

Guida all'implementazione

Test di carico distribuito su AWS



Test di carico distribuito su AWS: Guida all'implementazione

Copyright © 2024 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 delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Panoramica della soluzione	1
Funzionalità	2
Vantaggi	3
Casi d'uso	4
Concetti e definizioni	5
Panoramica dell'architettura	7
Diagramma architetturale	7
Considerazioni sulla progettazione di AWS Well-Architected	9
Eccellenza operativa	9
Sicurezza	9
Affidabilità	10
Efficienza delle prestazioni	10
Ottimizzazione dei costi	10
Sostenibilità	11
Dettagli architettonici	12
Front-end	12
API per il test di carico	12
Console Web	12
Backend	13
Pipeline di immagini del contenitore	13
Infrastruttura di test	13
Motore di test di carico	14
Servizi AWS in questa soluzione	14
Come funziona il test di carico distribuito su AWS	16
Considerazioni di natura progettuale	18
Applicazioni supportate	18
JMeter supporto per script	18
Pianificazione dei test	19
Test simultanei	20
Gestione degli utenti	20
Implementazione regionale	20
Pianifica la tua implementazione	21
Costo	21
Sicurezza	22

Ruoli IAM	22
Amazon CloudFront	23
Gruppo di sicurezza AWS Fargate	23
Test di stress della rete	23
Limitazione dell'accesso all'interfaccia utente pubblica	23
Regioni AWS supportate	24
Quote	24
Quote per i servizi AWS in questa soluzione	24
CloudFormation Quote AWS	25
Quote di test di carico	25
Test simultanei	20
Politica di EC2 test di Amazon	26
Politica di test CloudFront di carico di Amazon	26
Implementa la soluzione	27
Panoramica del processo di distribuzione	27
CloudFormation Modello AWS	27
Avvio dello stack	28
Implementazione in più regioni	31
Monitora la soluzione con Service Catalog AppRegistry	35
Attiva Application Insights CloudWatch	35
Conferma i cartellini dei costi associati alla soluzione	37
Attiva i tag di allocazione dei costi associati alla soluzione	38
AWS Cost Explorer	39
Aggiornare la soluzione	40
Quando si esegue l'aggiornamento da versioni DLT precedenti alla v3.2.6 alla versione più recente, l'aggiornamento dello stack non riesce	41
Risoluzione dei problemi	43
Risoluzione di problemi noti	43
Contattare AWS Support	43
Crea un caso	44
Come possiamo aiutarti?	44
Informazioni aggiuntive	44
Aiutaci a risolvere il tuo caso più velocemente	44
Risolvi subito o contattaci	44
Disinstalla la soluzione	45
Utilizzando la Console di gestione AWS	45

Utilizzo dell'interfaccia a riga di comando AWS	45
Eliminazione dei bucket Amazon S3	45
Usa la soluzione	47
Risultati del test	47
Flusso di lavoro di pianificazione dei test	48
Determina il numero di utenti	48
Dati in tempo reale	49
Flusso di lavoro di annullamento	50
Guida per sviluppatori	51
Codice sorgente	51
Personalizzazione dell'immagine del contenitore	51
API di test di carico distribuita	58
GET /scenarios	59
POST /scenari	60
OPZIONI/scenari	62
GET /scenarios/ {testID}	62
POST /scenarios/ {testID}	64
DELETE /scenarios/ {testID}	64
OPZIONI /scenarios/ {testID}	65
OTTIENI /tasks	66
OPZIONI/task	66
GET /regions	67
OPZIONI/regioni	68
Aumenta le risorse del contenitore	68
Crea una nuova revisione della definizione delle attività	69
Aggiornare la tabella DynamoDB	69
Documentazione di riferimento	70
Raccolta di dati anonimizzata	70
Collaboratori	71
Revisioni	72
Note	73
.....	Ixxiv

Automatizza il test delle tue applicazioni software su larga scala

Data di pubblicazione: novembre 2019

Distributed Load Testing on AWS ti aiuta ad automatizzare i test delle tue applicazioni software su larga scala e in fase di carico per identificare i colli di bottiglia prima di rilasciare l'applicazione. Questa soluzione crea e simula migliaia di utenti connessi che generano record transazionali a un ritmo costante senza la necessità di fornire server.

Questa soluzione sfrutta [Amazon Elastic Container Service \(Amazon ECS\) su AWS Fargate](#) per distribuire contenitori in grado di eseguire tutte le simulazioni e offre le seguenti funzionalità:

- Implementa Amazon ECS su contenitori AWS Fargate che possono essere eseguiti indipendentemente per testare le capacità di carico del software in fase di test.
- Simula decine di migliaia di utenti connessi, in più regioni AWS, generando record transazionali a un ritmo continuo.
- [Personalizza i test delle tue applicazioni creando script personalizzati. JMeter](#)
- Pianifica i test di carico in modo che inizino automaticamente in date future o in date ricorrenti.
- Esegui i test di carico delle applicazioni contemporaneamente o esegui più test contemporaneamente.

Questa guida all'implementazione fornisce una panoramica della soluzione Distributed Load Testing on AWS, della sua architettura e dei suoi componenti di riferimento, considerazioni per la pianificazione della distribuzione, i passaggi di configurazione per la distribuzione della soluzione nel cloud Amazon Web Services (AWS). Include collegamenti a un CloudFormation modello [AWS](#) che avvia e configura i servizi AWS necessari per distribuire questa soluzione utilizzando le best practice di AWS per la sicurezza e la disponibilità.

Il pubblico previsto per l'utilizzo delle caratteristiche e delle funzionalità di questa soluzione nel proprio ambiente include architetti di infrastrutture IT, amministratori e DevOps professionisti con esperienza pratica di architettura nel cloud AWS.

Utilizza questa tabella di navigazione per trovare rapidamente le risposte a queste domande:

Se vuoi.	Leggi..
Conosci il costo di esecuzione di questa soluzione.	Costo
Il costo stimato per l'esecuzione di questa soluzione nella regione Stati Uniti orientali (Virginia settentrionale) è di 30,90 USD al mese per le risorse AWS.	
Comprendi le considerazioni sulla sicurezza relative a questa soluzione.	Sicurezza
Scopri come pianificare le quote per questa soluzione.	Quote
Scopri quali regioni AWS supportano questa soluzione.	Regioni AWS supportate
Visualizza o scarica il CloudFormation modello AWS incluso in questa soluzione per distribuire automaticamente le risorse dell'infrastruttura (lo «stack») per questa soluzione.	CloudFormation Modello AWS
Accedi al codice sorgente e, facoltativamente, utilizza AWS Cloud Development Kit (AWS CDK) per distribuire la soluzione.	GitHub repository

Funzionalità

La soluzione offre le seguenti funzionalità:

Out-of-the-Box Test prestazionali configurabili

Include test prestazionali preconfigurati disponibili per l'uso immediato.

Test applicativi personalizzabili

Consente una personalizzazione flessibile e precisa dei test per identificare potenziali problemi.

Adatta i test a requisiti e scenari specifici utilizzando JMeter script.

Simula un carico utente elevato

In grado di simulare decine di migliaia di utenti connessi per sottoporre a stress test la tua applicazione.

Generazione continua di transazioni

Genera continuamente record transazionali per valutare le prestazioni in condizioni di carico costante.

monitoraggio in tempo reale

Fornisce il monitoraggio in tempo reale dell'avanzamento e dei risultati dei test. Pianifica l'avvio automatico dei test in date specifiche o a intervalli ricorrenti.

Simulazione di richieste regionali

Simula le richieste degli utenti da qualsiasi regione per valutare le prestazioni globali.

Flessibilità degli endpoint

Testa qualsiasi endpoint nelle regioni AWS, negli ambienti locali o in altri provider di servizi cloud.

Risultati dettagliati dei test

Visualizza i risultati completi dei test, tra cui il tempo di risposta medio, il numero di utenti simultanei, le richieste riuscite e le richieste non riuscite.

Console Web intuitiva

Offre una console easy-to-use web per la gestione e il monitoraggio dei test.

Supporta più protocolli

Compatibile con vari protocolli come HTTP WebSocket, HTTPS, JDBC, JMS, FTP e gRPC.

Integrazione con AWS Service Catalog AppRegistry e Application Manager, una funzionalità di AWS Systems Manager

Questa soluzione include una AppRegistry risorsa [Service Catalog](#) per registrare il CloudFormation modello della soluzione e le relative risorse sottostanti come applicazione sia in Service Catalog AppRegistry che in [Application Manager](#). Con questa integrazione, gestisci centralmente le risorse della soluzione e abilita le azioni di ricerca, reportistica e gestione delle applicazioni.

Vantaggi

La soluzione offre i seguenti vantaggi:

Supporta test completi delle prestazioni

Facilita i test di carico, stress e resistenza per una valutazione approfondita dell'applicazione.

Individuazione precoce dei problemi di prestazioni

Identifica i problemi di prestazioni e i punti deboli prima del rilascio in produzione.

Simulazione di utilizzo nel mondo reale

Rispecchia accuratamente i modelli di utilizzo del mondo reale per evidenziare i punti deboli e le aree di ottimizzazione.

Performance Insights dettagliati

Fornisce informazioni dettagliate sulle prestazioni e sulla resilienza del software in caso di carico significativo.

Valutazione automatizzata delle prestazioni

Consente valutazioni regolari delle prestazioni senza intervento manuale.

Test efficienti in termini di costi

Offre un pay-as-you-go modello che elimina la necessità di un'infrastruttura di test dedicata e di canoni di abbonamento.

Casi d'uso

Simula il carico di produzione

Testa le applicazioni web e mobili in condizioni simili a quelle di produzione prima di lanciare una nuova versione.

Convalida le prestazioni delle applicazioni

Assicurati che la tua applicazione sia in grado di gestire il traffico utente previsto senza degradazioni. Verifica i limiti delle applicazioni utilizzando risorse predefinite e valuta la scalabilità dell'infrastruttura.

Gestisci i carichi di picco

Verifica che la tua infrastruttura sia in grado di gestire i picchi di carico o i picchi di traffico imprevisti, garantendo la stabilità in caso di forte domanda.

Ottimizza le prestazioni

Comprendi il profilo prestazionale della tua applicazione e identifica i punti deboli come l'esecuzione inefficiente del codice, le query sul database e la latenza di rete.

Avvio rapido del test

Inizia rapidamente i test con i test out-of-the-box delle prestazioni.

Test personalizzabili

Personalizza i test in base a scenari e requisiti specifici, adattando il numero di utenti e attività simultanee avviate.

Test programmati

Pianifica i test per i test di regressione e il monitoraggio continuo delle prestazioni, garantendo prestazioni costanti delle applicazioni.

Valutazione delle prestazioni geografiche

Valuta le prestazioni delle applicazioni in diverse aree geografiche per garantire l'efficienza globale.

Integrazione della pipeline CI/CD

Integra i test delle prestazioni nella tua pipeline CI/CD per test automatizzati e senza interruzioni durante i cicli di sviluppo.

Concetti e definizioni

Questa sezione descrive i concetti chiave e definisce la terminologia specifica di questa soluzione:

scenario

Definizione del test che include nome, descrizione, numero di attività, concorrenza, regione AWS, ramp up, hold for, tipo di test, data di pianificazione e configurazioni di ricorrenza.

conteggio delle attività

Numero di container che verranno lanciati nel cluster Fargate per eseguire lo scenario di test. Non verranno create attività aggiuntive una volta raggiunto il limite dell'account per le risorse Fargate. Tuttavia, le attività già in esecuzione continueranno.

concurrency

Il numero di utenti virtuali simultanei generati per attività. Il limite consigliato in base alle impostazioni predefinite è di 200 utenti virtuali. La concorrenza è limitata dalla CPU e dalla memoria.

aumentare

Il tempo necessario per raggiungere la concorrenza prefissata.

tieni premuto per

È ora di mantenere la concorrenza tra gli obiettivi.

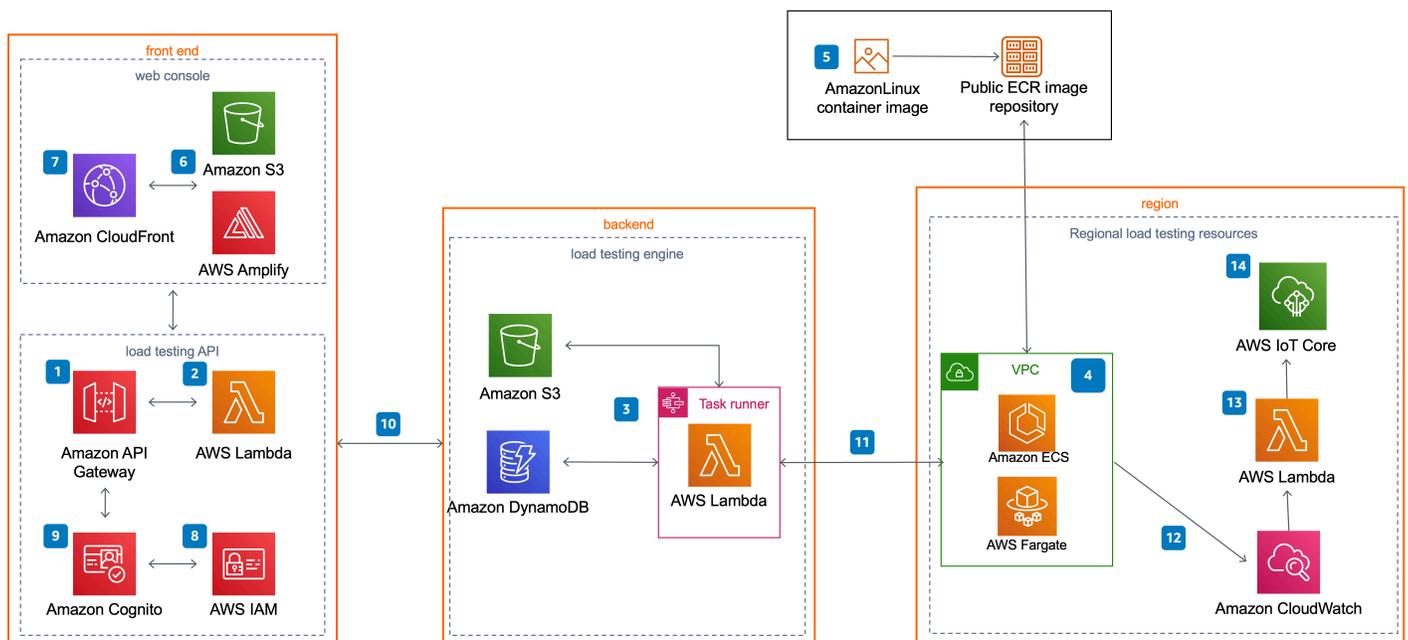
Per un riferimento generale ai termini di AWS, consulta il [Glossario AWS](#).

Panoramica dell'architettura

Diagramma architetturale

La distribuzione di questa soluzione con i parametri predefiniti distribuisce i seguenti componenti nel tuo account AWS.

Test di carico distribuito sull'architettura AWS su AWS



Note

Le CloudFormation risorse AWS vengono create a partire da costrutti di AWS Cloud Development Kit (AWS CDK).

Il flusso di processo di alto livello per i componenti della soluzione distribuiti con il CloudFormation modello AWS è il seguente:

1. [Un'API distribuita per testare il carico, che sfrutta Amazon API Gateway per richiamare i microservizi della soluzione \(funzioni AWS Lambda\).](#)
2. I microservizi forniscono la logica aziendale per gestire i dati di test ed eseguire i test.

