipi all'interno della AZ selezionata: istanze EC2, attività ECS o pod EKS.
- Tutte le destinazioni dello scenario devono trovarsi nello stesso VPC.

Permissions

Per eseguire questo scenario è necessario un ruolo IAM con una policy di fiducia che consenta a FIS di assumere il ruolo e le policy gestite per i tipi di risorse scelti nell'esperimento: EC2, ECS ed EKS. Quando crei un modello di esperimento dallo scenario Cross-AZ: Traffic Slowdown, FIS crea il ruolo per te con la policy di fiducia e le seguenti policy gestite da AWS:

- [AWSFaultInjectionSimulatorEC2Accesso](#)
- [AWSFaultInjectionSimulatorECSAccess](#)

- [AWSFaultInjectionSimulatorEKSAccess](#)

Se utilizzi un [ruolo IAM](#) esistente per eseguire lo scenario Cross-AZ: Traffic Slowdown, puoi allegare la seguente policy per concedere ad AWS FIS le autorizzazioni necessarie:

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

```

    },
    {
      "Sid": "SendCommand",
      "Effect": "Allow",
      "Action": [
        "ssm:SendCommand"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/*"
      ]
    },
    {
      "Sid": "ListCommands",
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommands"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CancelCommand",
      "Effect": "Allow",
      "Action": [
        "ssm:CancelCommand"
      ],
      "Resource": "*"
    }
  ]
}

```

Contenuto dello scenario

Il seguente contenuto definisce lo scenario. Questo codice JSON può essere salvato e utilizzato per creare un [modello di esperimento](#) utilizzando il [create-experiment-template](#) comando dell'AWS Command Line Interface (AWS CLI). Per la versione più recente dello scenario, visita la libreria di scenari nella console FIS e vai alla scheda Contenuto.

```

{
  "tags": {
    "Name": "Cross-AZ: Traffic Slowdown"
  },

```

```

"description": "Inject packet loss to disrupt and slow down traffic between AZs.",
"actions": {
  "PacketLossForEC2": {
    "actionId": "aws:ssm:send-command",
    "parameters": {
      "duration": "PT30M",
      "documentArn": "arn:aws:ssm:us-east-1::document/AWSFIS-Run-Network-
Packet-Loss-Sources",
      "documentParameters": "{\"Sources\": \"us-east-1b,us-east-1c,us-
east-1d,us-east-1e,us-east-1f\", \"LossPercent\": \"15\", \"Interface\": \"DEFAULT\",
\"TrafficType\": \"egress\", \"DurationSeconds\": \"1800\", \"FlowsPercent\": \"100\",
\"InstallDependencies\": \"True\"}"
    },
    "targets": {
      "Instances": "TargetsForEC2"
    }
  },
  "PacketLossForECS": {
    "actionId": "aws:ecs:task-network-packet-loss",
    "parameters": {
      "sources": "us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f",
      "lossPercent": "15",
      "duration": "PT30M",
      "flowsPercent": "100",
      "installDependencies": "true",
      "useEcsFaultInjectionEndpoints": "true"
    },
    "targets": {
      "Tasks": "TargetsForECS"
    }
  },
  "PacketLossForEKS": {
    "actionId": "aws:eks:pod-network-packet-loss",
    "parameters": {
      "sources": "us-east-1b,us-east-1c,us-east-1d,us-east-1e,us-east-1f",
      "lossPercent": "15",
      "duration": "PT30M",
      "flowsPercent": "100",
      "interface": "DEFAULT",
      "kubernetesServiceAccount": "fis-service-account"
    },
    "targets": {
      "Pods": "TargetsForEKS"
    }
  }
}

```

```
    }
  },
  "targets": {
    "TargetsForEC2": {
      "filters": [
        {
          "path": "Placement.AvailabilityZone",
          "values": [
            "us-east-1a"
          ]
        }
      ],
      "resourceTags": {
        "CrossAZTrafficSlowdown": "PacketLossForEC2"
      },
      "resourceType": "aws:ec2:instance",
      "selectionMode": "ALL"
    },
    "TargetsForECS": {
      "filters": [
        {
          "path": "AvailabilityZone",
          "values": [
            "us-east-1a"
          ]
        }
      ],
      "resourceTags": {
        "CrossAZTrafficSlowdown": "PacketLossForECS"
      },
      "resourceType": "aws:ecs:task",
      "selectionMode": "ALL"
    },
    "TargetsForEKS": {
      "parameters": {
        "availabilityZoneIdentifier": "us-east-1a",
        "clusterIdentifier": "",
        "namespace": "default",
        "selectorType": "labelSelector",
        "selectorValue": "CrossAZTrafficSlowdown=PacketLossForEKS"
      },
      "resourceType": "aws:eks:pod",
      "selectionMode": "ALL"
    }
  }
}
```

```
  },
  "experimentOptions": {
    "accountTargeting": "single-account",
    "emptyTargetResolutionMode": "skip"
  },
  "stopConditions": [
    {
      "source": "none"
    }
  ]
}
```

Cross-Region: Connectivity

Puoi utilizzare Cross-Region: Connectivity lo scenario per bloccare il traffico di rete dell'applicazione dalla regione dell'esperimento alla regione di destinazione e sospendere la replica tra regioni per le tabelle globali multiregione di Amazon S3 e Amazon DynamoDB. Interregione: la connettività influisce sul traffico delle applicazioni in uscita dalla regione in cui viene eseguito l'esperimento (regione sperimentale). Il traffico in entrata senza stato proveniente dalla regione che desideri isolare dalla regione dell'esperimento (regione di destinazione) potrebbe non essere bloccato. Il traffico proveniente dai servizi gestiti AWS potrebbe non essere bloccato.

Questo scenario può essere utilizzato per dimostrare che le applicazioni multiregionali funzionano come previsto quando le risorse nella regione di destinazione non sono accessibili dalla regione sperimentale. Include il blocco del traffico di rete dalla regione dell'esperimento alla regione di destinazione prendendo di mira i gateway di transito e le tabelle delle rotte. Inoltre, sospende la replica tra regioni per le tabelle globali S3 e DynamoDB. Per impostazione predefinita, le azioni per le quali non viene trovato alcun obiettivo verranno ignorate.

Azioni

Insieme, le seguenti azioni bloccano la connettività tra regioni per i servizi AWS inclusi. Le azioni vengono eseguite in parallelo. Per impostazione predefinita, lo scenario blocca il traffico per 3 ore, che è possibile aumentare fino a una durata massima di 12 ore.

Interrompi la connettività Transit Gateway

Cross Region: Connectivity include [aws:network: transit-gateway-disrupt-cross -region-connectivity](#) per bloccare il traffico di rete interregionale dalla regione dell'esperimento alla VPCs regione di destinazione collegata da un gateway VPCs di transito. Ciò non influisce sull'accesso agli endpoint

VPC all'interno della regione dell'esperimento, ma bloccherà il traffico proveniente dalla regione dell'esperimento destinato a un endpoint VPC nella regione di destinazione.

Questa azione ha come obiettivo i gateway di transito che collegano la regione dell'esperimento e la regione di destinazione. Per impostazione predefinita, ha come target i gateway di transito con un [tag](#) denominato `DisruptTransitGateway` con un valore di `Allowed`. Puoi aggiungere questo tag ai tuoi gateway di transito o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati gateway di transito validi, questa azione verrà ignorata.

Interrompi la connettività della sottorete

Cross Region: Connectivity include [aws:network: route-table-disrupt-cross -region-connectivity](#) per bloccare il traffico di rete interregionale dalle VPCs regione dell'esperimento ai blocchi IP AWS pubblici nella regione di destinazione. Questi blocchi IP pubblici includono gli endpoint dei servizi AWS nella regione di destinazione, ad esempio l'endpoint regionale S3, e i blocchi IP AWS per i servizi gestiti, ad esempio gli indirizzi IP utilizzati per i sistemi di bilanciamento del carico e Amazon API Gateway. Questa azione blocca anche la connettività di rete tramite connessioni peering VPC interregionali dalla regione dell'esperimento alla regione di destinazione. Non influisce sull'accesso agli endpoint VPC nella regione dell'esperimento, ma bloccherà il traffico proveniente dalla regione dell'esperimento destinato a un endpoint VPC nella regione di destinazione.

Questa azione si rivolge alle sottoreti nella regione dell'esperimento. Per impostazione predefinita, si rivolge alle sottoreti con un [tag](#) denominato `DisruptSubnet` con un valore di `Allowed`. Puoi aggiungere questo tag alle tue sottoreti o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovate sottoreti valide, questa azione verrà ignorata.

Interrompi la connettività degli endpoint VPC

Cross Region: Connectivity include [aws:network: disrupt-vpc-endpoint](#) interrompe la connettività a un servizio associato agli endpoint VPC di destinazione. Ad esempio, se un endpoint VPC crea un collegamento privato a `com.amazonaws.us-east-1.ec2`, la connettività a quel servizio verrà interrotta.

Questa azione si rivolge agli endpoint VPC nella regione dell'esperimento. Per impostazione predefinita, ha come target gli endpoint VPC di interfaccia con un [tag](#) denominato `DisruptVpcEndpoint` con un valore di `Allowed`. Puoi aggiungere questo tag ai tuoi endpoint VPC o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati endpoint VPC validi, questa azione verrà ignorata.

Mettili in pausa la replica S3

Cross Region: Connectivity include [aws:s3: bucket-pause-replication](#) per sospendere la replica S3 dalla regione dell'esperimento alla regione di destinazione per i bucket mirati. La replica dalla regione di destinazione alla regione dell'esperimento non sarà influenzata. Al termine dello scenario, la replica del bucket riprenderà dal punto in cui era stata messa in pausa. Tieni presente che il tempo necessario alla replica per mantenere sincronizzati tutti gli oggetti varierà in base alla durata dell'esperimento e alla velocità di caricamento degli oggetti nel bucket.

Questa azione riguarda i bucket S3 nella regione dell'esperimento con la [replica tra regioni \(CRR\) abilitata su un bucket S3 nella regione](#) di destinazione. [Per impostazione predefinita, si rivolge ai bucket con un tag denominato con un valore di](#). `DisruptS3 Allowed` Puoi aggiungere questo tag ai tuoi bucket o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati bucket validi, questa azione verrà ignorata.

Sospendi la replica di DynamoDB

Cross-Region: Connectivity include [aws:dynamodb: global-table-pause-replication](#) per sospendere la replica tra la regione dell'esperimento e tutte le altre regioni, inclusa la regione di destinazione. Ciò impedisce la replica all'interno e all'esterno della regione dell'esperimento ma non influisce sulla replica tra altre regioni. Al termine dello scenario, la replica della tabella riprenderà dal punto in cui era stata messa in pausa. Tieni presente che il tempo necessario alla replica per mantenere sincronizzati tutti i dati varierà in base alla durata dell'esperimento e alla frequenza delle modifiche alla tabella.

Questa azione ha come obiettivo principale sia le tabelle globali multiregionali di DynamoDB che, alla fine, le tabelle globali coerenti nella regione dell'esperimento. Per impostazione predefinita, si rivolge alle tabelle con un [tag](#) denominato `DisruptDynamoDb` con un valore di `Allowed` Puoi aggiungere questo tag alle tue tabelle o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovate tabelle globali valide, questa azione verrà ignorata.

Mettili in pausa la replica multiregionale di MemoryDB

Cross-Region: Connectivity include [aws:memorydb: multi-region-cluster-pause -replication per sospendere la replica](#) dal cluster membro regionale nella regione dell'esperimento al resto dei cluster nel cluster multiregionale di destinazione. La replica tra altri cluster membri regionali non sarà influenzata. Al termine dello scenario, la replica riprenderà dal punto in cui era stata messa in pausa. Tieni presente che il tempo impiegato per la replica per sincronizzare i dati tra i cluster membri varierà in base alla durata dell'esperimento e alla velocità di scrittura dei dati nei cluster.

Questa azione si rivolge ai cluster MemoryDB Multi-Region con un membro regionale nella regione dell'esperimento. Per impostazione predefinita, si rivolge ai cluster multiregionali con un [tag](#) denominato con un valore di `DisruptMemoryDB Allowed`. Puoi aggiungere questo tag ai tuoi cluster multiregionali o sostituire il tag predefinito con il tuo tag nel modello dell'esperimento. Per impostazione predefinita, se non vengono trovati cluster validi, questa azione verrà ignorata.

Limitazioni

- Questo scenario non include le condizioni di [arresto](#). Le condizioni di arresto corrette per l'applicazione devono essere aggiunte al modello dell'esperimento.

Requisiti

- Aggiungi l'autorizzazione richiesta al [ruolo sperimentale](#) AWS FIS.
- I tag delle risorse devono essere applicati alle risorse che devono essere prese di mira dall'esperimento. Questi possono utilizzare la propria convenzione di etichettatura o i tag predefiniti nello scenario.

Permissions

La seguente policy concede ad AWS FIS le autorizzazioni necessarie per eseguire un esperimento con lo scenario. Cross-Region: Connectivity [Questa policy deve essere associata al ruolo dell'esperimento](#).

