Guía de implementación

Pruebas de carga distribuidas en AWS

Pruebas de carga distribuidas en AWS: Guía de implementación

Table of Contents

Automatice las pruebas de sus aplicaciones de software a escala

Fecha de publicación: diciembre de 2025

Las pruebas de carga distribuidas en AWS le ayudan a automatizar las pruebas de rendimiento de sus aplicaciones de software a escala para identificar los cuellos de botella antes de lanzar la aplicación. Esta solución simula que miles de usuarios conectados generan solicitudes HTTP a un ritmo sostenido sin necesidad de aprovisionar servidores.

Esta solución aprovecha Amazon Elastic Container Service (Amazon ECS) en AWS Fargate para implementar contenedores que ejecuten sus simulaciones de pruebas de carga y ofrece las siguientes capacidades:

- Implemente Amazon ECS en contenedores de AWS Fargate que se ejecutan de forma independiente para probar la capacidad de carga de su aplicación.
- Simule decenas de miles de usuarios simultáneos en varias regiones de AWS y genere solicitudes a un ritmo continuo.
- Personalice las pruebas de sus aplicaciones mediante K6 JMeter, los scripts de prueba de Locust o una sencilla configuración de punto final HTTP.
- Programa las pruebas de carga para que se ejecuten de forma inmediata, en una fecha y hora futuras o de forma periódica.
- Ejecuta varias pruebas de carga de forma simultánea en diferentes escenarios y regiones.

Esta guía de implementación proporciona una descripción general de la solución Distributed Load Testing en AWS, su arquitectura y componentes de referencia, las consideraciones para planificar la implementación y los pasos de configuración para implementar la solución en la nube de Amazon Web Services (AWS). Incluye enlaces a una CloudFormation plantilla de AWS que lanza y configura los servicios de AWS necesarios para implementar esta solución utilizando las prácticas recomendadas de AWS en materia de seguridad y disponibilidad.

El público al que va dirigido el uso de las funciones y capacidades de esta solución en su entorno incluye arquitectos, administradores y DevOps profesionales de infraestructuras de TI con experiencia práctica en la arquitectura en la nube de AWS.

Utilice esta tabla de navegación para encontrar rápidamente las respuestas a estas preguntas:

Si quiere…	Lea…
Conocer el costo de ejecutar esta solución. El costo estimado de ejecutar esta solución en la región EE.UU. Este (Norte de Virginia) es de 30,90 USD al mes para los recursos de AWS.	Costo
Comprender las consideraciones de seguridad de esta solución.	Seguridad
Saber cómo planificar las cuotas de esta solución.	Cuotas
Conozca qué regiones de AWS admiten esta solución.	Regiones de AWS admitidas
Obtenga información sobre el servidor MCP opcional para el análisis de pruebas de carga asistido por IA.	Integración del servidor MCP
Consulte o descargue la CloudFormation plantilla de AWS incluida en esta solución para implementar automáticamente los recursos de infraestructura (la «pila») de esta solución.	CloudFormation Plantilla de AWS
Acceder al código fuente y, opcionalmente, utilizar AWS Cloud Development Kit (AWS CDK) para implementar la solución.	GitHub repositorio

Características

La solución ofrece las siguientes características:

Soporte para múltiples marcos de pruebas

Admite JMeter scripts de prueba K6 y Locust, así como pruebas sencillas de puntos finales HTTP sin necesidad de scripts personalizados. Para obtener más información, consulte los tipos de pruebas en la sección de detalles de la arquitectura.

Simulación de alta carga de usuarios

Simula decenas de miles de usuarios virtuales simultáneos para poner a prueba su aplicación en condiciones de carga realistas.

Distribución de carga multirregional

Distribuye las pruebas de carga en varias regiones de AWS para simular el tráfico de usuarios distribuido geográficamente y evaluar el rendimiento global.

Programación de pruebas flexible

Programa las pruebas para que se ejecuten inmediatamente, en una fecha y hora futuras específicas, o en un horario recurrente mediante expresiones cron para las pruebas de regresión automatizadas.

Supervisión en tiempo real

Proporciona una transmisión de datos en vivo opcional para monitorear el progreso de las pruebas con métricas en tiempo real que incluyen los tiempos de respuesta, el recuento de usuarios virtuales y las tasas de éxito de las solicitudes.

Resultados completos de las pruebas

Muestra los resultados detallados de las pruebas con métricas de rendimiento, percentiles (p50, p90, p95, p99), análisis de errores y artefactos descargables para su análisis sin conexión a Internet.

Comparación de puntos de referencia

Designa las pruebas de referencia para comparar el rendimiento a fin de realizar un seguimiento de las mejoras o regresiones a lo largo del tiempo.

Flexibilidad de terminales

Prueba cualquier punto de conexión HTTP o HTTPS en regiones de AWS, entornos locales u otros proveedores de nube.

Consola web intuitiva

Proporciona una consola basada en la web para crear, gestionar y supervisar las pruebas sin necesidad de interactuar con la línea de comandos.

Análisis asistido por IA (opcional)

Se integra con las herramientas de desarrollo de IA a través del servidor Model Context Protocol (MCP) para un análisis inteligente de los datos de las pruebas de carga.

Multiple Protocol Support

Admite varios protocolos, incluidos HTTP, HTTPS, JDBC WebSocket, JMS, FTP y gRPC a través de scripts de prueba personalizados.

Ventajas

La solución ofrece las siguientes ventajas:

Pruebas de rendimiento integrales

Soporta pruebas de carga, stress testing y pruebas de resistencia para evaluar minuciosamente el rendimiento de la aplicación en diversas condiciones.

Detección temprana de problemas

Identifica los cuellos de botella en el rendimiento, las pérdidas de memoria y los problemas de escalabilidad antes de la implementación en producción, lo que reduce el riesgo de interrupciones.

Simulación de uso en el mundo real

Simula con precisión el comportamiento de los usuarios y los patrones de tráfico del mundo real para validar el rendimiento de las aplicaciones en condiciones realistas.

Performance Insights procesables

Proporciona métricas, percentiles y análisis de errores detallados para comprender el comportamiento de las aplicaciones y guiar los esfuerzos de optimización.

Flujos de trabajo de pruebas automati

Permite realizar pruebas programadas y recurrentes para la supervisión continua del rendimiento y las pruebas de regresión sin intervención manual.

Infraestructura rentable

Utiliza contenedores AWS Fargate sin servidor pay-per-use con precios, lo que elimina la necesidad de una infraestructura de pruebas dedicada y de cuotas de suscripción continuas.

Implementación de prueba rápida

Implementa y escala la infraestructura de pruebas en cuestión de minutos sin aprovisionar ni administrar servidores.

Interrogación sencilla de los resultados de las pruebas

Se integra con las herramientas de desarrollo de inteligencia artificial a través de un servidor de protocolo de contexto modelo (MCP) opcional, que permite realizar consultas en lenguaje natural y analizar de forma inteligente los datos de las pruebas de carga para obtener información y solucionar problemas más rápidamente.

Casos de uso

Validación previa a la producción

Pruebe las aplicaciones web y móviles en condiciones de carga similares a las de producción antes de lanzar una nueva versión para validar el rendimiento e identificar los problemas.

Planificación de la capacidad

Determine el número máximo de usuarios simultáneos que su aplicación puede admitir con la infraestructura actual e identifique cuándo es necesario escalarlo.

Verificación de picos de carga

Compruebe que su infraestructura puede gestionar los picos de carga, los picos de tráfico estacionales o los aumentos inesperados de la demanda sin que se degrade el rendimiento.

Optimización del rendimiento

Identifique los cuellos de botella en el rendimiento, como la lentitud de las consultas a las bases de datos, la ejecución ineficiente del código, la latencia de la red o las limitaciones de recursos.

Pruebas de regresión

Programe pruebas de carga recurrentes para detectar las regresiones de rendimiento provocadas por las nuevas implementaciones de código o los cambios en la infraestructura.

Evaluación del rendimiento global

Evalúe el rendimiento de las aplicaciones en varias regiones geográficas para garantizar una experiencia de usuario uniforme para una audiencia global.

Pruebas de carga de API

Pruebe REST APIs, puntos finales de GraphQL o microservicios para validar los tiempos de respuesta, el rendimiento y las tasas de error bajo carga.

Integración de canalizaciones de CI/CD

Integre las pruebas de rendimiento automatizadas en los procesos continuos de integración e implementación para detectar los problemas de rendimiento al principio del ciclo de desarrollo.

Pruebas de servicios de terceros

Pruebe el rendimiento y la fiabilidad de los servicios APIs o servicios de terceros de los que depende su aplicación en distintas condiciones de carga.

Conceptos y definiciones

En esta sección se describen los conceptos clave y se define la terminología específica de esta solución:

escenario

Definición de la prueba, que incluye el nombre de la prueba, la descripción, el recuento de tareas, la simultaneidad, la región de AWS, la aceleración, la espera, el tipo de prueba, la fecha de programación y las configuraciones de recurrencia.

recuento de tareas

Número de contenedores que se lanzarán en el clúster de Fargate para ejecutar el escenario de prueba. No se crearán tareas adicionales una vez que se alcance el límite de recursos de Fargate en la cuenta. Sin embargo, las tareas que ya están en ejecución continuarán.

concurrency

La simultaneidad (número de usuarios virtuales simultáneos por tarea). La simultaneidad recomendada según la configuración predeterminada es 200. La simultaneidad está limitada por la CPU y la memoria. En el caso de las pruebas basadas en Apache JMeter, una mayor simultaneidad aumenta la memoria utilizada por la JVM en la tarea de ECS. La definición de tareas ECS predeterminada crea tareas con 4 GB de memoria. Se recomienda empezar con valores de simultaneidad más bajos para 1 tarea y supervisar las CloudWatch métricas de ECS del clúster de tareas. Consulte las métricas de uso del clúster de Amazon ECS.

aumento

El período de tiempo que aumentará gradualmente desde cero hasta el nivel de simultaneidad objetivo.

mantenga presionado para

El período de tiempo para mantener el nivel de simultaneidad objetivo una vez finalizado el aumento.

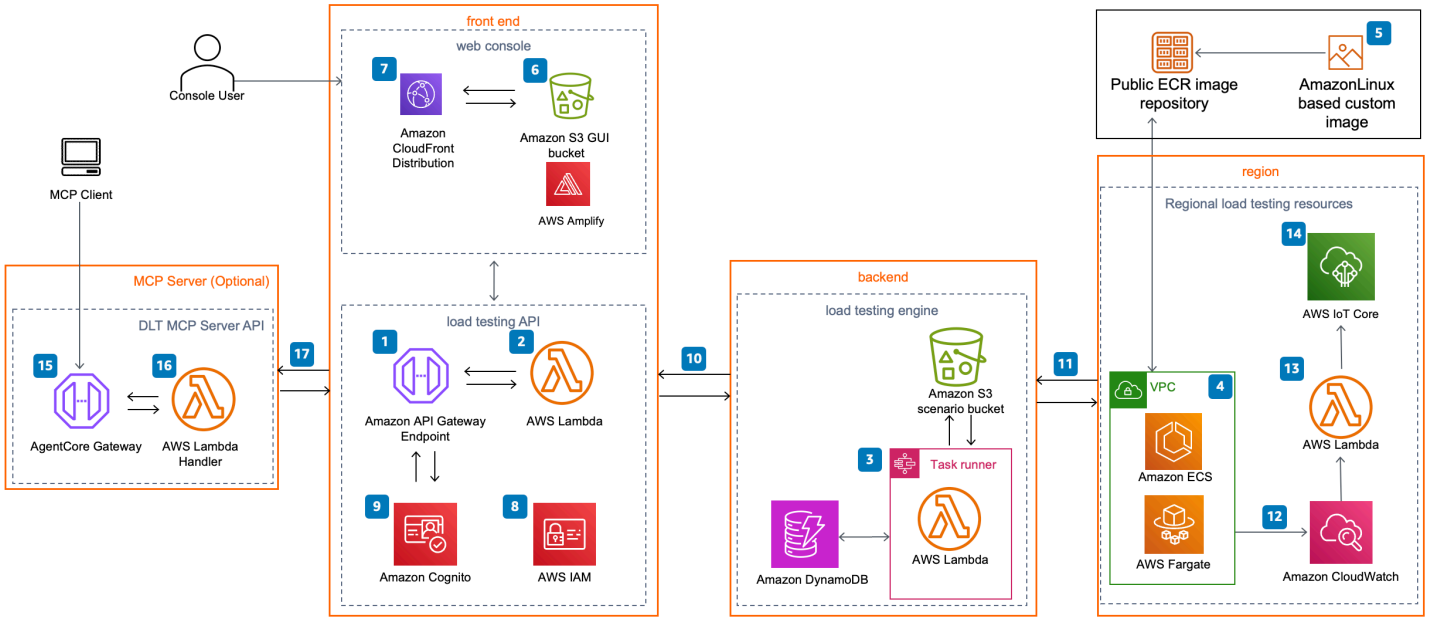
Para ver una referencia general de los términos de AWS, consulte el Glosario de AWS.

Información general de la arquitectura

Diagrama de arquitectura

Al implementar esta solución con los parámetros predeterminados, se implementan los siguientes componentes en su cuenta de AWS.

Pruebas de carga distribuidas en la arquitectura de AWS en AWS



> **Note**
>
> Los CloudFormation recursos de AWS se crean a partir de componentes del AWS Cloud Development Kit (AWS CDK).

El flujo de proceso de alto nivel para los componentes de la solución implementados con la CloudFormation plantilla de AWS es el siguiente:

1. Una API de comprobador de carga distribuida utiliza Amazon API Gateway para invocar los microservicios de la solución (funciones de AWS Lambda).
2. Los microservicios proporcionan la lógica empresarial necesaria para gestionar los datos de las pruebas y ejecutarlas.

3. Estos microservicios interactúan con Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB y AWS Step Functions para almacenar los detalles y los resultados del escenario de las pruebas y organizar la ejecución de las pruebas.
4. Se implementa una topología de red Amazon Virtual Private Cloud (Amazon VPC) que contiene los contenedores Amazon Elastic Container Service (Amazon ECS) de la solución que se ejecutan en AWS Fargate.
5. Los contenedores utilizan una imagen base de Amazon Linux 2023 con el marco de pruebas de carga Taurus instalado. Taurus es un marco de automatización de pruebas de código abierto que admite K6 JMeter, Locust y otras herramientas de prueba. La imagen del contenedor es compatible con Open Container Initiative (OCI) y está alojada por AWS en un repositorio público de Amazon Elastic Container Registry (Amazon ECR). Para obtener más información, consulte Personalización de imágenes de contenedores.
6. Se implementa una consola web con tecnología AWS Amplify en un bucket de S3 configurado para el alojamiento web estático.
7. Amazon CloudFront proporciona un acceso público y seguro al contenido del bucket del sitio web de la solución.
8. Durante la configuración inicial, la solución crea una función de administrador predeterminada (función de IAM) y envía una invitación de acceso a la dirección de correo electrónico de un usuario especificada por el cliente.
9. Un grupo de usuarios de Amazon Cognito administra el acceso de los usuarios a la consola, a la API del comprobador de carga distribuido y al servidor MCP.
10. Tras implementar esta solución, puede utilizar la consola web o APIs crear y ejecutar escenarios de prueba que definan una serie de tareas.
11. Los microservicios utilizan este escenario de prueba para ejecutar tareas de ECS en Fargate en las regiones especificadas.
12. Una vez finalizada la prueba, la solución almacena los resultados en S3 y DynamoDB y los registra en Amazon. CloudWatch
13. Si habilita la opción de datos en tiempo real, la solución envía CloudWatch los registros de las tareas de Fargate a una función Lambda durante la prueba para cada región en la que se ejecute la prueba.
14. La función Lambda publica los datos en el tema correspondiente en AWS IoT Core en la región en la que se implementó la pila principal. La consola web se suscribe al tema y muestra datos en tiempo real mientras se ejecuta la prueba.