3. Questi microservizi interagiscono con [Amazon Simple Storage Service](#) (Amazon S3), [Amazon DynamoDB](#) e AWS [Step Functions](#) per fornire storage per i dettagli e i risultati dello scenario di test ed eseguire scenari di test.
4. [Viene implementata una topologia di rete Amazon Virtual Private Cloud \(Amazon VPC\) contenente i contenitori Amazon Elastic Container Service \(Amazon ECS\) della soluzione in esecuzione su AWS Fargate.](#)
5. I contenitori includono l'immagine del [contenitore conforme a Open Container Initiative AmazonLinux](#)(OCI) (con il framework di test di carico blazemeter installato), che viene utilizzata per generare il carico per testare le prestazioni dell'applicazione. Taurus/Blazemeter è un framework di automazione dei test open source. L'immagine del contenitore è ospitata da AWS in un repository pubblico [Amazon Elastic Container Registry](#) (Amazon ECR). [Per ulteriori informazioni sull'archivio di immagini ECR, consulta la sezione Personalizzazione dell'immagine del contenitore.](#)
6. Una console Web basata su [AWS Amplify](#) viene distribuita in un bucket Amazon S3 configurato per l'hosting web statico.
7. [Amazon CloudFront](#) fornisce un accesso pubblico e sicuro ai contenuti del bucket del sito Web della soluzione.
8. Durante la configurazione iniziale, questa soluzione crea anche un ruolo di amministratore della soluzione predefinito (ruolo IAM) e invia un invito di accesso a un indirizzo e-mail utente specificato dal cliente.
9. Un pool di utenti di [Amazon Cognito](#) gestisce l'accesso degli utenti alla console e all'API del tester di carico distribuito.
10. Dopo aver distribuito questa soluzione, puoi utilizzare la console web per creare uno scenario di test che definisce una serie di attività.
11. I microservizi utilizzano questo scenario di test per eseguire attività di Amazon ECS su AWS Fargate nelle regioni specificate.
12. [Oltre a memorizzare i risultati in Amazon S3 e DynamoDB, una volta completato il test, l'output viene registrato in Amazon CloudWatch](#)
13. Se selezioni l'opzione live data, la soluzione invia CloudWatch i log di Amazon per le attività di AWS Fargate a una funzione Lambda durante il test, per ogni regione in cui è stato eseguito il test.
14. La funzione Lambda pubblica quindi i dati nell'argomento corrispondente in [AWS IoT Core](#) nella regione in cui è stato distribuito lo stack principale. La console Web sottoscrive l'argomento e puoi visualizzare i dati durante l'esecuzione del test nella console Web.

Considerazioni sulla progettazione di AWS Well-Architected

Questa soluzione utilizza le best practice di [AWS Well-Architected Framework](#), che aiuta i clienti a progettare e gestire carichi di lavoro affidabili, sicuri, efficienti ed economici nel cloud.

Questa sezione descrive in che modo i principi di progettazione e le migliori pratiche di Well-Architected Framework favoriscono questa soluzione.

Eccellenza operativa

Questa sezione descrive come abbiamo progettato questa soluzione utilizzando i principi e le migliori pratiche del pilastro dell'eccellenza [operativa](#).

- Risorse definite come infrastruttura come utilizzo di codice. CloudFormation
- La soluzione invia i parametri CloudWatch ad Amazon in varie fasi per garantire l'osservabilità nell'infrastruttura: funzioni Lambda, attività Amazon ECS, bucket Amazon S3 e il resto dei componenti della soluzione.

Sicurezza

[Questa sezione descrive come abbiamo progettato questa soluzione utilizzando i principi e le migliori pratiche del pilastro della sicurezza.](#)

- Amazon Cognito autentica e autorizza gli utenti delle app per l'interfaccia utente Web.
- Tutte le comunicazioni interservizi utilizzano ruoli [AWS Identity and Access Management](#) (IAM) applicabili.
- Tutti i ruoli utilizzati dalla soluzione seguono l'accesso con privilegi minimi. Contengono solo le autorizzazioni minime necessarie per eseguire il trasferimento.
- Tutti gli archivi di dati, inclusi i bucket S3, crittografano i dati inattivi.
- Un pool di utenti di Amazon Cognito gestisce l'accesso degli utenti alla console e agli endpoint API Gateway del tester di carico distribuito.
- La registrazione, il tracciamento e il controllo delle versioni sono attivati laddove applicabile.
- L'accesso alla rete è privato per impostazione predefinita e gli endpoint [Amazon Virtual Private Cloud](#) (Amazon VPC) vengono attivati laddove disponibili.

Affidabilità

[Questa sezione descrive come abbiamo progettato questa soluzione utilizzando i principi e le migliori pratiche del pilastro dell'affidabilità.](#)

- La soluzione utilizza AWS Serverless Services laddove possibile (esempi Lambda, API Gateway, Amazon S3, AWS Step Functions, Amazon DynamoDB e AWS Fargate) per garantire alta disponibilità e ripristino in caso di guasto del servizio.
- Tutte le elaborazioni di calcolo utilizzano funzioni Lambda o Amazon ECS su AWS Fargate.
- I dati vengono archiviati in DynamoDB e Amazon S3, quindi persistono in più zone di disponibilità per impostazione predefinita.

Efficienza delle prestazioni

[Questa sezione descrive come abbiamo progettato questa soluzione utilizzando i principi e le migliori pratiche del pilastro prestazione-efficienza.](#)

- La soluzione utilizza un'architettura serverless con la possibilità di scalare orizzontalmente in base alle esigenze.
- La soluzione può essere lanciata in qualsiasi regione che supporti i servizi AWS in questa soluzione, ad esempio: AWS Lambda, Amazon API Gateway, AWS S3, AWS Step Functions, Amazon DynamoDB, Amazon ECS, AWS Fargate e Amazon Cognito.
- La soluzione utilizza dappertutto servizi gestiti per ridurre l'onere operativo di approvvigionamento e gestione delle risorse.
- La soluzione viene testata e distribuita automaticamente ogni giorno per garantire la coerenza man mano che i servizi AWS cambiano. Inoltre, è stata esaminata da architetti di soluzioni ed esperti in materia per individuare le aree da sperimentare e migliorare.

Ottimizzazione dei costi

Questa sezione descrive come abbiamo progettato questa soluzione utilizzando i principi e le migliori pratiche del [pilastro dell'ottimizzazione dei costi](#).

- La soluzione utilizza un'architettura serverless, pertanto ai clienti viene addebitato solo ciò che utilizzano.

- Amazon DynamoDB ridimensiona la capacità su richiesta, quindi paghi solo per la capacità utilizzata.
- AWS ECS su AWS Fargate ti consente di pagare solo per le risorse di elaborazione che utilizzi, senza spese iniziali.

Sostenibilità

[Questa sezione descrive come abbiamo progettato questa soluzione utilizzando i principi e le migliori pratiche del pilastro della sostenibilità.](#)

- La soluzione utilizza servizi serverless gestiti per ridurre al minimo l'impatto ambientale dei servizi di backend rispetto ai servizi locali operativi in modo continuo.
- I servizi serverless consentono la scalabilità verso l'alto o verso il basso in base alle esigenze.

Dettagli dell'architettura

Questa sezione descrive i componenti e i [servizi AWS che compongono questa soluzione](#) e i dettagli dell'architettura su come questi componenti interagiscono.

La soluzione Distributed Load Testing on AWS è composta da due componenti di alto livello: un [front-end](#) e un [backend](#).

Front-end

Il front-end è costituito da un'API di test del carico e da una console web che utilizzi per interagire con il backend della soluzione.

API per il test di carico

Distributed Load Testing su AWS configura Amazon API Gateway per ospitare l' RESTful API della soluzione. Gli utenti possono interagire con i dati di test in modo sicuro tramite la console Web e RESTful l'API incluse. L'API funge da «porta d'ingresso» per l'accesso ai dati di test archiviati in Amazon DynamoDB. Puoi anche utilizzare il APIs per accedere a qualsiasi funzionalità estesa incorporata nella soluzione.

Questa soluzione sfrutta le funzionalità di autenticazione degli utenti dei pool di utenti di Amazon Cognito. Dopo aver autenticato correttamente un utente, Amazon Cognito emette un token web JSON che viene utilizzato per consentire alla console di inviare richieste alla soluzione (endpoint Amazon APIs API Gateway). Le richieste HTTPS vengono inviate dalla console a APIs con l'intestazione di autorizzazione che include il token.

In base alla richiesta, API Gateway richiama la funzione AWS Lambda appropriata per eseguire le attività necessarie sui dati archiviati nelle tabelle DynamoDB, archiviare scenari di test come oggetti JSON in Amazon S3, recuperare immagini dei parametri Amazon e inviare scenari di test alla macchina a stati AWS CloudWatch Step Functions.

[Per ulteriori informazioni sull'API della soluzione, consulta la sezione Distributed load testing API di questa guida.](#)

Console Web

Questa soluzione include una console Web che è possibile utilizzare per configurare ed eseguire test, monitorare i test in esecuzione e visualizzare i risultati dettagliati dei test. La console è

un'applicazione ReactJS ospitata in Amazon S3 e accessibile tramite Amazon CloudFront. L'applicazione sfrutta AWS Amplify per l'integrazione con Amazon Cognito per autenticare gli utenti. La console Web contiene anche un'opzione per visualizzare i dati in tempo reale per un test in esecuzione, in cui sottoscrive l'argomento corrispondente in AWS IoT Core.

La console web è progettata per dimostrare come interagire con questa soluzione di test di carico. In un ambiente di produzione, consigliamo di personalizzare la console Web per soddisfare esigenze specifiche o di creare una console personalizzata.

L'URL della console web è il nome del dominio di CloudFront distribuzione che può essere trovato negli CloudFormation output come Console. Dopo aver avviato il CloudFormation modello, riceverai anche un'e-mail contenente l'URL della console Web e la password monouso per accedervi.

Backend

Il backend è costituito da una pipeline di immagini del contenitore e da un motore di test di carico utilizzato per generare il carico per i test. Interagisci con il backend tramite il front-end. Inoltre, le attività di Amazon ECS su AWS Fargate avviate per ogni test sono contrassegnate con un identificatore di test (ID) univoco. Questi tag ID di test possono essere utilizzati per aiutarti a monitorare i costi di questa soluzione. Per ulteriori informazioni, consulta i [tag di allocazione dei costi definiti dall'utente](#) nella AWS Billing and Cost Management User Guide.

Pipeline di immagini del contenitore

Questa soluzione sfrutta un'immagine del contenitore creata [AmazonLinux](#) come immagine di base con il framework di test di carico blazemeter installato. Questa immagine è ospitata in un repository pubblico di Amazon Elastic Container Registry (Amazon ECR). L'immagine viene utilizzata per eseguire attività nel cluster Amazon ECS on AWS Fargate.

Per ulteriori informazioni, consulta la sezione sulla [personalizzazione dell'immagine del contenitore](#) di questa guida.

Infrastruttura di test

Oltre al modello principale, la soluzione crea un modello secondario per avviare le risorse necessarie per eseguire i test in più regioni. Il modello è archiviato in Amazon S3 e nella console Web viene fornito un collegamento al modello. I modelli secondari creano un VPC, un cluster AWS Fargate e una funzione Lambda per l'elaborazione di dati in tempo reale.

Per ulteriori informazioni su come avviare una regione secondaria, consulta la sezione [Distribuzione multiregionale](#) di questa guida.

Motore di test di carico

La soluzione Distributed Load Testing utilizza Amazon Elastic Container Service (Amazon ECS) e AWS Fargate per simulare migliaia di utenti connessi, in più regioni, generando un numero selezionato di transazioni al secondo.

Puoi definire i parametri per le attività che verranno eseguite come parte del test utilizzando la console web inclusa. La soluzione utilizza questi parametri per generare uno scenario di test JSON e lo archivia in Amazon S3.

Una macchina a stati AWS Step Functions esegue e monitora le attività di Amazon ECS in un cluster AWS Fargate. La macchina a stati AWS Step Functions include una funzione AWS Lambda ecr-checker, una funzione AWS Lambda, una funzione AWS Lambda task-runner, una funzione task-status-checker AWS Lambda con cancellazione delle attività e una funzione AWS Lambda con analisi dei risultati. [Per ulteriori informazioni sul flusso di lavoro, consulta la sezione Test workflow di questa guida.](#) Per ulteriori informazioni sui risultati dei test, consulta la sezione [Risultati dei test](#) di questa guida. Per ulteriori informazioni sul flusso di lavoro per l'annullamento del test, consulta la sezione [Flusso di lavoro per l'annullamento del test](#) di questa guida.

Se si selezionano dati in tempo reale, la soluzione avvia una funzione real-time-data-publisher Lambda in ogni regione tramite CloudWatch i log che corrispondono alle attività Fargate in quella regione. La soluzione elabora e pubblica quindi i dati su un argomento in AWS IoT Core all'interno della regione in cui è stato lanciato lo stack principale. Per ulteriori informazioni, consulta la sezione [Dati in tempo reale](#) di questa guida.

Servizi AWS in questa soluzione

I seguenti servizi AWS sono inclusi in questa soluzione:

Servizio AWS	Descrizione
Gateway Amazon API	Core. Ospita gli endpoint dell'API REST nella soluzione.
AWS CloudFormation	Nucleo. Gestisce le implementazioni per l'infrastruttura della soluzione.
Amazon CloudFront	Core. Serve i contenuti Web ospitati in Amazon S3.

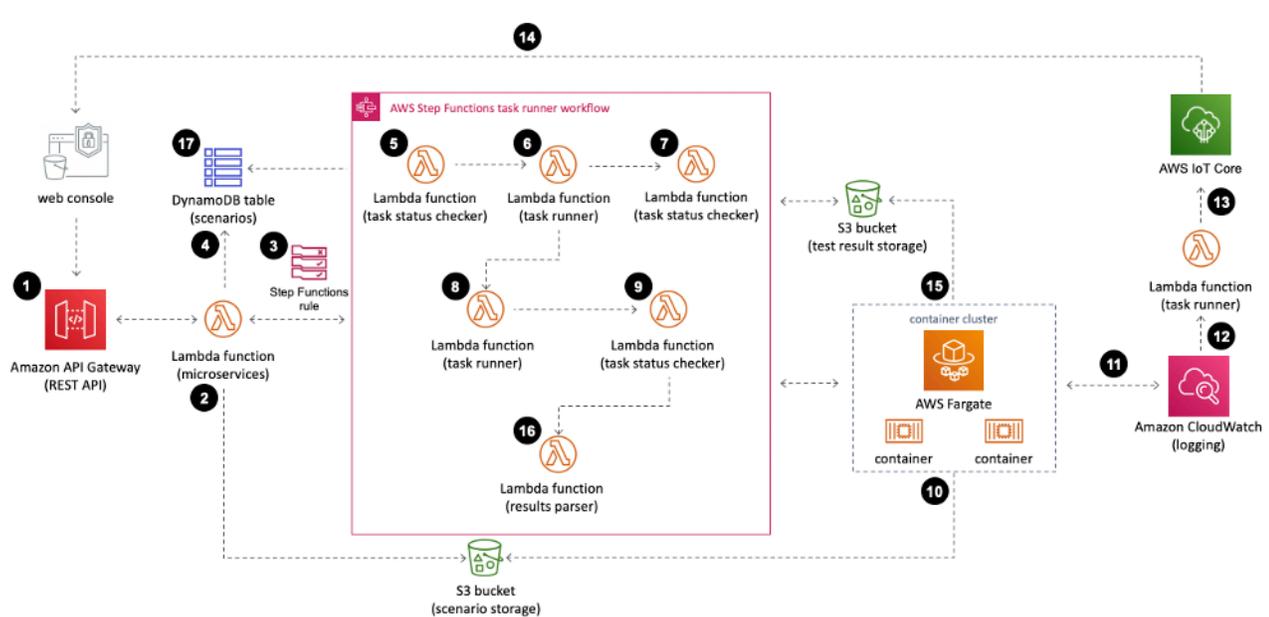
Servizio AWS	Descrizione
Amazon CloudWatch	Nucleo. Memorizza i log e le metriche della soluzione.
Amazon Cognito	Nucleo. Gestisce la gestione e l'autenticazione degli utenti per l'API.
Amazon DynamoDB	Nucleo. Archivia le informazioni sulla distribuzione e i dettagli e i risultati dello scenario di test.
Amazon Elastic Container Service	Nucleo. Distribuisce e gestisce attività Amazon ECS indipendenti su contenitori AWS Fargate.
AWS Fargate	Nucleo. Contenitori Amazon ECS della soluzione Hosts
AWS Identity and Access Management	Nucleo. Gestisce la gestione dei ruoli e delle autorizzazioni degli utenti.
AWS Lambda	Nucleo. Fornisce la logica per APIs l'implementazione, l'analisi dei risultati dei test e l'avvio delle attività dei lavoratori/dirigenti.
AWS Step Functions	Nucleo. Orchestra il provisioning dei contenitori Amazon ECS sulle attività di AWS Fargate nelle regioni specificate
AWS Amplify	Supporto. Fornisce una console Web basata su AWS Amplify .
CloudWatch Events Amazon	Supporto. Pianifica l'inizio automatico dei test in una data specificata o in date ricorrenti.
Amazon Elastic Container Registry	Supporto. Ospita l'immagine del contenitore in un repository ECR pubblico.
AWS IoT Core	Supporto. Consente la visualizzazione di dati in tempo reale per un test in esecuzione sottoscrivendo l'argomento corrispondente in AWS IoT Core.
AWS Systems Manager	Supporto. Fornisce il monitoraggio delle risorse a livello di applicazione e la visualizzazione delle operazioni relative alle risorse e dei dati sui costi.
Amazon S3	Supporto. Ospita contenuti web statici, log, metriche e dati di test.