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

```
    "Sid": "RouteTableDisruptConnectivity2",
    "Effect": "Allow",
    "Action": "ec2:CreateRouteTable",
    "Resource": "arn:aws:ec2:*:*:vpc/*"
  },
  {
    "Sid": "RouteTableDisruptConnectivity21",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:route-table/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateRouteTable",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity3",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateNetworkInterface",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity4",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:prefix-list/*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": "CreateManagedPrefixList",
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity5",
    "Effect": "Allow",
```

```

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

```

```

    "Effect": "Allow",
    "Action": "ec2:DeleteNetworkInterface",
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity10",
    "Effect": "Allow",
    "Action": "ec2:CreateManagedPrefixList",
    "Resource": "arn:aws:ec2:*:*:prefix-list/*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "RouteTableDisruptConnectivity11",
    "Effect": "Allow",
    "Action": [
      "ec2:DeleteManagedPrefixList",
      "ec2:ModifyManagedPrefixList"
    ],
    "Resource": "arn:aws:ec2:*:*:prefix-list/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "EC2DescribeResources",
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeVpcs",
      "ec2:DescribeVpcPeeringConnections",
      "ec2:DescribeManagedPrefixLists",
      "ec2:DescribeSubnets",
      "ec2:DescribeRouteTables",

```

```

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

```

```

    "Sid": "RouteTableDisruptConnectivity18",
    "Effect": "Allow",
    "Action": "ec2:DisassociateRouteTable",
    "Resource": "arn:aws:ec2:*:*:subnet/*"
  },
  {
    "Sid": "RouteTableDisruptConnectivity19",
    "Effect": "Allow",
    "Action": "ec2:ModifyVpcEndpoint",
    "Resource": "arn:aws:ec2:*:*:route-table/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/managedByFIS": "true"
      }
    }
  },
  {
    "Sid": "TransitGatewayDisruptConnectivity1",
    "Effect": "Allow",
    "Action": [
      "ec2:DisassociateTransitGatewayRouteTable",
      "ec2:AssociateTransitGatewayRouteTable"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:transit-gateway-route-table/*",
      "arn:aws:ec2:*:*:transit-gateway-attachment/*"
    ]
  },
  {
    "Sid": "S3CrossRegion1",
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  },
  {
    "Sid": "S3CrossRegion3",
    "Effect": "Allow",
    "Action": "s3:PauseReplication",
    "Resource": "arn:aws:s3::*:*",
    "Condition": {
      "StringLike": {
        "s3:DestinationRegion": "*"
      }
    }
  }
}

```

```

    },
    {
      "Sid": "S3CrossRegion4",
      "Effect": "Allow",
      "Action": [
        "s3:GetReplicationConfiguration",
        "s3:PutReplicationConfiguration"
      ],
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "BoolIfExists": {
          "s3:isReplicationPauseRequest": "true"
        }
      }
    },
    {
      "Sid": "DynamoDbPauseReplication",
      "Effect": "Allow",
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:PutResourcePolicy",
        "dynamodb:GetResourcePolicy",
        "dynamodb>DeleteResourcePolicy"
      ],
      "Resource": [
        "arn:aws:dynamodb:*:*:table/*"
      ]
    },
    {
      "Sid": "DynamoDbMrscPauseReplication",
      "Effect": "Allow",
      "Action": [
        "dynamodb:InjectError"
      ],
      "Resource": ["*"]
    },
    {
      "Sid": "ResolveResourcesViaTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "MemDbCrossRegion",

```

```

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

```

```

    {
      "Sid": "DisruptVPCE4",
      "Effect": "Allow",
      "Action": "vpce:AllowMultiRegion",
      "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*"
    },
    {
      "Sid": "ModifyVPCE",
      "Effect": "Allow",
      "Action": "ec2:ModifyVpcEndpoint",
      "Resource": [
        "arn:aws:ec2:*:*:vpc-endpoint/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    }
  ]
}

```

Contenuto dello scenario

Il seguente contenuto definisce lo scenario. Questo codice JSON può essere salvato e utilizzato per creare un [modello di esperimento](#) utilizzando il [create-experiment-template](#) comando dell'AWS Command Line Interface (AWS CLI). Per la versione più recente dello scenario, visita la libreria di scenari nella console 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": {}
    },
    "VPC-Endpoint": {

```

```

        "resourceType": "aws:ec2:vpc-endpoint",
        "resourceTags": {
            "DisruptPrivateLink": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "S3-Bucket": {
        "resourceType": "aws:s3:bucket",
        "resourceTags": {
            "S3Impact": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "DynamoDB-Global-Table": {
        "resourceType": "aws:dynamodb:global-table",
        "resourceTags": {
            "DisruptDynamoDb": "Allowed"
        },
        "selectionMode": "ALL"
    },
    "MemoryDB-Multi-Region-Cluster": {
        "resourceType": "aws:memorydb:multi-region-cluster",
        "resourceTags": {
            "DisruptMemoryDb": "Allowed"
        },
        "selectionMode": "ALL"
    }
},
"actions": {
    "Disrupt-Transit-Gateway-Connectivity": {
        "actionId": "aws:network:transit-gateway-disrupt-cross-region-
connectivity",
        "parameters": {
            "duration": "PT3H",
            "region": "eu-west-1"
        },
        "targets": {
            "TransitGateways": "Transit-Gateway"
        }
    },
    "Disrupt-Subnet-Connectivity": {
        "actionId": "aws:network:route-table-disrupt-cross-region-
connectivity",
        "parameters": {

```

```

        "duration": "PT3H",
        "region": "eu-west-1"
    },
    "targets": {
        "Subnets": "Subnet"
    }
},
"Disrupt-Vpc-Endpoint": {
    "actionId": "aws:network:disrupt-vpc-endpoint",
    "parameters": {
        "duration": "PT3H"
    },
    "targets": {
        "VPCEndpoints": "VPC-Endpoint"
    }
},
"Pause-S3-Replication": {
    "actionId": "aws:s3:bucket-pause-replication",
    "parameters": {
        "duration": "PT3H",
        "region": "eu-west-1"
    },
    "targets": {
        "Buckets": "S3-Bucket"
    }
},
"Pause-DynamoDB-Replication": {
    "actionId": "aws:dynamodb:global-table-pause-replication",
    "parameters": {
        "duration": "PT3H"
    },
    "targets": {
        "Tables": "DynamoDB-Global-Table"
    }
},
"Pause-MemoryDB-Multi-Region-Cluster-Replication": {
    "actionId": "aws:memorydb:multi-region-cluster-pause-replication",
    "parameters": {
        "duration": "PT3H",
        "region": "eu-west-1"
    },
    "targets": {
        "MultiRegionClusters": "MemoryDB-Multi-Region-Cluster"
    }
}

```

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

Utilizzo di esperimenti con più account per AWS FIS

Puoi creare e gestire modelli di esperimenti con più account utilizzando la AWS FIS console o la riga di comando. Puoi creare un esperimento con più account specificando l'opzione Account Targeting Experiment come "multi-account" e aggiungendo configurazioni di account di destinazione. Dopo aver creato un modello di esperimento con più account, puoi utilizzarlo per eseguire un esperimento.

Con un esperimento con più account, puoi configurare ed eseguire scenari di errore reali su un'applicazione che si estende su più AWS account all'interno di una regione. Esegui esperimenti su più account da un account orchestrator che influiscono sulle risorse di più account di destinazione.

Quando esegui un esperimento con più account, gli account target con risorse interessate riceveranno una notifica tramite i rispettivi dashboard di AWS Health, in modo da informare gli utenti degli account target. Con gli esperimenti su più account, puoi:

- Esegui scenari di errore reali su applicazioni che si estendono su più account con i controlli e i guardrail centralizzati che forniscono. AWS FIS
- Controlla gli effetti di un esperimento con più account utilizzando i ruoli IAM con autorizzazioni e tag dettagliati per definire l'ambito di ogni obiettivo.
- Visualizza centralmente le azioni AWS FIS intraprese in ogni account dai e attraverso i log. Console di gestione AWS AWS FIS
- Monitora e verifica le chiamate AWS FIS API effettuate in ogni account con AWS CloudTrail.

Questa sezione ti aiuta a iniziare con esperimenti su più account.

Argomenti

- [Concetti per esperimenti con più account](#)
- [Le migliori pratiche per esperimenti con più account](#)
- [Prerequisiti per esperimenti con più account](#)
- [Crea un modello di esperimento con più account](#)
- [Aggiorna la configurazione di un account di destinazione](#)
- [Eliminare la configurazione di un account di destinazione](#)

Concetti per esperimenti con più account

Di seguito sono riportati i concetti chiave per gli esperimenti con più account:

- **Account Orchestrator** - L'account orchestrator funge da account centrale per configurare e gestire l'esperimento nella AWS FIS console, nonché per centralizzare la registrazione. L'account orchestrator possiede il modello e l'esperimento dell'esperimento. AWS FIS
- **Account Target** - Un account Target è un account AWS individuale con risorse che possono essere influenzate da un esperimento con AWS FIS più account.
- **Configurazioni degli account Target** - È possibile definire gli account di destinazione che fanno parte di un esperimento aggiungendo le configurazioni degli account di destinazione al modello dell'esperimento. La configurazione di un account di destinazione è un elemento del modello di esperimento necessario per gli esperimenti con più account. Ne definisci uno per ogni account di destinazione impostando un ID AWS account, un ruolo IAM e una descrizione opzionale.

Le migliori pratiche per esperimenti con più account

Di seguito sono riportate le best practice per l'utilizzo di esperimenti con più account:

- Quando configuri gli obiettivi per esperimenti con più account, ti consigliamo di utilizzare tag di risorsa coerenti su tutti gli account di destinazione. Un AWS FIS esperimento risolverà le risorse con tag coerenti in ogni account di destinazione. Un'azione deve risolvere almeno una risorsa target in qualsiasi account di destinazione o avrà esito negativo, ad eccezione degli esperimenti con `emptyTargetResolutionMode` set to `skip`. Le quote di azione si applicano per account. Se desideri indirizzare le risorse per risorsa ARNs, si applica lo stesso limite per azione di un singolo account.
- Quando si assegnano risorse in una o più zone di disponibilità utilizzando parametri o filtri, è necessario specificare un ID AZ, non un nome AZ. L'ID AZ è un identificatore univoco e coerente per una zona di disponibilità tra gli account. Per sapere come trovare l'ID AZ per le zone di disponibilità nel tuo account, consulta [Zona di disponibilità IDs per le tue risorse AWS](#).

Prerequisiti per esperimenti con più account

Per utilizzare le condizioni di interruzione per un esperimento con più account, devi prima configurare gli allarmi tra account. I ruoli IAM vengono definiti quando crei un modello di esperimento con più account. Puoi creare i ruoli IAM necessari prima di creare il modello.

Contenuti

- [Autorizzazioni per esperimenti con più account](#)
- [Condizioni di interruzione degli esperimenti con più account \(opzionale\)](#)
- [Leve di sicurezza per esperimenti con più account \(opzionale\)](#)

Autorizzazioni per esperimenti con più account

Gli esperimenti su più account utilizzano il concatenamento dei ruoli IAM per concedere le autorizzazioni necessarie AWS FIS per intraprendere azioni sulle risorse negli account target. Per gli esperimenti su più account, configuri i ruoli IAM in ogni account di destinazione e nell'account orchestrator. Questi ruoli IAM richiedono una relazione di fiducia tra gli account di destinazione e l'account orchestrator e tra l'account orchestrator e AWS FIS.

I ruoli IAM per gli account di destinazione contengono le autorizzazioni necessarie per agire sulle risorse e vengono creati per un modello di esperimento aggiungendo configurazioni di account di destinazione. Creerai un ruolo IAM per l'account orchestrator con l'autorizzazione ad assumere i ruoli degli account di destinazione e stabilire una relazione di fiducia con AWS FIS. Questo ruolo IAM viene utilizzato come modello `roleArn` di esperimento.

Per ulteriori informazioni sul concatenamento dei [ruoli, consulta Roles Terms and concepts](#), nella Guida per l'utente di IAM.

Nell'esempio seguente, imposterai le autorizzazioni per un account orchestratore A per eseguire un esperimento con `aws:ecs:pause-volume-io` l'account di destinazione B.

1. Nell'account B, crea un ruolo IAM con le autorizzazioni necessarie per eseguire l'azione. Per le autorizzazioni richieste per ogni azione, consulta [Riferimento alle azioni](#). L'esempio seguente mostra le autorizzazioni concesse da un account di destinazione per eseguire l'azione EBS Pause Volume IO. [the section called "aws:ecs:pause-volume-io"](#)

JSON

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

```

    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:PauseVolumeIO"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "tag:GetResources"
    ],
    "Resource": "*"
  }
]
}

```

- Successivamente, aggiungi una politica di fiducia nell'account B che crei una relazione di fiducia con l'account A. Scegli un nome per il ruolo IAM per l'account A, che creerai nel passaggio 3.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "AccountIdA"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "sts:ExternalId":
            "arn:aws:fis:region:accountIdA:experiment/*"
        },
        "ArnEquals": {
          "aws:PrincipalArn":
            "arn:aws:iam:111122223333:role/role_name"
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

3. Nell'account A, crea un ruolo IAM. Questo nome di ruolo deve corrispondere al ruolo specificato nella politica di fiducia nel passaggio 2. Per scegliere come target più account, concedi all'orchestrator le autorizzazioni per assumere ogni ruolo. L'esempio seguente mostra le autorizzazioni per l'account A ad assumere l'account B. Se disponi di account target aggiuntivi, aggiungerai un ruolo aggiuntivo a questa politica. ARNs Puoi avere un solo ruolo ARN per account di destinazione.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::111122223333:role/role_name"
      ]
    }
  ]
}

```

4. Questo ruolo IAM per l'account A viene utilizzato come modello di esperimento. `roleArn`
L'esempio seguente mostra la policy di fiducia richiesta nel ruolo IAM che concede AWS FIS le autorizzazioni per assumere l'account A, l'account orchestrator.

JSON

```

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

```

```
    }  
  ]  
}
```

Puoi anche utilizzare Stacksets per fornire più ruoli IAM contemporaneamente. Per utilizzarlo CloudFormation StackSets, dovrai impostare le StackSet autorizzazioni necessarie nei tuoi account. AWS Per ulteriori informazioni, consulta [Working with AWS CloudFormation StackSets](#).

Condizioni di interruzione degli esperimenti con più account (opzionale)

Una condizione di arresto è un meccanismo per interrompere un esperimento se raggiunge una soglia definita come allarme. Per impostare una condizione di interruzione per un esperimento con più account, puoi utilizzare allarmi tra più account. È necessario abilitare la condivisione in ogni account di destinazione per rendere l'allarme disponibile all'account orchestrator utilizzando le autorizzazioni di sola lettura. Una volta condivisa, puoi combinare le metriche di diversi account di destinazione utilizzando Metric Math. Quindi, puoi aggiungere questo allarme come condizione di arresto dell'esperimento.

Per ulteriori informazioni sui dashboard tra account, consulta [Attivazione della funzionalità tra account in](#). CloudWatch

Leve di sicurezza per esperimenti con più account (opzionale)

Le leve di sicurezza vengono utilizzate per interrompere tutti gli esperimenti in corso e impedire l'avvio di nuovi esperimenti. È possibile utilizzare la leva di sicurezza per impedire gli esperimenti FIS in determinati periodi di tempo o in risposta agli allarmi sullo stato dell'applicazione. Ogni AWS account ha una leva di sicurezza per. Regione AWS Quando viene inserita una leva di sicurezza, influisce su tutti gli esperimenti eseguiti nello stesso account e nella stessa regione della leva di sicurezza. Per interrompere e prevenire gli esperimenti con più account, la leva di sicurezza deve essere inserita nello stesso account e nella stessa regione in cui sono in corso gli esperimenti.

Crea un modello di esperimento con più account

Per imparare a creare un modello di esperimento tramite Console di gestione AWS

Per informazioni, consulta [Crea un modello di esperimento](#).

Per creare un modello di esperimento utilizzando la CLI

1. Aprire il AWS Command Line Interface
2. Per creare un esperimento da un file JSON salvato con l'opzione `account targeting experiment` impostata su `"multi-account"` (ad esempio, `my-template.json`), sostituisci i valori segnaposto *italics* con i tuoi valori, quindi esegui il comando seguente. [create-experiment-template](#)

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

Ciò restituirà il modello dell'esperimento nella risposta. Copia il id dalla risposta, che è l'ID del modello di esperimento.