> **Note**
>
> Los siguientes pasos describen la integración opcional del servidor MCP para el análisis de las pruebas de carga asistido por IA. Este componente solo se implementa si selecciona la opción de servidor MCP durante la implementación de la solución.

15. Un cliente MCP (herramienta de desarrollo de IA) se conecta al punto final de AWS AgentCore Gateway para acceder a los datos de la solución de pruebas de carga distribuidas a través del protocolo Model Context. AgentCore Gateway valida el token de autenticación de Cognito del usuario para garantizar el acceso autorizado al servidor MCP.
16. Tras la autenticación correcta, AgentCore Gateway reenvía la solicitud de la herramienta MCP a la función Lambda del servidor MCP de DLT. La función Lambda devuelve los datos estructurados a AgentCore Gateway, que los envía de vuelta al cliente MCP para obtener análisis e información asistidos por IA.
17. La función Lambda procesa la solicitud y consulta los recursos de AWS correspondientes (tablas de DynamoDB, buckets de S3 o CloudWatch registros) para recuperar los datos de las pruebas de carga solicitados.

Consideraciones sobre el diseño de AWS Well-Architected

Esta solución utiliza las prácticas recomendadas del AWS Well-Architected Framework, que ayuda a los clientes a diseñar y operar cargas de trabajo confiables, seguras, eficientes y rentables en la nube.

En esta sección se describe cómo los principios de diseño y las prácticas recomendadas de Well-Architected Framework benefician a esta solución.

Excelencia operativa

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del pilar de excelencia operativa.

- Todos los recursos se definen como infraestructura como código mediante CloudFormation plantillas de AWS generadas a partir de construcciones de AWS CDK.

- La solución amplía las métricas CloudWatch en varias etapas para proporcionar observabilidad de las funciones de Lambda, las tareas de ECS, los depósitos de S3 y otros componentes de la solución.

Seguridad

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del pilar de seguridad.

- Cognito autentica y autoriza a los usuarios de la consola web y las solicitudes de API.
- Todas las comunicaciones entre servicios utilizan las funciones de AWS Identity and Access Management (IAM) con el acceso con privilegios mínimos y contienen solo los permisos mínimos necesarios.
- Todo el almacenamiento de datos, incluidos los depósitos de S3 y las tablas de DynamoDB, cifra los datos en reposo mediante claves administradas por AWS.
- El registro, el seguimiento y el control de versiones están habilitados cuando corresponde con fines de auditoría y conformidad.
- El acceso a la red es privado de forma predeterminada y los puntos de enlace de la VPC están habilitados cuando están disponibles para mantener el tráfico dentro de la red de AWS.

> **Note**
>
> La solución crea varios grupos de CloudWatch registros con períodos de retención variables según el volumen de registros y las consideraciones de costo:

Tipo de registro	Periodo de retención
Información sobre los contenedores ECS	1 día
Step Functions, registros personalizados de ECS, registros de acceso a API Gateway	1 año
Registros de tiempo de ejecución de Lambda	2 años
Registros de ejecución de API Gateway	Nunca caducan

Puede modificar estos períodos de retención en la CloudWatch consola en función de sus requisitos.

Fiabilidad

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del pilar de fiabilidad.

- La solución utiliza los servicios sin servidor de AWS siempre que es posible (por ejemplo: Lambda, API Gateway, Amazon S3, AWS Step Functions, Amazon DynamoDB y AWS Fargate) para garantizar una alta disponibilidad y recuperación en caso de fallo del servicio.
- Todo el procesamiento informático utiliza funciones de Lambda o Amazon ECS en AWS Fargate.
- Los datos se almacenan en DynamoDB y Amazon S3, por lo que permanecen en varias zonas de disponibilidad de forma predeterminada.

Eficiencia del rendimiento

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del pilar de eficiencia del rendimiento.

- La solución utiliza una arquitectura sin servidor con la capacidad de escalar horizontalmente según sea necesario.
- La solución se puede lanzar en cualquier región que admita los servicios de AWS de esta solución, como AWS Lambda, Amazon API Gateway, Amazon S3, AWS Step Functions, Amazon DynamoDB, Amazon ECS, AWS Fargate y Amazon Cognito.
- La solución utiliza servicios gestionados en todas partes para reducir la carga operativa que supone el aprovisionamiento y la administración de los recursos.
- La solución se prueba e implementa automáticamente a diario para lograr la coherencia a medida que cambian los servicios de AWS, y los arquitectos de soluciones y los expertos en la materia la revisan para detectar áreas que puedan experimentar y mejorar.

Optimización de costos

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las prácticas recomendadas del pilar de optimización de costos.

- La solución utiliza una arquitectura sin servidor; por lo tanto, a los clientes solo se les cobra por lo que utilizan.
- Amazon DynamoDB escala la capacidad a pedido, de modo que solo paga por la capacidad que utilice.
- AWS ECS en AWS Fargate le permite pagar únicamente por los recursos informáticos que utilice, sin gastos iniciales.
- AWS AgentCore Gateway actúa como un proxy rentable basado en Lambda para la API de pruebas de carga distribuidas, lo que elimina la necesidad de una infraestructura dedicada y reduce los costes gracias a los precios sin servidor. pay-per-request

Sostenibilidad

En esta sección se describe cómo diseñamos esta solución utilizando los principios y las mejores prácticas del pilar de sostenibilidad.

- La solución utiliza servicios gestionados sin servidor para minimizar el impacto medioambiental de los servicios de backend en comparación con los servicios locales que funcionan de forma continua.
- Los servicios sin servidor permiten escalar verticalmente u horizontalmente la solución según sea necesario.

Detalles de la arquitectura

En esta sección se describen los componentes y los servicios de AWS que componen esta solución y los detalles de la arquitectura sobre cómo funcionan juntos estos componentes.

La solución Distributed Load Testing en AWS consta de tres componentes de alto nivel: un front-end, un backend y un servidor MCP opcional.

Interfaz

La interfaz proporciona las interfaces para interactuar con la solución e incluye:

- Una API de pruebas de carga para el acceso programático
- Una consola web para crear, programar y ejecutar pruebas de rendimiento
- Un servidor MCP opcional para el análisis asistido por IA de los resultados y errores de las pruebas

API de pruebas de carga

Las pruebas de carga distribuidas en AWS configuran Amazon API Gateway para alojar la RESTful API de la solución. Los usuarios pueden interactuar con el sistema de pruebas de carga de forma segura a través de la consola web, la RESTful API y el servidor MCP opcional incluidos. La API actúa como una «puerta principal» para acceder a los datos de prueba almacenados en Amazon DynamoDB. También puede utilizarla APIs para acceder a cualquier funcionalidad ampliada que incorpore a la solución.

Esta solución aprovecha las funciones de autenticación de usuarios de los grupos de usuarios de Amazon Cognito. Tras autenticar correctamente a un usuario, Amazon Cognito emite un token web JSON que se utiliza para permitir que la consola envíe solicitudes a los puntos de enlace de la solución (puntos de enlace APIs de Amazon API Gateway). La consola envía las solicitudes HTTPS APIs con el encabezado de autorización que incluye el token.

En función de la solicitud, API Gateway invoca la función de AWS Lambda adecuada para realizar las tareas necesarias con los datos almacenados en las tablas de DynamoDB, almacenar los escenarios de prueba como objetos JSON en Amazon S3, recuperar imágenes de métricas de Amazon y enviar los escenarios de prueba a la máquina de estados de AWS Step Functions. CloudWatch

Para obtener más información sobre la API de la solución, consulte la sección sobre la API de pruebas de carga distribuidas de esta guía.

Consola web

Esta solución incluye una consola web que puede utilizar para configurar y ejecutar pruebas, supervisar las pruebas en ejecución y ver los resultados detallados de las pruebas. La consola es una aplicación ReactJS creada con Cloudscape, un sistema de diseño de código abierto para crear aplicaciones web intuitivas. La consola está alojada en Amazon S3 y se accede a ella a través de Amazon CloudFront. La aplicación utiliza AWS Amplify para integrarse con Amazon Cognito y autenticar a los usuarios. La consola web también incluye una opción para ver los datos en tiempo real de una prueba en ejecución, en la que se suscribe al tema correspondiente de AWS IoT Core.

La URL de la consola web es el nombre del dominio de CloudFront distribución que se encuentra en los CloudFormation resultados como Consola. Tras lanzar la CloudFormation plantilla, también recibirá un correo electrónico con la URL de la consola web y la contraseña de un solo uso para iniciar sesión en ella.

Servidor MCP (opcional)

El servidor opcional Model Context Protocol (MCP) proporciona una interfaz adicional para que las herramientas de desarrollo de IA accedan a los datos de las pruebas de carga y los analicen mediante interacciones en lenguaje natural. Este componente solo se implementa si selecciona la opción de servidor MCP durante la implementación de la solución.

El servidor MCP permite a los agentes de IA consultar los resultados de las pruebas, analizar las métricas de rendimiento y obtener información sobre los datos de las pruebas de carga mediante herramientas como Amazon Q, Claude y otros asistentes de IA compatibles con MCP. Para obtener información detallada sobre la arquitectura y la configuración del servidor MCP, consulte el servidor MCP en esta sección.

Backend

El backend consta de un contenedor, una imagen, una canalización y un motor de pruebas de carga que se utiliza para generar la carga para las pruebas. Interactúas con el backend a través del front-end. Además, las tareas de Amazon ECS en AWS Fargate lanzadas para cada prueba se etiquetan con un identificador (ID) de prueba único. Estas etiquetas de identificación de prueba se pueden usar para ayudarlo a monitorear los costos de esta solución. Para obtener información adicional, consulte

las etiquetas de asignación de costos definidas por el usuario en la Guía del usuario de AWS Billing and Cost Management.

Canalización de imágenes de contenedores

Esta solución utiliza una imagen de contenedor creada con Amazon Linux 2023 como imagen base con el marco de pruebas de carga Taurus instalado. Taurus es un marco de automatización de pruebas de código abierto que admite K6 JMeter, Locust y otras herramientas de prueba. AWS aloja esta imagen en un repositorio público de Amazon Elastic Container Registry (Amazon ECR). La solución usa esta imagen para ejecutar tareas en el clúster de Amazon ECS en AWS Fargate.

Para obtener más información, consulte la sección de personalización de imágenes de contenedores de esta guía.

Infraestructura de pruebas

Además de la CloudFormation plantilla principal, la solución proporciona una plantilla regional para lanzar los recursos necesarios para ejecutar las pruebas en varias regiones. La solución almacena esta plantilla en Amazon S3 y proporciona un enlace a ella en la consola web. Cada pila regional incluye una VPC, un clúster de AWS Fargate y una función Lambda para procesar datos en tiempo real.

Para obtener más información sobre cómo implementar una infraestructura de pruebas en otras regiones, consulte la sección de implementación multirregional de esta guía.

Motor de pruebas de carga

La solución Distributed Load Testing utiliza Amazon Elastic Container Service (Amazon ECS) y AWS Fargate para simular miles de usuarios simultáneos en varias regiones, lo que genera solicitudes HTTP a un ritmo sostenido.

Los parámetros de prueba se definen mediante la consola web incluida. La solución utiliza estos parámetros para generar un escenario de prueba de JSON y lo almacena en Amazon S3. Para obtener más información sobre los scripts de prueba y los parámetros de prueba, consulte los tipos de pruebas en esta sección.

Una máquina de estados de AWS Step Functions ejecuta y supervisa las tareas de Amazon ECS en un clúster de AWS Fargate. La máquina de estados de AWS Step Functions incluye una función

AWS Lambda comprobadora de errores electrónicos, una función AWS Lambda, una función task-status-checker AWS Lambda ejecutora de tareas, una función AWS Lambda canceladora de tareas y una función AWS Lambda analizadora de resultados. Para obtener más información sobre el flujo de trabajo, consulte la sección Probar el flujo de trabajo de esta guía. Para obtener más información sobre los resultados de las pruebas, consulte la sección Resultados de las pruebas de esta guía. Para obtener más información sobre el flujo de trabajo de cancelación de pruebas, consulte la sección Flujo de trabajo de cancelación de pruebas de esta guía.

Si selecciona datos en tiempo real, la solución inicia una función real-time-data-publisher Lambda en cada región mediante CloudWatch los registros que corresponden a las tareas de Fargate en esa región. A continuación, la solución procesa y publica los datos en un tema de AWS IoT Core dentro de la región en la que lanzó la pila principal. Para obtener más información, consulte la sección Datos en tiempo real de esta guía.

Servidor MCP

La integración opcional del servidor Model Context Protocol (MCP) permite a los agentes de IA acceder mediante programación a los datos de las pruebas de carga y analizarlos mediante interacciones en lenguaje natural. Este componente solo se implementa si selecciona la opción de servidor MCP durante la implementación de la solución.

El servidor MCP actúa como un puente entre las herramientas de desarrollo de IA y su implementación de DLT, ya que proporciona una interfaz estandarizada para el análisis inteligente de los resultados de las pruebas de rendimiento. La arquitectura integra varios servicios de AWS para crear una interfaz segura y escalable para las interacciones entre agentes de IA:

AWS AgentCore Gateway

AWS AgentCore Gateway es un servicio totalmente gestionado que proporciona alojamiento estandarizado y administración de protocolos para servidores MCP. En esta solución, AgentCore Gateway actúa como punto de enlace público al que se conectan los agentes de IA cuando solicitan acceso a los datos de las pruebas de carga.

El servicio gestiona todas las comunicaciones del protocolo MCP, incluida la detección de herramientas, la validación de los tokens de autenticación y el enrutamiento de las solicitudes. AgentCore Gateway funciona como un servicio multiusuario con protecciones de seguridad integradas contra las amenazas comunes a los puntos finales públicos, a la vez que valida las firmas y reclamos de los tokens de Cognito para cada solicitud.

Servidor DLT MCP Lambda

La función Lambda del servidor MCP de DLT es un componente sin servidor personalizado que procesa las solicitudes de MCP de los agentes de IA y las traduce en consultas contra sus recursos de DLT.

Esta función Lambda actúa como capa de inteligencia de la integración de MCP, recuperando los resultados de las pruebas de las tablas de DynamoDB, accediendo a los artefactos de rendimiento almacenados en los buckets de S3 y consultando los registros para obtener información de ejecución detallada. CloudWatch La función Lambda implementa patrones de acceso de solo lectura y transforma los datos DLT sin procesar en formatos estructurados y compatibles con la IA que los agentes pueden interpretar y analizar fácilmente.

Integración de autenticación

El sistema de autenticación aprovecha la infraestructura de grupos de usuarios de Cognito existente para mantener controles de acceso uniformes en las interfaces de la consola web y del servidor MCP.

Esta integración utiliza la autenticación OAuth 2.0 basada en tokens. Los usuarios se autentican una vez mediante el proceso de inicio de sesión de Cognito y reciben tokens que funcionan tanto para las interacciones de la interfaz de usuario como para el acceso al servidor MCP. El sistema mantiene los mismos límites de permisos y controles de acceso que la interfaz web, lo que garantiza que los usuarios solo puedan acceder a través de agentes de IA a los mismos datos de pruebas de carga a los que pueden acceder a través de la consola.

Los servicios de AWS en esta solución

Esta solución incluye los siguientes servicios de AWS:

Servicio de AWS	Description (Descripción)
Amazon API Gateway	Principal. Aloja los puntos de enlace de la API REST en la solución.
AWS CloudFormation	Principal. Administra las implementaciones de la infraestructura de la solución.
Amazon CloudFront	Principal. Sirve el contenido web alojado en Amazon S3.