Servizio AWS	Descrizione
Amazon Virtual Private Cloud	Supporto. Contiene i contenitori Amazon ECS della soluzione in esecuzione su AWS Fargate.

Come funziona il test di carico distribuito su AWS

La seguente suddivisione dettagliata mostra i passaggi necessari per l'esecuzione di uno scenario di test.

Workflow di test



1. Si utilizza la console Web per inviare uno scenario di test che include i dettagli di configurazione all'API della soluzione.
2. La configurazione dello scenario di test viene caricata su Amazon Simple Storage Service (Amazon S3) come file JSON (`s3://<bucket-name>/test-scenarios/<$TEST_ID>/<$TEST_ID>.json`).
3. Una macchina a stati AWS Step Functions viene eseguita utilizzando l'ID di test, il conteggio delle attività, il tipo di test e il tipo di file come input della macchina a stati AWS Step Functions. Se il test è pianificato, creerà innanzitutto una regola CloudWatch Events, che attiva AWS Step Functions alla data specificata. Per maggiori dettagli sul flusso di lavoro di pianificazione, consulta la sezione Flusso di [lavoro di pianificazione dei test](#) di questa guida.

4. I dettagli di configurazione sono archiviati nella tabella degli scenari Amazon DynamoDB.
5. Nel flusso di lavoro del task runner di AWS Step Functions, la funzione task-status-checker AWS Lambda verifica se le attività di Amazon Elastic Container Service (Amazon ECS) sono già in esecuzione per lo stesso ID di test. Se vengono rilevate attività con lo stesso ID di test in esecuzione, viene generato un errore. Se non ci sono attività Amazon ECS in esecuzione nel cluster AWS Fargate, la funzione restituisce l'ID di test, il conteggio delle attività e il tipo di test.
6. La funzione task-runner AWS Lambda ottiene i dettagli delle attività dal passaggio precedente ed esegue le attività dei worker di Amazon ECS nel cluster AWS Fargate. L'API Amazon ECS utilizza l' RunTask azione per eseguire le attività dei lavoratori. Queste attività lavorative vengono avviate e quindi attendono un messaggio di avvio dall'attività principale per iniziare il test. L' RunTask azione è limitata a 10 attività per definizione. Se il numero di attività è superiore a 10, la definizione dell'attività verrà eseguita più volte fino all'avvio di tutte le attività dei lavoratori. La funzione genera anche un prefisso per distinguere il test corrente nella funzione di analisi dei risultati di AWS Lambda.
7. La funzione task-status-checker AWS Lambda verifica se tutte le attività di lavoro di Amazon ECS vengono eseguite con lo stesso ID di test. Se le attività sono ancora in fase di provisioning, attende un minuto e verifica nuovamente. Una volta eseguite tutte le attività di Amazon ECS, restituisce l'ID di test, il conteggio delle attività, il tipo di test, tutte le attività IDs e il prefisso e li passa alla funzione task-runner.
8. La funzione task-runner AWS Lambda viene eseguita nuovamente, questa volta lanciando una singola attività Amazon ECS che funge da nodo leader. Questa attività ECS invia un messaggio di avvio del test a ciascuna delle attività del lavoratore per avviare i test contemporaneamente.
9. La funzione task-status-checker AWS Lambda verifica nuovamente se le attività di Amazon ECS sono in esecuzione con lo stesso ID di test. Se le attività sono ancora in esecuzione, attende un minuto e verifica nuovamente. Quando non ci sono attività Amazon ECS in esecuzione, restituisce l'ID di test, il conteggio delle attività, il tipo di test e il prefisso.
10. Quando la funzione task-runner AWS Lambda esegue le attività di Amazon ECS nel cluster AWS Fargate, ogni attività scarica la configurazione di test da Amazon S3 e avvia il test.
11. Una volta eseguiti i test, il tempo di risposta medio, il numero di utenti simultanei, il numero di richieste riuscite e il numero di richieste non riuscite per ogni attività vengono registrati in Amazon CloudWatch e possono essere visualizzati in una CloudWatch dashboard.
12. Se hai incluso dati in tempo reale nel test, la soluzione filtra i risultati dei test in tempo reale CloudWatch utilizzando un filtro di abbonamento. Quindi la soluzione passa i dati a una funzione Lambda.
13. La funzione Lambda struttura quindi i dati ricevuti e li pubblica su un argomento di AWS IoT Core.

- 14 La console Web sottoscrive l'argomento AWS IoT Core per il test e riceve i dati pubblicati sull'argomento per rappresentare graficamente i dati in tempo reale durante l'esecuzione del test.
- 15 Al termine del test, le immagini del contenitore esportano un report dettagliato come file XML in Amazon S3. A ogni file viene assegnato un UUID per il nome del file. Ad esempio, `s3://dlte-bucket/test-scenarios/ <$TEST_ID> /results/ <$UUID> .json`.
- 16 Quando i file XML vengono caricati su Amazon S3, la funzione `results-parser` AWS Lambda legge i risultati nei file XML a partire dal prefisso e analizza e aggrega tutti i risultati in un unico risultato riepilogativo.
- 17 La funzione `results-parser` AWS Lambda scrive il risultato aggregato in una tabella Amazon DynamoDB.

Considerazioni di natura progettuale

Applicazioni supportate

Questa soluzione supporta applicazioni basate su cloud e applicazioni locali purché sia disponibile una connessione di rete dal tuo account AWS all'applicazione. La soluzione supporta l'utilizzo APIs di HTTP o HTTPS. Hai anche il controllo sulle intestazioni delle richieste HTTP, quindi puoi aggiungere intestazioni di autorizzazione o personalizzate per passare token o chiavi API.

JMeter supporto per script

Quando si crea uno scenario di test utilizzando l'interfaccia utente (UI) di questa soluzione, è possibile utilizzare uno script di JMeter test. Dopo aver selezionato il file di JMeter script, questo viene caricato nel bucket `<stack-name>-scenariosbucket` Amazon Simple Storage Service (Amazon S3). Quando le attività di Amazon Elastic Container Service (Amazon ECS) sono in esecuzione, lo script viene scaricato dal bucket Amazon S3 `<stack-name>-scenariosbucket` e JMeter il test viene eseguito.

Se disponi di file JMeter di input, puoi comprimerli insieme allo script. JMeter È possibile scegliere il file zip quando si crea uno scenario di test.

Se desideri includere dei plugin, tutti i file.jar inclusi nella sottodirectory `a/plugining` del file zip fornito in bundle verranno copiati nella directory delle JMeter estensioni e saranno disponibili per il test di carico.

Note

Se JMeter includi file di input nel file di JMeter script, devi includere il percorso relativo dei file di input nel file di script. JMeter Inoltre, i file di input devono trovarsi nel percorso relativo. Ad esempio, se i file JMeter di input e il file di script si trovano invece in the /home/user directory and you refer to the input files in the JMeter script file, the path of input files must be `./INPUT_FILES`. If you use `/home/user/INPUT_FILES`, il test avrà esito negativo perché non sarà in grado di trovare i file di input.

Se includete JMeter dei plugin, i file.jar devono essere raggruppati in una sottodirectory `_PLUGIN.jar` named `/plugins` within the root of the zip file. Relative to the root of the zip file, the path to the jar files must be `./plugins/BUNDLED`

[Per ulteriori informazioni su come utilizzare JMeter gli script, consultate il Manuale dell'utente. JMeter](#)

Pianificazione dei test

Puoi pianificare i test da eseguire in date future o utilizzare l'opzione Esegui ora. È possibile pianificare un test come esecuzione unica in futuro o impostare un test ricorrente in cui specificare una data di prima esecuzione e una ricorrenza pianificata. Le opzioni per la ricorrenza includono: giornaliera, settimanale, bisettimanale e mensile. Per ulteriori informazioni su come funziona la pianificazione, consulta la sezione Flusso di lavoro di [pianificazione dei test di questa guida](#).

A partire dalla versione 3.3.0, Distributed Load Testing on AWS consente agli utenti di pianificare test di carico utilizzando espressioni cron. Seleziona Run of Schedule e poi la scheda CRON per inserire manualmente un valore cron o utilizzare i campi a discesa. `cronExpiryDate` Deve corrispondere alla data di esecuzione del test pianificata. Controlla le date della prossima esecuzione (UTC) per confermare la tua pianificazione.

Note

- **Durata del test:** durante la pianificazione, considera la durata totale dei test. Ad esempio, il completamento di un test con un tempo di accelerazione di 10 minuti e un tempo di attesa di 40 minuti richiederà circa 80 minuti.
- **Intervallo minimo:** assicurati che l'intervallo tra i test programmati sia più lungo della durata stimata del test. Ad esempio, se il test dura circa 80 minuti, programmallo in modo che venga eseguito non più di ogni 3 ore.

- Limitazione oraria: il sistema non consente di programmare i test con una differenza di solo un'ora, anche se la durata stimata del test è inferiore a un'ora.

Test simultanei

Questa soluzione include una CloudWatch dashboard Amazon per ogni test e visualizza l'output combinato di tutte le attività in esecuzione per quel test nel cluster Amazon ECS in tempo reale. La CloudWatch dashboard mostra il tempo di risposta medio, il numero di utenti simultanei, il numero di richieste riuscite e il numero di richieste non riuscite. Ogni metrica viene aggregata al secondo e la dashboard viene aggiornata ogni minuto.

Gestione degli utenti

Durante la configurazione iniziale, fornisci un nome utente e un indirizzo e-mail che Amazon Cognito utilizza per concederti l'accesso alla console web della soluzione. La console non fornisce l'amministrazione degli utenti. Per aggiungere altri utenti, devi utilizzare la console Amazon Cognito. Per ulteriori informazioni, consulta la sezione [Gestione degli utenti nei pool di utenti](#) nella Amazon Cognito Developer Guide.

Implementazione regionale

Questa soluzione utilizza Amazon Cognito, disponibile solo in regioni AWS specifiche. Pertanto, è necessario distribuire questa soluzione in una regione in cui è disponibile Amazon Cognito. Per la disponibilità dei servizi più aggiornata per regione, consulta l'[AWS Regional Services List](#).

Pianifica la tua implementazione

Questa sezione descrive i [costi](#), la [sicurezza](#), [le regioni](#) e altre considerazioni prima di implementare la soluzione.

Costo

Sei responsabile del costo dei servizi AWS utilizzati durante l'esecuzione di questa soluzione. Il costo totale per l'esecuzione di questa soluzione dipende dal numero di test di carico eseguiti, dalla durata di tali test e dalla quantità di dati utilizzati come parte dei test. A partire da questa revisione, il costo per l'esecuzione di questa soluzione con le impostazioni predefinite nella regione Stati Uniti orientali (Virginia settentrionale) è di circa 30,90 USD al mese.

La tabella seguente fornisce un esempio di ripartizione dei costi per l'implementazione di questa soluzione con i parametri predefiniti nella regione Stati Uniti orientali (Virginia settentrionale) per un mese.

Servizio AWS	Dimensioni	Costo [USD]
AWS Fargate	10 attività su richiesta (utilizzando due v CPUs e 4 GB di memoria) in esecuzione per 30 ore	29,62 USD
Amazon DynamoDB	1.000 unità con capacità di scrittura su richiesta 1.000 unità con capacità di lettura su richiesta	0,0015 USD
AWS Lambda	1.000 richieste Durata totale di 10 minuti	\$1,25
AWS Step Functions	1.000 transizioni di stato	0,025 USD
Totale:		\$30,90 al mese

Ti consigliamo di creare un [budget](#) tramite [AWS Cost Explorer](#) per gestire i costi. I prezzi sono soggetti a modifiche. Per tutti i dettagli, consulta la pagina web dei prezzi per ogni [servizio AWS utilizzato in questa soluzione](#).

Important

A partire dalla versione 1.3.0, la CPU viene aumentata a 2 vCPU e la memoria viene aumentata a 4 GB. Queste modifiche aumentano il costo stimato rispetto alle versioni precedenti di questa soluzione. Se i test di carico non richiedono questi aumenti delle risorse AWS, puoi ridurli. Per ulteriori informazioni, consulta la sezione [Aumentare le risorse dei container](#) di questa guida.

Note

Questa soluzione offre la possibilità di includere dati in tempo reale durante l'esecuzione di un test. Questa funzionalità richiede una funzione AWS Lambda aggiuntiva e un argomento AWS IoT Core che comporta costi aggiuntivi.

I prezzi sono soggetti a modifiche. Per tutti i dettagli, consulta la pagina web dei prezzi per ogni servizio AWS che utilizzerai in questa soluzione.

Sicurezza

Quando crei sistemi sull'infrastruttura AWS, le responsabilità di sicurezza vengono condivise tra te e AWS. Questo [modello di responsabilità condivisa](#) riduce il carico operativo perché AWS gestisce, gestisce e controlla i componenti, tra cui il sistema operativo host, il livello di virtualizzazione e la sicurezza fisica delle strutture in cui operano i servizi. Per ulteriori informazioni sulla sicurezza di AWS, visita [AWS Cloud Security](#).

Ruoli IAM

I ruoli di AWS Identity and Access Management (IAM) consentono ai clienti di assegnare policy e autorizzazioni di accesso granulari a servizi e utenti sul cloud AWS. Questa soluzione crea ruoli IAM che garantiscono l'accesso alle funzioni AWS Lambda della soluzione per creare risorse regionali.

Amazon CloudFront

Questa soluzione implementa una console Web [ospitata](#) in un bucket Amazon Simple Storage Service (Amazon S3). Per contribuire a ridurre la latenza e migliorare la sicurezza, questa soluzione include una CloudFront distribuzione Amazon con un'identità di accesso di origine, ovvero un CloudFront utente che fornisce l'accesso pubblico ai contenuti del bucket del sito Web della soluzione. Per ulteriori informazioni, consulta [Limitazione dell'accesso ai contenuti Amazon S3 utilizzando un'identità Origin Access](#) nella CloudFront Amazon Developer Guide.

Gruppo di sicurezza AWS Fargate

Per impostazione predefinita, questa soluzione apre al pubblico la regola in uscita del gruppo di sicurezza AWS Fargate. Se desideri impedire ad AWS Fargate di inviare traffico ovunque, modifica la regola in uscita con uno specifico Classless Inter-Domain Routing (CIDR).

Questo gruppo di sicurezza include anche una regola in entrata che consente il traffico locale sulla porta 50.000 verso qualsiasi fonte appartenente allo stesso gruppo di sicurezza. Viene utilizzato per consentire ai contenitori di comunicare tra loro.

Test di stress della rete

L'utente è responsabile dell'utilizzo di questa soluzione in base alla [politica di Network Stress Test](#). Questa policy copre situazioni come quelle in cui hai intenzione di eseguire test di rete ad alto volume direttamente dalle tue EC2 istanze Amazon su altre posizioni come altre EC2 istanze Amazon, proprietà/servizi AWS o endpoint esterni. Questi test sono talvolta chiamati stress test, test di carico o test del giorno di gioco. La maggior parte dei test effettuati dai clienti non rientra in questa politica, tuttavia, fai riferimento a questa politica se ritieni che genererai traffico che dura, in aggregato, per più di 1 minuto, oltre 1 Gbps (1 miliardo di bit al secondo) o oltre 1 Gpps (1 miliardo di pacchetti al secondo).