3. Esegui il [create-target-account-configuration](#) comando per aggiungere una configurazione di account di destinazione al modello di esperimento. Sostituisci i valori segnaposto *italics* con i tuoi valori, utilizzando il passaggio 2 id dal punto 2 come valore per il `--experiment-template-id` parametro, quindi esegui quanto segue. Il parametro `--description` è facoltativo. Ripeti questo passaggio per ogni account di destinazione.

```
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. Esegui il [get-target-account-configuration](#) comando per recuperare i dettagli per una configurazione specifica dell'account di destinazione.

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

5. Dopo aver aggiunto tutte le configurazioni dell'account di destinazione, puoi eseguire il [list-target-account-configurations](#) comando di comando per verificare che le configurazioni dell'account di destinazione siano state create.

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

Puoi anche verificare di aver aggiunto configurazioni di account di destinazione eseguendo il comando. [get-experiment-template](#) Il modello restituirà un campo di sola lettura `targetAccountConfigurationsCount` che rappresenta il conteggio di tutte le configurazioni dell'account di destinazione sul modello di esperimento.

6. [Quando sei pronto, puoi eseguire il modello di esperimento usando il comando start-experiment.](#)

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

Aggiorna la configurazione di un account di destinazione

Puoi aggiornare la configurazione di un account di destinazione esistente se desideri modificare l'ARN o la descrizione del ruolo per l'account. Quando aggiorni la configurazione di un account di destinazione, le modifiche non influiscono sugli esperimenti in esecuzione che utilizzano il modello.

Per aggiornare la configurazione di un account di destinazione utilizzando il Console di gestione AWS

1. Apri la AWS FIS console all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti
3. Seleziona il modello di esperimento e scegli Azioni, Aggiorna modello di esperimento.
4. Nel pannello laterale, scegli Passaggio 3, Configura l'accesso al servizio.
5. Modifica le configurazioni dell'account di destinazione e scegli Aggiorna modello di esperimento.
6. Seleziona Passaggio 5, Rivedi e crea.

Per aggiornare la configurazione di un account di destinazione utilizzando la CLI

Esegui da [update-target-account-configuration](#) comando a comando, sostituendo i valori segnaposto *italics* con i tuoi valori. I `--description` parametri `--role-arn` and sono opzionali e non verranno aggiornati se non inclusi.

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

Eliminare la configurazione di un account di destinazione

Se non hai più bisogno di una configurazione dell'account di destinazione, puoi eliminarla. Quando elimini la configurazione di un account di destinazione, gli esperimenti in corso che utilizzano il modello non vengono modificati. L'esperimento continua a funzionare fino al completamento o all'arresto.

Per eliminare la configurazione di un account di destinazione utilizzando il Console di gestione AWS

1. Apri la AWS FIS console all'indirizzo <https://console.aws.amazon.com/fis/>.
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Aggiorna.
4. Nel pannello laterale, scegli Passaggio 3, Configura l'accesso al servizio.
5. In Configurazioni dell'account di Target, seleziona Rimuovi per l'ARN del ruolo dell'account di destinazione che desideri eliminare.
6. Seleziona Passaggio 5, Rivedi e crea.
7. Rivedi il modello e scegli Aggiorna modello di esperimento. Quando viene richiesta la conferma, inserisci update e scegli Aggiorna modello di esperimento.

Per eliminare la configurazione di un account di destinazione utilizzando la CLI

Esegui il [delete-target-account-configuration](#) comando, sostituendo i valori segnaposto *italics* con i tuoi valori.

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

Pianificazione degli esperimenti

Con AWS Fault Injection Service (FIS), puoi eseguire esperimenti di fault injection sui tuoi carichi di lavoro AWS. Questi esperimenti vengono eseguiti su modelli che contengono una o più azioni da eseguire su obiettivi specifici. Ora puoi pianificare gli esperimenti come attività singola o come attività ricorrenti in modo nativo dalla console FIS. Oltre alle [regole pianificate](#), FIS offre ora una nuova funzionalità di pianificazione. FIS ora si integra con EventBridge Scheduler e crea regole per conto dell'utente. EventBridge Scheduler è uno strumento di pianificazione senza server che consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato.

Important

Experiment Scheduler with non AWS Fault Injection Service è disponibile in AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali).

Argomenti

- [Crea un ruolo di scheduler](#)
- [Crea una pianificazione dell'esperimento](#)
- [Aggiorna la pianificazione di un esperimento](#)
- [Disabilita o elimina la pianificazione di un esperimento](#)

Crea un ruolo di scheduler

Un ruolo di esecuzione è un ruolo IAM che, per interagire con lo scheduler, AWS FIS si assume che lo EventBridge scheduler di Event Bridge avvii FIS Experiment. A questo ruolo si allegano politiche di autorizzazione per concedere a EventBridge Scheduler l'accesso per richiamare FIS Experiment. I passaggi seguenti descrivono come creare un nuovo ruolo di esecuzione e una politica per consentire l'avvio EventBridge di un esperimento.

Crea un ruolo di scheduler utilizzando la CLI di AWS

Questo è il ruolo IAM necessario affinché Event Bridge sia in grado di pianificare l'esperimento per conto del cliente.

1. Copia la seguente policy JSON per assumere il ruolo e salvala localmente come `fis-execution-role.json`. Questa politica di fiducia consente a EventBridge Scheduler di assumere il ruolo per tuo conto.

JSON

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

2. Da AWS Command Line Interface (AWS CLI), inserisci il seguente comando per creare un nuovo ruolo. Sostituiscilo `FisSchedulerExecutionRole` con il nome che desideri assegnare a questo ruolo.

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

In caso di successo, verrà visualizzato il seguente risultato:

```
{
  "Role": {
    "Path": "/",
    "RoleName": "FisSchedulerExecutionRole",
    "RoleId": "AROAZL22PDN5A6WKRBNUN",
    "Arn": "arn:aws:iam::123456789012:role/FisSchedulerExecutionRole",
    "CreateDate": "2023-08-24T17:23:05+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
```



```
{
  "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. Esegui il comando seguente per allegare la policy al tuo ruolo di esecuzione. Sostituisci `your-policy-arn` con l'ARN della policy creata nel passaggio precedente. `FisSchedulerExecutionRole` sostituisilo con il nome del tuo ruolo di esecuzione.

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

L'`attach-role-policy` operazione non restituisce una risposta sulla riga di comando.

6. È possibile limitare lo scheduler all'esecuzione solo di modelli di AWS FIS esperimenti con un valore di tag specifico. Ad esempio, la seguente politica concede l'`fis:StartExperiment` autorizzazione per tutti gli AWS FIS esperimenti, ma limita lo scheduler all'esecuzione solo di modelli di esperimenti con tag. `Purpose=Schedule`

JSON

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

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

Crea una pianificazione dell'esperimento

Prima di pianificare un esperimento, ne hai bisogno uno o più [Componenti del modello di esperimento](#) da richiamare. Puoi utilizzare una risorsa AWS esistente o crearne una nuova.

Una volta creato il modello di esperimento, fai clic su Azioni e seleziona Pianifica esperimento. Verrai reindirizzato alla pagina di pianificazione dell'esperimento. Il nome del programma verrà inserito automaticamente.

Segui la sezione Schema di pianificazione e scegli una pianificazione una tantum o ricorrente. Compila i campi di input obbligatori e vai alle autorizzazioni.

The screenshot shows the AWS Scheduler console interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, a search bar, and a region selector. Below the navigation bar, a horizontal menu lists various AWS services: S3, Amazon EventBridge, AWS FIS, CloudFormation, CloudWatch, IAM, EC2, AWS Organizations, and Application Composer. The main content area is titled 'Schedule pattern' and contains several sections:

- Occurrence**: A section with an 'Info' link and a sub-header 'You can define an one-time or recurrent schedule.' It features two radio buttons: 'One-time schedule' (which is selected) and 'Recurring schedule'.
- Date and time**: A section with a sub-header 'The date and time to invoke the target.' It includes a date input field (YYYY/MM/DD), a calendar icon, a time input field (hh:mm), and a timezone dropdown menu (currently set to '(UTC -04:00) America/New...').
- Flexible time window**: A section with a sub-header 'If you choose a flexible time window, Scheduler invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time.' It features a dropdown menu labeled 'Select'.
- Schedule state**: A section with a sub-header 'Enable schedule' and a sub-header 'You can choose not to enable the schedule now. You will be able to enable the schedule after it has been created.' It includes a radio button labeled 'Enable' which is selected.

Lo stato di pianificazione sarà abilitato per impostazione predefinita. Nota: se disabiliti lo stato di pianificazione, l'esperimento non verrà pianificato anche se crei una pianificazione.

AWS FIS Experiment Scheduler è basato su [EventBridge Scheduler](#). È possibile fare riferimento alla documentazione per i vari [tipi di pianificazione supportati](#).

Per aggiornare la pianificazione utilizzando la console

1. Apri la [AWS FIS console](#).
2. Nel riquadro di navigazione a sinistra, scegli Modelli di esperimenti.
3. Scegli il modello di esperimento per il quale desideri creare la pianificazione.
4. Fai clic su Azioni e seleziona Pianifica esperimento dal menu a discesa.
 - a. In Schedule name, il nome viene compilato automaticamente.
 - b. In Schema di pianificazione, seleziona Pianificazione ricorrente.
 - c. [In Tipo di pianificazione, puoi selezionare una pianificazione basata sulla tariffa, vedi Tipi di pianificazione.](#)
 - d. In Espressione della frequenza, scegli una frequenza più lenta del tempo di esecuzione dell'esperimento, ad esempio 5 minuti.
 - e. In Intervallo di tempo, seleziona il tuo fuso orario.
 - f. In Data e ora di inizio, specifica una data e un'ora di inizio.
 - g. In Data e ora di fine, specifica una data e un'ora di fine.
 - h. In Schedule State, attiva l'opzione Abilita pianificazione.
 - i. In Autorizzazioni, seleziona Usa ruolo esistente, quindi cerca `FisSchedulerExecutionRole`.
 - j. Scegli Next (Successivo).
5. Seleziona Rivedi e crea pianificazione, esamina i dettagli dello scheduler, quindi scegli Crea pianificazione.

Aggiorna la pianificazione di un esperimento

Puoi aggiornare la pianificazione di un esperimento in modo che avvenga alla data e all'ora specifiche che preferisci.

Per aggiornare l'esecuzione di un esperimento utilizzando la console

1. Apri la [console Amazon FIS](#).
2. Nel riquadro di navigazione, scegli Experiment Templates.
3. Scegliete Tipo di risorsa: Modello di esperimento per il quale è già stata creata una pianificazione.
4. Fai clic sull'ID dell'esperimento per il modello. Quindi vai alla scheda Pianificazioni.
5. Controlla se esiste una pianificazione esistente associata all'esperimento. Seleziona la pianificazione associata e fai clic sul pulsante Aggiorna pianificazione.

Disabilita o elimina la pianificazione di un esperimento

Per interrompere l'esecuzione o l'esecuzione di un esperimento secondo una pianificazione, puoi eliminare o disabilitare la regola. I passaggi seguenti illustrano come eliminare o disabilitare l'esecuzione di un esperimento utilizzando la AWS console.

Per eliminare o disabilitare una regola

1. Apri la [console Amazon FIS](#).
2. Nel riquadro di navigazione, scegli Experiment Templates.
3. Scegliete Tipo di risorsa: Modello di esperimento per il quale è già stata creata una pianificazione.
4. Fai clic sull'ID dell'esperimento per il modello. Quindi vai alla scheda Pianificazioni.
5. Controlla se esiste una pianificazione esistente associata all'esperimento. Seleziona la pianificazione associata e fai clic sul pulsante Aggiorna pianificazione.
6. Esegui una delle seguenti operazioni:
 - a. Per eliminare la pianificazione, seleziona il pulsante accanto alla regola Elimina pianificazione. Digita `delete` e fai clic sul pulsante Elimina pianificazione.
 - b. Per disabilitare la pianificazione, seleziona il pulsante accanto alla regola Disabilita pianificazione. Digita `disable` e fai clic sul pulsante Disabilita pianificazione.

Leve di sicurezza per AWS FIS

Le leve di sicurezza vengono utilizzate per interrompere tutti gli esperimenti in corso e impedire l'avvio di nuovi esperimenti. È possibile utilizzare la leva di sicurezza per impedire gli esperimenti FIS in determinati periodi di tempo o in risposta agli allarmi sullo stato dell'applicazione. Ogni AWS account ha una leva di sicurezza per. Regione AWS

Per gli esperimenti in corso che vengono interrotti dalla leva di sicurezza, si paga solo per la durata dell'azione eseguita prima dell'interruzione dell'esperimento. Gli esperimenti di cui viene impedito l'avvio non comporteranno alcun costo. Le seguenti sezioni forniscono informazioni su come iniziare a utilizzare le leve di sicurezza.

Argomenti

- [Concetti relativi alle leve di sicurezza](#)
- [Lavorare con le leve di sicurezza](#)

Concetti relativi alle leve di sicurezza

Una leva di sicurezza può essere innestata o disinnestata.

- Se disattivata, sono consentiti esperimenti FIS. Per impostazione predefinita, le leve di sicurezza sono disattivate.
- Se attivati, gli esperimenti in corso vengono interrotti e non è consentito iniziare nuovi esperimenti.

Un esperimento su cui agisce una leva di sicurezza terminerà in uno dei seguenti stati:

- Interrotto, se l'esperimento era in corso quando la leva di sicurezza era innestata.
- Annullato se l'esperimento è stato avviato quando la leva di sicurezza era già innestata.

Non è possibile riprendere o rieseguire un esperimento che è stato interrotto o annullato. Tuttavia, è possibile iniziare un nuovo esperimento utilizzando lo stesso modello di esperimento una volta disinserita la leva di sicurezza.

Risorsa della leva di sicurezza

La leva di sicurezza è una risorsa definita dall'Amazon Resource Name (ARN). Le leve di sicurezza includono i seguenti parametri:

- Stato, che può essere attivato o disattivato.
- Motivo, ovvero una stringa inserita dall'utente per registrare il motivo per cui lo stato della leva di sicurezza è stato modificato.

Lavorare con le leve di sicurezza

Questa sezione spiegherà in dettaglio come visualizzare, attivare e disattivare le leve di sicurezza utilizzando la AWS FIS console o la riga di comando.

Visualizzazione di una leva di sicurezza

Puoi visualizzare lo stato della leva di sicurezza per il tuo account e la tua regione seguendo i passaggi seguenti.

Per visualizzare una leva di sicurezza utilizzando la console

1. [Aprire la console AWS FIS](#)
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Se la leva di sicurezza è inserita, vedrai un banner di avviso nella parte superiore della pagina. Se non è presente alcun banner di avviso, la leva di sicurezza si disattiva.