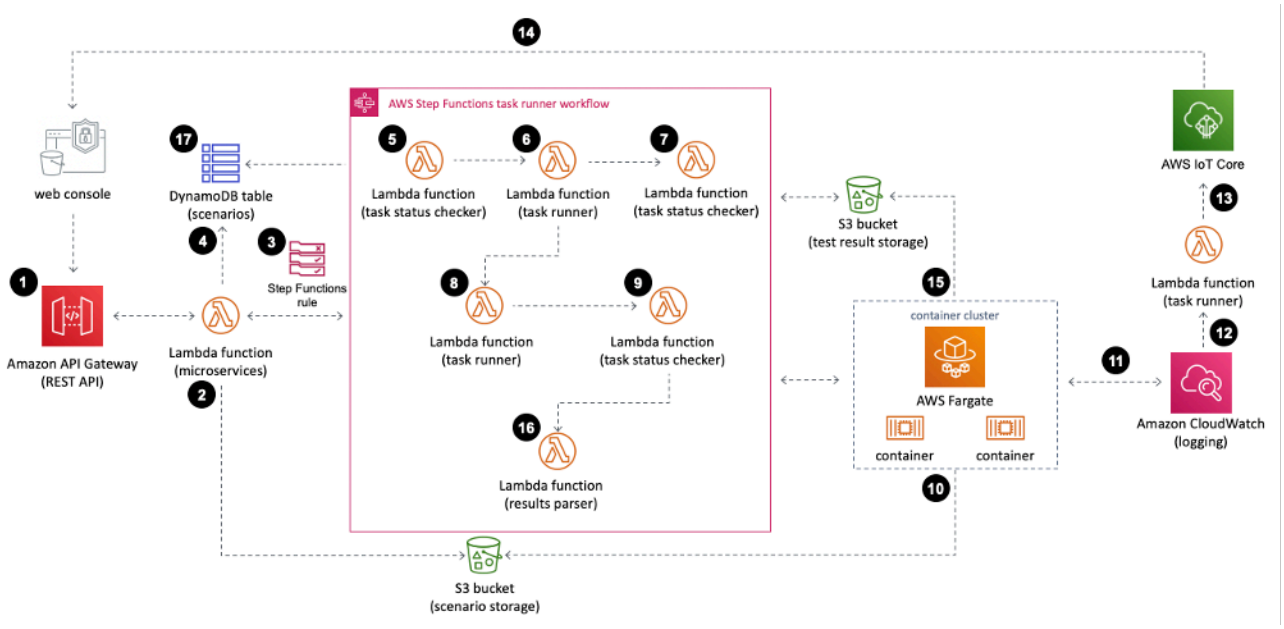
Servicio de AWS	Description (Descripción)
Amazon CloudWatch	Principal. Almacena los registros y las métricas de la solución.
Amazon Cognito	Principal. Gestiona la administración de usuarios y la autenticación de la API.
Amazon DynamoDB	Principal. Almacena la información de implementación y prueba los detalles y los resultados del escenario.
Amazon Elastic Container Service	Principal. Implementa y administra tareas independientes de Amazon ECS en contenedores de AWS Fargate.
AWS Fargate	Principal. Aloja los contenedores Amazon ECS de la solución
AWS Identity and Access Management	Principal. Gestiona las funciones de los usuarios y la gestión de los permisos.
AWS Lambda	Principal. Proporciona la lógica para APIs la implementación, las pruebas, el análisis de los resultados y el inicio de workers/leader las tareas.
AWS Step Functions	Principal. Organiza el aprovisionamiento de contenedores de Amazon ECS en las tareas de AWS Fargate en las regiones especificadas
AWS Amplify	Admite. Proporciona una consola web con tecnología de AWS Amplify.
CloudWatch Eventos de Amazon	Admite. Programa las pruebas para que comiencen automáticamente en una fecha específica o en fechas recurrentes.
Amazon Elastic Container Registry	Admite. Aloja la imagen del contenedor en un repositorio ECR público.
AWS IoT Core	Admite. Permite ver los datos en tiempo real de una prueba en ejecución suscribiéndose al tema correspondiente en AWS IoT Core.
AWS Systems Manager	Admite. Proporciona monitoreo de recursos a nivel de aplicación y visualización de las operaciones de los recursos y los datos de costos.

Servicio de AWS	Description (Descripción)
Amazon S3	Admite. Aloja el contenido web estático, los registros, las métricas y los datos de las pruebas.
Amazon Virtual Private Cloud	Admite. Contiene los contenedores Amazon ECS de la solución que se ejecutan en AWS Fargate.
Amazon Bedrock AgentCore	Soporte, opcional. Alberga el servidor de protocolo de contexto de modelo remoto (MCP) opcional de la solución para la integración de los agentes de IA con la API.

Cómo funcionan las pruebas de carga distribuidas en AWS

En el siguiente desglose detallado se muestran los pasos necesarios para ejecutar un escenario de prueba.

Flujo de trabajo de prueba



1. Utiliza la consola web para enviar un escenario de prueba que incluye los detalles de configuración a la API de la solución.

2. La configuración del escenario de prueba se carga en el Amazon Simple Storage Service (Amazon S3) como un archivo `s3://<bucket-name>/test-scenarios/<$TEST_ID>/<$TEST_ID>.json` JSON ().
3. Una máquina de estados de AWS Step Functions se ejecuta con el ID de prueba, el recuento de tareas, el tipo de prueba y el tipo de archivo como entrada de la máquina de estados de AWS Step Functions. Si la prueba está programada, primero se creará una regla de CloudWatch eventos, que activará AWS Step Functions en la fecha especificada. Para obtener más información sobre el flujo de trabajo de programación, consulte la sección Flujo de trabajo de programación de pruebas de esta guía.
4. Los detalles de configuración se almacenan en la tabla Amazon DynamoDB de escenarios.
5. En el flujo de trabajo del ejecutor de tareas de AWS Step Functions, la función task-status-checker AWS Lambda comprueba si las tareas de Amazon Elastic Container Service (Amazon ECS) ya se están ejecutando para el mismo ID de prueba. Si se encuentran en ejecución tareas con el mismo ID de prueba, se produce un error. Si no hay ninguna tarea de Amazon ECS en ejecución en el clúster de AWS Fargate, la función devuelve el ID de la prueba, el recuento de tareas y el tipo de prueba.
6. La función AWS Lambda ejecutora de tareas obtiene los detalles de la tarea del paso anterior y ejecuta las tareas de trabajo de Amazon ECS en el clúster de AWS Fargate. La API de Amazon ECS utiliza la RunTask acción para ejecutar las tareas de los trabajadores. Estas tareas de trabajo se lanzan y, a continuación, esperan un mensaje de inicio de la tarea principal para comenzar la prueba. La RunTask acción está limitada a 10 tareas por definición. Si el número de tareas es superior a 10, la definición de la tarea se ejecutará varias veces hasta que se hayan iniciado todas las tareas de los trabajadores. La función también genera un prefijo para distinguir la prueba actual en la función AWS Lambda de análisis de resultados.
7. La función task-status-checker AWS Lambda comprueba si todas las tareas de trabajo de Amazon ECS se ejecutan con el mismo ID de prueba. Si las tareas aún se están aprovisionando, espera un minuto y vuelve a comprobarlo. Una vez ejecutadas todas las tareas de Amazon ECS, devuelve el identificador de la prueba, el recuento de tareas, el tipo de prueba, todas las tareas IDs y el prefijo y los pasa a la función de ejecución de tareas.
8. La función AWS Lambda ejecutora de tareas se vuelve a ejecutar y, esta vez, lanza una única tarea de Amazon ECS para que actúe como nodo líder. Esta tarea de ECS envía un mensaje de inicio de la prueba a cada una de las tareas del trabajador para iniciar las pruebas simultáneamente.
9. La función task-status-checker AWS Lambda vuelve a comprobar si las tareas de Amazon ECS se ejecutan con el mismo ID de prueba. Si las tareas siguen ejecutándose, espera un minuto y

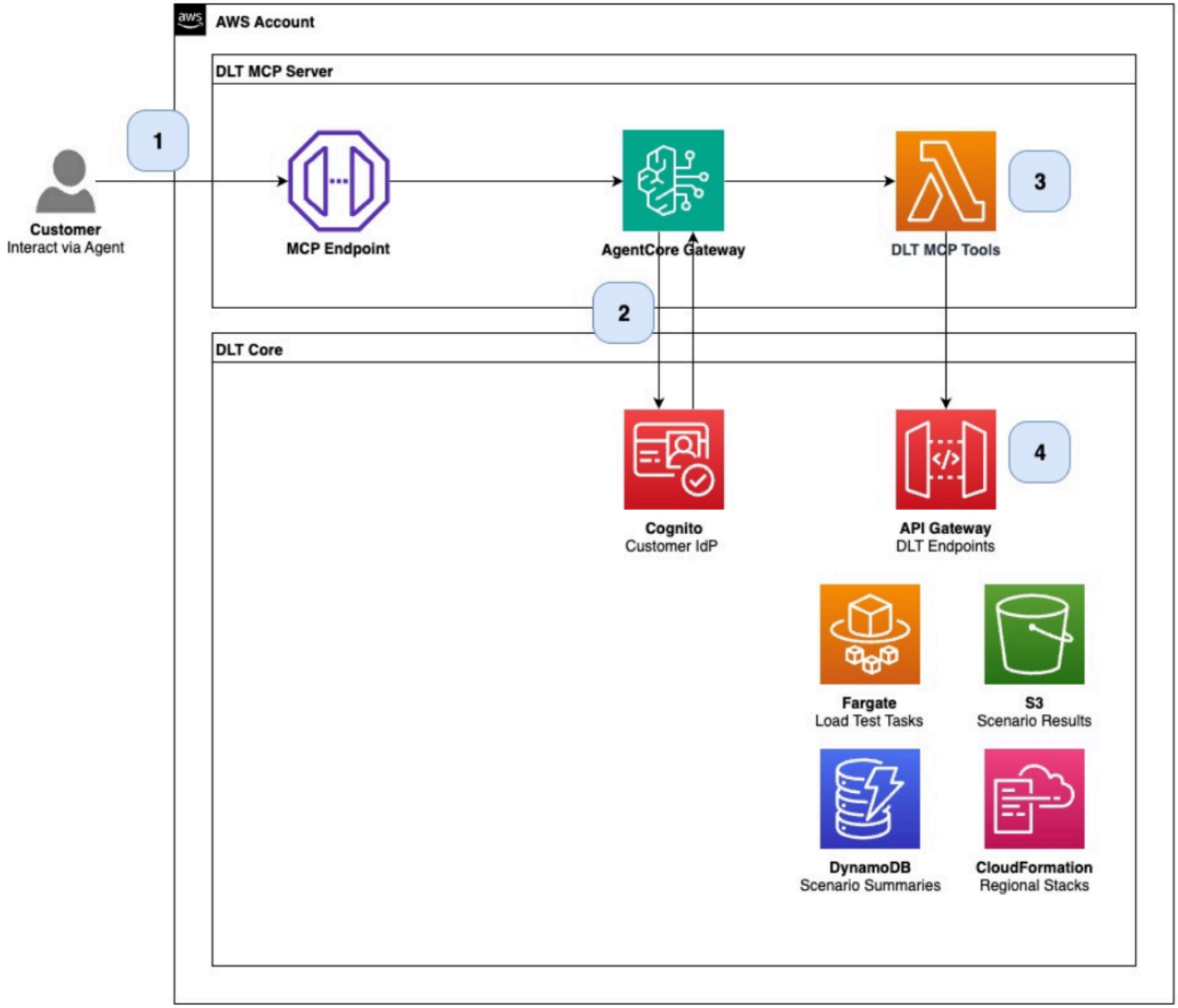
vuelve a comprobarlas. Cuando no hay tareas de Amazon ECS en ejecución, devuelve el ID de la prueba, el recuento de tareas, el tipo de prueba y el prefijo.

10. Cuando la función AWS Lambda ejecutora de tareas ejecuta las tareas de Amazon ECS en el clúster de AWS Fargate, cada tarea descarga la configuración de prueba de Amazon S3 e inicia la prueba.
11. Una vez ejecutadas las pruebas, el tiempo medio de respuesta, el número de usuarios simultáneos, el número de solicitudes satisfactorias y el número de solicitudes fallidas de cada tarea se registran en Amazon CloudWatch y se pueden ver en un CloudWatch panel de control.
12. Si incluiste datos en tiempo real en la prueba, la solución filtra los resultados de las pruebas en tiempo real CloudWatch mediante un filtro de suscripción. A continuación, la solución pasa los datos a una función Lambda.
13. A continuación, la función Lambda estructura los datos recibidos y los publica en un tema de AWS IoT Core.
14. La consola web se suscribe al tema AWS IoT Core de la prueba y recibe los datos publicados en el tema para representar gráficamente los datos en tiempo real mientras se ejecuta la prueba.
15. Una vez finalizada la prueba, las imágenes del contenedor exportan un informe detallado como un archivo XML a Amazon S3. A cada archivo se le asigna un UUID para el nombre del archivo. Por ejemplo, s3://dlte-bucket/test-scenarios/ <$TEST_ID> /results/ <$UUID> .json.
16. Cuando los archivos XML se cargan en Amazon S3, la función AWS Lambda del analizador de resultados lee los resultados de los archivos XML empezando por el prefijo y los analiza y agrega todos los resultados en un resultado resumido.
17. La función AWS Lambda del analizador de resultados escribe el resultado agregado en una tabla de Amazon DynamoDB.

Flujo de trabajo del servidor MCP (opcional)

Si implementa la integración opcional del servidor MCP, los agentes de IA pueden acceder a los datos de las pruebas de carga y analizarlos mediante el siguiente flujo de trabajo:

Arquitectura de servidor MCP



1. Interacción con el cliente: el cliente interactúa con el MCP de DLT a través del punto de conexión MCP alojado en AWS Gateway. AgentCore Los agentes de IA se conectan a este punto final para solicitar acceso a los datos de las pruebas de carga.
2. Autorización: AgentCore Gateway gestiona la autorización contra el cliente de la aplicación del grupo de usuarios de Solution Cognito. La puerta de enlace valida el token de Cognito del usuario para garantizar que tiene permiso para acceder al servidor DLT MCP. Se concede el acceso a los usuarios autorizados y el acceso a las herramientas del agente se limita a las operaciones de solo lectura.

3. Especificación de la herramienta: la AgentCore puerta de enlace se conecta a la función Lambda del servidor MCP de DLT. Una especificación de herramienta define las herramientas disponibles que los agentes de IA pueden utilizar para interactuar con los datos de las pruebas de carga.
4. Acceso a la API de solo lectura: la función Lambda se limita al acceso a la API de solo lectura a través de los puntos finales de DLT API Gateway existentes. La función proporciona cuatro operaciones principales:
   - Listar escenarios: recupera una lista de escenarios de prueba de la tabla de escenarios de DynamoDB
   - Obtenga los resultados de las pruebas de escenarios: acceda a los resultados detallados de las pruebas para escenarios específicos de DynamoDB y S3
   - Obtenga ejecutores de pruebas de carga de Fargate: consulte información sobre la ejecución de tareas de Fargate en el clúster de ECS
   - Obtenga las pilas regionales disponibles: recupere información sobre la infraestructura regional implementada en CloudFormation

La integración del servidor MCP aprovecha la infraestructura DLT existente (API Gateway, Cognito, DynamoDB, S3) para proporcionar un acceso seguro y de solo lectura a los datos de prueba para análisis e información basados en inteligencia artificial.

Consideraciones sobre el diseño

En esta sección, se describen las decisiones de diseño y las opciones de configuración importantes para la solución Distributed Load Testing on AWS, incluidas las aplicaciones compatibles, los tipos de pruebas, las opciones de programación y las consideraciones de implementación.

Aplicaciones compatibles

Esta solución permite probar aplicaciones basadas en la nube y aplicaciones locales siempre que tenga conectividad de red desde su cuenta de AWS a su aplicación. La solución admite APIs el uso de protocolos HTTP o HTTPS.

Tipos de pruebas

Las pruebas de carga distribuidas en AWS admiten varios tipos de pruebas: pruebas de punto final HTTP simples JMeter, K6 y Locust.