Limitazione dell'accesso all'interfaccia utente pubblica

Per limitare l'accesso all'interfaccia utente pubblica oltre ai meccanismi di autenticazione e autorizzazione forniti da IAM e Amazon Cognito, utilizza la soluzione AWS [WAF \(web application firewall\)](#) Security Automations.

Questa soluzione implementa automaticamente una serie di regole AWS WAF che filtrano i comuni attacchi basati sul Web. Gli utenti possono scegliere tra funzionalità di protezione preconfigurate che definiscono le regole incluse in una lista di controllo degli accessi Web AWS WAF (Web ACL).

Regioni AWS supportate

Questa soluzione utilizza il servizio Amazon Cognito, che attualmente non è disponibile in tutte le regioni AWS. Per la disponibilità più aggiornata dei servizi AWS per regione, consulta l'[AWS Regional Services List](#).

Il test di carico distribuito su AWS è disponibile nelle seguenti regioni AWS:

Nome della Regione	
Stati Uniti orientali (Ohio)	Asia Pacifico (Tokyo)
Stati Uniti orientali (Virginia settentrionale)	Canada (Centrale)
Stati Uniti occidentali (California settentrionale)	Europa (Francoforte)
US West (Oregon)	Europa (Irlanda)
Asia Pacifico (Mumbai)	Europa (Londra)
Asia Pacifico (Osaka-Locale)	Europa (Parigi)
Asia Pacifico (Seoul)	Europa (Stoccolma)
Asia Pacifico (Singapore)	Sud America (San Paolo)
Asia Pacifico (Sydney)	

Quote

Le quote di servizio, anche denominate limiti, rappresentano il numero massimo di risorse di servizio o operazioni per l'account AWS.

Quote per i servizi AWS in questa soluzione

Assicurati di disporre di una quota sufficiente per ciascuno dei [servizi implementati in questa soluzione](#). Per ulteriori informazioni, consulta [Quote di servizio AWS](#).

Utilizza i seguenti collegamenti per accedere alla pagina relativa al servizio. Per visualizzare le quote di servizio per tutti i servizi AWS nella documentazione senza cambiare pagina, visualizza invece le informazioni nella pagina [Endpoint e quote del servizio](#) nel PDF.

CloudFormation Quote AWS

Il tuo account AWS ha CloudFormation quote AWS di cui dovresti essere a conoscenza quando [avvii lo stack](#) di questa soluzione. Comprendendo queste quote, puoi evitare errori di limitazione che potrebbero impedirti di implementare questa soluzione con successo. Per ulteriori informazioni, consulta le [CloudFormation quote AWS](#) nella AWS CloudFormation User's Guide.

Quote di test di carico

Il numero massimo di attività che possono essere eseguite in Amazon ECS utilizzando il tipo di avvio AWS Fargate si basa sulla dimensione della vCPU delle attività. La dimensione predefinita dell'attività in Distributed Load Testing on AWS è di 2 vCPU. Per visualizzare le quote predefinite correnti, consulta le quote dei [servizi Amazon ECS](#). Le quote dei conti correnti possono differire dalle quote elencate. Per controllare le quote specifiche di un account, controlla la quota di servizio per il conteggio delle risorse vCPU on-demand di Fargate nella Console di gestione AWS. Per istruzioni su come richiedere un aumento, consulta le [quote dei servizi AWS nella AWS](#) General Reference Guide.

L' AmazonLinux immagine del contenitore (con Blazemeter installato) non limita le connessioni simultanee per attività, ma ciò non significa che possa supportare un numero illimitato di utenti. Per determinare il numero di utenti simultanei che i contenitori possono generare per un test, consulta la sezione [Determinare il numero di utenti di questa guida](#).

Note

Il limite consigliato per gli utenti simultanei in base alle impostazioni predefinite è di 200 utenti.

Test simultanei

Questa soluzione include una CloudWatch dashboard Amazon per ogni test e visualizza l'output combinato di tutte le attività in esecuzione per quel test nel cluster Amazon ECS in tempo reale. La CloudWatch dashboard mostra il tempo di risposta medio, il numero di utenti simultanei, il numero di richieste riuscite e il numero di richieste non riuscite. Ogni metrica viene aggregata al secondo e la dashboard viene aggiornata ogni minuto.

Politica di EC2 test di Amazon

Non è necessaria l'approvazione di AWS per eseguire test di carico utilizzando questa soluzione purché il traffico di rete rimanga inferiore a 1 Gbps. Se il test genererà più di 1 Gbps, contatta AWS. Per ulteriori informazioni, consulta la [Amazon EC2 Testing Policy](#).

Politica di test CloudFront di carico di Amazon

Se prevedi di testare il carico di un CloudFront endpoint, consulta le [linee guida per i test di carico](#) nell'Amazon CloudFront Developer Guide. Consigliamo inoltre di distribuire il traffico tra più attività e regioni. Fornisci almeno 30 minuti di accelerazione per il test di carico. Per i test di carico che inviano più di 500.000 richieste al secondo o richiedono dati superiori a 300 Gbps, consigliamo di ottenere prima un'approvazione preventiva per l'invio del traffico. CloudFront può limitare il traffico di test di carico non approvato che influisce sulla disponibilità del servizio. CloudFront

Implementa la soluzione

Questa soluzione utilizza [CloudFormation modelli e stack AWS](#) per automatizzarne l'implementazione. I CloudFormation modelli specificano le risorse AWS incluse in questa soluzione e le relative proprietà. Lo CloudFormation stack fornisce le risorse descritte nei modelli.

Panoramica del processo di distribuzione

Segui le step-by-step istruzioni in questa sezione per configurare e distribuire la soluzione nel tuo account.

Prima di avviare la soluzione, esamina i [costi](#), l'[architettura](#), la [sicurezza della rete](#) e altre considerazioni discusse in precedenza in questa guida.

Tempo di implementazione: circa 15 minuti

CloudFormation Modello AWS

Puoi scaricare il CloudFormation modello per questa soluzione prima di distribuirla. Questa soluzione utilizza AWS CloudFormation per automatizzare l'implementazione di Distributed Load Testing su AWS. Include il seguente CloudFormation modello AWS, che puoi scaricare prima della distribuzione:

[View template](#)

[load-testing-on-aws.template](#): utilizza questo modello per avviare la soluzione e tutti i componenti associati. La configurazione predefinita distribuisce i servizi di base e di supporto disponibili nei [servizi AWS in questa sezione della soluzione](#), ma puoi personalizzare il modello per soddisfare le tue esigenze specifiche.

Note

Le CloudFormation risorse AWS vengono create a partire da costrutti di AWS Cloud Development Kit (AWS CDK). Se hai già distribuito questa soluzione, consulta [Aggiornare la soluzione per le istruzioni di aggiornamento](#).

Avvio dello stack

Important

Se stai aggiornando lo stack da una versione precedente alla v3.2.6 alla versione più recente, leggi [questa sezione](#) prima di aggiornare lo stack.

Prima di avviare la distribuzione automatizzata, esamina l'architettura e le altre considerazioni discusse in questa guida. Segui le step-by-step istruzioni in questa sezione per configurare e distribuire Distributed Load Testing on AWS nel tuo account.

Tempo di implementazione: circa 15 minuti

Important

Questa soluzione include un'opzione per inviare metriche operative anonime ad AWS. Utilizziamo questi dati per comprendere meglio come i clienti utilizzano questa soluzione e i servizi e i prodotti correlati. AWS è proprietaria dei dati raccolti tramite questo sondaggio. La raccolta dei dati è soggetta all'[Informativa sulla privacy di AWS](#).

Per disattivare questa funzionalità, scarica il modello, modifica la sezione di CloudFormation mappatura AWS, quindi utilizza la CloudFormation console AWS per caricare il modello aggiornato e distribuire la soluzione. Per ulteriori informazioni, consulta la sezione [Raccolta di dati anonimi](#) di questa guida.

Questo CloudFormation modello AWS automatizzato implementa Distributed Load Testing su AWS.

Note

Sei responsabile del costo dei servizi AWS utilizzati durante l'esecuzione di questa soluzione. Per ulteriori dettagli, visita la sezione [Costo](#) di questa guida e consulta la pagina web dei prezzi per ogni servizio AWS utilizzato in questa soluzione.

1. Accedi alla Console di gestione AWS e seleziona il pulsante in basso per avviare il CloudFormation modello distributed-load-testing-on -aws AWS.

Launch solution

In alternativa, puoi anche [scaricare il modello](#) come punto di partenza per la tua implementazione.

- Questo modello viene avviato nella regione Stati Uniti orientali (Virginia settentrionale), per impostazione predefinita. Per avviare questa soluzione in un'altra regione AWS, utilizza il selettore di regione nella barra di navigazione della console.

Note

Questa soluzione utilizza Amazon Cognito, che al momento è disponibile solo in regioni AWS specifiche. Pertanto, è necessario avviare questa soluzione in una regione AWS in cui è disponibile Amazon Cognito. Per la disponibilità dei servizi più aggiornata per regione, consulta l'[AWS Regional Services List](#).

- Nella pagina Create stack, verifica che l'URL del modello corretto sia visualizzato nella casella di testo URL Amazon S3 e scegli Avanti.
- Nella pagina Specificare i dettagli dello stack, assegna un nome allo stack di soluzioni.
- In Parametri, esamina i parametri per il modello e modificali se necessario. Questa soluzione utilizza i seguenti valori predefiniti.

Parametro	Predefinito	Descrizione
Nome dell'amministratore	<Da configurare da parte dell'utente>	Nome utente per l'amministratore della soluzione iniziale.
Email dell'amministratore	<i><Requires input></i>	Indirizzo e-mail dell'utente amministratore. Dopo il lancio, verrà inviata un'e-mail a questo indirizzo con le istruzioni per l'accesso alla console.
ID VPC esistente	<Optional input>	Se hai un VPC che desideri utilizzare ed è già stato

Parametro	Predefinito	Descrizione
		creato, inserisci l'ID di un VPC esistente nella stessa regione in cui è stato distribuito lo stack. Ad esempio, vpc-1a2b3c4d5e6f.
Prima sottorete esistente	<Optional input>	L'ID della prima sottorete all'interno del VPC esistente . Questa sottorete necessita di un percorso verso Internet per recuperare l'immagine del contenitore per eseguire i test. Ad esempio, subnet-7h8i9j0k.
Seconda sottorete esistente	<Optional input>	L'ID della seconda sottorete all'interno del VPC esistente . Questa sottorete necessita di un percorso verso Internet per recuperare l'immagine del contenitore per eseguire i test. Ad esempio, subnet-1x2y3z.
Blocco CIDR VPC AWS Fargate	192.168.0.0/16	Se non fornisci valori per un VPC esistente, il blocco CIDR per Amazon VPC creato dalla soluzione contiene l'indirizzo IP per AWS Fargate.
Blocco CIDR di AWS Fargate Subnet A	192.168.0.0/20	Se non fornisci valori per un VPC esistente, il blocco CIDR contiene l'indirizzo IP per la sottorete Amazon VPC A.

Parametro	Predefinito	Descrizione
Blocco CIDR AWS Fargate Subnet B	192.168.16.0/20	Se non fornisci valori per un VPC esistente, il blocco CIDR contiene l'indirizzo IP per la sottorete Amazon VPC B.
Blocco CIDR del gruppo di sicurezza AWS Fargate	0.0.0.0/0	Blocco CIDR che limita l'accesso in uscita ai container Amazon ECS.

6. Scegli Next (Successivo).
7. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).
8. Nella pagina Rivedi, verifica e conferma le impostazioni. Seleziona la casella per confermare che il modello creerà risorse AWS Identity and Access Management (IAM).
9. Seleziona Create (Crea) per implementare lo stack.

Puoi visualizzare lo stato dello stack nella CloudFormation console AWS nella colonna Status. Dovresti ricevere lo status CREATE_COMPLETE in circa 15 minuti.

Note

Oltre alla funzione principale di AWS Lambda, questa soluzione include la funzione Lambda con risorse personalizzate, che viene eseguita solo durante la configurazione iniziale o quando le risorse vengono aggiornate o eliminate.

Quando si esegue questa soluzione, la funzione Lambda della risorsa personalizzata è inattiva. Tuttavia, non eliminate questa funzione poiché è necessaria per gestire le risorse associate.

Implementazione in più regioni

Durata dell'implementazione: circa cinque minuti

È possibile eseguire test in più regioni. Quando si distribuisce la soluzione Distributed Load Testing, vengono creati tre bucket Amazon S3. La soluzione crea uno stack regionale secondario e lo archivia nel bucket di scenari Amazon S3.

Note

La convenzione di denominazione dei bucket è `<stack-name> - dltestrunnerstoragedltsenariosbucket <_[0-9][0-9].-<[0-9][0-9].->` con la parola chiave `scenarios` nel nome del bucket, che puoi individuare accedendo alla console S3, quindi Buckets.

Per eseguire una distribuzione multiregionale, devi distribuire il CloudFormation modello regionale, archiviato nel bucket degli scenari di Amazon S3, nelle regioni in cui desideri eseguire il test. Puoi installare il modello regionale effettuando le seguenti operazioni:

1. Nella console web della soluzione, vai a Gestisci regioni nel menu in alto.
2. Usa l'icona degli appunti per copiare il link del CloudFormation modello in Amazon S3.
3. Accedi alla [CloudFormation console AWS](#) e seleziona la regione corretta.
4. Nella pagina Create stack, verifica che l'URL del modello corretto sia visualizzato nella casella di testo URL Amazon S3 e scegli Avanti.
5. Nella pagina Specificare i dettagli dello stack, assegna un nome allo stack di soluzioni.
6. In Parametri, esamina i parametri per il modello e modificali se necessario. Questa soluzione utilizza i seguenti valori predefiniti.

Parametro	Predefinito	Descrizione
ID VPC esistente	<Optional input>	Se hai un VPC che desideri utilizzare ed è già stato creato, inserisci l'ID di un VPC esistente nella stessa regione in cui è stato distribuito lo stack. Ad esempio, vpc-1a2b3c4d5e6f.
Prima sottorete esistente	<Optional input>	L'ID della prima sottorete all'interno del VPC esistente. Questa sottorete necessita di un percorso verso Internet per recuperare l'immagine

Parametro	Predefinito	Descrizione
		del contenitore per eseguire i test. Ad esempio, subnet-7h8i9j0k.
Seconda sottorete esistente	<Optional input>	L'ID della seconda sottorete all'interno del VPC esistente . Questa sottorete necessita di un percorso verso Internet per recuperare l'immagine del contenitore per eseguire i test. Ad esempio, subnet-1x2y3z.
Blocco CIDR VPC AWS Fargate	192.168.0.0/16	Se non fornisci valori per un VPC esistente, il blocco CIDR per Amazon VPC creato dalla soluzione contiene l'indirizzo IP per AWS Fargate.
Blocco CIDR di AWS Fargate Subnet A	192.168.0.0/20	Se non fornisci valori per un VPC esistente, il blocco CIDR contiene l'indirizzo IP per la sottorete Amazon VPC A.
Blocco CIDR AWS Fargate Subnet B	192.168.16.0/20	Se non fornisci valori per un VPC esistente, il blocco CIDR contiene l'indirizzo IP per la sottorete Amazon VPC B.
Blocco CIDR del gruppo di sicurezza AWS Fargate	0.0.0.0/0	Blocco CIDR che limita l'accesso in uscita ai container Amazon ECS.

7. Scegli Next (Successivo).

8. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).

9. Nella pagina Rivedi, verifica e conferma le impostazioni. Assicurati di selezionare la casella per confermare che il modello creerà risorse AWS Identity and Access Management (IAM).

10. Seleziona Create (Crea) per implementare lo stack.

Puoi visualizzare lo stato dello stack nella CloudFormation console AWS nella colonna Status. Dovresti ricevere lo status `CREATE_COMPLETE` in circa cinque minuti.