Per visualizzare una leva di sicurezza utilizzando la CLI

- Utilizza il seguente comando:

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

Una leva di sicurezza può trovarsi in uno dei seguenti stati:

- Disinnestata - La leva di sicurezza non influisce su nessun esperimento. Gli esperimenti possono essere eseguiti liberamente. Le leve di sicurezza sono disattivate per impostazione predefinita.

- **Attivante** - La leva di sicurezza sta passando da disinnestata a innestata. Potrebbero esserci ancora esperimenti che non sono stati ancora interrotti. La leva di sicurezza non può essere cambiata in questo stato.
- **Attivata** - La leva di sicurezza è attiva e non sono in corso esperimenti. Qualsiasi nuovo esperimento che tenti di iniziare mentre la leva di sicurezza è inserita verrà annullato.

Inserimento di una leva di sicurezza

Attivare una leva di sicurezza utilizzando la console

1. [Aprire la console AWS FIS](#)
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Scegli il pulsante Interrompi tutti gli esperimenti.
4. Inserisci il motivo per cui è stata inserita la leva di sicurezza.
5. Scegli Conferma.

Per attivare una leva di sicurezza utilizzando la CLI

- Utilizza il seguente comando. Compila il campo del motivo con la tua risposta.

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

Disinnestare una leva di sicurezza

Per disinnestare una leva di sicurezza utilizzando la console

1. [Aprire la console AWS FIS](#)
2. Nel riquadro di navigazione, scegli Esperimenti.
3. Scegli il pulsante della leva di sicurezza Disengage.
4. Inserisci il motivo del disinnesto della leva di sicurezza.
5. Scegli Conferma.

Per disattivare una leva di sicurezza utilizzando la CLI

- Utilizza il seguente comando:

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

Monitoraggio degli AWS esperimenti FIS

È possibile utilizzare i seguenti strumenti per monitorare l'avanzamento e l'impatto degli esperimenti del AWS AWS Fault Injection Service (FIS).

AWS Console FIS e AWS CLI

Usa la console AWS FIS o il AWS CLI per monitorare l'avanzamento di un esperimento in esecuzione. È possibile visualizzare lo stato di ogni azione nell'esperimento e i risultati di ogni azione. Per ulteriori informazioni, consulta [the section called “Visualizza i tuoi esperimenti”](#).

CloudWatch metriche di utilizzo e allarmi

Utilizza le metriche di CloudWatch utilizzo per fornire visibilità sull'utilizzo delle risorse da parte del tuo account. AWS Le metriche di utilizzo FIS corrispondono alle AWS quote di servizio. È possibile configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni, consulta [Monitora utilizzando CloudWatch](#).

È inoltre possibile creare condizioni di interruzione per gli esperimenti AWS FIS creando CloudWatch allarmi che definiscono quando un esperimento supera i limiti. Quando viene attivato l'allarme, l'esperimento si interrompe. Per ulteriori informazioni, consulta [Condizioni di arresto](#). Per ulteriori informazioni sulla creazione di CloudWatch allarmi, consulta [Creazione di un CloudWatch allarme basato su una soglia statica](#) e [Creazione di un CloudWatch allarme basato sul rilevamento di anomalie](#) nella Amazon CloudWatch User Guide.

AWS Registrazione degli esperimenti FIS

Abilita la registrazione dell'esperimento per acquisire informazioni dettagliate sull'esperimento durante l'esecuzione. Per ulteriori informazioni, consulta [Registrazione degli esperimenti](#).

Eventi di modifica dello stato dell'esperimento

Amazon ti EventBridge consente di rispondere automaticamente agli eventi di sistema o alle modifiche delle risorse. AWS FIS emette una notifica quando lo stato di un esperimento cambia. Puoi creare regole per gli eventi che ti interessano che specificano l'azione automatica da intraprendere quando un evento corrisponde a una regola. Ad esempio, inviare una notifica a un argomento Amazon SNS o richiamare una funzione Lambda. Per ulteriori informazioni, consulta [Monitora utilizzando EventBridge](#).

CloudTrail registri

AWS CloudTrail Utilizzalo per acquisire informazioni dettagliate sulle chiamate effettuate all'API AWS FIS e archivarle come file di registro in Amazon S3. CloudTrail registra anche le chiamate effettuate al servizio APIs per le risorse su cui esegui gli esperimenti. È possibile utilizzare questi CloudTrail registri per determinare quali chiamate sono state effettuate, l'indirizzo IP di origine da cui proviene la chiamata, chi ha effettuato la chiamata, quando è stata effettuata la chiamata e così via.

AWS Notifiche Health Dashboard

AWS Health offre una visibilità continua sulle prestazioni delle risorse e sulla disponibilità dei AWS servizi e degli account. Quando inizi un esperimento, AWS FIS invia una notifica al tuo AWS Health Dashboard. La notifica è presente per tutta la durata dell'esperimento in ogni account che contiene risorse destinate a un esperimento, inclusi gli esperimenti con più account. Gli esperimenti con più account con solo azioni che non includono obiettivi, come `aws:ssm:start-automation-execution` e `aws:fis:wait`, non genereranno alcuna notifica. Le informazioni sul ruolo utilizzato per consentire l'esperimento verranno elencate in Risorse interessate. Per ulteriori informazioni su AWS Health Dashboard, consulta [AWS Health Dashboard](#) nella AWS Health User Guide.

Note

AWS Health offre eventi con il massimo impegno possibile.

Monitora AWS i parametri di utilizzo FIS con Amazon CloudWatch

Puoi usare Amazon CloudWatch per monitorare l'impatto degli esperimenti AWS FIS sugli obiettivi. Puoi anche monitorare l'utilizzo del AWS FIS.

Per ulteriori informazioni sulla visualizzazione dello stato di un esperimento, vedere [Visualizza i tuoi esperimenti](#).

Monitora gli AWS esperimenti FIS

Mentre pianifichi gli esperimenti AWS FIS, identifica le CloudWatch metriche che puoi utilizzare per identificare la linea di base o lo «stato stazionario» per i tipi di risorse target per l'esperimento. Dopo aver iniziato un esperimento, puoi monitorare tali CloudWatch metriche per gli obiettivi selezionati tramite il modello di esperimento.

Per ulteriori informazioni sulle CloudWatch metriche disponibili per un tipo di risorsa target supportato da AWS FIS, consulta quanto segue:

- [Monitora le tue istanze utilizzando CloudWatch](#)
- [Metriche di Amazon ECS CloudWatch](#)
- [Monitoraggio dei parametri di Amazon RDS tramite CloudWatch](#)
- [Monitoraggio dei parametri di Run Command tramite CloudWatch](#)

AWS Metriche di utilizzo FIS

Puoi utilizzare le metriche di CloudWatch utilizzo per fornire visibilità sull'utilizzo delle risorse da parte del tuo account. Utilizza queste metriche per visualizzare l'utilizzo corrente del servizio su CloudWatch grafici e dashboard.

AWS Le metriche di utilizzo FIS corrispondono alle quote di servizio. AWS È possibile configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sugli CloudWatch allarmi, consulta la [Amazon CloudWatch User Guide](#).

AWS FIS pubblica la seguente metrica nello spazio dei nomi AWS/Usage.

Parametro	Descrizione
ResourceCount	Il numero totale delle risorse specificate in esecuzione nell'account. La risorsa è definita dalle dimensioni associate attraverso il parametro.

Le seguenti dimensioni vengono utilizzate per perfezionare le metriche di utilizzo pubblicate da FIS.
AWS

Dimensione	Descrizione
Service	Il nome del AWS servizio che contiene la risorsa. Per le metriche di utilizzo AWS FIS, il valore per questa dimensione è. FIS

Dimensione	Descrizione
Type	Il tipo di entità che viene segnalato. Attualmente, l'unico valore valido per le metriche di utilizzo AWS FIS è. Resource
Resource	Il tipo di risorsa in esecuzione. I valori possibili sono ExperimentTemplates per i modelli di esperimenti e ActiveExperiments per gli esperimenti attivi.
Class	Questa dimensione è riservata per utilizzi futuri.

Monitora gli esperimenti AWS FIS con Amazon EventBridge

Quando lo stato di un esperimento cambia, AWS FIS emette una notifica. Queste notifiche vengono rese disponibili come eventi tramite Amazon EventBridge (precedentemente chiamato CloudWatch Events). AWS FIS emette questi eventi con la massima diligenza possibile. Gli eventi vengono consegnati quasi EventBridge in tempo reale.

Con EventBridge, puoi creare regole che attivano azioni programmatiche in risposta a un evento. Ad esempio, puoi configurare una regola che richiama un argomento SNS per inviare una notifica e-mail o richiama una funzione Lambda per eseguire alcune azioni.

Per ulteriori informazioni EventBridge, consulta la sezione Guida [introduttiva ad Amazon EventBridge](#) nella Amazon EventBridge User Guide.

Di seguito è riportata la sintassi di un evento di modifica dello stato dell'esperimento:

```
{
  "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

L'ID dell'esperimento il cui stato è cambiato.

experiment-template-id

L'ID del modello di esperimento utilizzato dall'esperimento.

new_value

Il nuovo stato dell'esperimento. I valori possibili sono:

- completed
- failed
- initiating
- running
- stopped
- stopping

old_value

Lo stato precedente dell'esperimento. I valori possibili sono:

- initiating
- pending
- running

- stopping

Registrazione degli esperimenti per AWS FIS

È possibile utilizzare la registrazione dell'esperimento per acquisire informazioni dettagliate sull'esperimento mentre viene eseguito.

I costi per la registrazione dell'esperimento vengono calcolati in base ai costi associati a ciascun tipo di destinazione del registro. [Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon \(sotto Paid Tier, Logs, Vending Logs\) e i prezzi di Amazon S3.](#)

Autorizzazioni

Devi concedere le autorizzazioni AWS FIS per inviare i log a ogni destinazione di log che configuri. Per ulteriori informazioni, consulta quanto segue nella Amazon CloudWatch Logs User Guide:

- [Log inviati a Logs CloudWatch](#)
- [Registri inviati ad Amazon S3](#)

Schema di registro

Di seguito è riportato lo schema utilizzato nella registrazione degli esperimenti. La versione attuale dello schema è la 2. I campi per `details` dipendono dal valore di `log_type`. I campi per `resolved_targets` dipendono dal valore di `target_type`. Per ulteriori informazioni, consulta [the section called "Esempi di record di 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
}
}

```

Note di rilascio

- La versione 2 introduce:
 - Il `target_type` campo e lo `resolved_targets` modifica da un elenco di ARNs a un elenco di oggetti. I campi validi per l'`resolved_targets` soggetto dipendono dal valore di `target_type`, che è il [tipo di risorsa](#) degli obiettivi.
 - I `action-error` e i tipi di `target-resolution-detail` eventi che aggiungono il `account_id` campo.
- La versione 1 è la versione iniziale.

Destinazioni di registro

AWS FIS supporta la consegna dei log alle seguenti destinazioni:

- Un bucket Amazon S3
- Un gruppo di CloudWatch log di Amazon Logs

Consegna dei log S3

I registri vengono consegnati nella seguente posizione.

```
bucket-and-optional-prefix/AWSLogs/account-id/fis/region/experiment-id/YYYY/MM/DD/account-id_awsfislogs_region_experiment-id_YYYYMMDDHHMMZ_hash.log
```

Possono essere necessari alcuni minuti prima che i registri vengano consegnati al bucket.

CloudWatch Registri, consegna dei log

I log vengono consegnati a un flusso named /aws/fis di log/. *experiment-id*

I log vengono consegnati al gruppo di log in meno di un minuto.

Esempi di record di registro

Di seguito sono riportati alcuni esempi di record di registro per un esperimento che esegue l'aws:ec2:reboot-istancesazione su un' EC2 istanza selezionata a caso.

Registri

- [inizio dell'esperimento](#)
- [target-resolution-start](#)
- [target-resolution-detail](#)
- [target-resolution-end](#)
- [inizio azione](#)
- [azione-fine](#)
- [errore di azione](#)
- [fine dell'esperimento](#)

inizio esperimento

Di seguito è riportato un record di esempio per l'`experiment-start`evento.

```
{
  "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

Di seguito è riportato un record di esempio per l'`target-resolution-start`evento.

```
{
  "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

Di seguito è riportato un record di esempio per l'`target-resolution-detail`evento. Se la risoluzione dell'obiettivo fallisce, il record include anche il `error_information` campo.

```
{
  "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

Se la risoluzione dell'obiettivo fallisce, il record include anche il `error_information` campo. Se `total_pages` è maggiore di 1, il numero di obiettivi risolti ha superato il limite di dimensione per un record. Esistono `target-resolution-end` record aggiuntivi che contengono gli obiettivi risolti rimanenti.

Di seguito è riportato un esempio di record per l'`target-resolution-end` evento relativo a un' EC2 azione.

```

{
  "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
  }
}

```

Di seguito è riportato un record di esempio per l'`target-resolution-end` evento di un'azione 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
  }
}
```

inizio-azione

Di seguito è riportato un record di esempio per l'`action-start` evento. Se il modello di esperimento specifica i parametri per l'azione, il record include anche il `parameters` campo.

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

azione-errore

Di seguito è riportato un record di esempio per l'`action-errore`evento. Questo evento viene restituito solo quando un'azione fallisce. Viene restituito per ogni account in cui l'azione ha esito negativo.

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

fine dell'azione

Di seguito è riportato un record di esempio per l'`action-ende`evento.

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

```
}
```

fine dell'esperimento

Di seguito è riportato un record di esempio per l'`experiment-end` evento.

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

Abilita la registrazione degli esperimenti

La registrazione degli esperimenti è disattivata per impostazione predefinita. Per ricevere i registri degli esperimenti, è necessario creare l'esperimento da un modello di esperimento con la registrazione abilitata. La prima volta che esegui un esperimento configurato per utilizzare una destinazione che non è stata utilizzata in precedenza per la registrazione, ritardiamo l'esperimento per configurare l'invio dei log a questa destinazione, operazione che richiede circa 15 secondi.

Per abilitare la registrazione degli esperimenti tramite la console

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Aggiorna modello di esperimento.
4. Per i registri, configura le opzioni di destinazione. Per inviare i log a un bucket S3, scegli Invia a un bucket Amazon S3 e inserisci il nome e il prefisso del bucket. Per inviare i log a Logs, scegli Send to CloudWatch Logs e inserisci il gruppo di log. CloudWatch
5. Scegliete Aggiorna modello di esperimento.

Per abilitare la registrazione dell'esperimento utilizzando il AWS CLI

Utilizzate il [update-experiment-template](#) comando e specificate una configurazione di registro.

Disabilita la registrazione degli esperimenti

Se non desideri più ricevere i registri degli esperimenti, puoi disabilitare la registrazione degli esperimenti.