Pruebas sencillas de puntos finales HTTP

La consola web proporciona una interfaz de configuración de puntos finales HTTP que permite probar cualquier punto final HTTP o HTTPS sin necesidad de escribir scripts personalizados. Defina la URL del punto final, seleccione el método HTTP (GET, POST, PUT, DELETE, etc.) en un menú desplegable y, si lo desea, añada encabezados de solicitud y cargas útiles de cuerpo personalizados. Esta configuración te permite realizar pruebas APIs con identificadores de autorización personalizados, tipos de contenido o cualquier otro encabezado HTTP y cuerpo de solicitud que requiera tu aplicación.

JMeter pruebas

Al crear un escenario de prueba mediante la consola web, puede cargar un script JMeter de prueba. La solución carga el script en el bucket S3 del escenario. Cuando se ejecutan las tareas de Amazon ECS, descargan el JMeter script de S3 y ejecutan la prueba.

> **Important**
>
> Si bien su JMeter script puede definir la simultaneidad (usuarios virtuales), las tasas de transacción (TPS), los tiempos de aceleración y otros parámetros de carga, la solución anulará estas configuraciones con los valores que especifique en la pantalla Traffic Shape durante la creación de la prueba. La configuración de Traffic Shape controla el recuento de tareas, la simultaneidad (usuarios virtuales por tarea), la duración del aumento y el tiempo de espera de la ejecución de la prueba.

Si tiene archivos JMeter de entrada, puede comprimirlos junto con el script. JMeter Puede elegir el archivo zip al crear un escenario de prueba.

Si quieres incluir complementos, cualquier archivo.jar que esté incluido en un subdirectorio /plugins del archivo zip incluido se copiará en el directorio de JMeter extensiones y estará disponible para las pruebas de carga.

> **Note**
>
> Si incluye archivos JMeter de entrada en el archivo de JMeter script, debe incluir la ruta relativa de los archivos de entrada en el archivo de script. JMeter Además, los archivos de entrada deben estar en la ruta relativa. Por ejemplo, si los archivos de JMeter entrada y el

archivo de script están en su lugar en/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, la prueba fallará porque no podrá encontrar los archivos de entrada.

Si incluye JMeter complementos, los archivos.jar deben estar agrupados en un subdirectorio denominado /plugins dentro de la raíz del archivo zip. En relación con la raíz del archivo zip, la ruta a los archivos jar debe ser. /plugins/bundled_plugin.jar.

Para obtener más información sobre cómo utilizar los scripts, consulte el Manual del usuario. JMeter JMeter

Pruebas K6

La solución admite las pruebas basadas en el marco K6. K6 se publica bajo la licencia AGPL-3.0. La solución muestra un mensaje de confirmación de licencia al crear una nueva prueba K6. Puede cargar el archivo de prueba K6 junto con los archivos de entrada necesarios en un archivo de almacenamiento.

Important

Si bien su script K6 puede definir la simultaneidad (usuarios virtuales), las etapas, los umbrales y otros parámetros de carga, la solución anulará estas configuraciones con los valores que especifique en la pantalla Traffic Shape durante la creación de la prueba. La configuración de Traffic Shape controla el recuento de tareas, la simultaneidad (usuarios virtuales por tarea), la duración de la aceleración y la duración de la espera durante la ejecución de la prueba.

Pruebas de langostas

La solución es compatible con las pruebas basadas en el marco Locust. Puede cargar el archivo de prueba de Locust junto con los archivos de entrada necesarios en un archivo de almacenamiento.

Important

Si bien su script de Locust puede definir la simultaneidad (número de usuarios), la velocidad de aparición y otros parámetros de carga, la solución anulará estas configuraciones con los

valores que especifique en la pantalla Traffic Shape durante la creación de la prueba. La configuración de Traffic Shape controla el recuento de tareas, la simultaneidad (usuarios virtuales por tarea), la duración de la aceleración y la duración de la espera durante la ejecución de la prueba.

Programar pruebas

La solución ofrece tres opciones de temporización de ejecución para ejecutar pruebas de carga:

- Ejecutar ahora: ejecuta la prueba de carga inmediatamente después de la creación
- Ejecutar una vez: ejecute la prueba en una fecha y hora específicas en el futuro
- Ejecute según un cronograma: cree pruebas recurrentes utilizando expresiones cron para definir el cronograma

Al seleccionar Ejecutar una vez, se especifica el tiempo de ejecución en formato de 24 horas y la fecha de ejecución en la que debe empezar a ejecutarse la prueba de carga.

Al seleccionar Ejecutar según una programación, puede introducir manualmente una expresión cron o seleccionar uno de los patrones cron más comunes (por ejemplo, cada hora, todos los días a una hora específica, los días de la semana o todos los meses). La expresión cron utiliza un formato de programación detallado con campos para los minutos, las horas, el día del mes, el mes, el día de la semana y el año. También debe especificar una fecha de caducidad, que defina cuándo debe dejar de ejecutarse la prueba programada. Para obtener más información sobre cómo funciona la programación, consulte la sección Flujo de trabajo de programación de pruebas de esta guía.

> **Note**
>
> - Duración de la prueba: tenga en cuenta la duración total de las pruebas al programarlas. Por ejemplo, una prueba con un tiempo de preparación de 10 minutos y un tiempo de espera de 40 minutos tardará aproximadamente 80 minutos en completarse.
> - Intervalo mínimo: asegúrese de que el intervalo entre las pruebas programadas sea superior a la duración estimada de la prueba. Por ejemplo, si la prueba dura unos 80 minutos, prográmela para que no se ejecute con más frecuencia que cada 3 horas.
> - Limitación horaria: el sistema no permite programar las pruebas con una diferencia de solo una hora, incluso si la duración estimada de la prueba es inferior a una hora.

Pruebas simultáneas

Esta solución crea un CloudWatch panel de Amazon para cada prueba que muestra el resultado combinado de todas las tareas que se ejecutan en el clúster de Amazon ECS en tiempo real. El CloudWatch panel muestra el tiempo medio de respuesta, el número de usuarios simultáneos, el número de solicitudes satisfactorias y el número de solicitudes fallidas. La solución agrega cada métrica por segundo y actualiza el panel cada minuto.

Administración de usuarios

Durante la configuración inicial, debe proporcionar un nombre de usuario y una dirección de correo electrónico que Amazon Cognito utiliza para concederle acceso a la consola web de la solución. La consola no proporciona administración de usuarios. Para añadir usuarios adicionales, debe utilizar la consola de Amazon Cognito. Para obtener más información, consulte Administración de usuarios en grupos de usuarios en la Guía para desarrolladores de Amazon Cognito.

Para migrar los usuarios existentes a los grupos de usuarios de Amazon Cognito, consulte el blog de AWS Approaches para migrar usuarios a los grupos de usuarios de Amazon Cognito.

Implementación regional

Esta solución utiliza Amazon Cognito, que solo está disponible en regiones específicas de AWS. Por lo tanto, debe implementar esta solución en una región en la que Amazon Cognito esté disponible. Para obtener la disponibilidad de servicios más reciente por región, consulte la lista de servicios regionales de AWS.

Planificación de la implementación

En esta sección se describen los costes, la seguridad, las regiones compatibles, las cuotas y otras consideraciones que debe tener en cuenta antes de implementar la solución.

Costo

Usted es responsable del coste de los servicios de AWS utilizados durante la ejecución de esta solución. El costo total depende del número de pruebas de carga ejecutadas, su duración y la cantidad de datos generados. A partir de esta revisión, el costo estimado de ejecutar esta solución con la configuración predeterminada en la región EE.UU. Este (Virginia del Norte) es de aproximadamente 30,90$ al mes.

La siguiente tabla proporciona un ejemplo de desglose de los costes de la implementación de esta solución con los parámetros predeterminados en la región EE.UU. Este (Virginia del Norte) durante un mes.

Servicio de AWS	Dimensiones	Coste [USD]
AWS Fargate	10 tareas bajo demanda (con dos v CPUs y 4 GB de memoria) ejecutándose durante 30 horas	29,62$
Amazon DynamoDB	1000 unidades de capacidad de escritura bajo demanda 1000 unidades de capacidad de lectura bajo demanda	0,0015$
AWS Lambda	1000 solicitudes 10 minutos de duración total	1,25$
AWS Step Functions	1000 transiciones estatales	0,025 USD
Total:		30,90$ al mes

Los recursos de la solución se etiquetan con Key= y Value=SolutionId . SO0062 Puede activar la clave de etiqueta SolutionId siguiendo la documentación sobre activating-tags. Una vez activada la etiqueta, puede crear una regla de categoría de costes siguiendo la documentación para crear categorías de costes. Puede ver el costo incurrido por la solución supervisando la consola de categorías de costos y seleccionando el nombre de la categoría de costos.

Recomendamos crear un presupuesto a través de AWS Cost Explorer para ayudar a administrar los costos. Los precios están sujetos a cambios. Para obtener más información, consulte la página web de precios de cada servicio de AWS utilizado en esta solución.

> **Note**
>
> La configuración de tareas predeterminada utiliza 2 v CPUs y 4 GB de memoria por tarea. Si sus pruebas de carga no requieren estos recursos, puede reducirlos para reducir los costes. Por el contrario, puedes aumentar los recursos para permitir una mayor simultaneidad por tarea. Para obtener más información, consulta la sección Aumentar los recursos de contenedores de esta guía.

> **Note**
>
> Esta solución ofrece la opción de incluir datos en tiempo real al ejecutar una prueba. Esta función requiere una función adicional de AWS Lambda y un tema sobre AWS IoT Core que conllevan costes adicionales.

Costos adicionales del servidor MCP (opcional)

La siguiente tabla proporciona un desglose de los costos de la integración del servidor MCP con los precios en la región de EE. UU. del Este (Virginia del Norte) durante un mes.

Componente de servicio	Dimensiones	Coste [USD]
AgentCore Gateway: herramienta de indexación	10 herramientas × 0,02$ por cada 100 herramientas	0,002$
AgentCore Gateway: API de búsqueda	10 000 interacciones × 0,025$ por cada 1000	0,25$

Componente de servicio	Dimensiones	Coste [USD]
AgentCore Gateway: invocaciones a la API	50 000 invocaciones × 0,005$ por 1000	0,25$
AWS Lambda Función	Variable en función del uso (cargas de trabajo típicas)	5,00$ - 20,00$
Coste adicional total estimado:		De 5,50 a 20,50 dólares al mes

Los precios están sujetos a cambios. Para obtener más información sobre los precios de AgentCore Gateway, consulte los precios de Amazon Bedrock (sección AgentCore Gateway). Para conocer los precios de Lambda, consulte los precios de AWS Lambda.

Seguridad

Cuando crea sistemas en la infraestructura de AWS, las responsabilidades de seguridad se comparten entre usted y AWS. Este modelo de responsabilidad compartida reduce su carga operativa, ya que AWS opera, administra y controla los componentes, desde el sistema operativo host y la capa de virtualización hasta la seguridad física en las instalaciones en las que operan los servicios. Para obtener más información sobre la seguridad de AWS, visite Seguridad en la nube de AWS.

Roles de IAM

Las funciones de AWS Identity and Access Management (IAM) permiten a los clientes asignar políticas y permisos de acceso detallados a los servicios y usuarios de la nube de AWS. Esta solución crea funciones de IAM que otorgan acceso a las funciones de AWS Lambda de la solución para crear recursos regionales.

Amazon CloudFront

Esta solución implementa una interfaz de usuario web alojada en un bucket de Amazon S3, distribuido por Amazon CloudFront. Para ayudar a reducir la latencia y mejorar la seguridad, esta solución incluye una CloudFront distribución con una identidad de acceso de origen, es decir, un CloudFront usuario que proporciona acceso público al contenido del bucket del sitio web de la

solución. De forma predeterminada, la CloudFront distribución usa TLS 1.2 para aplicar el nivel más alto de protocolo de seguridad. Para obtener más información, consulte Restringir el acceso a un origen de Amazon S3 en la Guía para CloudFront desarrolladores de Amazon.

CloudFront activa medidas de seguridad adicionales para añadir encabezados de seguridad HTTP a cada respuesta del espectador. Para obtener más información, consulta Añadir o eliminar encabezados HTTP en las respuestas. CloudFront

Esta solución usa el CloudFront certificado predeterminado, que tiene un protocolo de seguridad mínimo admitido de TLS v1.0. Para imponer el uso de TLS v1.2 o TLS v1.3, debe usar un certificado SSL personalizado en lugar del certificado predeterminado. CloudFront Para obtener más información, consulte Cómo configuro mi CloudFront distribución para usar un certificado. SSL/TLS

Amazon API Gateway

Esta solución implementa puntos de enlace de Amazon API Gateway optimizados RESTful APIs para proporcionar la funcionalidad de pruebas de carga mediante el punto de enlace de API Gateway predeterminado en lugar de un dominio personalizado. Para la optimización perimetral APIs mediante el punto final predeterminado, API Gateway utiliza la política de seguridad TLS-1-0. Para obtener más información, consulte Cómo trabajar con REST APIs en la Guía para desarrolladores de Amazon API Gateway.

Esta solución usa el certificado API Gateway predeterminado, que tiene un protocolo de seguridad mínimo admitido de TLS v1.0. Para imponer el uso de TLS v1.2 o TLS v1.3, debes usar un dominio personalizado con un certificado SSL personalizado en lugar del certificado API Gateway predeterminado. Para obtener más información, consulta Cómo configurar nombres de dominio personalizados para REST. APIs

Grupo de seguridad AWS Fargate

De forma predeterminada, esta solución abre al público la regla de salida del grupo de seguridad de AWS Fargate. Si quiere impedir que AWS Fargate envíe tráfico a todas partes, cambie la regla de salida por un enrutamiento entre dominios sin clase (CIDR) específico.

Este grupo de seguridad también incluye una regla de entrada que permite el tráfico local en el puerto 50.000 a cualquier fuente que pertenezca al mismo grupo de seguridad. Esto se usa para permitir que los contenedores se comuniquen entre sí.

Amazon VPC

VPC: una nube privada virtual (VPC) basada en el servicio Amazon VPC le proporciona una red privada y aislada de forma lógica en la nube de AWS.

Puede especificar su propia VPC en los CloudFormation parámetros de AWS durante la implementación. La VPC la utilizan exclusivamente las tareas de ECS que generan carga; la consola web y la API no se implementan en esta VPC. Si no especifica una VPC existente, la solución creará una nueva VPC con la configuración de red requerida. Si decide utilizar una VPC existente, debe cumplir los siguientes requisitos para ejecutar correctamente las tareas de pruebas de carga.

Requisitos de la VPC

A continuación, se indican los requisitos mínimos para que una VPC se utilice con las pruebas de carga distribuidas en AWS.

- La VPC debe contener al menos dos AZs
- La VPC debe contener al menos dos subredes, cada una en una zona de disponibilidad independiente
- Las subredes de VPC pueden ser públicas o privadas, pero deben usar la misma configuración (tanto pública como privada)
- La VPC debe proporcionar acceso a los puntos finales para ECR, CloudWatch Logs, S3 e IoT Core.
- La VPC debe proporcionar acceso a los servicios a los que se dirigen las pruebas de carga.

> **Note**
>
> Si no tiene una VPC que cumpla estos criterios, puede crear una VPC rápidamente con el asistente de VPC. Para obtener más información, consulte Creación de una VPC.

Las subredes públicas pueden cumplir estos requisitos al incluir lo siguiente:

- Una puerta de enlace a Internet conectada a la VPC
- Una ruta a la puerta de enlace a Internet (0.0.0.0/0)

Las subredes privadas pueden cumplir estos requisitos mediante el uso de puertas de enlace NAT o puntos finales de VPC, como se describe a continuación.