Quando le regioni sono state distribuite correttamente, vengono visualizzate nella console Web. Quando crei un test, la nuova regione verrà elencata nella modalità Gestisci regioni. È possibile utilizzare questa regione in un test selezionandola al momento della creazione del test. La soluzione crea un elemento DynamoDB per ogni regione lanciata nella tabella degli scenari, che contiene le informazioni necessarie relative alle risorse di test in quella regione. È possibile ordinare i risultati dei test nella console Web per regione. A causa dei vincoli delle API, puoi visualizzare solo i risultati aggregati di tutte le regioni in un test multiregionale rappresentandoli graficamente nelle metriche di Amazon CloudWatch. Puoi trovare il codice sorgente del grafico nei risultati del test una volta terminato il test.

Note

Puoi avviare lo stack regionale senza la console web. Ottieni un link al modello regionale nel bucket di scenari Amazon S3 e forniscilo come origine quando avvii lo stack regionale nella regione richiesta. In alternativa, puoi scaricare il modello e caricarlo come origine per la regione che desideri.

Monitora la soluzione con Service Catalog AppRegistry

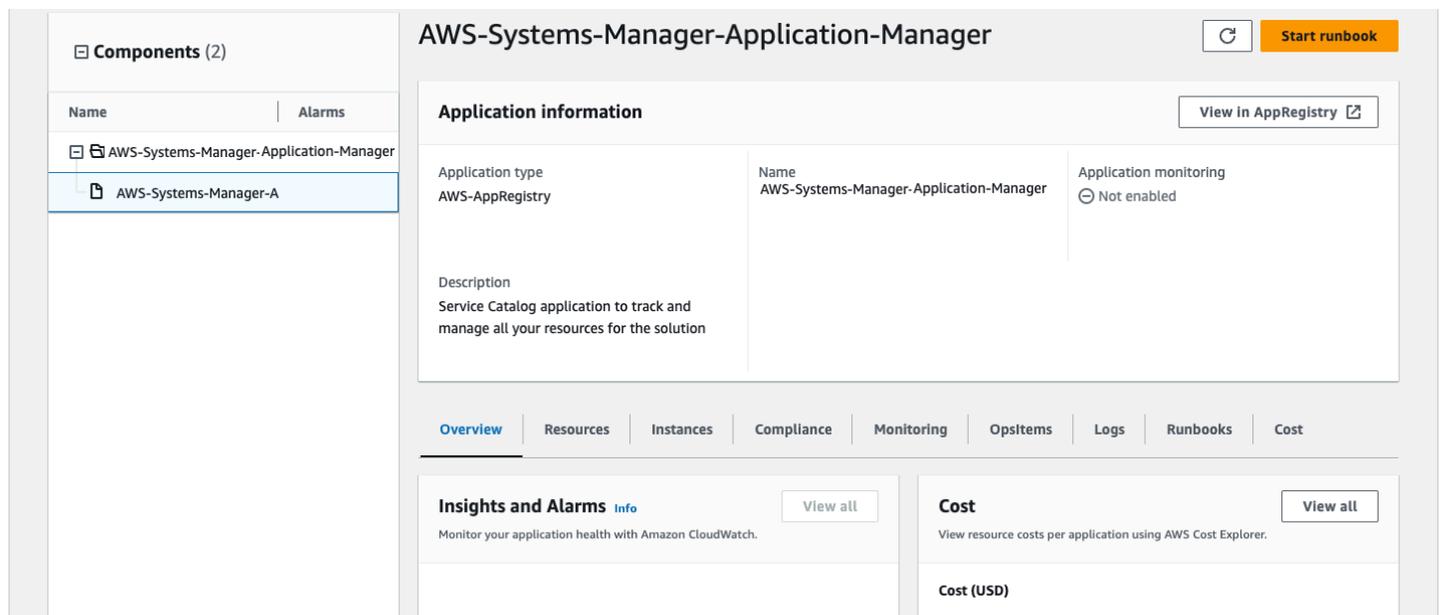
Questa soluzione include una AppRegistry risorsa Service Catalog per registrare il CloudFormation modello e le risorse sottostanti come applicazione sia in [Service Catalog AppRegistry](#) che in [AWS Systems Manager Application Manager](#).

AWS Systems Manager Application Manager ti offre una visione a livello di applicazione di questa soluzione e delle sue risorse in modo da poter:

- Monitora le risorse, i costi delle risorse distribuite su stack e account AWS e i log associati a questa soluzione da una posizione centrale.
- Visualizza i dati operativi relativi alle risorse di questa soluzione (come lo stato della distribuzione, gli CloudWatch allarmi, le configurazioni delle risorse e i problemi operativi) nel contesto di un'applicazione.

La figura seguente mostra un esempio di visualizzazione delle applicazioni per lo stack di soluzioni in Application Manager.

Rappresenta uno stack di soluzioni AWS in Application Manager



The screenshot displays the AWS Systems Manager Application Manager console. On the left, a sidebar shows a list of components under 'Components (2)', with 'AWS-Systems-Manager-A' selected. The main panel is titled 'AWS-Systems-Manager-Application-Manager' and includes a 'Start runbook' button. Below the title, there is an 'Application information' section with a 'View in AppRegistry' link. The application type is 'AWS-AppRegistry', the name is 'AWS-Systems-Manager-Application-Manager', and application monitoring is 'Not enabled'. A description states: 'Service Catalog application to track and manage all your resources for the solution'. A navigation bar at the bottom of the main panel includes tabs for Overview, Resources, Instances, Compliance, Monitoring, OpsItems, Logs, Runbooks, and Cost. Below this, there are two summary cards: 'Insights and Alarms' (with a 'View all' button) and 'Cost' (with a 'View all' button). The cost card shows 'View resource costs per application using AWS Cost Explorer.' and a 'Cost (USD)' field with a dash.

Attiva Application Insights CloudWatch

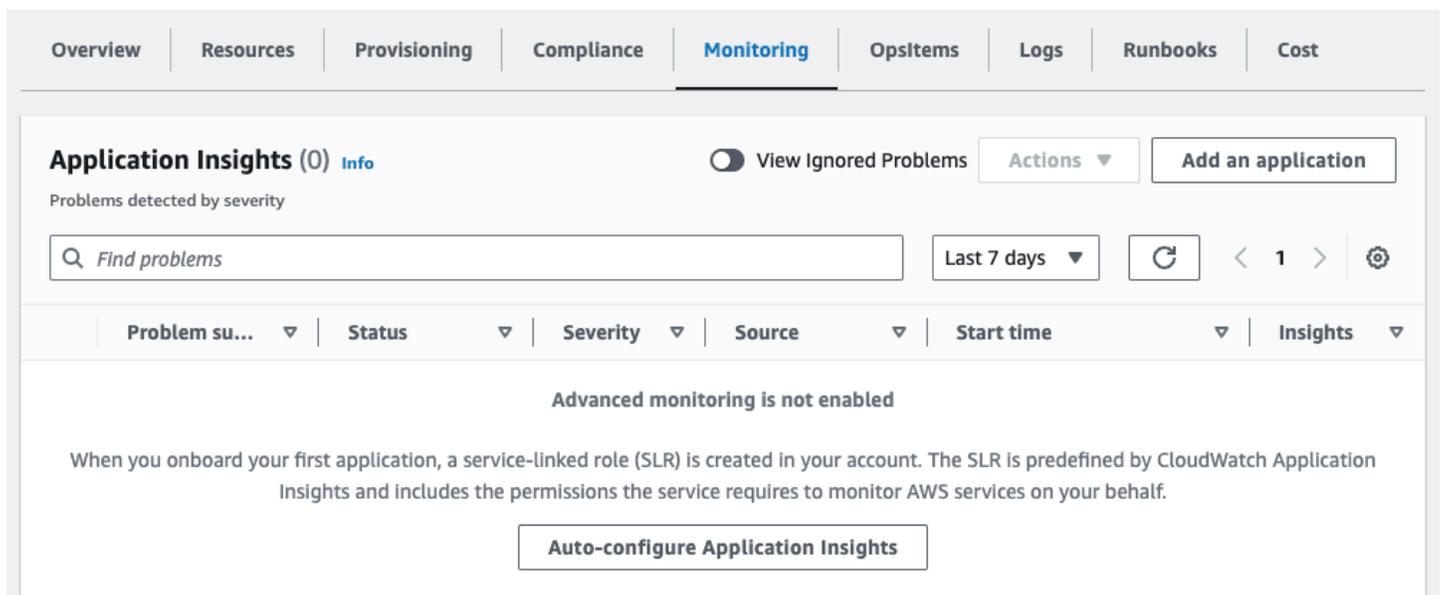
1. Accedere alla [console Systems Manager](#).

2. Nel riquadro di navigazione, scegli Application Manager.
3. In Applicazioni, cerca il nome dell'applicazione per questa soluzione e selezionalo.

Il nome dell'applicazione avrà il registro delle app nella colonna Origine dell'applicazione e avrà una combinazione del nome della soluzione, della regione, dell'ID dell'account o del nome dello stack.

4. Nell'albero dei componenti, scegliete lo stack di applicazioni che desiderate attivare.
5. Nella scheda Monitoraggio, in Application Insights, seleziona Configura automaticamente Application Insights.

La dashboard di Application Insights non mostra problemi rilevati e il monitoraggio avanzato non è abilitato



Overview | Resources | Provisioning | Compliance | **Monitoring** | Opsitems | Logs | Runbooks | Cost

Application Insights (0) [Info](#) View Ignored Problems [Actions](#) [Add an application](#)

Problems detected by severity

[Last 7 days](#) [<](#) [1](#) [>](#)

Problem su...	Status	Severity	Source	Start time	Insights
---------------	--------	----------	--------	------------	----------

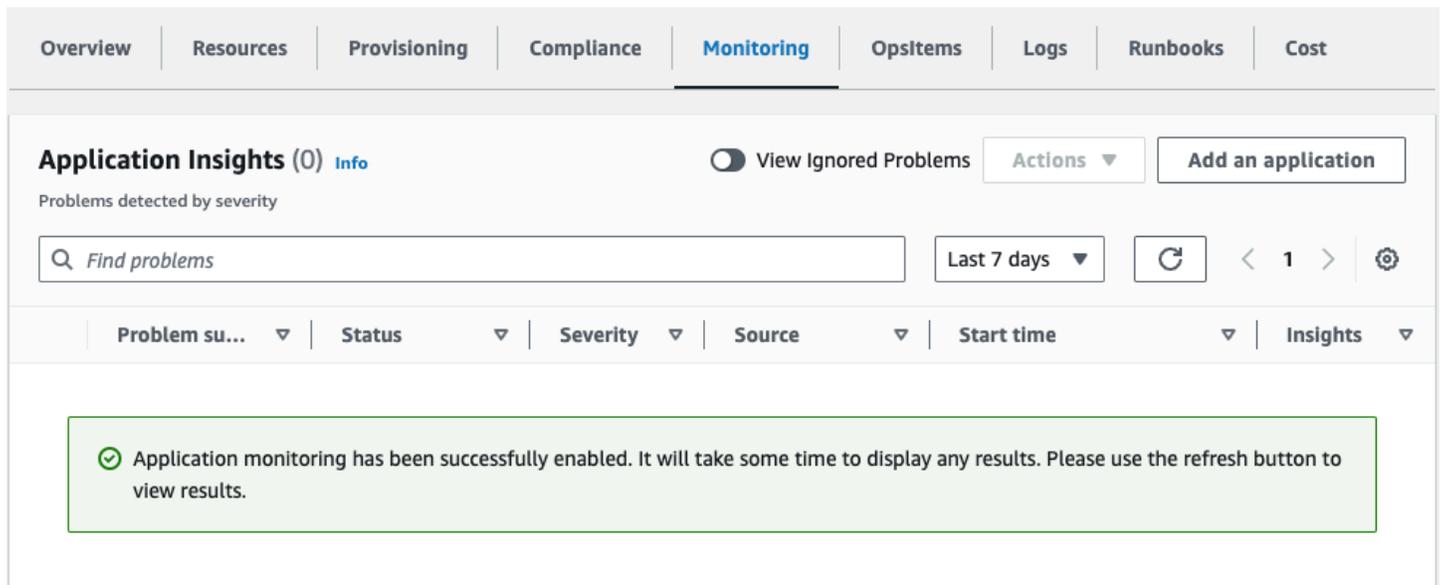
Advanced monitoring is not enabled

When you onboard your first application, a service-linked role (SLR) is created in your account. The SLR is predefined by CloudWatch Application Insights and includes the permissions the service requires to monitor AWS services on your behalf.

[Auto-configure Application Insights](#)

Il monitoraggio delle applicazioni è ora attivato e viene visualizzata la seguente casella di stato:

La dashboard di Application Insights mostra il messaggio di avvenuta attivazione del monitoraggio



Overview | Resources | Provisioning | Compliance | **Monitoring** | OpsItems | Logs | Runbooks | Cost

Application Insights (0) [info](#) View Ignored Problems Actions ▾ Add an application

Problems detected by severity

Last 7 days ▾ ↻ < 1 > ⚙️

Problem su... ▾ | **Status** ▾ | **Severity** ▾ | **Source** ▾ | **Start time** ▾ | **Insights** ▾

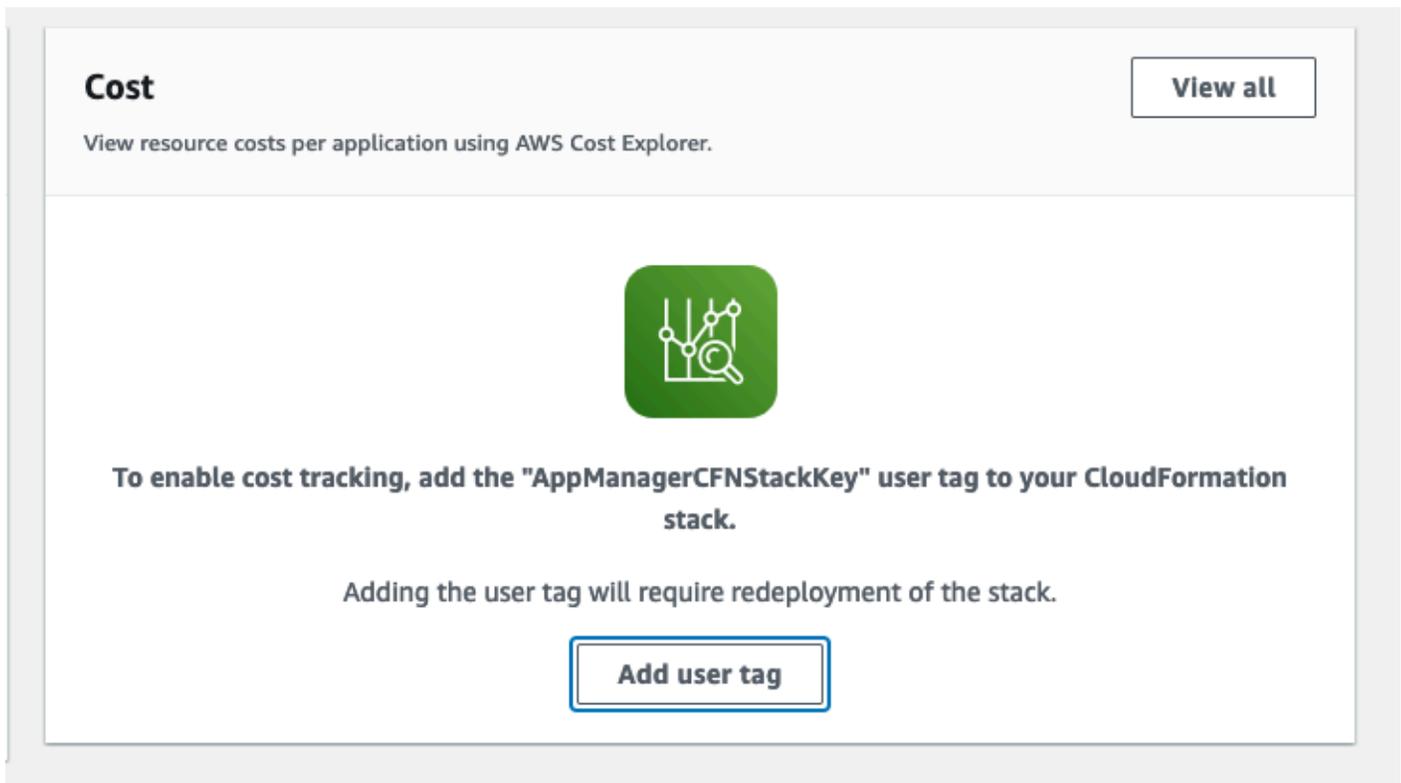
✔️ Application monitoring has been successfully enabled. It will take some time to display any results. Please use the refresh button to view results.