Per disabilitare la registrazione degli esperimenti tramite la console

1. Aprire la console AWS FIS all'indirizzo. <https://console.aws.amazon.com/fis/>
2. Nel riquadro di navigazione, scegli Modelli di esperimenti.
3. Seleziona il modello di esperimento e scegli Azioni, Aggiorna modello di esperimento.
4. Per i log, deseleziona Invia a un bucket Amazon S3 e Invia ai registri. CloudWatch
5. Scegli Aggiorna modello di esperimento.

Per disabilitare la registrazione dell'esperimento utilizzando il AWS CLI

Utilizzate il [update-experiment-template](#) comando e specificate una configurazione di registro vuota.

Registra le chiamate API con AWS CloudTrail

AWS AWS Fault Injection Service (FIS) è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in AWS FIS. CloudTrail acquisisce tutte le chiamate API per AWS FIS come eventi. Le chiamate acquisite includono chiamate dalla console AWS FIS e chiamate di codice alle operazioni dell'API AWS FIS. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per FIS. AWS Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, è possibile determinare la richiesta effettuata a AWS FIS, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la [Guida per l'AWS CloudTrail utente](#).

Usa CloudTrail

CloudTrail è abilitato sul tuo Account AWS quando crei l'account. Quando si verifica un'attività in AWS FIS, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio

nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti in Account AWS. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi del tuo Account AWS, compresi gli eventi per la AWS FIS, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Crea un percorso per il tuo account AWS](#)
- [CloudTrail Servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Tutte le azioni AWS FIS vengono registrate CloudTrail e documentate nel [AWS Fault Injection Service](#) API Reference. Per le azioni sperimentali eseguite su una risorsa di destinazione, consultate la documentazione di riferimento dell'API per il servizio proprietario della risorsa. Ad esempio, per le azioni eseguite su un' EC2 istanza Amazon, consulta l'[Amazon EC2 API Reference](#).

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Comprendi le AWS voci dei file di registro FIS

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro.

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

Di seguito è riportato un esempio di voce di CloudTrail registro per una chiamata all'azione AWS FISStopExperiment.

```
{
  "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"
}

```

Di seguito è riportato un esempio di voce di CloudTrail registro per un'azione API richiamata da AWS FIS come parte di un esperimento che include l'aws:ssm:send-command AWS azione FIS. L'userIdentity elemento riflette una richiesta effettuata con credenziali temporanee ottenute assumendo un ruolo. Il nome del ruolo assunto appare in. userName L'ID dell'esperimento, EXP21n T17 WMz A6DnuGz, appare nell'ARN del ruolo principalId assunto e come parte di esso.

```

{
  "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"
}
```

Risoluzione dei problemi AWS FIS

Per risolvere gli errori, AWS FIS restituisce errori dettagliati dai registri degli esperimenti `GetExperiment` API e FIS. Gli errori vengono restituiti come parte dello stato dell'esperimento quando lo stato dell'esperimento non è riuscito. Quando più azioni falliscono, la prima azione fallita viene restituita come errore dell'esperimento. È possibile esaminare i registri degli esperimenti FIS per verificare la presenza di altri errori. Per informazioni su come registrare e monitorare AWS FIS gli esperimenti, consulta [Monitoraggio degli AWS esperimenti FIS](#)

A seconda del tipo di errore, è possibile che si verifichi uno dei seguenti errori:

- **Motivo:** descrizione dettagliata dell'errore specifico. I valori del motivo non devono essere utilizzati per l'automazione, poiché sono soggetti a modifiche.
- **Codice:** il tipo di errore. I valori del codice non devono essere utilizzati per l'automazione, poiché sono soggetti a modifiche, se non diversamente specificato nella tabella seguente.
- **Posizione:** contesto per la sezione del modello di esperimento che ha avuto esito negativo, ad esempio l'azione o il bersaglio.
- **ID account:** l' AWS account in cui si è verificato l'errore.

Codici di errore

Codice di errore	Descrizione del codice
<code>ConfigurationFailure</code>	L'azione, l'obiettivo, l'esperimento o il registro non sono configurati correttamente. Controlla l'errore <code>location</code> e assicurati che i parametri e le configurazioni siano corretti.
<code>DependentServiceFailure</code>	Si è verificato un errore da un altro AWS servizio. Prova a eseguire di nuovo l'esperimento.
<code>InternalFailure</code>	Si è verificato un errore interno durante l'esecuzione dell'esperimento. È possibile eseguire l'automazione in base a questo codice di errore.

Codice di errore	Descrizione del codice
InvalidTarget	<p data-bbox="831 226 1481 403">Non è stato possibile risolvere un obiettivo durante la risoluzione dell'obiettivo o all'inizio di un'azione. Ciò può essere dovuto a uno dei seguenti motivi:</p> <ul data-bbox="831 449 1497 739" style="list-style-type: none"><li data-bbox="831 449 1497 529">• La destinazione non esiste, ad esempio se è stata eliminata o se l'ARN non è corretto.<li data-bbox="831 554 1497 634">• Esiste un tag per l'obiettivo che non risolve alcuna risorsa.<li data-bbox="831 659 1497 739">• Esiste un'azione che non è collegata a un obiettivo. <p data-bbox="831 819 1481 1041">Per risolvere i problemi, esamina i log per identificare quali obiettivi non sono stati risolti. Verifica che tutte le azioni siano collegate agli obiettivi e che l'ID o i tag della risorsa esistano e non siano stati scritti male.</p>

Codice di errore	Descrizione del codice
AuthorizationFailure	<p>Esistono due cause principali del fallimento dell'esperimento dovuto a errori di autorizzazione:</p> <ul style="list-style-type: none">• Il ruolo IAM a cui ti rivolgi non dispone delle autorizzazioni appropriate per risolvere obiettivi o intervenire sulle tue risorse. Per correggere questo errore, esamina le autorizzazioni richieste per le tue azioni nel FIS Actions Reference e aggiungile al tuo ruolo IAM sperimentale.• La creazione del ruolo SLR (AWS Service-linked role) per FIS è stata negata da una policy di controllo dei servizi (SCP) dell'organizzazione. FIS utilizza la SLR per gestire il monitoraggio e la selezione delle risorse per gli esperimenti. Per ulteriori informazioni, consulta Autorizzazioni di ruolo collegate al servizio per FIS AWS.
QuotaExceededFailure	<p>La quota per il tipo di risorsa è stata superata. Per determinare se la quota può essere aumentata, vedere Quote e limitazioni per il servizio AWS Fault Injection.</p>

Sicurezza nel servizio AWS Fault Injection

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità che si applicano al servizio di iniezione di AWS errori, vedere [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza AWS FIS. I seguenti argomenti mostrano come configurare il AWS FIS per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse AWS FIS.

Indice

- [Protezione dei dati in AWS Fault Injection Service](#)
- [Gestione delle identità e degli accessi per AWS Fault Injection Service](#)
- [Sicurezza dell'infrastruttura in AWS Fault Injection Service](#)
- [Accedere a AWS FIS utilizzando un'interfaccia VPC endpoint \(\)AWS PrivateLink](#)

Protezione dei dati in AWS Fault Injection Service

Il modello di [responsabilità AWS condivisa modello](#) si applica alla protezione dei dati in AWS Fault Injection Service. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo

dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [AWS Modello di responsabilità condivisa e GDPR](#) nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò vale anche quando lavori con AWS FIS o altri dispositivi Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Crittografia dei dati a riposo

AWS FIS crittografa sempre i dati inattivi. I dati in AWS FIS vengono crittografati quando sono inattivi utilizzando una crittografia trasparente lato server. Questo consente di ridurre gli oneri operativi e

la complessità associati alla protezione dei dati sensibili. La crittografia dei dati inattivi consente di creare applicazioni sicure che rispettano rigorosi requisiti normativi e di conformità per la crittografia.

Crittografia dei dati in transito

AWS FIS crittografa i dati in transito tra il servizio e altri servizi integrati. AWS Tutti i dati che passano tra AWS FIS e i servizi integrati vengono crittografati utilizzando Transport Layer Security (TLS). Per ulteriori informazioni su altri AWS servizi integrati, vedere. [Supportato Servizi AWS](#)

Gestione delle identità e degli accessi per AWS Fault Injection Service

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse FIS. AWS IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Indice

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [Come funziona AWS Fault Injection Service con IAM](#)
- [AWS Esempi di policy relative al servizio Fault Injection](#)
- [Utilizza ruoli collegati ai servizi per Fault Injection Service AWS](#)
- [AWS politiche gestite per AWS Fault Injection Service](#)

Destinatari

Il modo in cui si utilizza AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in AWS FIS.

Utente del servizio: se utilizzi il servizio AWS FIS per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità AWS FIS per

svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso consente di richiedere le autorizzazioni corrette all'amministratore.

Amministratore del servizio: se sei responsabile delle risorse AWS FIS della tua azienda, probabilmente hai pieno accesso a FIS. AWS È tuo compito determinare a quali funzionalità e risorse AWS FIS devono accedere gli utenti del servizio. Devi quindi inviare le richieste all'amministratore IAM per modificare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM.

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso al AWS FIS.

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- **Limiti delle autorizzazioni:** imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Politiche di controllo del servizio (SCPs):** specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in AWS Organizations. Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .

- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

Come funziona AWS Fault Injection Service con IAM

Prima di utilizzare IAM per gestire l'accesso al AWS FIS, scopri quali funzionalità IAM sono disponibili per l'uso con AWS FIS.

Funzionalità IAM che puoi utilizzare con AWS Fault Injection Service

Funzionalità IAM	AWS Supporto FIS
Policy basate sull'identità	Sì
Policy basate su risorse	No
Operazioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì

Funzionalità IAM	AWS Supporto FIS
Ruoli di servizio	Sì
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come AWS FIS e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

Politiche basate sull'identità per FIS AWS

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Esempi di politiche basate sull'identità per FIS AWS

Per visualizzare esempi di politiche AWS FIS basate sull'identità, vedere. [AWS Esempi di policy relative al servizio Fault Injection](#)

Politiche basate sulle risorse all'interno della FIS AWS

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Azioni politiche per la FIS AWS

Supporta le operazioni di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di azioni AWS FIS, vedere [Azioni definite da AWS Fault Injection Service nel Service Authorization](#) Reference.

Le azioni politiche in AWS FIS utilizzano il seguente prefisso prima dell'azione:

```
fis
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
  "fis:action1",  
  "fis:action2"  
]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `List`, includi la seguente azione:

```
"Action": "fis:List*"
```

Risorse politiche per la FIS AWS

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Alcune azioni dell'API AWS FIS supportano più risorse. Per specificare più risorse in un'unica istruzione, separare ARNs con virgole.

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

Per visualizzare un elenco dei tipi di risorse AWS FIS e relativi ARNs, vedere Tipi di [risorse definiti da AWS Fault Injection Service nel Service](#) Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, vedere [Azioni definite da AWS Fault Injection Service](#).

Chiavi relative alle condizioni delle politiche per AWS FIS

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione AWS FIS, consulta [Condition keys for AWS Fault Injection Service nel Service](#) Authorization Reference. Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, vedere [Azioni definite da AWS Fault Injection Service](#).

Per visualizzare esempi di politiche AWS FIS basate sull'identità, vedere. [AWS Esempi di policy relative al servizio Fault Injection](#)

ACLs AWS in FIS

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con FIS AWS

Supporta ABAC (tag nelle policy): sì

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Per visualizzare un esempio di politica basata sull'identità per limitare l'accesso a una risorsa in base ai tag di quella risorsa, consulta. [Esempio: utilizzare i tag per controllare l'utilizzo delle risorse](#)

Utilizzo di credenziali temporanee con FIS AWS

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per

ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nella Guida per l'utente IAM.

Autorizzazioni principali multiservizio per FIS AWS

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso diretto (FAS) utilizzano le autorizzazioni del principale chiamante an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

Ruoli di servizio per FIS AWS

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

Ruoli collegati ai servizi per FIS AWS

Supporta i ruoli collegati ai servizi: sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi AWS FIS, vedere [Utilizza ruoli collegati ai servizi per Fault Injection Service AWS](#)

AWS Esempi di policy relative al servizio Fault Injection

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse AWS FIS. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM \(console\)](#) nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti dalla AWS FIS, incluso il formato di ARNs per ciascun tipo di risorsa, vedere [Azioni, risorse e chiavi di condizione per il servizio AWS Fault Injection nel Service](#) Authorization Reference.

Indice

- [Best practice per le policy](#)
- [Esempio: utilizza la console FIS AWS](#)
- [Esempio: elenca le azioni FIS disponibili AWS](#)
- [Esempio: creare un modello di esperimento per un'azione specifica](#)
- [Esempio: avvia un esperimento](#)
- [Esempio: utilizzare i tag per controllare l'utilizzo delle risorse](#)
- [Esempio: eliminare un modello di esperimento con un tag specifico](#)
- [Esempio: consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Esempio: utilizza i tasti condizionali per ec2:InjectApiError](#)
- [Esempio: utilizzare i tasti condizionali per aws:s3:bucket-pause-replication](#)

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse AWS FIS nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS. Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.

- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Esempio: utilizza la console FIS AWS

Per accedere alla console AWS Fault Injection Service, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentire all'utente di elencare e visualizzare i dettagli sulle risorse AWS FIS del proprio Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l'API. AWS Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

La seguente politica di esempio concede l'autorizzazione a elencare e visualizzare tutte le risorse AWS FIS utilizzando la console AWS FIS, ma non a crearle, aggiornarle o eliminarle. Concede inoltre le autorizzazioni per visualizzare le risorse disponibili utilizzate da tutte le azioni AWS FIS che è possibile specificare in un modello di esperimento.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FISReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "fis:List*",
        "fis:Get*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AdditionalReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "ssm:Describe*",
        "ssm:Get*",
        "ssm:List*",
        "ec2:DescribeInstances",
        "rds:DescribeDBClusters",
        "ecs:DescribeClusters",
        "ecs:ListContainerInstances",
        "eks:DescribeNodegroup",
        "cloudwatch:DescribeAlarms",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PermissionsToCreateServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "fis.amazonaws.com"
        }
      }
    }
  ]
}
```

```
}
```

Esempio: elenca le azioni FIS disponibili AWS

La seguente politica concede l'autorizzazione a elencare le azioni AWS FIS disponibili.