Opción 1: puerta de enlace NAT

- Implemente una puerta de enlace NAT en cada zona de disponibilidad con subredes privadas
- Configure las tablas de enrutamiento para enrutar el tráfico con destino a Internet (0.0.0.0/0) a través de la puerta de enlace NAT

Opción 2: puntos finales de VPC

Cree los siguientes puntos de enlace de VPC en su VPC:

- Punto final de la API Amazon ECR: `com.amazonaws.<region>.ecr.api`
- Punto de conexión DKR de Amazon ECR: `com.amazonaws.<region>.ecr.dkr`
- Punto final CloudWatch de Amazon Logs: `com.amazonaws.<region>.logs`
- Punto final de Amazon S3 Gateway: `com.amazonaws.<region>.s3`
- Punto final AWS IoT Core (obligatorio si se utilizan los gráficos de datos en tiempo real) `com.amazonaws.<region>.iot.data`

Es posible que también funcionen otras configuraciones de VPC.

> **Important**
>
> El grupo de seguridad adjunto a cada interfaz de punto final de la VPC debe permitir el tráfico TCP entrante en el puerto 443 desde el grupo de seguridad de tareas de ECS.

Configuración del grupo de seguridad

Durante la implementación, la solución creará un grupo de seguridad dentro de la VPC para permitir el siguiente tráfico con las tareas del clúster de ECS:

- Todo el tráfico saliente
- El tráfico entrante en el puerto 50000 proviene de otras tareas del mismo grupo de seguridad, para facilitar la coordinación entre las tareas del trabajador y del líder.

Prueba de stress de red

Usted es responsable de usar esta solución según la política de pruebas de esfuerzo de red. Esta política cubre situaciones como cuando planea ejecutar pruebas de red de gran volumen directamente desde sus instancias de Amazon EC2 a otras ubicaciones, como otras instancias de Amazon EC2, propiedades o servicios de AWS o puntos de enlace externos. Estas pruebas a veces se denominan pruebas de stress, pruebas de carga o pruebas diurnas. La mayoría de las pruebas realizadas con clientes no están sujetas a esta política; sin embargo, consulte esta política si cree que generará tráfico que se mantendrá, en total, durante más de 1 minuto, a más de 1 Gbps (mil millones de bits por segundo) o más de 1 Gbps (mil millones de paquetes por segundo).

Restringir el acceso a la interfaz de usuario pública

Para restringir el acceso a la interfaz de usuario pública más allá de los mecanismos de autenticación y autorización proporcionados por IAM y Amazon Cognito, utilice la solución de automatización de seguridad AWS WAF (firewall de aplicaciones web).

Esta solución implementa automáticamente un conjunto de reglas de AWS WAF que filtran los ataques habituales basados en la web. Los usuarios pueden seleccionar entre las funciones de protección preconfiguradas que definen las reglas incluidas en una lista de control de acceso web (ACL web) de AWS WAF.

Seguridad del servidor MCP (opcional)

Si implementa la integración opcional del servidor MCP, la solución utiliza AWS AgentCore Gateway para proporcionar un acceso seguro a los datos de las pruebas de carga para los agentes de IA. AgentCore Gateway valida los tokens de autenticación de Amazon Cognito para cada solicitud, lo que garantiza que solo los usuarios autorizados puedan acceder al servidor MCP. La función Lambda del servidor MCP implementa patrones de acceso de solo lectura, lo que impide que los agentes de IA modifiquen las configuraciones o los resultados de las pruebas. Todas las interacciones del servidor MCP utilizan los mismos límites de permisos y controles de acceso que la consola web.

Regiones de AWS admitidas

Esta solución utiliza el servicio Amazon Cognito, que actualmente no está disponible en todas las regiones de AWS. Para obtener la disponibilidad más reciente de los servicios de AWS por región, consulte la lista de servicios regionales de AWS.

Las pruebas de carga distribuidas en AWS están disponibles en las siguientes regiones de AWS:

Nombre de la región	
Este de EE. UU. (Ohio)	Asia-Pacífico (Tokio)
Este de EE. UU. (Norte de Virginia)	Canadá (centro)
EE.UU. Oeste (Norte de California)	Europa (Fráncfort)
Oeste de EE. UU. (Oregón)	Europa (Irlanda)
Asia-Pacífico (Mumbai)	Europa (Londres)
Asia-Pacífico (Seúl)	Europa (París)
Asia-Pacífico (Singapur)	Europa (Estocolmo)
Asia-Pacífico (Sídney)	América del Sur (São Paulo)

Regiones de AWS compatibles con el servidor MCP (opcional)

Si planea implementar la integración opcional del servidor MCP, debe implementar la solución en una región de AWS en la que esté disponible AWS AgentCore Gateway. La función de servidor MCP solo está disponible en las siguientes regiones de AWS:

Nombre de región	Código de región
Este de EE. UU. (Norte de Virginia)	us-east-1
Oeste de EE. UU. (Oregón)	us-west-2
Asia-Pacífico (Singapur)	ap-southeast-1
Asia-Pacífico (Sídney)	ap-southeast-2
Asia-Pacífico (Tokio)	ap-northeast-1
Europa (Fráncfort)	eu-central-1

Nombre de región	Código de región
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (París)	eu-west-3

Para obtener la disponibilidad más reciente de AWS AgentCore Gateway por región, consulte los puntos de enlace y las cuotas de AWS AgentCore Gateway en la Guía para desarrolladores de AWS AgentCore Gateway.

Cuotas

Las cuotas de servicio (que también se denominan límites) establecen el número máximo de recursos u operaciones de servicio para su cuenta de AWS.

Cuotas para servicios de AWS en esta solución

Asegúrese de tener una cuota suficiente para cada uno de los servicios implementados en esta solución. Para obtener más información, consulte Service Quotas de AWS.

Use los enlaces siguientes para ir a la página de ese servicio. Para ver las cuotas de servicio para todos los servicios de AWS en la documentación sin cambiar de página, consulte la información de la página de Cuotas y puntos de conexión del servicio del PDF.

CloudFormation Cuotas de AWS

Su cuenta de AWS tiene CloudFormation cuotas de AWS que debe tener en cuenta al lanzar la pila de esta solución. Si comprende estas cuotas, puede evitar errores de limitación que le impidan implementar esta solución correctamente. Para obtener más información, consulte CloudFormation las cuotas de AWS en la Guía del CloudFormation usuario de AWS.

Cuotas de pruebas de carga

El número máximo de tareas que se pueden ejecutar en Amazon ECS mediante el tipo de lanzamiento de AWS Fargate se basa en el tamaño de la vCPU de las tareas. El tamaño predeterminado de la tarea en las pruebas de carga distribuidas en AWS es de 2 vCPU. Para ver las

cuotas predeterminadas actuales, consulte las cuotas de servicio de Amazon ECS. Las cuotas de las cuentas corrientes pueden diferir de las cuotas enumeradas. Para comprobar las cuotas específicas de una cuenta, compruebe la cuota de servicio para el recuento de recursos de vCPU bajo demanda de Fargate en la consola de administración de AWS. Para obtener instrucciones sobre cómo solicitar un aumento, consulte las cuotas de servicio de AWS en la Guía de referencia general de AWS.

La imagen del contenedor de Amazon Linux 2023 (con Taurus instalado) no limita las conexiones simultáneas por tarea, pero eso no significa que pueda admitir un número ilimitado de usuarios. Para determinar el número de usuarios simultáneos que los contenedores pueden generar para una prueba, consulte la sección Determine el número de usuarios de esta guía.

> **Note**
>
> El límite recomendado para los usuarios simultáneos según la configuración predeterminada es de 200 usuarios.

Pruebas simultáneas

Esta solución crea un CloudWatch panel de Amazon para cada prueba que muestra el resultado combinado de todas las tareas que se ejecutan en el clúster de Amazon ECS en tiempo real. El CloudWatch panel muestra el tiempo medio de respuesta, el número de usuarios simultáneos, el número de solicitudes satisfactorias y el número de solicitudes fallidas. La solución agrega cada métrica por segundo y actualiza el panel cada minuto.

Política de pruebas de Amazon EC2

No necesita la aprobación de AWS para ejecutar pruebas de carga con esta solución siempre y cuando el tráfico de red se mantenga por debajo de 1 Gbps. Si la prueba generará más de 1 Gbps, póngase en contacto con AWS. Para obtener más información, consulte la Política de pruebas de Amazon EC2.

Política de pruebas CloudFront de carga de Amazon

Si tienes pensado realizar pruebas de carga en un CloudFront punto final, consulta las directrices de pruebas de carga de la Guía para CloudFront desarrolladores de Amazon. También recomendamos distribuir el tráfico entre varias tareas y regiones. Proporcione al menos 30 minutos de tiempo de aceleración para la prueba de carga. Para las pruebas de carga que envíen más de 500 000

solicitudes por segundo o que requieran datos de más de 300 Gbps, recomendamos obtener primero una aprobación previa para enviar el tráfico. CloudFront puede limitar el tráfico no aprobado en las pruebas de carga, lo que afecta a la disponibilidad del servicio. CloudFront

Supervisión de la solución después de la implementación

Tras implementar la solución, recomendamos monitorizar continuamente los recursos de la solución mediante CloudWatch alarmas y métricas de Amazon.

Configurar CloudWatch alarmas

Puede configurar CloudWatch alarmas para monitorear las métricas clave y recibir notificaciones cuando se superen los umbrales. Considere la posibilidad de configurar alarmas para los siguientes recursos:

Métricas CloudFront de distribución de Amazon

Supervisa el rendimiento y los errores de la CloudFront distribución. Para obtener más información, consulta las estadísticas CloudFront de distribución en la Guía para CloudFront desarrolladores de Amazon.

Métricas de Amazon API Gateway

Supervise las tasas de solicitudes de API, la latencia y los errores. Para obtener más información, consulte las dimensiones y métricas de Amazon API Gateway en la Guía para desarrolladores de Amazon API Gateway.

Métricas de funciones de AWS Lambda

Supervise las invocaciones, la duración, los errores y las limitaciones de las funciones Lambda para los microservicios de la solución.

Métricas de Amazon ECS y AWS Fargate

Supervise el uso de la CPU y la memoria de las tareas durante las pruebas de carga para garantizar que los recursos sean adecuados.

Métricas de Amazon DynamoDB

Supervise el consumo de capacidad de lectura y escritura, las solicitudes limitadas y la latencia.

Contrate a un experto

Contratos a corto plazo de AWS Countdown Premium para pruebas de carga distribuida en AWS

Nuestros ingenieros de AWS ofrecen orientación experta sobre los aspectos fundamentales de las pruebas de rendimiento, el desarrollo de scripts y el análisis de resultados. Inscríbase ahora.

Información general

Los contratos a corto plazo de AWS Countdown Premium (CDP) proporcionan orientación experta a las organizaciones que realizan pruebas de rendimiento a escala. A través de un modelo colaborativo do-it-yourself «», los ingenieros de AWS ofrecen supervisión estratégica y experiencia técnica, mientras que su equipo mantiene la responsabilidad de ejecución. Los ingenieros expertos de AWS están disponibles en el plazo de una semana a partir de la inscripción, sin necesidad de contratos a largo plazo.

Modelo de servicio

Los ingenieros de CDP trabajan junto con su equipo para brindarle orientación y supervisión durante la implementación de las pruebas de rendimiento. Este enfoque no intervencionista garantiza que reciba orientación experta a la vez que desarrolla sus capacidades internas. El servicio es ideal para las organizaciones con capacidades de prueba existentes que necesitan experiencia especializada en AWS para implementar las pruebas de carga distribuidas en AWS de manera efectiva.

¿Qué ofrecen los ingenieros de CDP

Los ingenieros de CDP lo guían a través de los aspectos básicos de las pruebas de rendimiento y las pruebas de carga distribuida en la arquitectura de AWS. Proporcionan orientación sobre la JMeter estructura de los scripts de K6 y Locust y el desarrollo de los scripts de prueba, ayudan a implementar las CloudFormation plantillas y evalúan los resultados de las pruebas con recomendaciones de optimización del rendimiento. Support incluye el análisis de la utilización de los recursos, la alineación de las mejores prácticas y la end-to-end orientación desde la configuración inicial hasta el análisis de los resultados, lo que permite la transferencia de conocimientos a su equipo.

Responsabilidades del cliente

Su equipo se encarga de las configuraciones a nivel de aplicación, del desarrollo de los scripts de prueba y de la verificación de los escenarios de prueba. Usted es responsable de la ejecución y

las operaciones reales de las pruebas, incluidas todas las actividades de prueba antes, durante y después de las pruebas de rendimiento.

Ventajas principales

Los compromisos a corto plazo de CDP reducen el riesgo gracias a la supervisión de expertos, la orientación contextual específica para su carga de trabajo, las recomendaciones de optimización del rendimiento, la resolución más rápida de los problemas, la alineación de las mejores prácticas y un apoyo integral, al tiempo que mantienen la propiedad y el desarrollo de las capacidades de su equipo.

Arquitecturas compatibles

Las pruebas de carga distribuidas en AWS permiten probar aplicaciones web APIs, microservicios y arquitecturas sin servidor a escala, aprovechando la solución de pruebas de carga distribuidas en AWS. Las capacidades de prueba van mucho más allá de estos casos de uso habituales e incluyen bases de datos, TCP/UDP protocolos, directorios LDAP, servidores de correo SMTP y muchos otros sistemas y protocolos que requieren la validación del rendimiento bajo carga.

Introducción

Las organizaciones interesadas en contrataciones a corto plazo de CDP para las pruebas de carga distribuidas en AWS pueden inscribirse directamente a través del sitio web de AWS aquí y seleccionar «Implementación de casos de uso» como área de interés.

Fuera de alcance

CDP no proporciona el desarrollo de guiones de prueba personalizados (solo con orientación), ni gestiona las operaciones de ejecución de las pruebas ni crea laboratorios o talleres prácticos personalizados. El soporte in situ también está fuera del alcance.

Implementación de la solución

El método de implementación recomendado para esta solución es AWS Launch Wizard. Proporciona:

- Una experiencia de configuración guiada con paneles de ayuda detallados en cada paso
- Una página centralizada para supervisar el estado de todas sus implementaciones
- Indica si hay una versión más reciente de la solución disponible para su implementación o actualización

Como alternativa, puede implementar la solución directamente mediante una CloudFormation plantilla de AWS.

Información general del proceso de implementación

Antes de implementar la solución, revise el costo, la arquitectura, la seguridad y otras consideraciones analizadas anteriormente en esta guía.

Tiempo de implementación: aproximadamente 15 minutos para la pila principal, más 5 minutos para cada región adicional

> **Note**
>
> Esta solución incluye métricas de recopilación de datos para AWS. Utilizamos estos datos para comprender mejor cómo utilizan los clientes esta solución, así como los servicios y productos relacionados. Los datos recopilados a través de esta encuesta son propiedad de AWS. La recopilación de datos está sujeta al Aviso de privacidad de AWS.

> **Note**
>
> Usted es responsable del coste de los servicios de AWS utilizados durante la ejecución de esta solución. Para obtener más información, visite la sección de costos de esta guía y consulte la página web de precios de cada servicio de AWS utilizado en esta solución.

Implemente con AWS Launch Wizard

Esta solución incluye un proceso de implementación guiado mediante AWS Launch Wizard. Siga estos pasos para implementar las pruebas de carga distribuidas en AWS en su cuenta.

1. Inicie sesión en la consola de administración de AWS y seleccione el botón de abajo para iniciar el proceso de implementación.

   Launch solution

2. Si hay más de un patrón de implementación disponible para la solución, seleccione el que mejor se adapte a su caso de uso.
3. Seleccione la versión que desee implementar. Se recomienda utilizar la versión más reciente.
4. Haga clic en el botón Iniciar el asistente de despliegue.

A continuación, deberá seguir una serie de pasos para recopilar la información necesaria para implementar la solución. El aprovisionamiento de los recursos necesarios tardará aproximadamente 15 minutos.

Seleccione su despliegue en la lista de despliegues para ver su estado.

Implemente con AWS CloudFormation

Esta solución utiliza CloudFormation plantillas y pilas de AWS para automatizar su implementación. Las CloudFormation plantillas especifican los recursos de AWS incluidos en esta solución y sus propiedades. La CloudFormation pila aprovisiona los recursos que se describen en las plantillas.