Conferma i cartellini dei costi associati alla soluzione

Dopo aver attivato i tag di allocazione dei costi associati alla soluzione, è necessario confermare i tag di allocazione dei costi per visualizzare i costi di questa soluzione. Per confermare i tag di allocazione dei costi:

1. Accedere alla [console Systems Manager](#).
2. Nel riquadro di navigazione, scegli Application Manager.
3. In Applicazioni, scegli il nome dell'applicazione per questa soluzione e selezionala.
4. Nella scheda Panoramica, in Costo, seleziona Aggiungi tag utente.

Schermata che mostra la schermata Application Cost Add User Tag



Cost View all

View resource costs per application using AWS Cost Explorer.



To enable cost tracking, add the "AppManagerCFNStackKey" user tag to your CloudFormation stack.

Adding the user tag will require redeployment of the stack.

Add user tag

5. Nella pagina Aggiungi tag utente, inserisci `confirm`, quindi seleziona Aggiungi tag utente.

Il completamento del processo di attivazione può richiedere fino a 24 ore e la visualizzazione dei dati del tag.

Attiva i tag di allocazione dei costi associati alla soluzione

Dopo aver confermato i tag dei costi associati a questa soluzione, è necessario attivare i tag di allocazione dei costi per visualizzare i costi di questa soluzione. I tag di allocazione dei costi possono essere attivati solo dall'account di gestione dell'organizzazione.

Per attivare i tag di allocazione dei costi:

1. Accedi alla console [AWS Billing and Cost Management and Cost Management](#).
2. Nel riquadro di navigazione, seleziona Cost Allocation Tags.
3. Nella pagina Tag di allocazione dei costi, filtra il AppManagerCFNStackKey tag, quindi seleziona il tag dai risultati visualizzati.
4. Seleziona Activate (Attiva).

AWS Cost Explorer

È possibile visualizzare la panoramica dei costi associati all'applicazione e ai componenti dell'applicazione all'interno della console di Application Manager tramite l'integrazione con AWS Cost Explorer. Cost Explorer ti aiuta a gestire i costi fornendo una panoramica dei costi e dell'utilizzo delle risorse AWS nel tempo.

1. Accedi alla [console AWS Cost Management](#).
2. Nel menu di navigazione, seleziona Cost Explorer per visualizzare i costi e l'utilizzo della soluzione nel tempo.

Aggiorna la soluzione

Se hai già distribuito la soluzione, segui questa procedura per aggiornare lo CloudFormation stack della soluzione e ottenere la versione più recente del framework della soluzione.

1. Accedi alla [CloudFormation console](#), seleziona lo CloudFormation stack esistente e seleziona **Aggiorna**.
2. Seleziona **Sostituisci modello corrente**.
3. In **Specificare il modello**:
 - a. Seleziona l'URL di Amazon S3.
 - b. Copia il link del [modello più recente](#).
 - c. Incolla il link nella casella dell'URL di Amazon S3.
 - d. Verifica che l'URL del modello corretto sia visualizzato nella casella di testo dell'URL di Amazon S3.
 - e. Scegli **Next (Successivo)**.
 - f. Scegliere **Next (Successivo)** di nuovo.
4. In **Parametri**, esamina i parametri del modello e modificali se necessario. Per informazioni dettagliate sui parametri, [consulta Launch the stack](#).
5. Scegli **Next (Successivo)**.
6. Nella pagina **Configure stack options (Configura opzioni pila)**, scegliere **Next (Successivo)**.
7. Nella pagina **Rivedi**, verifica e conferma le impostazioni.
8. Seleziona la casella riconoscendo che il modello potrebbe creare risorse IAM.
9. Scegli **Visualizza set di modifiche e verifica le modifiche**.
10. Scegli **Aggiorna stack** per distribuire lo stack.

Puoi visualizzare lo stato dello stack nella CloudFormation console AWS nella colonna **Status**. Dovresti ricevere uno `UPDATE_COMPLETE` status in circa 15 minuti.

Quando si esegue l'aggiornamento da versioni DLT precedenti alla v3.2.6 alla versione più recente, l'aggiornamento dello stack non riesce

1. Scarica il [distributed-load-testing-on-aws.template](https://github.com/aws-samples/distributed-load-testing-on-aws-template).
2. Apri il modello e vai a Condizioni: e cerca DLTCommonResourcesAppRegistryCondition
3. Dovresti vedere qualcosa di simile al seguente:

```
Conditions:
DLTCommonResourcesAppRegistryConditionCCEF54F8:
Fn::Equals:
- "true"
- "true"
```

4. Cambia il secondo valore vero in falso:

```
Conditions:
DLTCommonResourcesAppRegistryConditionCCEF54F8:
Fn::Equals:
- "true"
- "false"
```

5. Usa il modello personalizzato per aggiornare il tuo stack.
6. Questo stack rimuove le risorse relative al registro delle app dallo stack. Quindi l'aggiornamento dovrebbe essere completato.
7. Esegui un altro aggiornamento dello stack utilizzando l'URL del modello più recente per aggiungere nuovamente le risorse dell'applicazione del registro delle app allo stack.

Note

AWS Systems Manager Application Manager ti offre una visione a livello di applicazione di questa soluzione e delle sue risorse in modo da poter:

1. Monitora le risorse, i costi delle risorse distribuite su stack e account AWS e i log associati a questa soluzione da una posizione centrale.

2. Visualizza i dati operativi per le risorse di questa soluzione nel contesto di un'applicazione, come lo stato della distribuzione, gli CloudWatch allarmi, le configurazioni delle risorse e i problemi operativi.

Risoluzione dei problemi

La [risoluzione dei problemi noti](#) fornisce istruzioni per mitigare gli errori noti. Se queste istruzioni non risolvono il problema, [Contatta AWS Support](#) fornisce istruzioni per aprire un caso AWS Support per questa soluzione.

Risoluzione di problemi noti

Problema: stai utilizzando un VPC esistente e i tuoi test hanno esito negativo con lo stato Non riuscito, con conseguente messaggio di errore seguente:

```
Test might have failed to run.
```

- Risoluzione: *

[Assicurati che le sottoreti esistano nel VPC specificato e che abbiano un percorso verso Internet con un gateway Internet o un gateway NAT.](#) AWS Fargate necessita dell'accesso per estrarre l'immagine del contenitore dall'archivio pubblico per eseguire correttamente i test.

Problema: i test richiedono troppo tempo per essere eseguiti o sono bloccati a tempo indeterminato

- Risoluzione: *

Annulla il test e controlla AWS Fargate per assicurarti che tutte le attività siano state interrotte. Se non si sono interrotti, interrompi manualmente tutte le attività di Fargate. Controlla i limiti delle attività Fargate su richiesta sul tuo account per assicurarti di poter avviare il numero di attività desiderato. Puoi anche controllare CloudWatch i log della funzione Lambda task-runner per maggiori informazioni sugli errori durante l'avvio delle attività di Fargate. Controlla i log CloudWatch ECS per i dettagli su ciò che accade nei container Fargate in esecuzione.

Contattare AWS Support

Se disponi di [AWS Developer Support](#), [AWS Business Support](#) o [AWS Enterprise Support](#), puoi utilizzare il Support Center per ottenere l'assistenza di esperti su questa soluzione. Le istruzioni per eseguire tali operazioni sono fornite nelle sezioni seguenti.

Crea un caso

1. Accedi al [Support Center](#).
2. Scegli Crea caso.

Come possiamo aiutarti?

1. Scegli Technical
2. Per Assistenza, seleziona Soluzioni.
3. Per Categoria, seleziona Distributed Load Testing on AWS.
4. Per Severity, seleziona l'opzione più adatta al tuo caso d'uso.
5. Quando si inseriscono i campi Servizio, Categoria e Severità, l'interfaccia compila i collegamenti alle domande più comuni per la risoluzione dei problemi. Se non riesci a risolvere le tue domande con questi link, scegli Passaggio successivo: Informazioni aggiuntive.

Informazioni aggiuntive

1. In Oggetto, inserisci il testo che riassume la domanda o il problema.
2. Per Descrizione, descrivi il problema in dettaglio.
3. Scegli Allega file.
4. Allega le informazioni di cui AWS Support ha bisogno per elaborare la richiesta.

Aiutaci a risolvere il tuo caso più velocemente

1. Inserisci le informazioni richieste.
2. Scegli Passaggio successivo: risolvi ora o contattaci.

Risolvi subito o contattaci

1. Rivedi le soluzioni Solve now.
2. Se non riesci a risolvere il problema con queste soluzioni, scegli Contattaci, inserisci le informazioni richieste e scegli Invia.

Disinstalla la soluzione

Puoi disinstallare la soluzione Distributed Load Testing on AWS dalla Console di gestione AWS o utilizzando l'interfaccia a riga di comando AWS. È necessario eliminare manualmente la console, lo scenario e i bucket di registrazione di Amazon Simple Storage Service (Amazon S3) creati da questa soluzione. Le implementazioni delle soluzioni AWS non le eliminano automaticamente nel caso in cui siano archiviati dati da conservare.

Note

Se hai distribuito stack regionali, devi eliminare gli stack in quelle regioni prima di eliminare lo stack principale.

Utilizzando la Console di gestione AWS

1. Accedi alla [CloudFormation console AWS](#).
2. Nella pagina Stacks, seleziona lo stack di installazione di questa soluzione.
3. Scegli Elimina.

Utilizzo dell'interfaccia a riga di comando AWS

Determina se l'AWS Command Line Interface (AWS CLI) è disponibile nel tuo ambiente. Per istruzioni di installazione, consulta [What Is the AWS Command Line Interface](#) nella AWS CLI User Guide. Dopo aver verificato che la CLI di AWS è disponibile, esegui il comando seguente.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Eliminazione dei bucket Amazon S3

Questa soluzione è configurata per conservare il bucket Amazon S3 creato dalla soluzione (per la distribuzione in una regione opt-in) se decidi di eliminare lo stack CloudFormation AWS per prevenire la perdita accidentale di dati. Dopo aver disinstallato la soluzione, puoi eliminare manualmente questo bucket S3 se non hai bisogno di conservare i dati. Segui questi passaggi per eliminare il bucket Amazon S3.

1. Accedere alla [console Amazon S3](#).
2. Scegli Bucket dal riquadro di navigazione a sinistra.
3. Nel campo Trova i bucket per nome, inserisci il nome dello stack di questa soluzione.
4. Seleziona uno dei bucket S3 della soluzione e scegli Empty.
5. Inserisci l'eliminazione definitiva nel campo di verifica e scegli Vuoto.
6. Scegli il nome del bucket S3 che hai appena svuotato e scegli Elimina.
7. Inserisci il nome del bucket S3 nel campo di verifica e scegli Elimina bucket.

Ripeti i passaggi da 3 a 7 fino a eliminare tutti i bucket S3.

Per eliminare il bucket S3 utilizzando AWS CLI, esegui il seguente comando:

```
$ aws s3 rb s3://<bucket-name> --force
```

Usa la soluzione

Questa sezione include informazioni su come utilizzare la soluzione Distributed Load Testing on AWS, inclusi [i risultati dei test](#), il [flusso di lavoro di pianificazione dei test](#) e [i dati in tempo reale](#).

Risultati del test

Distributed Load Testing su AWS sfrutta il framework Load Testing per eseguire test di applicazioni su larga scala. Una volta completato un test, viene generato un rapporto dettagliato contenente i seguenti risultati.

- Tempo di risposta medio: il tempo di risposta medio, in secondi, per tutte le richieste generate dal test.
- Latenza media: la latenza media, in secondi, per tutte le richieste generate dal test.
- Tempo medio di connessione: il tempo medio, in secondi, impiegato per connettersi all'host per tutte le richieste generate dal test.
- Larghezza di banda media: la larghezza di banda media per tutte le richieste generate dal test.
- Conteggio totale: il numero totale di richieste.
- Numero di richieste riuscite: il numero totale di richieste riuscite.
- Conteggio errori: il numero totale di errori.
- Richieste al secondo: la media delle richieste al secondo per tutte le richieste generate dal test.
- Percentile: il percentile del tempo di risposta per il test. Il tempo di risposta massimo è del 100%; il tempo di risposta minimo è dello 0%.

Note

I risultati dei test vengono visualizzati nella console. Puoi visualizzare i file XML per i risultati non elaborati del test nella Results cartella del bucket Scenarios Amazon S3.

Per ulteriori informazioni sui risultati dei test Taurus, consulta [Generazione di report sui test](#) nel manuale utente di Taurus.

Flusso di lavoro di pianificazione dei test

Usa la console web per pianificare un test di carico. Quando si pianifica un test, viene eseguito il seguente flusso di lavoro:

- Quando viene creato un test di carico con l'opzione di pianificazione, i parametri di pianificazione vengono inviati all'API della soluzione tramite Amazon API Gateway.
- L'API passa quindi i parametri a una funzione Lambda che crea una regola CloudWatch Events, la cui esecuzione sarà pianificata nella data specificata.
- Se il test è monouso, la regola CloudWatch Events viene eseguita nella data specificata. La funzione `api-services` Lambda esegue un nuovo test tramite il flusso di lavoro di test.
- Se il test è ricorrente, la regola CloudWatch Events si attiva nella data specificata. Viene eseguita la funzione `api-services` Lambda, che elimina la regola CloudWatch Events corrente e crea un'altra regola che viene eseguita immediatamente al momento della creazione e successivamente in modo ricorrente in base alla frequenza di ricorrenza specificata.

Determina il numero di utenti

Il numero di utenti che un container può supportare per un test può essere determinato aumentando gradualmente il numero di utenti e monitorando le prestazioni in Amazon CloudWatch. Una volta notato che le prestazioni della CPU e della memoria si stanno avvicinando ai limiti, hai raggiunto il numero massimo di utenti che un container può supportare per quel test nella configurazione predefinita (2 vCPU e 4 GB di memoria). Puoi iniziare a determinare i limiti di utenti simultanei per il test utilizzando il seguente esempio:

1. Crea un test con non più di 200 utenti.
2. Durante l'esecuzione del test, monitora la CPU e la memoria utilizzando la [CloudWatch console](#):
 - a. Dal riquadro di navigazione a sinistra, in Container Insights, seleziona Performance Monitoring.
 - b. Nella pagina di monitoraggio delle prestazioni, dal menu a discesa a sinistra, seleziona ECS Clusters.
 - c. Dal menu a discesa a destra, seleziona il tuo cluster Amazon Elastic Container Service (Amazon ECS).
3. Durante il monitoraggio, controlla la CPU e la memoria. Se la CPU non supera il 75% o la memoria non supera l'85% (ignora i picchi occasionali), puoi eseguire un altro test con un numero maggiore di utenti.

Ripetere i passaggi 1-3 se il test non ha superato i limiti di risorse. Facoltativamente, le risorse dei contenitori possono essere aumentate per consentire un numero maggiore di utenti simultanei. Tuttavia, ciò comporta un costo più elevato. Per i dettagli, consulta la sezione [Aumentare le risorse del contenitore](#) di questa guida.

Note

Per risultati accurati, esegui solo un test alla volta per determinare i limiti degli utenti simultanei. Tutti i test utilizzano lo stesso cluster e CloudWatch Container Insights aggrega i dati sulle prestazioni in base al cluster. Ciò fa sì che entrambi i test vengano segnalati contemporaneamente a CloudWatch Container Insights, il che si traduce in metriche di utilizzo delle risorse imprecise per un singolo test.