JSON

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

Esempio: creare un modello di esperimento per un'azione specifica

La seguente politica concede l'autorizzazione a creare un modello di esperimento per l'azione `aws:ec2:stop-instances`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PolicyExample",
      "Effect": "Allow",
      "Action": [
        "fis:CreateExperimentTemplate"
      ],
      "Resource": [
        "arn:aws:fis:*:*:action/aws:ec2:stop-instances",

```

```

        "arn:aws:fis:*:*:experiment-template/*"
    ]
},
{
    "Sid": "PolicyPassRoleExample",
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam:*:*:role/role-name"
    ]
}
]
}

```

Esempio: avvia un esperimento

La seguente politica concede il permesso di avviare un esperimento utilizzando il ruolo IAM e il modello di esperimento specificati. Consente inoltre a AWS FIS di creare un ruolo collegato al servizio per conto dell'utente. Per ulteriori informazioni, consulta [Utilizza ruoli collegati ai servizi per Fault Injection Service AWS](#).

JSON

```

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

```

```

    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": "fis.amazonaws.com"
      }
    }
  }
]
}

```

Esempio: utilizzare i tag per controllare l'utilizzo delle risorse

La seguente politica concede il permesso di eseguire esperimenti da modelli di esperimenti che dispongono del tag `Purpose=Test`. Non concede il permesso di creare o modificare modelli di esperimento o eseguire esperimenti utilizzando modelli che non hanno il tag specificato.

JSON

```

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

```

Esempio: eliminare un modello di esperimento con un tag specifico

La seguente politica concede il permesso di eliminare un modello di esperimento con `tagPurpose=Test`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "fis:DeleteExperimentTemplate"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

Esempio: consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    }
  ]
}
```

```

    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}

```

Esempio: utilizza i tasti condizionali per **ec2:InjectApiError**

La seguente politica di esempio utilizza la chiave `ec2:FisTargetArns` condition per definire l'ambito delle risorse di destinazione. Questa politica consente le azioni AWS FIS `aws:ec2:api-insufficient-instance-capacity-error` e `aws:ec2:asg-insufficient-instance-capacity-error`.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:InjectApiError",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:FisActionId": [
            "aws:ec2:api-insufficient-instance-capacity-error",
            "aws:ec2:asg-insufficient-instance-capacity-error"
          ]
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:InjectApiError",
    "Resource": "*",
    "Condition": {
      "ForAllValues:ArnLike": {
        "ec2:FisTargetArns": [
          "arn:aws:autoscaling:*:*:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
        ]
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "autoscaling:DescribeAutoScalingGroups",
    "Resource": "*"
  }
]
}

```

Esempio: utilizzare i tasti condizionali per **aws:s3:bucket-pause-replication**

La politica di esempio seguente utilizza la chiave `S3:IsReplicationPauseRequest` condition per consentire `PutReplicationConfiguration` e `GetReplicationConfiguration` solo se utilizzata da AWS FIS nel contesto dell'azione AWS FIS. `aws:s3:bucket-pause-replication`

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "S3:PauseReplication"
      ],
      "Resource": "arn:aws:s3:::mybucket",
      "Condition": {

```

```

        "StringEquals": {
            "s3:DestinationRegion": "region"
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "S3:PutReplicationConfiguration",
            "S3:GetReplicationConfiguration"
        ],
        "Resource": "arn:aws:s3:::mybucket",
        "Condition": {
            "BoolIfExists": {
                "s3:IsReplicationPauseRequest": "true"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "S3:ListBucket"
        ],
        "Resource": "arn:aws:s3:::*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "tag:GetResources"
        ],
        "Resource": "*"
    }
]
}

```

Utilizza ruoli collegati ai servizi per Fault Injection Service AWS

AWS [Fault Injection Service](#) utilizza ruoli collegati ai servizi [AWS Identity and Access Management \(IAM\)](#). Un ruolo collegato al servizio è un tipo unico di ruolo IAM collegato direttamente al FIS. AWS I ruoli collegati ai servizi sono predefiniti da AWS FIS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per conto dell'utente. AWS

Un ruolo collegato ai servizi semplifica la configurazione del AWS FIS perché non è necessario aggiungere manualmente le autorizzazioni necessarie per gestire il monitoraggio e la selezione delle risorse per gli esperimenti. AWS FIS definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo FIS può assumerne i ruoli. AWS Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni. Una policy delle autorizzazioni specifica non può essere collegata a un'altra entità IAM.

Oltre al ruolo collegato al servizio, è necessario specificare anche un ruolo IAM che conceda l'autorizzazione a modificare le risorse specificate come destinazioni in un modello di esperimento. Per ulteriori informazioni, consulta [Ruoli IAM per AWS esperimenti FIS](#).

È possibile eliminare un ruolo collegato ai servizi solo dopo avere eliminato le risorse correlate. In questo modo proteggi le tue risorse AWS FIS perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Autorizzazioni di ruolo collegate al servizio per FIS AWS

AWS FIS utilizza il ruolo collegato al servizio denominato AWSServiceRoleForFIS per consentirgli di gestire il monitoraggio e la selezione delle risorse per gli esperimenti.

Il ruolo collegato al servizio AWSServiceRoleForFIS prevede che i seguenti servizi assumano il ruolo:

- `fis.amazonaws.com`

Il ruolo collegato al servizio AWSServiceRoleForFIS utilizza la politica gestita Amazon. FISServiceRolePolicy Questa policy consente alla AWS FIS di gestire il monitoraggio e la selezione delle risorse per gli esperimenti. Per ulteriori informazioni, consulta [Amazon FISService RolePolicy](#) nel AWS Managed Policy Reference.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio è necessario configurare le relative autorizzazioni. AWSServiceRoleForAffinché il ruolo collegato al servizio FIS venga creato correttamente, l'identità IAM con cui si utilizza AWS FIS deve disporre delle autorizzazioni richieste. Per concedere le autorizzazioni richieste, collega la seguente policy all'identità IAM.

JSON

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

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": "iam:CreateServiceLinkedRole",  
    "Resource": "*",  
    "Condition": {  
      "StringLike": {  
        "iam:AWSServiceName": "fis.amazonaws.com"  
      }  
    }  
  }  
]
```

Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente IAM.

Crea un ruolo collegato al servizio per FIS AWS

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando si avvia un esperimento AWS FIS nell' Console di gestione AWS, nella o nell' AWS API AWS CLI, AWS FIS crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando si avvia un esperimento FIS, AWS FIS crea nuovamente il ruolo collegato al servizio.

Modifica un ruolo collegato al servizio per FIS AWS

AWS FIS non consente di modificare il ruolo collegato al servizio FIS. `AWSServiceRoleFor` Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

Eliminare un ruolo collegato al servizio per FIS AWS

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Note

Se il servizio AWS FIS utilizza il ruolo quando si tenta di ripulire le risorse, la pulizia potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per pulire le risorse AWS FIS utilizzate dal FIS AWSService RoleFor

Assicurati che nessuno dei tuoi esperimenti sia attualmente in esecuzione. Se necessario, interrompi gli esperimenti. Per ulteriori informazioni, consulta [Interrompere un esperimento](#).

Eliminazione manuale del ruolo collegato al servizio con IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al servizio AWSServiceRoleForFIS. Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato ai servizi](#) nella Guida per l'utente IAM.

Regioni supportate per AWS i ruoli collegati ai servizi FIS

AWS FIS supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, vedere [Endpoint e quote del AWS Fault Injection Service](#).

AWS politiche gestite per AWS Fault Injection Service

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. Le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Si consiglia pertanto di ridurre ulteriormente le autorizzazioni definendo [policy gestite dal cliente](#) specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando nel Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

AWS politica gestita: Amazon FISService RolePolicy

Questa politica è allegata al ruolo collegato al servizio denominato AWSServiceRoleForFIS per consentire a FIS di gestire il monitoraggio e AWS la selezione delle risorse per gli esperimenti. Per ulteriori informazioni, consulta [Utilizza ruoli collegati ai servizi per Fault Injection Service AWS](#).

AWS politica gestita: accesso AWSFault InjectionSimulator EC2

Utilizza questa policy in un ruolo di esperimento per concedere l'autorizzazione AWS FIS a eseguire esperimenti che utilizzano le [azioni AWS FIS per Amazon EC2](#). Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).

Per visualizzare le autorizzazioni per questa politica, consulta [AWSFaultInjectionSimulatorEC2Access](#) nel Managed Policy Reference.AWS

AWS politica gestita: AWSFault InjectionSimulator ECSAccess

Usa questa policy in un ruolo di esperimento per concedere l'autorizzazione AWS FIS a eseguire esperimenti che utilizzano le [azioni AWS FIS per Amazon ECS](#). Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).

Per vedere le autorizzazioni per questa policy, consulta [AWSFaultInjectionSimulatorECSAccess](#) nella Guida di riferimento sulle policy gestite da AWS .

AWS politica gestita: AWSFault InjectionSimulator EKSAccess

Utilizza questa politica in un ruolo di esperimento per concedere l'autorizzazione AWS FIS a eseguire esperimenti che utilizzano le [azioni AWS FIS per Amazon EKS](#). Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).

Per vedere le autorizzazioni per questa policy, consulta [AWSFaultInjectionSimulatorEKSAccess](#) nella Guida di riferimento sulle policy gestite da AWS .

AWS politica gestita: AWSFault InjectionSimulatorNetworkAccess

Utilizza questo criterio in un ruolo di esperimento per concedere l'autorizzazione AWS FIS a eseguire esperimenti che utilizzano le azioni di [rete AWS FIS](#). Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).

Per vedere le autorizzazioni per questa policy, consulta [AWSFaultInjectionSimulatorNetworkAccess](#) nella Guida di riferimento sulle policy gestite da AWS .

AWS politica gestita: AWSFault Injection Simulator RDSAccess

Usa questa policy in un ruolo di esperimento per concedere l'autorizzazione AWS FIS a eseguire esperimenti che utilizzano le [azioni AWS FIS per Amazon RDS](#). Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).

Per vedere le autorizzazioni per questa policy, consulta [AWSFaultInjectionSimulatorRDSAccess](#) nella Guida di riferimento sulle policy gestite da AWS .

AWS politica gestita: AWSFault Injection Simulator SSMAccess

Utilizzate questo criterio in un ruolo di esperimento per concedere l'autorizzazione AWS FIS a eseguire esperimenti che utilizzano le [azioni AWS FIS per Systems Manager](#). Per ulteriori informazioni, consulta [the section called "Ruolo dell'esperimento"](#).

Per vedere le autorizzazioni per questa policy, consulta [AWSFaultInjectionSimulatorSSMAccess](#) nella Guida di riferimento sulle policy gestite da AWS .

AWS Aggiornamenti FIS alle politiche gestite AWS

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per AWS FIS da quando questo servizio ha iniziato a tenere traccia di queste modifiche.

Modifica	Descrizione	Data
AWSFaultInjectionSimulatorEC2Accesso : aggiornamento a una politica esistente	È stata aggiunta l'autorizzazione richiesta per gli scenari «AZ: Application Slowdown» e «Cross-AZ: Traffic Slowdown». Le autorizzazioni sono: ec2: DescribeSubnets	12 novembre 2025
AWSFaultInjectionSimulatorEC2Accesso : aggiornamento di una policy esistente	Sono state aggiunte le autorizzazioni necessarie per gli scenari «AZ: Application Slowdown» e «Cross-AZ: Traffic Slowdown». Le autorizzazioni sono: ecs:, ec2: ed ec2: DescribeContainerInstances DescribeSubnets DescribeInstances	12 novembre 2025

Modifica	Descrizione	Data
AWSFaultInjectionSimulatorECSAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire al AWS FIS di risolvere gli obiettivi ECS.	25 gennaio 2024
AWSFaultInjectionSimulatorNetworkAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di eseguire esperimenti utilizzando le azioni <code>aws:network:route-table-disrupt-cross-region-connectivity</code> e <code>aws:network:transit-gateway-disrupt-cross-region-connectivity</code> .	25 gennaio 2024
AWSFaultInjectionSimulatorEC2Access : aggiornamento a una politica esistente	Autorizzazioni aggiunte per consentire a AWS FIS di risolvere le istanze EC2.	13 novembre 2023
AWSFaultInjectionSimulatorEKSAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire a FIS di risolvere gli obiettivi AWS EKS.	13 novembre 2023
AWSFaultInjectionSimulatorRDSAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di risolvere gli obiettivi RDS.	13 novembre 2023
AWSFaultInjectionSimulatorEC2Access : aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di eseguire documenti SSM su istanze EC2 e di terminare le istanze EC2.	2 giugno 2023
AWSFaultInjectionSimulatorSSMAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire AWS a FIS di eseguire documenti SSM su istanze EC2.	2 giugno 2023

Modifica	Descrizione	Data
AWSFaultInjectionSimulatorECSAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di eseguire esperimenti utilizzando le nuove azioni. <code>aws:ecs:task</code>	1 giugno 2023
AWSFaultInjectionSimulatorEKSAccess : aggiornamento di una policy esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di eseguire esperimenti utilizzando le nuove azioni. <code>aws:eks:pod</code>	1 giugno 2023
AWSFaultInjectionSimulatorEC2Access : nuova politica	È stata aggiunta una policy per consentire a AWS FIS di eseguire un esperimento che utilizza azioni AWS FIS per Amazon EC2.	26 ottobre 2022
AWSFaultInjectionSimulatorCSAccess : nuova policy	È stata aggiunta una policy per consentire a AWS FIS di eseguire un esperimento che utilizza azioni AWS FIS per Amazon ECS.	26 ottobre 2022
AWSFaultInjectionSimulatorKSAccess : nuova policy	È stata aggiunta una policy per consentire a AWS FIS di eseguire un esperimento che utilizza azioni AWS FIS per Amazon EKS.	26 ottobre 2022
AWSFaultInjectionSimulatorNetworkAccess : nuova policy	È stata aggiunta una policy per consentire a AWS FIS di eseguire un esperimento che utilizza azioni di rete AWS FIS.	26 ottobre 2022
AWSFaultInjectionSimulatorRDSAccess : nuova policy	È stata aggiunta una policy per consentire a AWS FIS di eseguire un esperimento che utilizza azioni AWS FIS per Amazon RDS.	26 ottobre 2022

Modifica	Descrizione	Data
AWSFaultInjectionSimulatorSMAccess : nuova policy	È stata aggiunta una policy per consentire a AWS FIS di eseguire un esperimento che utilizza azioni AWS FIS per Systems Manager.	26 ottobre 2022
Amazon FISService RolePolicy : aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di descrivere le sottoreti.	26 ottobre 2022
Amazon FISService RolePolicy : aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di descrivere i cluster EKS.	7 luglio 2022
Amazon FISService RolePolicy : aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di elencare e descrivere le attività nei cluster.	7 febbraio 2022
Amazon FISService RolePolicy : aggiornamento a una politica esistente	È stata rimossa la <code>events:ManagedBy</code> condizione per l' <code>events:DescribeRule</code> azione.	6 gennaio 2022
Amazon FISService RolePolicy : aggiornamento a una politica esistente	Sono state aggiunte autorizzazioni per consentire a AWS FIS di recuperare la cronologia degli CloudWatch allarmi utilizzati in condizioni di arresto.	30 giugno 2021
AWS FIS ha iniziato a tenere traccia delle modifiche	AWS La FIS ha iniziato a tenere traccia delle modifiche alle sue politiche AWS gestite	1° marzo 2021

Sicurezza dell'infrastruttura in AWS Fault Injection Service

In quanto servizio gestito, AWS Fault Injection Service è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS](#)

[Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere a AWS FIS attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Accedere a AWS FIS utilizzando un'interfaccia VPC endpoint ([AWS PrivateLink](#))

Puoi stabilire una connessione privata tra il tuo VPC e il servizio AWS Fault Injection creando un endpoint VPC di interfaccia. Gli endpoint VPC sono alimentati da [AWS PrivateLink](#) una tecnologia che consente di accedere privatamente al AWS FIS APIs senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione Direct Connect. AWS Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per comunicare con AWS FIS. APIs

Ogni endpoint di interfaccia è rappresentato da una o più [interfacce di rete elastiche](#) nelle sottoreti.

Per ulteriori informazioni, consulta [Access Servizi AWS through AWS PrivateLink nella Guida](#).AWS PrivateLink

Considerazioni sugli endpoint AWS VPC FIS

Prima di configurare un endpoint VPC di interfaccia per AWS FIS, consulta [Access an using Servizio AWS an interface VPC](#) endpoint nella Guida.AWS PrivateLink

AWS FIS supporta l'esecuzione di chiamate a tutte le sue azioni API dal tuo VPC.

Crea un endpoint VPC di interfaccia per FIS AWS

Puoi creare un endpoint VPC per il servizio AWS FIS utilizzando la console Amazon VPC o ([AWS Command Line Interface AWS CLI](#)). Per ulteriori informazioni, consulta la sezione [Creazione di un endpoint VPC](#) nella Guida di AWS PrivateLink .

Crea un endpoint VPC per AWS FIS utilizzando il seguente nome di servizio:

`com.amazonaws.region.fis`

Se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API a AWS FIS utilizzando il nome DNS predefinito per la regione, per esempio, `fis.us-east-1.amazonaws.com`

Crea una policy per gli endpoint VPC per FIS AWS

Puoi allegare una policy per gli endpoint all'endpoint VPC che controlla l'accesso al FIS. AWS La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire azioni.

Per ulteriori informazioni, consulta [Control access to VPC endpoints using endpoint policies](#) nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per azioni FIS specifiche AWS

La seguente policy sugli endpoint VPC concede l'accesso alle azioni AWS FIS elencate su tutte le risorse a tutti i principali attori.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "fis:ListExperimentTemplates",
        "fis:StartExperiment",
        "fis:StopExperiment",
        "fis:GetExperiment"
      ],
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

Esempio: policy degli endpoint VPC che nega l'accesso da uno specifico Account AWS

La seguente politica degli endpoint VPC nega l' Account AWS accesso specificato a tutte le azioni e le risorse, ma concede a tutti gli altri Account AWS accessi a tutte le azioni e risorse.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Principal": {
        "AWS": [ "123456789012" ]
      }
    }
  ]
}
```