CloudFormation Plantilla de AWS

Puede descargar la CloudFormation plantilla de esta solución antes de implementarla. Esta solución utiliza AWS CloudFormation para automatizar la implementación de las pruebas de carga distribuidas en AWS. Incluye la siguiente CloudFormation plantilla de AWS, que puede descargar antes de la implementación:

View template

distribu

load-testing-on-aws.template: utilice esta plantilla para lanzar la solución y todos los componentes

asociados. La configuración predeterminada implementa los servicios principales y de soporte que se encuentran en los servicios de AWS en esta sección de soluciones, pero puede personalizar la plantilla para que se adapte a sus necesidades específicas.

> **Note**
>
> Los CloudFormation recursos de AWS se crean a partir de componentes del AWS Cloud Development Kit (AWS CDK). Si ya implementó esta solución anteriormente, consulte Actualizar la solución para obtener instrucciones de actualización.

Lanzar la pila

Siga estos pasos para implementar la solución Distributed Load Testing on AWS en su cuenta. Esta CloudFormation plantilla de AWS automatizada implementa las pruebas de carga distribuidas en AWS.

1. Inicie sesión en la consola de administración de AWS y seleccione el botón para lanzar la CloudFormation plantilla.

   Como alternativa, puede descargar la plantilla como punto de partida para su propia implementación.
2. La plantilla se activa en la región Este de EE. UU. (Norte de Virginia) de forma predeterminada. Para lanzar esta solución en una región de AWS diferente, utilice el selector de regiones de la barra de navegación de la consola.

   > **Note**
   >
   > Esta solución utiliza Amazon Cognito, que actualmente solo está disponible en regiones específicas de AWS. Por lo tanto, debe lanzar esta solución en una región de AWS en la que Amazon Cognito esté disponible. Para obtener la disponibilidad de servicios más reciente por región, consulte la lista de servicios regionales de AWS.
3. En la página Crear pila, verifique que la URL de la plantilla correcta aparezca en el cuadro de texto URL de Amazon S3 y seleccione Siguiente.

4. En la página Especificar los detalles de la pila, especifique un nombre para la pila.
5. En Parámetros, revise los parámetros de la plantilla y modifíquelos según sea necesario. Esta solución utiliza los siguientes valores predeterminados.

Parámetro	Predeterminado	Description (Descripción)
Nombre del administrador	<Requiere que se introduzcan datos>	Nombre de usuario del administrador de la solución inicial.
Correo electrónico del administrador	*<Requires input>*	Dirección de correo electrónico del usuario administrador. Tras el lanzamiento, se enviará un correo electrónico a esta dirección con las instrucciones de inicio de sesión en la consola.
ID de VPC existente	<Optional input>	Si tiene una VPC que quiere usar y ya está creada, introduzca el ID de una VPC existente en la misma región en la que se implementó la pila. Por ejemplo, vpc-1a2b3c4d5e6f.
Primera subred existente	<Optional input>	El ID de la primera subred de la VPC existente. Esta subred necesita una ruta a Internet para obtener la imagen del contenedor para ejecutar las pruebas. Por ejemplo, la subnet-7h8i9j0k.
Segunda subred existente	<Optional input>	El ID de la segunda subred de la VPC existente. Esta subred necesita una ruta

Parámetro	Predeterminado	Description (Descripción)
		a Internet para obtener la imagen del contenedor para ejecutar las pruebas. Por ejemplo, subnet-1x2y3z.
Proporcione un bloque CIDR válido para que la solución cree la VPC	192.168.0.0/16	Puede dejar este parámetro en blanco si utiliza una VPC existente
Proporcione un bloque CIDR válido para la subred A para que la solución cree la VPC	192.168.0.0/20	Bloque CIDR para la subred A de la VPC de AWS Fargate
Proporcione un bloque CIDR válido para la subred B para que la solución cree la VPC	192.168.16.0/20	Bloque CIDR para la subred B de la VPC de AWS Fargate
Proporcione un bloque CIDR para permitir el tráfico saliente de las tareas de Fargate	0.0.0.0/0	Bloque CIDR que restringe el acceso saliente a los contenedores Amazon ECS.
Actualización automática de la imagen del contenedor	No	Utilice automáticamente la imagen más actualizada y segura hasta la próxima versión secundaria. Al seleccionarNo, se mostrará la imagen tal y como se publicó originalmente, sin ninguna actualización de seguridad.

Parámetro	Predeterminado	Description (Descripción)
Implemente un servidor MCP opcional	No	Implemente el servidor MCP remoto opcional mediante AgentCore Gateway para conectar las aplicaciones de IA a las pruebas de carga distribuidas en AWS.

6. Elija Siguiente.
7. En la página Configurar opciones de pila, elija Siguiente.
8. En la página Revisar, revise y confirme la configuración. Seleccione la casilla para reconocer que la plantilla creará recursos de AWS Identity and Access Management (AWS IAM).
9. Elija Create stack (Crear pila) para implementar la pila.

Puede ver el estado de la pila en la CloudFormation consola de AWS en la columna Estado. Debería recibir el estado CREATE_COMPLETE en aproximadamente 15 minutos.

> **Note**
>
> Además de la función principal de AWS Lambda, esta solución incluye la función Lambda de recursos personalizados, que se ejecuta únicamente durante la configuración inicial o cuando se actualizan o eliminan los recursos.

Al ejecutar esta solución, la función Lambda de recursos personalizados está inactiva. Sin embargo, no elimine esta función, ya que es necesaria para administrar los recursos asociados.

Despliegue multirregional

Tiempo de implementación: aproximadamente 5 minutos por región

Puede realizar pruebas en varias regiones.

Al implementar la solución de pruebas de carga distribuidas, se crea una CloudFormation plantilla regional en el bucket de S3 de los escenarios. La URL de esta plantilla aparece en los CloudFormation resultados de la pila principal con la clave «RegionalCFTemplate».

Para realizar una prueba multirregional, debe implementar la CloudFormation plantilla regional en cada región en la que desee realizar la prueba.

> **Note**
>
> Cada cuenta de AWS solo puede usar una pila regional por región. Además, la pila regional no se puede utilizar en la misma región que la pila principal.

Puede instalar la plantilla regional de la siguiente manera:

1. En la consola web de la solución, vaya al panel de control en el menú de la izquierda.
2. Utilice el icono del portapapeles para copiar el enlace CloudFormation de la plantilla en Amazon S3.
3. Inicie sesión en la CloudFormation consola de AWS y seleccione la región correcta.
4. En la página Crear pila, verifique que la URL de la plantilla correcta aparezca en el cuadro de texto URL de Amazon S3 y seleccione Siguiente.
5. En la página Especificar los detalles de la pila, especifique un nombre para la pila.
6. En Parámetros, revise los parámetros de la plantilla y modifíquelos según sea necesario. Esta solución utiliza los siguientes valores predeterminados.

Parámetro	Predeterminado	Description (Descripción)
ID de VPC existente	<Optional input>	Si tiene una VPC que quiere usar y ya está creada, introduzca el ID de una VPC existente en la misma región en la que se implementó la pila. Por ejemplo, vpc-1a2b3 c4d5e6f.
Primera subred existente	<Optional input>	El ID de la primera subred de la VPC existente. Esta subred necesita una ruta a Internet para obtener la imagen del contenedor para ejecutar

Parámetro	Predeterminado	Description (Descripción)
		las pruebas. Por ejemplo, la subnet-7h8i9j0k.
Segunda subred existente	<Optional input>	El ID de la segunda subred de la VPC existente. Esta subred necesita una ruta a Internet para obtener la imagen del contenedor para ejecutar las pruebas. Por ejemplo, subnet-1x2y3z.
Proporcione un bloque CIDR válido para que la solución cree la VPC	192.168.0.0/16	Si no proporciona valores para una VPC existente, el bloque CIDR de la Amazon VPC creada por la solución contiene la dirección IP de AWS Fargate.
Proporcione un bloque CIDR para permitir el tráfico saliente de las tareas de Fargate	0.0.0.0/0	Bloque CIDR que restringe el acceso saliente a los contenedores Amazon ECS.

7. Elija Siguiente.
8. En la página Configurar opciones de pila, elija Siguiente.
9. En la página Revisar, revise y confirme la configuración. Asegúrese de marcar la casilla para confirmar que la plantilla creará recursos de AWS Identity and Access Management (IAM).
10. Elija Create stack (Crear pila) para implementar la pila.

    Puede ver el estado de la pila en la CloudFormation consola de AWS en la columna Estado. Debería recibir el estado CREATE_COMPLETE en aproximadamente cinco minutos.

Cuando las regiones se hayan implementado correctamente, aparecerán en la consola web. Al crear una prueba, todas las regiones disponibles aparecen en el panel de control y en la sección Creación de escenarios. Puede añadir una región a una prueba en el paso Traffic Shape de la creación del escenario.

La solución crea un elemento de DynamoDB para cada región implementada en la tabla de escenarios, que contiene la información necesaria sobre los recursos de prueba en esa región. Puede ordenar los resultados de las pruebas en la consola web por región. Para ver los resultados agregados de todas las regiones en una prueba multirregional, usa CloudWatch las métricas de Amazon. Encontrarás el código fuente del gráfico en los resultados de la prueba una vez finalizada la prueba.

> **Note**
>
> Puede lanzar la pila regional sin la consola web. Obtenga un enlace a la plantilla regional en el paquete de escenarios de Amazon S3 e indíquelo como fuente al lanzar la pila regional en la región requerida. Como alternativa, puede descargar la plantilla y subirla como fuente para la región que desee.

Actualización de la solución

Al actualizar la solución, se aplican las funciones, los parches de seguridad y las correcciones de errores más recientes a la implementación. Para actualizar a la versión más reciente, consulte la sección correspondiente según su método de implementación original: AWS Launch Wizard o AWS CloudFormation.

> **Important**
>
> Antes de realizar la actualización, asegúrese de que no se estén realizando pruebas de carga en este momento. El proceso de actualización puede interrumpir temporalmente la disponibilidad de la solución.

Actualización mediante AWS Launch Wizard

La consola muestra automáticamente la versión más reciente disponible de la solución en el menú desplegable de versiones de implementación. Si ya implementó la solución anteriormente, siga este procedimiento para actualizar la implementación a la versión más reciente.

1. Vaya a Launch Wizard Deployments.
2. Seleccione la implementación que desee actualizar.
3. Seleccione Acciones y, a continuación, Actualizar la versión de despliegue.
4. Seleccione la versión más reciente de entre las versiones de despliegue disponibles.
5. Revise la configuración.
6. Realice los cambios necesarios en cada paso.
7. Confirme la actualización.

Actualización mediante AWS CloudFormation

Si ya implementó la solución anteriormente, siga este procedimiento para actualizar la CloudFormation pila a la versión más reciente.

1. Inicie sesión en la CloudFormation consola, seleccione la CloudFormation pila existente y seleccione Actualizar pila.

2. Seleccione Realizar una actualización directa.
3. Seleccione Sustituir plantilla existente.
4. En Especificar plantilla:
   a. Seleccione URL de Amazon S3.
   b. Copie el enlace de la plantilla más reciente.
   c. Pegue el enlace en el cuadro URL de Amazon S3.
   d. Compruebe que la URL de la plantilla correcta aparezca en el cuadro de texto URL de Amazon S3.
   e. Elija Siguiente.
   f. Vuelva a seleccionar Siguiente.
5. En Parámetros, revise los parámetros de la plantilla y modifíquelos según sea necesario. Consulte Lanzar la pila para obtener más información sobre los parámetros.
6. Elija Siguiente.
7. En la página Configurar opciones de pila, elija Siguiente.
8. En la página Revisar, revise y confirme la configuración.
9. Seleccione la casilla para aceptar que la plantilla puede crear recursos de IAM.
10. Seleccione Ver conjunto de cambios y verifique los cambios.
11. Seleccione Crear pila para implementar la pila.

Puede ver el estado de la pila en la CloudFormation consola de AWS en la columna Estado. Debería recibir un `UPDATE_COMPLETE` estado en aproximadamente 15 minutos.

> **Note**
>
> Si tiene problemas de autenticación de Amazon Cognito al iniciar sesión desde su navegador después de actualizar la pila, actualice su navegador (Ctrl+Shift+R activado Windows/Linux o Cmd+Shift+R en Mac) para borrar los datos en caché e inténtelo de nuevo.

Solución de problemas con las actualizaciones de versiones anteriores a la v3.3.0

> **Note**
>
> Esta sección se aplica únicamente a las actualizaciones de versiones anteriores a la v3.3.0. Si va a actualizar desde la versión 3.3.0 o posterior, siga el procedimiento de actualización estándar mediante AWS Launch Wizard o AWS. CloudFormation

1. Descargue la plantilla -aws.templatedistributed-load-testing-on.
2. Abre la plantilla y navega hasta ella `Conditions:` y busca. `DLTCommonResourcesAppRegistryCondition`
3. Debería ver algo similar a lo siguiente:

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

4. Cambie el segundo `true` valor `afalse`:

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

5. Utilice la plantilla personalizada para actualizar su pila siguiendo los pasos que se indican en Actualizar mediante AWS CloudFormation.
6. Esta actualización elimina de la pila los recursos relacionados con el registro de aplicaciones, lo que permite que la actualización se complete correctamente.
7. Realiza otra actualización de la pila utilizando la URL de plantilla más reciente.

Actualización de pilas regionales

Si ha implementado la solución en varias regiones, debe actualizar cada pila regional por separado. Siga el procedimiento de actualización estándar para cada CloudFormation paquete regional en las regiones en las que haya implementado la infraestructura de pruebas.

Administrador de aplicaciones de AWS Systems Manager

Tras actualizar la solución, AWS Systems Manager Application Manager proporciona una vista a nivel de aplicación de la solución y sus recursos. Puede usar Application Manager para:

- Supervise los recursos, los costos de los recursos implementados en todas las pilas y cuentas de AWS y los registros desde una ubicación central.
- Vea los datos operativos de los recursos de la solución en el contexto de una aplicación, como el estado de la implementación, CloudWatch las alarmas, las configuraciones de los recursos y los problemas operativos.

Resolución de problemas

La resolución de problemas conocidos proporciona instrucciones para mitigar los errores conocidos. Si estas instrucciones no resuelven el problema, Contact AWS Support proporciona instrucciones para abrir un caso de AWS Support para esta solución.

Resolución de problemas conocidos

Problema: está utilizando una VPC existente y sus pruebas fallan con el estado Fallido, lo que genera el siguiente mensaje de error:

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

- Solución:

Asegúrese de que las subredes existan en la VPC especificada y de que tengan una ruta a Internet con una puerta de enlace de Internet o una puerta de enlace NAT. AWS Fargate necesita acceso para extraer la imagen del contenedor del repositorio público y poder ejecutar las pruebas correctamente.

Problema: las pruebas tardan demasiado en ejecutarse o se quedan indefinidamente ejecutándose

- Solución:

Cancele la prueba y compruebe AWS Fargate para asegurarse de que se hayan detenido todas las tareas. Si no se han detenido, detenga manualmente todas las tareas de Fargate. Comprueba los límites de tareas de Fargate bajo demanda en tu cuenta para asegurarte de que puedes lanzar la cantidad de tareas que desees. También puede consultar los CloudWatch registros de la función de ejecución de tareas de Lambda para obtener más información sobre los errores al lanzar las tareas de Fargate. Consulte los registros de CloudWatch ECS para obtener detalles sobre lo que sucede en los contenedores Fargate que están en funcionamiento.