Per ulteriori informazioni sulla calibrazione degli utenti per motore, consulta la sezione [Calibrazione di un test Taurus](#) nella documentazione. BlazeMeter

Dati in tempo reale

Facoltativamente, puoi includere dati in tempo reale durante l'esecuzione di un test per ottenere informazioni in tempo reale su ciò che sta accadendo. Il gruppo di CloudWatch log per le attività di Fargate contiene un filtro di sottoscrizione per i risultati dei test che includono l'opzione live data. Se la soluzione trova il modello, avvia una funzione Lambda che struttura i dati e li pubblica su un argomento di AWS IoT Core. La console Web sottoscrive l'argomento, riceve i dati in entrata e rappresenta graficamente i dati aggregati a intervalli di un secondo. La console Web contiene quattro grafici: tempo di risposta medio, utenti virtuali, successi e fallimenti.

Note

I dati sono effimeri e vengono utilizzati solo per vedere cosa succede durante l'esecuzione del test. Una volta completato il test, la soluzione archivia i dati dei risultati in DynamoDB e Amazon S3. La console web mantiene un massimo di 5.000 punti dati, dopodiché i dati più vecchi vengono sostituiti con i più recenti. Se la pagina si aggiorna, i grafici saranno vuoti e inizieranno dal successivo punto dati disponibile.

Flusso di lavoro di annullamento

Quando annulli un test di caricamento dalla console Web, la soluzione esegue il seguente flusso di lavoro di annullamento del test.

1. La richiesta di annullamento viene inviata all'`microservicesAPI`.
2. L'`microservicesAPI` richiama la funzione `task-canceller` Lambda che annulla le attività finché tutte le attività attualmente avviate non vengono interrotte.
3. Se la funzione `task-runner` Lambda continua a essere eseguita dopo la chiamata iniziale alla funzione `task-canceller` Lambda, le attività continueranno ad essere avviate. Al termine della funzione `task-runner` Lambda, AWS Step Functions continua con la `Cancel Test` fase, che esegue nuovamente la funzione `task-canceller` Lambda per interrompere le attività rimanenti.

Guida per sviluppatori

Questa sezione fornisce il codice sorgente della soluzione e personalizzazioni aggiuntive.

Codice sorgente

Visita il nostro [GitHub repository](#) per scaricare i modelli e gli script per questa soluzione e per condividere le tue personalizzazioni con altri.

Personalizzazione dell'immagine del contenitore

Questa soluzione utilizza un repository di immagini Amazon Elastic Container Registry (Amazon ECR) pubblico gestito da AWS per archiviare l'immagine utilizzata per eseguire i test configurati. Se desideri personalizzare l'immagine del contenitore, puoi ricostruirla e inserirla in un repository di immagini ECR nel tuo account AWS.

Se desideri personalizzare questa soluzione, puoi utilizzare l'immagine del contenitore predefinita o modificare questo contenitore in base alle tue esigenze. Se personalizzi la soluzione, utilizza il seguente esempio di codice per dichiarare le variabili di ambiente prima di creare la soluzione personalizzata.

```
#!/bin/bash
export REGION=aws-region-code # the AWS region to launch the solution (e.g. us-east-1)
export BUCKET_PREFIX=my-bucket-name # prefix of the bucket name without the region code
export BUCKET_NAME=$BUCKET_PREFIX-$REGION # full bucket name where the code will reside
export SOLUTION_NAME=my-solution-name
export VERSION=my-version # version number for the customized code
export PUBLIC_ECR_REGISTRY=public.ecr.aws/awssolutions/distributed-load-testing-on-aws-load-tester # replace with the container registry and image if you want to use a different container image
export PUBLIC_ECR_TAG=v3.1.0 # replace with the container image tag if you want to use a different container image
```

Se scegli di personalizzare l'immagine del contenitore, puoi ospitarla in un repository di immagini privato o in un repository di immagini pubblico nel tuo account AWS. Le risorse delle immagini si trovano nella `deployment/ecr/distributed-load-testing-on-aws-load-tester` directory, situata nella base di codice.

È possibile creare e inviare l'immagine alla destinazione dell'host.

- Per gli archivi e le immagini private di Amazon ECR, consulta i [repository privati e le immagini private di Amazon ECR nella Guida per l'utente di Amazon ECR](#).
- Per gli archivi e le immagini pubbliche di Amazon ECR, consulta gli [archivi pubblici e le immagini pubbliche di Amazon ECR nella Amazon ECR Public User Guide](#).

Dopo aver creato la tua immagine, puoi dichiarare le seguenti variabili di ambiente prima di creare la tua soluzione personalizzata.

```
#!/bin/bash
export PUBLIC_ECR_REGISTRY=YOUR_ECR_REGISTRY_URI # e.g. YOUR_ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/YOUR_IMAGE_NAME
export PUBLIC_ECR_TAG=YOUR_ECR_TAG # e.g. latest, v2.0.0
```

L'esempio seguente mostra il file contenitore.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023-minimal

RUN dnf update -y && \
    dnf install -y python3.11 python3.11-pip java-21-amazon-corretto bc procps jq
    findutils unzip && \
    dnf clean all

ENV PIP_INSTALL="pip3.11 install --no-cache-dir"

# install bzt
RUN $PIP_INSTALL --upgrade bzt awscli setuptools==70.0.0

# install bzt tools
RUN bzt -install-tools -o modules.install-checker.exclude=selenium,gatling,tsung,siege,ab,k6,external-results-loader,locust,junit,testng,rSpec,mocha,nunit,xunit,wdio
RUN rm -rf /root/.bzt/selenium-taurus
RUN mkdir /bzt-configs /tmp/artifacts
ADD ./load-test.sh /bzt-configs/
ADD ./*.jar /bzt-configs/
ADD ./*.py /bzt-configs/

RUN chmod 755 /bzt-configs/load-test.sh
RUN chmod 755 /bzt-configs/ecslister.py
RUN chmod 755 /bzt-configs/ecscontroller.py
```

```

RUN chmod 755 /bzt-configs/jar_updater.py
RUN python3.11 /bzt-configs/jar_updater.py

# Remove jar files from /tmp
RUN rm -rf /tmp/jmeter-plugins-manager-1.7*

# Add settings file to capture the output logs from bzt cli
RUN mkdir -p /etc/bzt.d && echo '{"settings": {"artifacts-dir": "/tmp/artifacts"}}' > /
etc/bzt.d/90-artifacts-dir.json

WORKDIR /bzt-configs
ENTRYPOINT ["/load-test.sh"]

```

Oltre a un file contenitore, la directory contiene il seguente script bash che scarica la configurazione di test da Amazon S3 prima di eseguire il programma Taurus/Blazemeter.

```

#!/bin/bash

# set a uuid for the results xml file name in S3
UUID=$(cat /proc/sys/kernel/random/uuid)
pypid=0
echo "S3_BUCKET:: ${S3_BUCKET}"
echo "TEST_ID:: ${TEST_ID}"
echo "TEST_TYPE:: ${TEST_TYPE}"
echo "FILE_TYPE:: ${FILE_TYPE}"
echo "PREFIX:: ${PREFIX}"
echo "UUID:: ${UUID}"
echo "LIVE_DATA_ENABLED:: ${LIVE_DATA_ENABLED}"
echo "MAIN_STACK_REGION:: ${MAIN_STACK_REGION}"

cat /proc/self/cgroup
TASK_ID=$(cat /proc/self/cgroup | grep -oE '[a-f0-9]{32}' | head -n 1)
echo $TASK_ID

sigterm_handler() {
    if [ $pypid -ne 0 ]; then
        echo "container received SIGTERM."
        kill -15 $pypid
        wait $pypid
        exit 143 #128 + 15
    fi
}

trap 'sigterm_handler' SIGTERM

```

```
echo "Download test scenario"
aws s3 cp s3://$S3_BUCKET/test-scenarios/$TEST_ID-$AWS_REGION.json test.json --region
  $MAIN_STACK_REGION

# Set the default log file values to jmeter
LOG_FILE="jmeter.log"
OUT_FILE="jmeter.out"
ERR_FILE="jmeter.err"
KPI_EXT="jtl"

# download JMeter jmx file
if [ "$TEST_TYPE" != "simple" ]; then
  # setting the log file values to the test type
  LOG_FILE="${TEST_TYPE}.log"
  OUT_FILE="${TEST_TYPE}.out"
  ERR_FILE="${TEST_TYPE}.err"

  # set variables based on TEST_TYPE
  if [ "$TEST_TYPE" == "jmeter" ]; then
    EXT="jmx"
    TYPE_NAME="JMeter"
    # Copy *.jar to JMeter library path. See the Taurus JMeter path: https://
gettaurus.org/docs/JMeter/
    JMETER_LIB_PATH=`find ~/.bzt/jmeter-taurus -type d -name "lib"`
    echo "cp $PWD/*.jar $JMETER_LIB_PATH"
    cp $PWD/*.jar $JMETER_LIB_PATH

  fi

  if [ "$FILE_TYPE" != "zip" ]; then
    aws s3 cp s3://$S3_BUCKET/public/test-scenarios/$TEST_TYPE/$TEST_ID.$EXT ./ --
region $MAIN_STACK_REGION
  else
    aws s3 cp s3://$S3_BUCKET/public/test-scenarios/$TEST_TYPE/$TEST_ID.zip ./ --region
$MAIN_STACK_REGION
    unzip $TEST_ID.zip
    echo "UNZIPPED"
    ls -l
    # only looks for the first test script file.
    TEST_SCRIPT=`find . -name "*.${EXT}" | head -n 1`
    echo $TEST_SCRIPT
    if [ -z "$TEST_SCRIPT" ]; then
      echo "There is no test script (}.${EXT}) in the zip file."
```

```

    exit 1
fi

sed -i -e "s|${TEST_ID}.${EXT}|${TEST_SCRIPT}|g" test.json

# copy bundled plugin jars to jmeter extension folder to make them available to
jmeter
BUNDLED_PLUGIN_DIR=`find $PWD -type d -name "plugins" | head -n 1`
# attempt to copy only if a /plugins folder is present in upload
if [ -z "$BUNDLED_PLUGIN_DIR" ]; then
    echo "skipping plugin installation (no /plugins folder in upload)"
else
    # ensure the jmeter extensions folder exists
    JMETER_EXT_PATH=`find ~/.bzt/jmeter-taurus -type d -name "ext"`
    if [ -z "$JMETER_EXT_PATH" ]; then
        # fail fast - if plugins bundled they will be needed for the tests
        echo "jmeter extension path (~/.bzt/jmeter-taurus/**/ext) not found - cannot
install bundled plugins"
        exit 1
    fi
    cp -v $BUNDLED_PLUGIN_DIR/*.jar $JMETER_EXT_PATH
fi
fi
fi

#Download python script
if [ -z "$IPNETWORK" ]; then
    python3.11 -u $SCRIPT $TIMEOUT &
    pypid=$!
    wait $pypid
    pypid=0
else
    aws s3 cp s3://$S3_BUCKET/Container_IPs/${TEST_ID}_IPHOSTS_${AWS_REGION}.txt ./ --
region $MAIN_STACK_REGION
    export IPHOSTS=$(cat ${TEST_ID}_IPHOSTS_${AWS_REGION}.txt)
    python3.11 -u $SCRIPT $IPNETWORK $IPHOSTS
fi

echo "Running test"

stdbuf -i0 -o0 -e0 bzt test.json -o modules.console.disable=true | stdbuf -i0 -o0 -e0
tee -a result.tmp | sed -u -e "s|^|${TEST_ID} $LIVE_DATA_ENABLED |"
CALCULATED_DURATION=`cat result.tmp | grep -m1 "Test duration" | awk -F ' ' '{ print
$5 }' | awk -F ':' '{ print ($1 * 3600) + ($2 * 60) + $3 }`

```

```
# upload custom results to S3 if any
# every file goes under $TEST_ID/$PREFIX/$UUID to distinguish the result correctly
if [ "$TEST_TYPE" != "simple" ]; then
  if [ "$FILE_TYPE" != "zip" ]; then
    cat $TEST_ID.$EXT | grep filename > results.txt
  else
    cat $TEST_SCRIPT | grep filename > results.txt
  fi

  if [ -f results.txt ]; then
    sed -i -e 's/<stringProp name="filename">//g' results.txt
    sed -i -e 's/<\/stringProp>//g' results.txt
    sed -i -e 's/ //g' results.txt

    echo "Files to upload as results"
    cat results.txt

    files=(`cat results.txt`)
    extensions=()
    for f in "${files[@]}"; do
      ext="${f##*.}"
      if [[ ! " ${extensions[@]} " =~ " ${ext} " ]]; then
        extensions+=("${ext}")
      fi
    done

    # Find all files in the current folder with the same extensions
    all_files=()
    for ext in "${extensions[@]}"; do
      for f in *."$ext"; do
        all_files+=("$f")
      done
    done

    for f in "${all_files[@]}"; do
      p="s3://$S3_BUCKET/results/$TEST_ID/${TYPE_NAME}_Result/$PREFIX/$UUID/$f"
      if [[ $f = /* ]]; then
        p="s3://$S3_BUCKET/results/$TEST_ID/${TYPE_NAME}_Result/$PREFIX/$UUID$f"
      fi

      echo "Uploading $p"
      aws s3 cp $f $p --region $MAIN_STACK_REGION
    done
  fi
fi
```

```

    fi
fi

if [ -f /tmp/artifacts/results.xml ]; then

    # Insert the Task ID at the same level as <FinalStatus>
    curl -s $ECS_CONTAINER_METADATA_URI_V4/task
    Task_CPU=$(curl -s $ECS_CONTAINER_METADATA_URI_V4/task | jq '.Limits.CPU')
    Task_Memory=$(curl -s $ECS_CONTAINER_METADATA_URI_V4/task | jq '.Limits.Memory')
    START_TIME=$(curl -s "$ECS_CONTAINER_METADATA_URI_V4/task" | jq -r
'.Containers[0].StartedAt')
    # Convert start time to seconds since epoch
    START_TIME_EPOCH=$(date -d "$START_TIME" +%s)
    # Calculate elapsed time in seconds
    CURRENT_TIME_EPOCH=$(date +%s)
    ECS_DURATION=$((CURRENT_TIME_EPOCH - START_TIME_EPOCH))

    sed -i.bak 's/<\FinalStatus>/<TaskId>"$TASK_ID"<\TaskId><\FinalStatus>/' /tmp/
artifacts/results.xml
    sed -i 's/<\FinalStatus>/<TaskCPU>"$Task_CPU"<\TaskCPU><\FinalStatus>/' /tmp/
artifacts/results.xml
    sed -i 's/<\FinalStatus>/<TaskMemory>"$Task_Memory"<\TaskMemory><\
FinalStatus>/' /tmp/artifacts/results.xml
    sed -i 's/<\FinalStatus>/<ECSDuration>"$ECS_DURATION"<\ECSDuration><\
FinalStatus>/' /tmp/artifacts/results.xml

    echo "Validating Test Duration"
    TEST_DURATION=$(grep -E '<TestDuration>[0-9]+.[0-9]+</TestDuration>' /tmp/artifacts/
results.xml | sed -e 's/<TestDuration> //' | sed -e 's/<\TestDuration> //')

    if (( $(echo "$TEST_DURATION > $CALCULATED_DURATION" | bc -l) )); then
        echo "Updating test duration: $CALCULATED_DURATION s"
        sed -i.bak.td 's/<TestDuration>[0-9]*\.[0-9]*<\TestDuration>/
<TestDuration>"$CALCULATED_DURATION"<\TestDuration>/' /tmp/artifacts/results.xml
    fi

    if [ "$TEST_TYPE" == "simple" ]; then
        TEST_TYPE="jmeter"
    fi

    echo "Uploading results, bzt log, and JMeter log, out, and err files"
    aws s3 cp /tmp/artifacts/results.xml s3://$S3_BUCKET/results/${TEST_ID}/${PREFIX}-
${UUID}-${AWS_REGION}.xml --region $MAIN_STACK_REGION

```