Etichettatura delle risorse AWS FIS

Un tag è un'etichetta di metadati che tu o AWS assegnate a una AWS risorsa. Ciascun tag è formato da una chiave e da un valore. Per i tag assegnati da te, puoi definire la chiave e il valore. Ad esempio, è possibile definire la chiave come `purpose` e il valore come `test` per una risorsa.

I tag consentono di:

- Identifica e organizza AWS le tue risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate.
- Controlla l'accesso alle tue risorse. AWS Per ulteriori informazioni, consulta [Controllo dell'accesso alle risorse AWS mediante i tag](#) nella Guida per l'utente di .

Restrizioni di tagging

Le seguenti restrizioni di base si applicano ai tag sulle risorse AWS FIS:

- Numero massimo di tag che è possibile assegnare a una risorsa: 50
- lunghezza massima della chiave: 128 caratteri Unicode;
- lunghezza massima del valore: 256 caratteri Unicode;
- Caratteri validi per chiavi e valori: a-z, A-Z, 0-9, spazio e i seguenti caratteri: `_`:`/`= `+` `-` e `@`
- Per chiavi e valori viene fatta distinzione tra maiuscole e minuscole
- Non è possibile utilizzarlo `aws:` come prefisso per le chiavi, poiché è riservato all'uso AWS

Lavora con i tag

Le seguenti risorse del AWS AWS Fault Injection Service (FIS) supportano l'etichettatura:

- Operazioni
- Esperimenti
- Modelli di esperimenti

È possibile utilizzare la console per lavorare con i tag per esperimenti e i modelli di esperimenti. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Etichetta un esperimento](#)
- [Modelli di esperimenti con tag](#)

Puoi usare i seguenti AWS CLI comandi per lavorare con i tag per azioni, esperimenti e modelli di esperimenti:

- [tag-resource](#): aggiunge tag a una risorsa.
- [untag-resource](#) — Rimuove i tag da una risorsa.
- [list-tags-for-resource](#)— Elenca i tag per una risorsa specifica.

Quote e limitazioni per il servizio AWS Fault Injection

Hai Account AWS delle quote predefinite, precedentemente denominate limiti, per ogni servizio. AWS Salvo diversa indicazione, ogni quota è specifica della regione. È possibile richiedere aumenti per le quote contrassegnate come regolabili nella tabella seguente.

Per visualizzare le quote per AWS FIS nel tuo account, apri la console [Service Quotas](#). Nel riquadro di navigazione, scegli AWS servizi e seleziona AWS Fault Injection Service. I valori fino alle quote approvate automaticamente vengono applicati immediatamente. Le quote approvate automaticamente sono descritte nella colonna di descrizione della tabella seguente. Se hai bisogno di quote che superano i limiti approvati automaticamente, invia una richiesta. I valori superiori ai limiti approvati automaticamente vengono esaminati dall'assistenza clienti e approvati ove possibile.

Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

Hai Account AWS le seguenti quote relative alla AWS FIS.

Name	Predefinita	Adattate	Description
Durata operazione in ore	Ogni regione supportata: 12	No	Il numero massimo di ore consentite per eseguire un'operazione in questo account nella regione corrente.
Minuti di azione per tabella globale Multi-region Strongly Consistent (MRSC) per aws:dynamodb: action global-table-pause-replication	Ogni regione supportata: 5.040	Sì	Il numero massimo di minuti di azione cumulativi per tabella globale MRSC a cui aws:dynamodb: può indirizzarsi in una finestra di 7 giorni. global-table-pause-replication
Operazioni per modello di esperimento	Ogni regione supportata: 20	No	Il numero massimo di operazioni che è possibile

Name	Predefinita	Adattate	Description
			creare in un modello di esperimento in questo account nella regione corrente.
Esperimenti attivi	Ogni Regione supportata: 5	No	Il numero massimo di esperimenti attivi che è possibile eseguire simultaneamente in questo account nella regione corrente.
Conservazione dei dati degli esperimenti completati in giorni	Ogni regione supportata: 120	No	Il numero massimo di giorni consentito alla AWS FIS per conservare i dati sugli esperimenti completati in questo account nella regione corrente.
Durata esperimento in ore	Ogni regione supportata: 12	No	Il numero massimo di ore consentite per eseguire un esperimento in questo account nella regione corrente.
Modelli di esperimenti	Ogni regione supportata: 500	No	Il numero massimo di modelli di esperimento che è possibile creare in questo account nella regione corrente.

Name	Predefinita	Adattate	Description
Numero massimo di elenchi di prefissi gestiti in <code>aws:network: -region-connectivity route-table-disrupt-cross</code>	Ogni regione supportata: 15	No	Il numero massimo di elenchi di prefissi gestiti che <code>aws:network: -region-connectivity</code> consentirà, per azione. <code>route-table-disrupt-cross</code>
Numero massimo di tabelle di routing in <code>aws:network: -region-connectivity route-table-disrupt-cross</code>	Ogni regione supportata: 10	No	Il numero massimo di tabelle di routing consentito da <code>aws:network: route-table-disrupt-cross -region-connectivity</code> , per azione.
Numero massimo di percorsi in <code>aws:network: -region-connectivity route-table-disrupt-cross</code>	Ogni Regione supportata: 200	No	Il numero massimo di percorsi consentiti da <code>aws:network: route-table-disrupt-cross -region-connectivity</code> , per azione.
Operazioni parallele per esperimento	Ogni regione supportata: 10	No	Il numero massimo di operazioni che è possibile eseguire in parallelo in un esperimento in questo account nella regione corrente.
Condizioni di interruzione per modello di esperimento	Ogni Regione supportata: 5	No	Il numero massimo di condizioni di arresto che è possibile aggiungere a un modello di esperimento in questo account nella regione corrente.

Name	Predefinita	Adattate	Description
Scegli come target i cluster DSQL Aurora per aws:dsq: cluster-connection-failure	Ogni regione supportata: 100	Sì	Il numero massimo di cluster Aurora DSQL a cui aws:dsq: cluster-connection-failure può indirizzare quando si identificano gli obiettivi utilizzando i tag, per esperimento.
Gruppi Target Auto Scaling per aws:ec2: -error asg-insufficient-instance-capacity	Ogni regione supportata: 500	Sì	Il numero massimo di gruppi di Auto Scaling a cui aws:ec2: asg-insufficient-instance-capacity -error può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.
Target Bucket per aws:s3: bucket-pause-replication	Ogni regione supportata: 25	Sì	Il numero massimo di bucket S3 a cui aws:s3: bucket-pause-replication può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.
Cluster di destinazione per aws:ecs: drain-container-instances	Ogni regione supportata: 100	Sì	Il numero massimo di cluster a cui aws:ecs: drain-container-instances può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.

Name	Predefinita	Adatta e	Description
Cluster di destinazione per aws:rds: failover-db-cluster	Ogni regione supportata: 160	Sì	Il numero massimo di cluster a cui aws:rds: failover-db-cluster può indirizzare quando si identificano gli obiettivi utilizzando i tag, per esperimento.
Obiettivo per aws:rds: DBInstances reboot-db-instances	Ogni regione supportata: 130	Sì	Il numero massimo di bersagli DBInstances che aws:rds: reboot-db-instances può avere come target quando si identificano i bersagli utilizzando i tag, per esperimento.
Istanze di destinazione per aws:ec2:reboot-instances	Ogni regione supportata: 600	Sì	Il numero massimo di cluster a cui aws:ec2:reboot-instances può indirizzare quando identifichi le destinazioni tramite i tag, per esperimento.
Istanze di destinazione per aws:ec2:s top-instances	Ogni regione supportata: 400	Sì	Il numero massimo di istanze a cui aws:ec2:s top-instances può indirizzare quando identifichi le destinazioni tramite i tag, per esperimento.

Name	Predefinita	Adattate	Description
Istanze di destinazione per aws:ec2:terminated-instances	Ogni Regione supportata: 300	Sì	Il numero massimo di istanze a cui aws:ec2:terminated-instances può indirizzare quando identifichi le destinazioni tramite i tag, per esperimento.
Istanze di destinazione per aws:ssm:send-command	Ogni Regione supportata: 200	Sì	Il numero massimo di istanze a cui aws:ssm:send-command può indirizzare quando identifichi le destinazioni tramite i tag, per esperimento.
Scegli come target i flussi di dati Kinesis per aws:kinesis: action. stream-expired-iterator-exception	Ogni regione supportata: 280	Sì	Il numero massimo di flussi di dati Kinesis che aws:kinesis: stream-expired-iterator-exception può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.
Scegli come target i flussi di dati Kinesis per aws:kinesis: action. stream-provisioned-throughput-exception	Ogni regione supportata: 280	Sì	Il numero massimo di flussi di dati Kinesis che aws:kinesis: stream-provisioned-throughput-exception può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.

Name	Predefinita	Adattate	Description
Obiettivo ManagedResources per aws:arc: action. start-zonal-autoshift	Ogni Regione supportata: 200	Sì	Il numero massimo di risorse gestite a cui aws:arc: start-zonal-autoshift può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.
Target Nodegroups per aws:eks: terminate-nodegroup-instances	Ogni regione supportata: 100	Sì	Il numero massimo di Nodegroup a cui aws:eks: terminate-nodegroup-instances può indirizzare quando identifichi i target utilizzando i tag, per esperimento.
Target Pods per aws:eks: pod-cpu-stress	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks: pod-cpu-stress può scegliere come target quando identifichi i bersagli utilizzando i parametri, per esperimento.
Target Pods per aws:eks:pod-delete	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks:pod-delete può utilizzare come target quando si identificano i bersagli utilizzando i parametri, per esperimento.

Name	Predefinita	Adattate	Description
Target Pods per aws:eks: pod-io-stress	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks: pod-io-stress può scegliere come target quando identifica i bersagli utilizzando i parametri, per esperimento.
Target Pods per aws:eks: pod-memory-stress	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks: pod-memory-stress può scegliere come target quando identifica i bersagli utilizzando i parametri, per esperimento.
Target Pods per aws:eks: pod-network-blackhole-port	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks: pod-network-blackhole-port può scegliere come target quando identifica i bersagli utilizzando i parametri, per esperimento.
Target Pods per aws:eks: pod-network-latency	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks: pod-network-latency può scegliere come target quando identifica i bersagli utilizzando i parametri, per esperimento.

Name	Predefinita	Adattate	Description
Target Pods per aws:eks: pod-network-packet-loss	Ogni regione supportata: 1.000	Sì	Il numero massimo di Pod che aws:eks: pod-network-packet-loss può scegliere come target quando identifica i bersagli utilizzando i parametri, per esperimento.
Obiettivo ReplicationGroups per aws:elasticache: - Deprecazione pianificata interrupt-cluster-az-power	Ogni regione supportata: 5	Sì	Il numero massimo di bersagli ReplicationGroups che aws:elasticache: interrupt-cluster-az-power può prendere di mira quando si identificano gli obiettivi utilizzando tag/parametri, per esperimento.
ReplicationGroups Obiettivo per aws:elasticache: replicationgroup-interrupt-az-power	Ogni regione supportata: 20	Sì	Il numero massimo di bersagli ReplicationGroups che aws:elasticache: può avere come target per esperimento. replicationgroup-interrupt-az-power Al targeting si applica un limite giornaliero. ReplicationGroups Per ulteriori informazioni, visita: https://docs.aws.amazon.com/fis/latest/userguide/fis-quotas.html .