Problema: las pruebas comienzan pero no se completan o se desconoce el estado de las tareas del ECS

- Solución:

Si seleccionó la opción de proporcionar una VPC existente en la cuenta en la que se implementó la solución, asegúrese de que la VPC que utilizan las tareas de ECS tenga suficientes direcciones IP libres para iniciar la cantidad de tareas proporcionada en la entrada de prueba. La definición de tareas de ECS utiliza la imagen de ECR que necesita una puerta de enlace a Internet o una ruta a Internet para que el servicio de ECS pueda aprovisionar las tareas descargando la imagen de ECR de la solución desde aws-solutions/ -. distributed-load-testing-on aws-load-tester Si no puede proporcionar una ruta a Internet porque todas las subredes de la VPC son privadas, puede alojar la imagen de ECR en su cuenta mediante la memoria caché de extracción de ECR. Actualice la definición de la tarea con el nuevo URI de la imagen ECR y cree una nueva revisión. Una vez actualizada la definición de la tarea, es necesario actualizar la configuración de la solución en la tabla de DynamoDB para poder utilizar la nueva revisión. El nombre de la tabla de DynamoDB se encuentra en la pestaña de resultados de CloudFormation la pila, debajo de la clave. ScenariosTable Actualice el atributo taskDefinition del elemento con la clave TestID y el valor region- [SOLUTION-DEPLOYED-REGION].

Problema: Las pruebas deben utilizar un punto final que sea privado o que no esté disponible a través de la pasarela de Internet

- Solución:

Al probar puntos finales de API privados a los que no se puede acceder a través de la puerta de enlace de Internet, tenga en cuenta los siguientes enfoques:

1. Configuración de red: asegúrese de que las tablas de rutas de subred utilizadas por las tareas de ECS estén actualizadas con una ruta al rango de direcciones IP del punto final privado que se está probando. Esto permite que el tráfico de prueba llegue al punto final privado de su VPC.
2. Resolución de DNS: en el caso de los dominios personalizados, configure los ajustes de DNS de la VPC para resolver el nombre de dominio del punto final privado. Consulte la documentación de DNS de VPC para obtener instrucciones detalladas.
3. Puntos de enlace de VPC: si está probando servicios de AWS, considere usar puntos de enlace de VPC (PrivateLinkAWS) para establecer una conectividad privada. Por ejemplo, para probar una API Gateway privada, puedes crear un punto de conexión de VPC para API Gateway. Consulte la documentación de Private API Gateway.
4. Emparejamiento de VPC: si el punto de enlace privado está en una VPC diferente, establezca el emparejamiento de VPC entre la VPC en la que se implementa la solución y la VPC que contiene el punto de enlace privado. Configure las tablas de enrutamiento adecuadas en ambas. VPCs Consulte la documentación de emparejamiento de VPC.

5. Transit Gateway: para escenarios de redes más complejos que involucren múltiples conexiones VPCs, considere usar AWS Transit Gateway para enrutar el tráfico entre la VPC de la solución y la VPC que contiene el punto final privado. Consulte la documentación de Transit Gateway.
6. Grupos de seguridad: asegúrese de que los grupos de seguridad asociados a sus tareas de ECS permitan el tráfico saliente al punto final privado y que los grupos de seguridad del punto final privado permitan el tráfico entrante desde las tareas de ECS.

Para probar instancias EC2 o balanceadores de carga de aplicaciones internos, asegúrese de que los rangos CIDR de la VPC no se superpongan y de que las rutas necesarias estén configuradas en las tablas de enrutamiento.

Problema: las pruebas se están completando, pero los resultados no están disponibles en la interfaz de usuario

- Solución:

Si la prueba se ha completado pero los resultados no están disponibles en la interfaz de usuario, los archivos de resultados deberían seguir estando disponibles en el bucket de S3 desde las tareas de ECS en las que se ejecutaron las pruebas. Esta es una limitación conocida de la solución. En la arquitectura actual, la solución utiliza una función Lambda de análisis de resultados para resumir los resultados de varias tareas de ECS, que luego se almacenan como un elemento en la tabla de DynamoDB. La tabla de DynamoDB tiene un límite de 400 KB de tamaño máximo de elemento. Esta limitación se alcanza en función de la complejidad del script de prueba, la simultaneidad y el número de tareas que se utilicen. El error no significa que la prueba esté fallando, sino que el proceso para resumir los resultados y almacenarlos en la tabla de DynamoDB para las operaciones CRUD ha fallado. Los resultados siguen disponibles en el depósito de S3 para el escenario de prueba.

Póngase en contacto con AWS Support.

Si tiene AWS Business Support+, AWS Enterprise Support o Unified Operations, puede utilizar AWS Support Center para obtener asistencia de expertos con esta solución. En las siguientes secciones, encontrará instrucciones.

Crear caso

1. Inicie sesión y vaya al Centro de soporte.
2. Seleccione Crear caso.

¿Cómo podemos ayudarle?

1. Elija una opción técnica
2. En Servicio, seleccione Soluciones.
3. Para la categoría, seleccione Pruebas de carga distribuidas en AWS.
4. En Gravedad, seleccione la opción que mejor se adapte a su caso de uso.
5. Al especificar los valores de Servicio, Categoría y Gravedad, la interfaz rellena los enlaces a las preguntas más frecuentes de solución de problemas. Si no puede resolver sus dudas con estos enlaces, elija Siguiente paso: información adicional.

Información adicional

1. En Asunto, introduzca un texto que resuma su pregunta o problema.
2. Para obtener una descripción, describa el problema en detalle, incluido el nombre de este producto y la versión que utiliza, como en este ejemplo: Distributed Load Testing on AWS Vx.y.z.
3. Elija Adjuntar archivos.
4. Adjunte la información que AWS Support necesita para procesar la solicitud.

Ayúdenos a resolver su caso más rápido

1. Especifique la información requerida.
2. Elija Siguiente paso: Resuelva ahora o póngase en contacto con nosotros.

Resuelva ahora o póngase en contacto con nosotros

1. Revise las soluciones de Resolver ahora.
2. Si estas no le ayudan a resolver su problema, elija Contactar con nosotros, especifique la información solicitada y seleccione Enviar.

Desinstalar la solución

Puede desinstalar la solución Distributed Load Testing on AWS desde la consola de administración de AWS o mediante la interfaz de línea de comandos de AWS. Debe eliminar manualmente la consola, el escenario y el registro de los buckets de Amazon Simple Storage Service (Amazon S3) creados por esta solución. Las implementaciones de las soluciones de AWS no las eliminan automáticamente en caso de que tenga datos que conservar.

> **Note**
>
> Si ha implementado pilas regionales, debe eliminar las pilas de esas regiones antes de eliminar la pila principal.

Uso de Consola de administración de AWS

AWS CloudFormation

1. Inicie sesión en la CloudFormation consola de AWS.
2. En la página Pilas, seleccione la pila de instalación de esta solución.
3. Elija Eliminar.

AWS Launch Wizard

1. Inicie sesión en la consola AWS Launch Wizard.
2. En la página Implementaciones de Launch Wizard, seleccione la implementación de esta solución.
3. Elija Acciones y, a continuación, elija Eliminar.
4. Confirme la eliminación.

Uso de la Interfaz de la línea de comandos de AWS

Determine si la Interfaz de la línea de comandos de AWS (AWS CLI) está disponible en su entorno. Para obtener instrucciones de instalación, consulte ¿Qué es la Interfaz de la línea de comandos de AWS? de la Guía del usuario de AWS CLI. Tras confirmar que la CLI de AWS está disponible, ejecute el siguiente comando:

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

Eliminar los buckets de Amazon S3

Esta solución está configurada para conservar los buckets de Amazon S3 creados por la solución (para implementarlos en una región opcional) si decide eliminar la CloudFormation pila de AWS para evitar la pérdida accidental de datos. Tras desinstalar la solución, puede eliminar manualmente este depósito de S3 si no necesita conservar los datos. Siga estos pasos para eliminar el bucket de Amazon S3.

1. Inicie sesión en la consola de Amazon S3.
2. En el panel de navegación izquierdo, elija Buckets.
3. En el campo Buscar buckets por nombre, introduce el nombre de la pila de esta solución.
4. Seleccione uno de los depósitos S3 de la solución y elija Vacío.
5. Introduzca eliminar permanentemente en el campo de verificación y seleccione Vacío.
6. Selecciona el depósito de S3 que acabas de vaciar y selecciona Eliminar.
7. Introduce el nombre del depósito de S3 en el campo de verificación y selecciona Eliminar depósito.

Repita los pasos 4 a 7 hasta que elimine todos los depósitos de S3.

Para eliminar el bucket de S3 mediante la AWS CLI, ejecute el siguiente comando:

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

Uso de la solución

En esta sección se proporciona una guía completa sobre el uso de la solución Distributed Load Testing en AWS, desde la creación del primer escenario de prueba hasta el análisis de los resultados detallados. El flujo de trabajo incluye la creación de un escenario de prueba, la ejecución de una prueba y la exploración de los resultados de la prueba.

Cree un escenario de prueba

La creación de un escenario de prueba implica cuatro pasos principales: configurar los ajustes generales, definir el escenario, configurar los patrones de tráfico y revisar la configuración.

Paso 1: Configuración general

Configure los parámetros básicos de la prueba de carga, incluidos el nombre de la prueba, la descripción y las opciones de configuración generales.

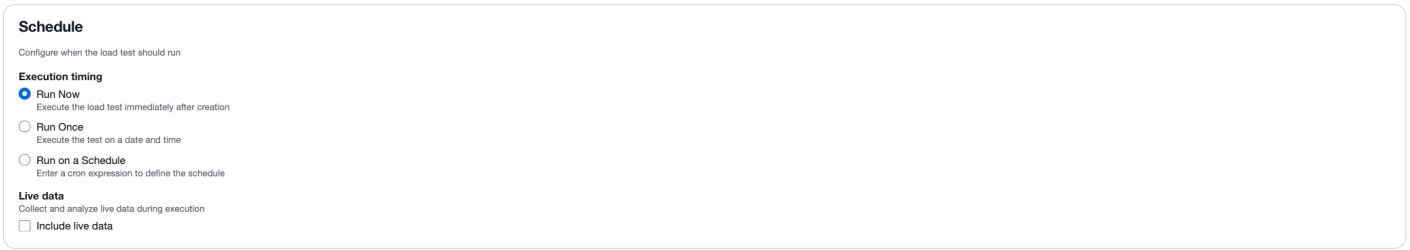
Identificación de la prueba

- Nombre de la prueba (obligatorio): un nombre descriptivo para el escenario de la prueba
- Descripción de la prueba (obligatoria): detalles adicionales sobre el propósito y la configuración de la prueba
- Etiquetas (opcional): añada hasta 5 etiquetas para clasificar y organizar los escenarios de prueba

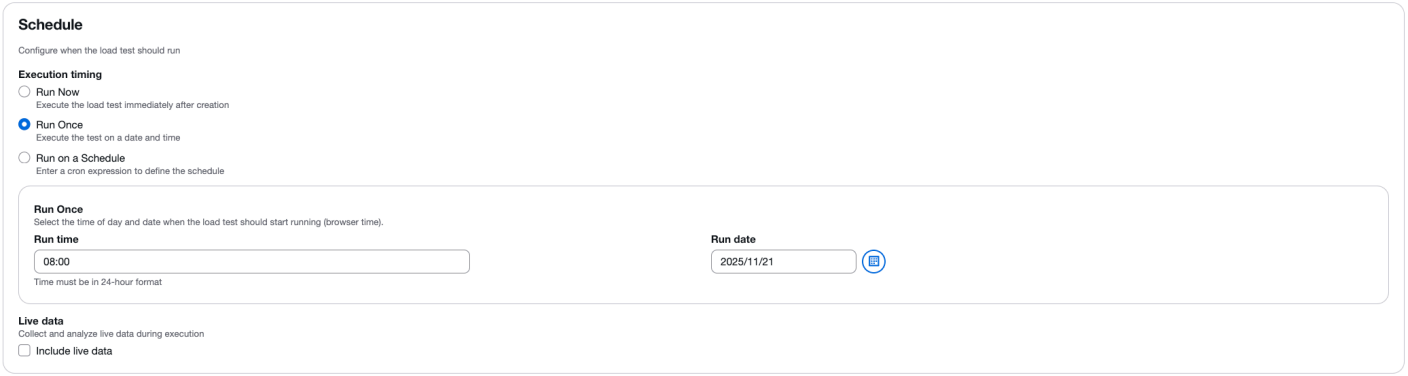
Opciones de programación

Configure cuándo debe ejecutarse la prueba:

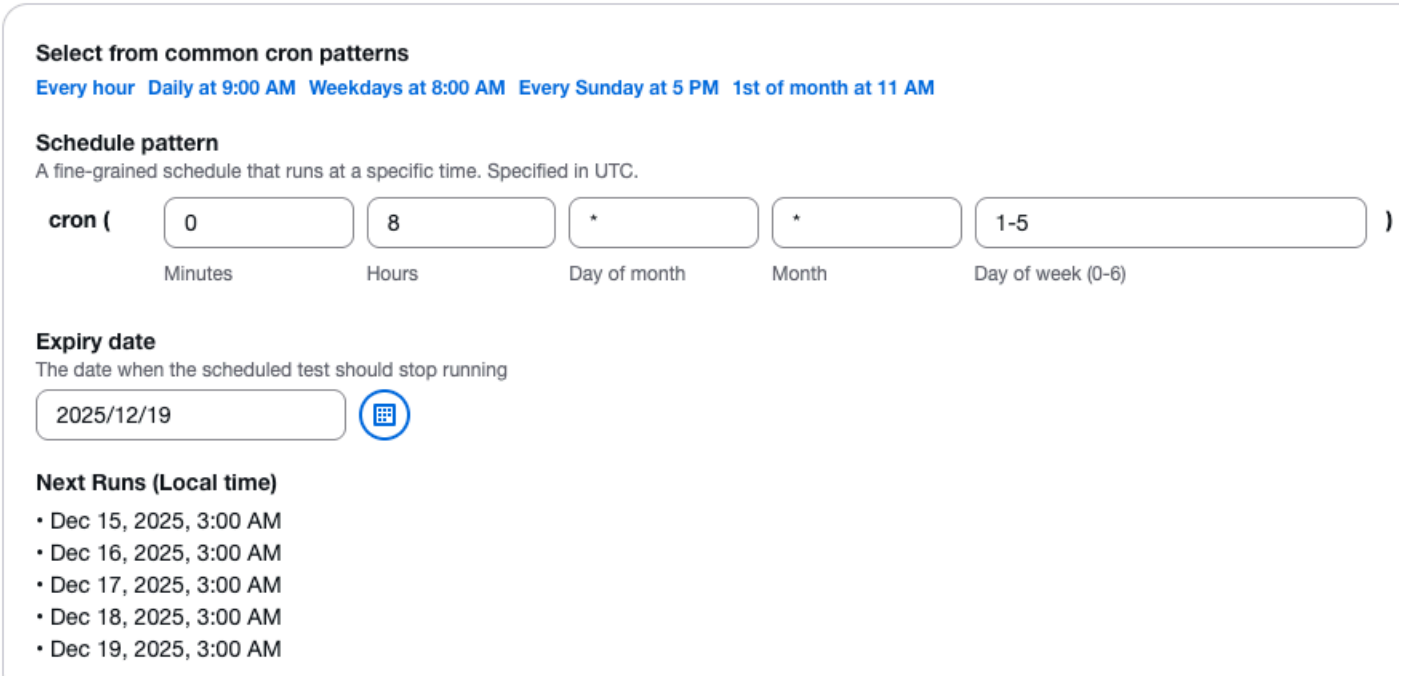
- Ejecutar ahora: ejecuta la prueba inmediatamente después de crearla.



- Ejecutar una vez: programa la prueba para que se ejecute en una fecha y hora específicas.



- Ejecute según un cronograma: utilice una programación basada en cronogramas para ejecutar las pruebas automáticamente a intervalos regulares. Puede seleccionar patrones comunes (cada hora, todos los días o semanalmente) o definir una expresión cron personalizada.