```
aws s3 cp /tmp/artifacts/bzt.log s3://$S3_BUCKET/results/${TEST_ID}/bzt-${PREFIX}-
${UUID}-${AWS_REGION}.log --region $MAIN_STACK_REGION
aws s3 cp /tmp/artifacts/$LOG_FILE s3://$S3_BUCKET/results/${TEST_ID}/${TEST_TYPE}-
${PREFIX}-${UUID}-${AWS_REGION}.log --region $MAIN_STACK_REGION
aws s3 cp /tmp/artifacts/$OUT_FILE s3://$S3_BUCKET/results/${TEST_ID}/${TEST_TYPE}-
${PREFIX}-${UUID}-${AWS_REGION}.out --region $MAIN_STACK_REGION
aws s3 cp /tmp/artifacts/$ERR_FILE s3://$S3_BUCKET/results/${TEST_ID}/${TEST_TYPE}-
${PREFIX}-${UUID}-${AWS_REGION}.err --region $MAIN_STACK_REGION
aws s3 cp /tmp/artifacts/kpi.${KPI_EXT} s3://$S3_BUCKET/results/${TEST_ID}/kpi-
${PREFIX}-${UUID}-${AWS_REGION}.${KPI_EXT} --region $MAIN_STACK_REGION

else
  echo "An error occurred while the test was running."
fi
```

Oltre al [Dockerfile](#) e allo script bash, nella directory sono inclusi anche due script Python. Ogni attività esegue uno script Python dall'interno dello script bash. Le attività di lavoro eseguono lo `ecslister.py` script, mentre l'attività principale eseguirà lo script `ecscontroller.py`. Lo `ecslister.py` script crea un socket sulla porta 50000 e attende un messaggio. Lo `ecscontroller.py` script si connette al socket e invia il messaggio di test di avvio alle attività del lavoratore, che consente loro di avviarsi contemporaneamente.

API di test di carico distribuita

Questa soluzione di test di carico consente di esporre i dati dei risultati del test in modo sicuro. L'API funge da «porta d'ingresso» per l'accesso ai dati di test archiviati in Amazon DynamoDB. Puoi anche utilizzare le API per accedere a qualsiasi funzionalità estesa incorporata nella soluzione.

Questa soluzione utilizza un pool di utenti Amazon Cognito integrato con Amazon API Gateway per l'identificazione e l'autorizzazione. Quando un pool di utenti viene utilizzato con l'API, i client possono chiamare i metodi attivati dal pool di utenti solo dopo aver fornito un token di identità valido.

Per ulteriori informazioni sull'esecuzione dei test direttamente tramite l'API, consulta [Signing Requests](#) nella documentazione di riferimento dell'API REST di Amazon API Gateway.

Le seguenti operazioni sono disponibili nell'API della soluzione.

Note

Per ulteriori informazioni testScenario e altri parametri, consulta [gli scenari e gli esempi di payload](#) nel GitHub repository.

Scenari

- [GET /scenarios](#)
- [POST /scenari](#)
- [OPZIONI/scenari](#)
- [OTTIENI /scenarios/ {testID}](#)
- [PUBBLICA /scenarios/ {testID}](#)
- [ELIMINA /scenarios/ {testID}](#)
- [OPZIONI /scenarios/ {testID}](#)

Attività

- [OTTIENI /tasks](#)
- [OPZIONI /tasks](#)

Regioni

- [GET /regions](#)
- [OPZIONI/regioni](#)

GET /scenarios**Descrizione**

L'GET /scenariosoperazione consente di recuperare un elenco di scenari di test.

Risposta

Nome	Descrizione
data	Un elenco di scenari che include l'ID, il nome, la descrizione, lo stato e il tempo di esecuzione per ogni test

POST /scenari

Descrizione

L'POST /scenariosoperazione consente di creare o pianificare uno scenario di test.

Corpo della richiesta

Nome	Descrizione
testName	Il nome del test
testDescription	La descrizione del test
testTaskConfigs	Un oggetto che specifica concurrency (il numero di esecuzioni parallele), taskCount (il numero di attività necessarie region per eseguire un test) e lo scenario
testScenario	La definizione del test che include concorrenza, tempo di test, host e metodo per il test
testType	Il tipo di test (ad esempio simple, jmeter)
fileType	Il tipo di file da caricare (ad esempio none, script, zip)
scheduleDate	La data in cui eseguire un test. Fornito solo se si pianifica un test (ad esempio, 2021-02-28)

Nome	Descrizione
<code>scheduleTime</code>	Il tempo necessario per eseguire un test. Fornito solo se si pianifica un test (ad esempio, 21:07)
<code>scheduleStep</code>	Fase del processo di pianificazione. Fornito solo se si pianifica un test ricorrente. (I passaggi disponibili includono e) create start
<code>cronvalue</code>	Il valore cron per personalizzare la pianificazione ricorrente. Se usato, ometti Scheduledate e ScheduleTime.
<code>cronExpiryDate</code>	Data obbligatoria in modo che il cron scada e non venga eseguito all'infinito.
<code>recurrence</code>	La ricorrenza di un test programmato. Fornito solo se si pianifica un test ricorrente (ad esempio, ,, daily o) weekly biweekly monthly

Risposta

Nome	Descrizione
<code>testId</code>	L'ID univoco del test
<code>testName</code>	Il nome del test
<code>status</code>	Lo stato del test

OPZIONI/scenari

Descrizione

L'OPTIONS /scenariosoperazione fornisce una risposta alla richiesta con le intestazioni di risposta CORS corrette.

Risposta

Nome	Descrizione
testId	L'ID univoco del test
testName	Il nome del test
status	Lo stato del test

GET /scenarios/ {testID}

Descrizione

L'GET /scenarios/{testId}operazione consente di recuperare i dettagli di uno scenario di test specifico.

Parametro di richiesta

testId

- L'ID univoco del test

Tipo: stringa

Campo obbligatorio: sì

Risposta

Nome	Descrizione
testId	L'ID univoco del test

Nome	Descrizione
testName	Il nome del test
testDescription	La descrizione del test
testType	Il tipo di test che viene eseguito (ad esempio, <code>simple</code> , <code>jmeter</code>)
fileType	Il tipo di file che viene caricato (ad esempio, <code>json</code> , <code>script</code> , <code>zip</code>)
status	Lo stato del test
startTime	L'ora e la data di inizio dell'ultimo test
endTime	L'ora e la data in cui è terminato l'ultimo test
testScenario	La definizione del test che include concorrenza, ora del test, host e metodo per il test
taskCount	Il numero di attività necessarie per eseguire il test
taskIds	Un elenco di attività IDs per l'esecuzione dei test
results	I risultati finali del test
history	Un elenco dei risultati finali dei test precedenti
errorReason	Un messaggio di errore generato quando si verifica un errore
nextRun	La prossima esecuzione pianificata (ad esempio, <code>2017-04-22 17:18:00</code>)
scheduleRecurrence	La ricorrenza del test (ad esempio, <code>daily</code> , <code>weekly</code> , <code>biweekly</code>) <code>monthly</code>

POST /scenarios/ {testID}

Descrizione

L'POST /scenarios/{testId}operazione consente di annullare uno scenario di test specifico.

Parametro di richiesta

testId

- L'ID univoco del test

Tipo: stringa

Campo obbligatorio: sì

Risposta

Nome	Descrizione
status	Lo stato del test

DELETE /scenarios/ {testID}

Descrizione

L'DELETE /scenarios/{testId}operazione consente di eliminare tutti i dati relativi a uno scenario di test specifico.

Parametro di richiesta

testId

- L'ID univoco del test

Tipo: stringa

Campo obbligatorio: sì

Risposta

Nome	Descrizione
status	Lo stato del test

OPZIONI /scenarios/ {testID}

Descrizione

L'OPTIONS /scenarios/{testId}operazione fornisce una risposta alla richiesta con le intestazioni di risposta CORS corrette.

Risposta

Nome	Descrizione
testId	L'ID univoco del test
testName	Il nome del test
testDescription	La descrizione del test
testType	Il tipo di test che viene eseguito (ad esempio simple, jmeter)
fileType	Il tipo di file che viene caricato (ad esempio none, script, zip)
status	Lo stato del test
startTime	L'ora e la data di inizio dell'ultimo test
endTime	L'ora e la data in cui è terminato l'ultimo test
testScenario	La definizione del test che include concorrenza, ora del test, host e metodo per il test

Nome	Descrizione
taskCount	Il numero di attività necessarie per eseguire il test
taskIds	Un elenco di attività IDs per l'esecuzione dei test
results	I risultati finali del test
history	Un elenco dei risultati finali dei test precedenti
errorReason	Un messaggio di errore generato quando si verifica un errore

OTTIENI /tasks

Descrizione

L'GET /tasksoperazione consente di recuperare un elenco di attività Amazon Elastic Container Service (Amazon ECS) in esecuzione.

Risposta

Nome	Descrizione
tasks	Un elenco di attività IDs per l'esecuzione dei test

OPZIONI/task

Descrizione

L'operazione OPTIONS /tasks tasks fornisce una risposta alla richiesta con le intestazioni di risposta CORS corrette.

Risposta

Nome	Descrizione
taskIds	Un elenco di attività IDs per l'esecuzione dei test

GET /regions

Descrizione

L'GET /regionsoperazione consente di recuperare le informazioni sulle risorse regionali necessarie per eseguire un test in quella regione.

Risposta

Nome	Descrizione
testId	L'ID della regione
ecsCloudWatchLogGroup	Il nome del gruppo di CloudWatch log di Amazon per le attività di Amazon Fargate nella regione
region	La regione in cui esistono le risorse della tabella
subnetA	L'ID di una delle sottoreti della regione
subnetB	L'ID di una delle sottoreti nella regione
taskCluster	Il nome del cluster AWS Fargate nella regione
taskDefinition	L'ARN della definizione dell'attività nella regione
taskImage	Il nome dell'immagine dell'attività nella regione
taskSecurityGroup	L'ID del gruppo di sicurezza nella regione

OPZIONI/regioni

Descrizione

L'OPTIONS /regionsoperazione fornisce una risposta alla richiesta con le intestazioni di risposta CORS corrette.

Risposta

Nome	Descrizione
testId	L'ID della regione
ecsCloudWatchLogGroup	Il nome del gruppo di CloudWatch log di Amazon per le attività di Amazon Fargate nella regione
region	La regione in cui esistono le risorse della tabella
subnetA	L'ID di una delle sottoreti della regione
subnetB	L'ID di una delle sottoreti nella regione
taskCluster	Il nome del cluster AWS Fargate nella regione
taskDefinition	L'ARN della definizione dell'attività nella regione
taskImage	Il nome dell'immagine dell'attività nella regione
taskSecurityGroup	L'ID del gruppo di sicurezza nella regione

Aumenta le risorse del contenitore

Per aumentare il numero di utenti attualmente supportati, aumenta le risorse del container. Ciò consente di aumentare la memoria CPUs necessaria per gestire l'aumento degli utenti simultanei.

Crea una nuova revisione della definizione delle attività

1. Accedi alla [console Amazon Elastic Container Service](#).
2. Nel menu di navigazione a sinistra, seleziona Task Definitions.
3. Seleziona la casella di controllo accanto alla definizione dell'attività che corrisponde a questa soluzione. Ad esempio, `[replaceable] <stackName>`- EcsTaskDefinition -<system-generated-random-Hash>`.
4. Scegliere Create new revision (Crea nuova revisione).
5. Nella pagina Crea nuova revisione, esegui le seguenti azioni:
 - a. In Dimensioni dell'attività, modifica la memoria dell'attività e la CPU dell'attività.
 - b. In Definizioni dei contenitori, esamina i limiti di memoria hard/soft. Se questo limite è inferiore alla memoria desiderata, scegli il contenitore.
 - c. Nella finestra di dialogo Modifica contenitore, vai a Limiti di memoria e aggiorna il limite rigido alla memoria desiderata.
 - d. Scegli Aggiorna.
6. Nella pagina Crea nuova revisione, scegli Crea.
7. Dopo aver creato correttamente la definizione dell'attività, registra il nome della nuova definizione dell'attività. Questo nome include il numero di versione, ad esempio:
`[replaceable] <stackName>`- EcsTaskDefinition -<system-generated-random-Hash>:
[sostituibile]<system-generated-versionNumber>`.

Aggiornare la tabella DynamoDB

1. Accedere alla console [DynamoDB](#).
2. Dal riquadro di navigazione a sinistra, seleziona Esplora gli elementi in Tabelle.
3. Seleziona la tabella `scenarios-table` DynamoDB associata a questa soluzione. Ad esempio, `[replaceable] <stackName>`- DLTTest RunnerStorage DLTScenarios Tabella- <system-generated-random-Hash>`
4. Seleziona l'elemento che corrisponde alla regione in cui hai modificato la definizione dell'attività. Ad esempio, `region-<region-name>`.
5. Aggiornare l'attributo TaskDefinition con la nuova definizione dell'attività.

Riferimento

Questa sezione include informazioni su una funzionalità opzionale per la raccolta di metriche uniche per questa soluzione, riferimenti a risorse correlate e un elenco di sviluppatori che hanno contribuito a questa soluzione.

Raccolta di dati anonimizzata

Questa soluzione include un'opzione per inviare metriche operative anonime ad AWS. Utilizziamo questi dati per comprendere meglio come i clienti utilizzano questa soluzione e i servizi e i prodotti correlati. Quando vengono richiamate, le seguenti informazioni vengono raccolte e inviate ad AWS:

- ID della soluzione: l'identificatore della soluzione AWS
- ID univoco (UUID): identificatore univoco generato casualmente per ogni implementazione della soluzione
- Timestamp: timestamp di raccolta dati
- Tipo di test: il tipo di test che viene eseguito
- Tipo di file: il tipo di file che viene caricato
- Numero attività: il conteggio delle attività per ogni test inviato tramite l'API della soluzione
- Durata dell'attività: il tempo di esecuzione totale per tutte le attività necessarie per eseguire un test
- Risultato del test: il risultato del test eseguito

AWS possiede i dati raccolti tramite questo sondaggio. La raccolta dei dati è soggetta alla [politica sulla privacy di AWS](#). Per disattivare questa funzionalità, completa i seguenti passaggi prima di avviare il CloudFormation modello AWS.

1. Scarica il [CloudFormation modello AWS](#) sul tuo disco rigido locale.
2. Apri il CloudFormation modello AWS con un editor di testo.
3. Modifica la sezione di mappatura dei CloudFormation modelli AWS da:

```
Solution:  
  Config:  
    SendAnonymousData: "Yes"
```

to:

```
Solution:  
Config:  
  SendAnonymousData: "No"
```

4. Accedi alla [CloudFormation console AWS](#).
5. Seleziona Crea stack.
6. Nella pagina Crea stack, sezione Specificare il modello, seleziona Carica un file modello.
7. In Carica un file modello, scegli Scegli file e seleziona il modello modificato dall'unità locale.
8. Scegli Avanti e segui i passaggi indicati in [Avvia lo stack](#) nella sezione Implementa la soluzione di questa guida.

Collaboratori

- Tom Nightingale
- Fernando Dingler
- Lee Beomseok
- Georg Lenz
- Erin McGill
- Dimitri López
- Kamyar Ziabari
- Bassem Wanis
- Garvit Singh
- Nikhil Reddy

Revisioni

Visita [Changelog.md](#) nel nostro GitHub repository per tenere traccia dei miglioramenti e delle correzioni specifici della versione.

Note

I clienti sono responsabili della propria valutazione indipendente delle informazioni contenute nel presente documento. Questo documento: (a) è solo a scopo informativo, (b) rappresenta le attuali offerte e pratiche di prodotti AWS, che sono soggette a modifiche senza preavviso, e (c) non crea alcun impegno o garanzia da parte di AWS e delle sue affiliate, fornitori o licenzianti. I prodotti o i servizi AWS sono forniti «così come sono» senza garanzie, dichiarazioni o condizioni di alcun tipo, esplicite o implicite. Le responsabilità e gli obblighi di AWS nei confronti dei propri clienti sono controllati da accordi AWS e questo documento non fa parte né modifica alcun accordo tra AWS e i suoi clienti.

Distributed Load Testing on AWS è concesso in licenza secondo i termini della versione 2.0 della licenza Apache, disponibile presso [The Apache](#) Software Foundation.

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.