Name	Predefinita	Adattate	Description
Target SpotInstances per aws:ec2: send-spot-instance-interruptions	Ogni regione supportata: 100	Sì	Il numero massimo di bersagli SpotInstances che aws:ec2: send-spot-instance-interruptions può avere come target quando si identificano gli obiettivi utilizzando i tag, per esperimento.
Sottoreti di destinazione per aws:network: disrupt-connectivity	Ogni regione supportata: 100	Sì	Il numero massimo di sottoreti a cui aws:network: disrupt-connectivity può indirizzare quando identifichi le destinazioni tramite i tag, per esperimento. Le quote superiori a 5 si applicano solo al parametro scope:all. Se hai bisogno di una quota più alta per un altro tipo di oscilloscopio, contatta l'assistenza clienti.
Sottoreti Target per aws:network: - region-connectivity route-table-disrupt-cross	Ogni Regione supportata: 50	Sì	Il numero massimo di sottoreti a cui aws:network: route-table-disrupt-cross -region-connectivity può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.

Name	Predefinita	Adattate	Description
Attività di destinazione per aws:ecs:s-top-task	Ogni regione supportata: 500	Sì	Il numero massimo di istanze a cui aws:ecs:s-top-task può indirizzare quando identifichi le destinazioni tramite i tag, per esperimento.
Target Tasks per aws:ecs: task-cpu-stress	Ogni Regione supportata: 200	Sì	Il numero massimo di attività a cui aws:ecs: task-cpu-stress può indirizzare quando identifichi obiettivi utilizzando tag/parametri, per esperimento.
Target Tasks per aws:ecs: task-io-stress	Ogni Regione supportata: 200	Sì	Il numero massimo di attività a cui aws:ecs: task-io-stress può indirizzare quando identifichi obiettivi utilizzando tag/parametri, per esperimento.
Target Tasks per aws:ecs: task-kill-process	Ogni Regione supportata: 200	Sì	Il numero massimo di attività a cui aws:ecs: task-kill-process può indirizzare quando identifichi obiettivi utilizzando tag/parametri, per esperimento.

Name	Predefinita	Adattate	Description
Target Tasks per aws:ecs: task-network-blackhole-port	Ogni Regione supportata: 200	Sì	Il numero massimo di attività a cui aws:ecs: task-network-blackhole-port può indirizzare quando identifichi obiettivi utilizzando tag/parametri, per esperimento.
Target Tasks per aws:ecs: task-network-latency	Ogni Regione supportata: 200	Sì	Il numero massimo di attività a cui aws:ecs: task-network-latency può indirizzare quando identifichi obiettivi utilizzando tag/parametri, per esperimento.
Target Tasks per aws:ecs: task-network-packet-loss	Ogni Regione supportata: 200	Sì	Il numero massimo di attività a cui aws:ecs: task-network-packet-loss può indirizzare quando identifichi obiettivi utilizzando tag/parametri, per esperimento.
Target per aws:network: -region-connectivity TransitGateways transit-gateway-disrupt-cross	Ogni Regione supportata: 50	Sì	Il numero massimo di gateway di transito che aws:network: transit-gateway-disrupt-cross -region-connectivity può utilizzare come target quando identifichi gli obiettivi utilizzando i tag, per esperimento.

Name	Predefinita	Adattate	Description
Interfacce virtuali di destinazione per <code>aws:directconnect:virtual-interface</code>	Ogni regione supportata: 5	<u>Si</u>	Il numero massimo di interfacce virtuali che <code>aws:directconnect:virtual-interface</code> può utilizzare e come target quando identifichi gli obiettivi utilizzando i tag, per esperimento.
Volumi di destinazione per <code>aws:ebs:pause-volume-io</code>	Ogni regione supportata: 160	<u>Si</u>	Il numero massimo di volumi a cui <code>aws:ebs:pause-volume-io</code> può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.
Configurazioni degli account Target per modello di esperimento	Ogni regione supportata: 40	<u>Si</u>	Il numero massimo di configurazioni di account di destinazione che è possibile creare per un modello di esperimento in questo account nella regione corrente.
Funzioni target per <code>aws:lambda:action.invocation-add-delay</code>	Ogni regione supportata: 140	<u>Si</u>	Il numero massimo di funzioni Lambda a cui <code>aws:lambda: invocation-add-delay</code> può indirizzare quando si identificano gli obiettivi utilizzando i tag, per esperimento.

Name	Predefinita	Adattate	Description
Funzioni di destinazione per l'azione <code>aws:lambda:invocation-error</code> .	Ogni regione supportata: 140	Sì	Il numero massimo di funzioni Lambda a cui <code>aws:lambda:invocation-error</code> può indirizzare quando identifichi gli obiettivi utilizzando i tag, per esperimento.
<code>invocation-http-integration-response</code> Funzioni target per <code>aws:lambda: action</code> .	Ogni regione supportata: 140	Sì	Il numero massimo di funzioni Lambda a cui <code>aws:lambda: invocation-http-integration-response</code> può indirizzare quando si identificano gli obiettivi utilizzando i tag, per esperimento.
Scegli come target i cluster multiregionali per l'azione <code>aws:memorydb: -replication. multi-region-cluster-pause</code>	Ogni Regione supportata: 50	Sì	Il numero massimo di cluster MemoryDB multi-regione a cui <code>aws:memorydb: -replication</code> può indirizzare quando identifichi i target utilizzando i tag, per esperimento. <code>multi-region-cluster-pause</code> Se hai bisogno di una quota più alta, contatta l'assistenza clienti.

Name	Predefinita	Adattate	Description
Tabelle di destinazione per aws:dynamodb: action global-table-pause-replication	Ogni regione supportata: 60	Sì	Il numero massimo di tabelle globali che aws:dynamodb: può indirizzare, per esperimento. global-table-pause-replication

L'utilizzo di AWS FIS è soggetto alle seguenti limitazioni aggiuntive:

Name	Limitazione
Limite giornaliero sugli obiettivi ReplicatedOnGroups per aws:elasticache: replicatedongroup-interrupt-az-power	Il limite è di 20 ReplicatedOnGroups obiettivi per account per regione al giorno. Puoi richiedere un aumento creando un caso di supporto nella console di AWS Support Center .

Cronologia dei documenti

La tabella seguente descrive gli importanti aggiornamenti della documentazione nella Guida per l'utente del servizio AWS Fault Injection.

Modifica	Descrizione	Data
Nuova azione per AWS Direct Connect	Puoi utilizzare l'azione <code>aws:directconnect:virtual-interface</code> per testare la resilienza della AWS Direct Connect connessione interrompendo temporaneamente le sessioni del Border Gateway Protocol tra le reti locali e i peer associati alle interfacce virtuali di destinazione.	18 dicembre 2025
Nuovi scenari	Ora puoi utilizzare i nuovi scenari «AZ: Application Slowdown» e «Cross-AZ: Traffic Slowdown».	12 novembre 2025
Nuovo parametro per i documenti SSM	I documenti SSM <code>AWSFIS-Run-Network-Latency-Sources</code> e <code>-Network-Packet-Loss-Sources</code> ora supportano il parametro <code>AWSFIS-RunFlowsPercent</code> .	12 novembre 2025
Nuovo parametro per le azioni ECS ed EKS	Le azioni <code>aws:ecs:task-network-latency</code> , <code>aws:ecs:</code> , <code>aws:eks: pod-network-latency</code> e <code>aws:eks: task-network-packet-loss</code> ora supportano il parametro <code>FlowsPercent</code> . <code>pod-network-packet-loss</code>	12 novembre 2025

AWSaggiornamenti delle politiche gestiti	AWSFaultInjectionSimulatorE C2Accesso alla politica gestita aggiornata: aggiunto ec2: DescribeSubnets autorizzazione.	12 novembre 2025
AWSaggiornamenti delle politiche gestiti	Politica gestita aggiornata AWSFaultInjectionSimulatorE CSAccess : aggiunte le autorizzazioni ecs: DescribeContainerInstances, ec2: DescribeSubnets ed ec2: DescribeInstances	12 novembre 2025
Nuove azioni per Amazon Kinesis Data Streams	Puoi usare le azioni aws:kinesis:stream.	15 ottobre 2025
Nuovo esperimento per Amazon EBS	Puoi utilizzare l'volume-io-latencyazione aws:ebs: per simulare una latenza elevata I/O sui tuoi volumi Amazon EBS.	16 settembre 2025
Nuovo ambito d'azione AWS FIS	Puoi utilizzare l'azione aws:network:disrupt-connectivity per interrompere la connettività ai bucket di directory S3 Express One Zone. Questo ambito è ora incluso anche nello scenario AZ Availability: Power Interruption.	18 agosto 2025

Supporto MemoryDB in FIS AWS	Puoi utilizzare AWS FIS per testare in che modo la tua applicazione con i cluster multiregione di Amazon MemoryDB risponde alla sospensione della replica dei dati durante un'interruzione della rete tra regioni.	15 luglio 2025
Supporto ARC in AWS FIS	È possibile utilizzare AWS FIS per verificare in che modo ARC zonal autoshift ripristina automaticamente l'applicazione durante un'interruzione di corrente in AZ.	26 marzo 2025
Nuova configurazione del rapporto sull'esperimento	Ora puoi consentire a AWS FIS di generare report per esperimenti che riassumono le azioni e le risposte degli esperimenti dai CloudWatch dashboard.	12 novembre 2024
Nuove azioni Lambda	Ora puoi usare le azioni <code>aws:lambda:function</code> per inserire errori nelle chiamate delle tue funzioni Lambda.	31 ottobre 2024
Nuova funzione di leva di sicurezza	AWSLa FIS ora supporta leve di sicurezza che consentono di interrompere rapidamente tutti gli esperimenti in corso e impedire l'avvio di nuovi esperimenti.	3 settembre 2024

Nuovo capitolo sulla risoluzione dei problemi	AWS FIS ha aggiunto una guida alla risoluzione dei problemi che include codici di errore e contesto per gli esperimenti falliti.	13 agosto 2024
Nuova azione	È ora possibile utilizzare l' <code>aws:dynamodb:global-table-pause-replication</code> azione per sospendere la replica dei dati tra la tabella globale di destinazione e le relative tabelle di replica. L' <code>aws:dynamodb:encrypted-global-table-pause-replication</code> azione non sarà più supportata.	24 aprile 2024
Nuova opzione sperimentale in modalità azioni	È possibile impostare la modalità azioni in modo <code>skip-all</code> da generare un'anteprima del bersaglio prima di eseguire un esperimento.	13 marzo 2024
AWSaggiornamenti delle politiche gestite	AWS FIS ha aggiornato le politiche gestite esistenti.	25 gennaio 2024
Nuovi scenari e azioni	Ora puoi utilizzare gli scenari AWS FIS Cross-Region:Connectivity e AZ Availability: Power Interruption.	30 novembre 2023

Nuova azione	Ora puoi usare l'aws:ec2:asg-insufficient-instance-capacity-errorazione.	30 novembre 2023
Nuova azione	Ora puoi usare l'aws:ec2:api-insufficient-instance-capacity-errorazione.	30 novembre 2023
Nuova azione	Ora puoi usare l'aws:network:route-table-disrupt-cross-region-connectivityazione.	30 novembre 2023
Nuova azione	Ora puoi usare l'aws:network:transit-gateway-disrupt-cross-region-connectivityazione.	30 novembre 2023
Nuova azione	Ora puoi usare l'aws:dynamodb:encrypted-global-table-pause-replicationazione.	30 novembre 2023
Nuova azione	Ora puoi usare l'aws:s3:bucket-pause-replicationazione.	30 novembre 2023
Nuova azione	Ora puoi usare l'aws:elasticache:interrupt-cluster-az-powerazione.	30 novembre 2023
Nuove opzioni di esperimento	Ora puoi utilizzare le opzioni sperimentali AWS FIS per il targeting degli account e la risoluzione degli obiettivi vuoti.	27 novembre 2023
Cambio di nome del FIS AWS	Nome del servizio aggiornato a AWS Fault Injection Service.	15 novembre 2023
AWSaggiornamenti delle politiche gestiti	AWSFIS ha aggiornato le politiche gestite esistenti.	13 novembre 2023

Nuova libreria di scenari	È ora possibile utilizzare la funzionalità di libreria AWS di scenari FIS.	7 novembre 2023
Nuovo programmatore di esperimenti	È ora possibile utilizzare la funzionalità di pianificazione degli esperimenti AWS FIS.	7 novembre 2023
AWSaggiornamenti delle politiche gestiti	AWSFIS ha aggiornato le politiche gestite esistenti.	2 giugno 2023
Nuove azioni	Puoi usare le nuove aws:ecs:task aws:eks:pod azioni.	1 giugno 2023
AWSaggiornamenti delle politiche gestiti	AWSFIS ha aggiornato le politiche gestite esistenti.	1 giugno 2023
Nuovo documento SSM preconfigurato	È possibile utilizzare il seguente documento SSM preconfigurato: -Disk-Fill. AWSFIS-Run	28 aprile 2023
Nuova azione	È possibile utilizzare l'aws:ebs:pause-volume-azione per fare una pausa I/O tra i volumi di destinazione e le istanze a cui sono collegati.	27 gennaio 2023
Nuova azione	È possibile utilizzare l'aws:network:disrupt-connectivityazione per negare tipi specifici di traffico alle sottoreti di destinazione.	26 ottobre 2022

Nuova azione	È possibile utilizzare l'aws:eks:inject-kubernetes-custom-resourceazione per eseguire un esperimento ChaosMesh or Litmus su un singolo cluster di destinazione.	7 luglio 2022
Registrazione dell'esperimento	Puoi configurare i modelli di esperimento per inviare i registri delle attività degli esperimenti ai CloudWatch registri o a un bucket S3.	28 febbraio 2022
Nuove notifiche	Quando lo stato di un esperimento cambia, AWS FIS emette una notifica. Queste notifiche vengono rese disponibili come eventi tramite Amazon EventBridge.	24 febbraio 2022
Nuova azione	È possibile utilizzare l'aws:ecs:stop-taskazione per interrompere l'attività specificata.	9 febbraio 2022
Nuova azione	È possibile utilizzare l'aws:cloudwatch:assert-alarm-stateazione per verificare che gli allarmi specificati si trovino in uno degli stati di allarme specificati.	5 novembre 2021

Nuovi documenti SSM preconfigurati	È possibile utilizzare i seguenti documenti SSM preconfigurati: AWSFIS-Run -IO-Stress, -Network-Blackhold-Port, -Network-Latency-Sources, -Network-Packet-Loss e AWSFIS-Run -Network-Packet-Loss-Sources. AWSFIS-Run AWSFIS-Run AWSFIS-Run	4 novembre 2021
Nuova azione	È possibile utilizzare l'aws:ec2:send-spot-instance-interruptionsazione per inviare un avviso di interruzione dell'istanza Spot alle istanze Spot di destinazione e quindi interrompere le istanze Spot di destinazione.	20 ottobre 2021
Nuova azione	È possibile utilizzare l'aws:ssm:start-automation-executionazione per avviare l'esecuzione di un runbook di automazione.	17 settembre 2021
Versione iniziale	La versione iniziale della Guida per l'utente del servizio AWS Fault Injection.	15 marzo 2021

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.