Programar flujo de trabajo

Al programar una prueba, se produce el siguiente flujo de trabajo:

- Los parámetros de programación se envían a la API de la solución a través de Amazon API Gateway.
- La API pasa los parámetros a una función Lambda que crea una regla de CloudWatch eventos programada para ejecutarse en la fecha especificada.

- Para las pruebas únicas (ejecutar una vez), la regla CloudWatch Events se ejecuta en la fecha especificada y la función `api-services` Lambda ejecuta la prueba.
- Para las pruebas recurrentes (ejecutadas según un programa), la regla CloudWatch Eventos se activa en la fecha especificada y la función `api-services` Lambda crea una nueva regla que se ejecuta de forma inmediata y recurrente en función de la frecuencia especificada.

Datos en directo

Seleccione la casilla de verificación Incluir datos en tiempo real para ver las métricas en tiempo real mientras se ejecuta la prueba. Cuando está habilitada, puedes monitorear:

- Tiempo medio de respuesta.
- Recuentos de usuarios virtuales.
- Las solicitudes realizadas correctamente cuentan.
- Recuentos de solicitudes fallidas.

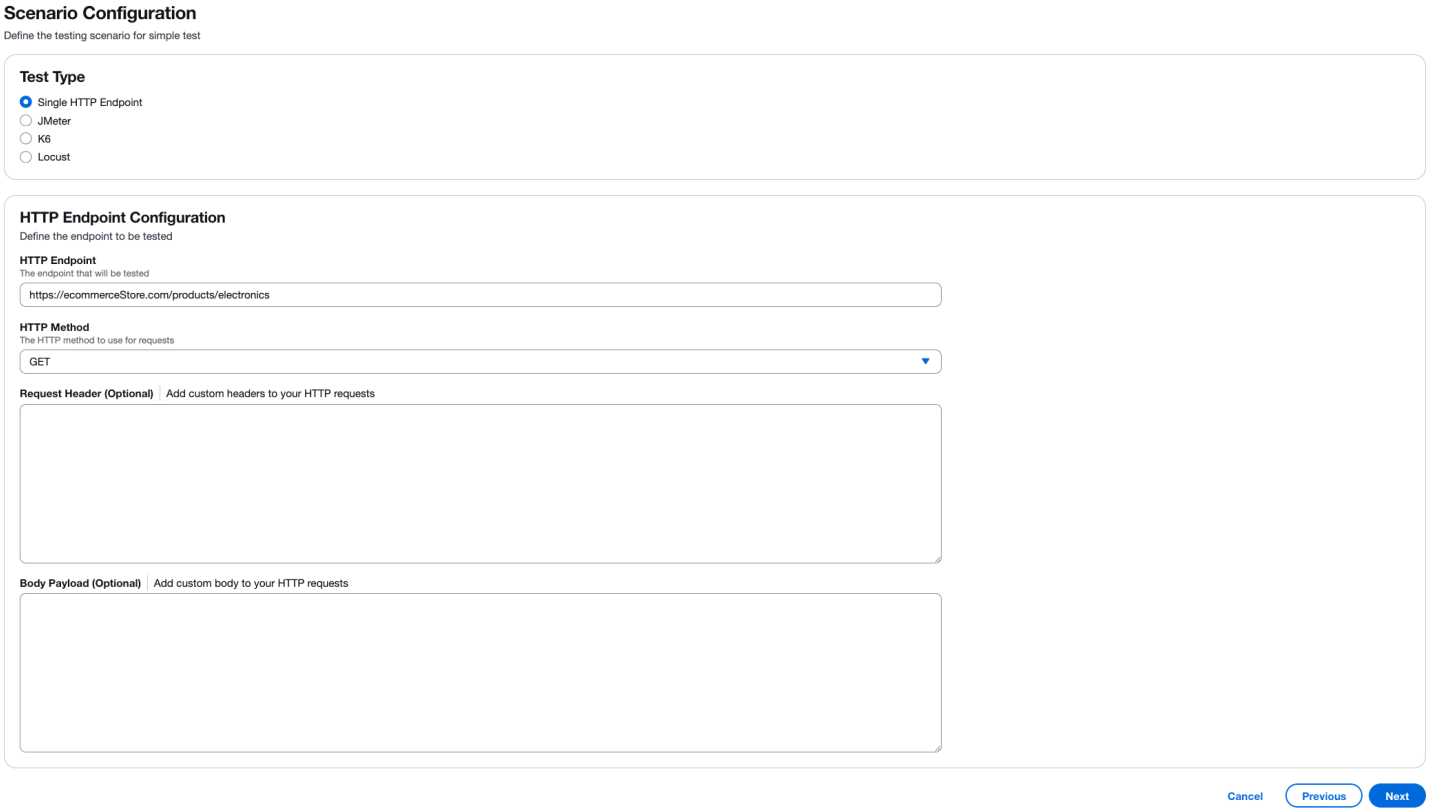
La función de datos en tiempo real proporciona gráficos en tiempo real con datos agregados en intervalos de un segundo. Para obtener más información, consulte Supervisión con datos en tiempo real.

Paso 2: Configuración del escenario

Defina el escenario de prueba específico y seleccione el marco de pruebas que prefiera.

Selección del tipo de prueba

Elija el tipo de prueba de carga que desea realizar:



- Punto final HTTP único: pruebe un único punto final de API o página web con una configuración sencilla.
- JMeter- Cargue scripts JMeter de prueba (archivos.jmx o archivos.zip).
- K6 - Cargue los scripts de prueba de K6 (archivos.js o archivos.zip).
- Locust: cargue los scripts de prueba de Locust (archivos.py o archivos.zip).

Imagen de configuración del punto final HTTP: :images/test-types.png [Seleccione el tipo de prueba que desea ejecutar] Cuando seleccione «Punto de enlace HTTP único», configure los siguientes ajustes:

Punto final HTTP (obligatorio)

> Introduzca la URL completa del punto final que desee probar. Por ejemplo, `https://api.example.com/users`. Asegúrese de que se pueda acceder al punto final desde la infraestructura de AWS.

Método HTTP (obligatorio)

Seleccione el método HTTP para sus solicitudes. El valor predeterminado es `GET`. Otras opciones incluyen `POSTPUT`,`DELETE`, `PATCHHEAD`, y`OPTIONS`.

Encabezado de solicitud (opcional)

Agrega encabezados HTTP personalizados a tus solicitudes. Los ejemplos comunes incluyen:

- `Content-Type: application/json`
- `Authorization: Bearer <token>`
- `User-Agent: LoadTest/1.0`

Elija Agregar encabezado para incluir varios encabezados.

Carga útil corporal (opcional)

Agregue el contenido del cuerpo de la solicitud para las solicitudes POST o PUT. Admite los formatos JSON, XML o texto sin formato. Por ejemplo: `{"userId": 123, "action": "test"}`.

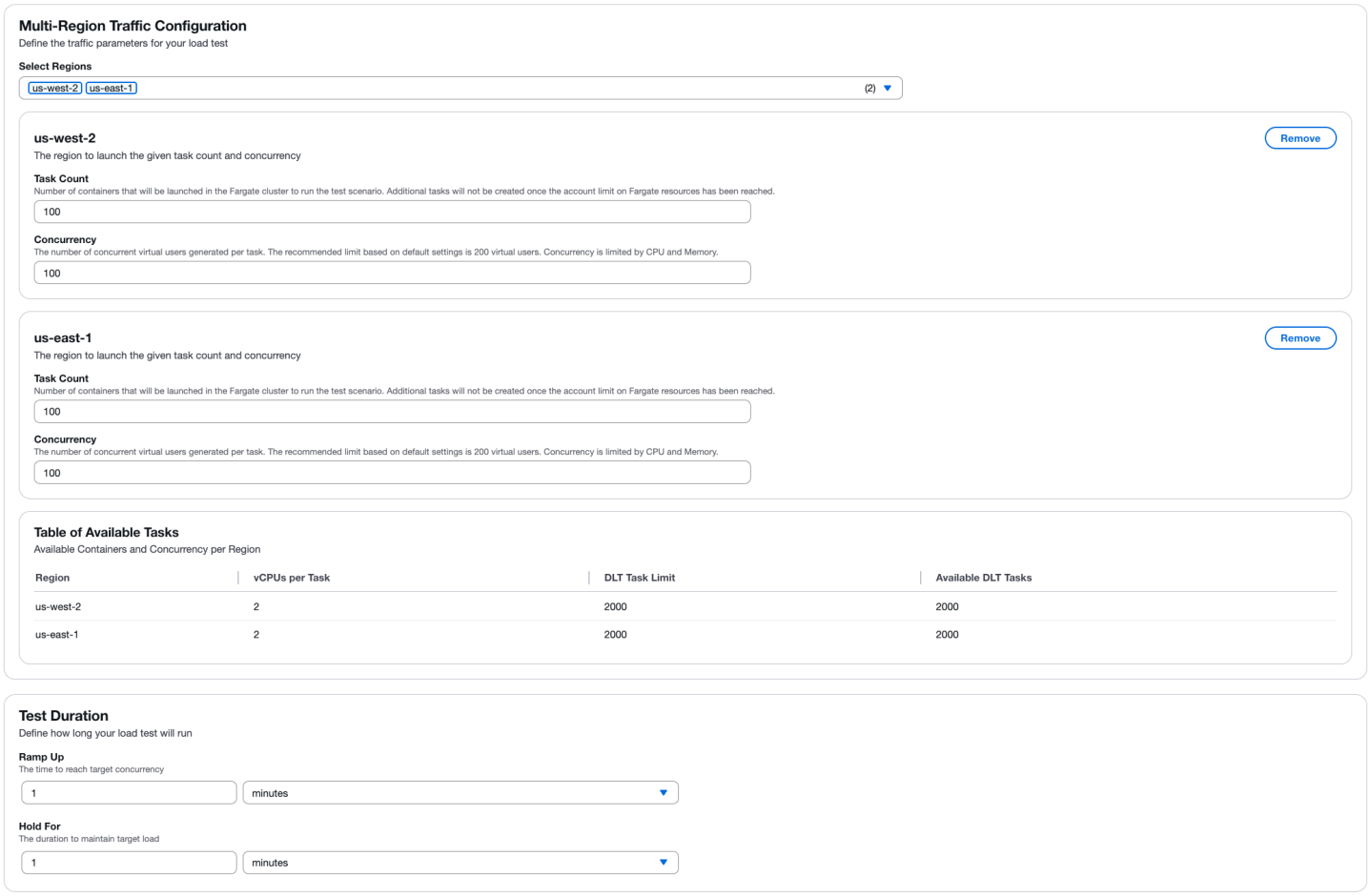
Pruebe los scripts del framework

Cuando utilice JMeter K6 o Locust, cargue el archivo del script de prueba o un archivo.zip que contenga el guion de prueba y los archivos auxiliares. Por ejemplo JMeter, puedes incluir complementos personalizados en una `/plugins` carpeta dentro de tu archivo.zip.

> **Important**
>
> Si bien su script de prueba (JMeter, K6 o Locust) puede definir la simultaneidad (usuarios virtuales), las tasas de transacción (TPS), los tiempos de aceleración y otros parámetros de carga, la solución anulará estas configuraciones con los valores que especifique en la pantalla Traffic Shape durante la creación de la prueba. La configuración de Traffic Shape controla el recuento de tareas, la simultaneidad (usuarios virtuales por tarea), la duración del aumento y el tiempo de espera durante la ejecución de la prueba.

Paso 3: Forma del tráfico

Configure la forma en que se distribuirá el tráfico durante la prueba, incluida la compatibilidad multirregional.



Configuración de tráfico multirregional

Seleccione una o más regiones de AWS para distribuir geográficamente la prueba de carga. Para cada región seleccionada, configure:

Recuento de tareas

- El número de contenedores (tareas) que se lanzarán en el clúster de Fargate para el escenario de prueba. No se crearán tareas adicionales una vez que la cuenta haya alcanzado el límite de «Se ha alcanzado el recurso de Fargate».

Concurrency (Simultaneidad)

- El número de usuarios virtuales simultáneos generados por tarea. El límite recomendado se basa en la configuración predeterminada de 2 v CPUs por tarea. La simultaneidad está limitada por los recursos de CPU y memoria.

Determine el número de usuarios

El número de usuarios que puede admitir un contenedor para una prueba se puede determinar aumentando gradualmente el número de usuarios y supervisando el rendimiento en Amazon CloudWatch. Una vez que observe que el rendimiento de la CPU y la memoria se acerca a sus límites, habrá alcanzado el número máximo de usuarios que un contenedor puede admitir para esa prueba en su configuración predeterminada (2 vCPU y 4 GB de memoria).

Proceso de calibración

Puede empezar a determinar los límites de usuarios simultáneos para la prueba mediante el siguiente ejemplo:

1. Cree una prueba con un máximo de 200 usuarios.
2. Mientras se ejecuta la prueba, supervise la CPU y la memoria mediante la CloudWatch consola:
   a. En el panel de navegación izquierdo, en Container Insights, selecciona Supervisión del rendimiento.
   b. En la página de supervisión del rendimiento, en el menú desplegable de la izquierda, selecciona ECS Clusters.
   c. En el menú desplegable de la derecha, selecciona tu clúster de Amazon Elastic Container Service (Amazon ECS).
3. Mientras monitorea, observe la CPU y la memoria. Si la CPU no supera el 75% o la memoria no supera el 85% (ignore los picos únicos), puede realizar otra prueba con un número mayor de usuarios.

Repita los pasos 1 a 3 si la prueba no superó los límites de recursos. Si lo desea, puede aumentar los recursos del contenedor para permitir un mayor número de usuarios simultáneos. Sin embargo, esto se traduce en un coste mayor. Para obtener más información, consulta la Guía para desarrolladores.

> **Note**
>
> Para obtener resultados precisos, ejecute solo una prueba a la vez para determinar los límites de usuarios simultáneos. Todas las pruebas utilizan el mismo clúster, y CloudWatch Container Insights agrega los datos de rendimiento en función del clúster. Esto hace que ambas pruebas se notifiquen a CloudWatch Container Insights simultáneamente, lo que da como resultado métricas de uso de recursos inexactas en una sola prueba.

Para obtener más información sobre la calibración de los usuarios por motor, consulte Calibración de una prueba Taurus en la documentación. BlazeMeter

> **Note**
>
> La solución muestra la información sobre la capacidad disponible para cada región, lo que le ayuda a planificar la configuración de la prueba dentro de los límites disponibles.

Tabla de tareas disponibles

La tabla de tareas disponibles muestra la disponibilidad de recursos para cada región seleccionada:

- Región: el nombre de la región de AWS.
- v CPUs por tarea: la cantidad de elementos virtuales CPUs asignados a cada tarea (predeterminado: 2).
- Límite de tareas de DLT: el número máximo de tareas que se pueden crear en función de los límites de Fargate de tu cuenta (predeterminado: 2000).
- Tareas de DLT disponibles: el número actual de tareas disponibles para su uso en la región (predeterminado: 2000).

Table of Available Tasks

Available Containers and Concurrency per Region

Region	vCPUs per Task	DLT Task Limit	Available DLT Tasks
us-west-2	2	2000	2000
us-east-1	2	2000	2000

Para aumentar el número de tareas disponibles o v CPUs por tarea, consulta la Guía para desarrolladores.

Duración de la prueba

Defina cuánto tiempo durará la prueba de carga:

Aumente

El tiempo necesario para alcanzar el objetivo de simultaneidad. La carga aumenta gradualmente desde 0 hasta el nivel de simultaneidad configurado durante este período.

Mantenga pulsado durante

El tiempo necesario para mantener la carga objetivo. La prueba continúa con total simultaneidad durante este período.

Paso 4: Revisar y crear

Revise todas las configuraciones antes de crear el escenario de prueba. Verificar:

- Configuración general (nombre, descripción, horario).
- Configuración del escenario (tipo de prueba, punto final o script).
- Forma del tráfico (tareas, usuarios, duración, regiones).

Tras revisarlo, selecciona Crear para guardar el escenario de prueba.

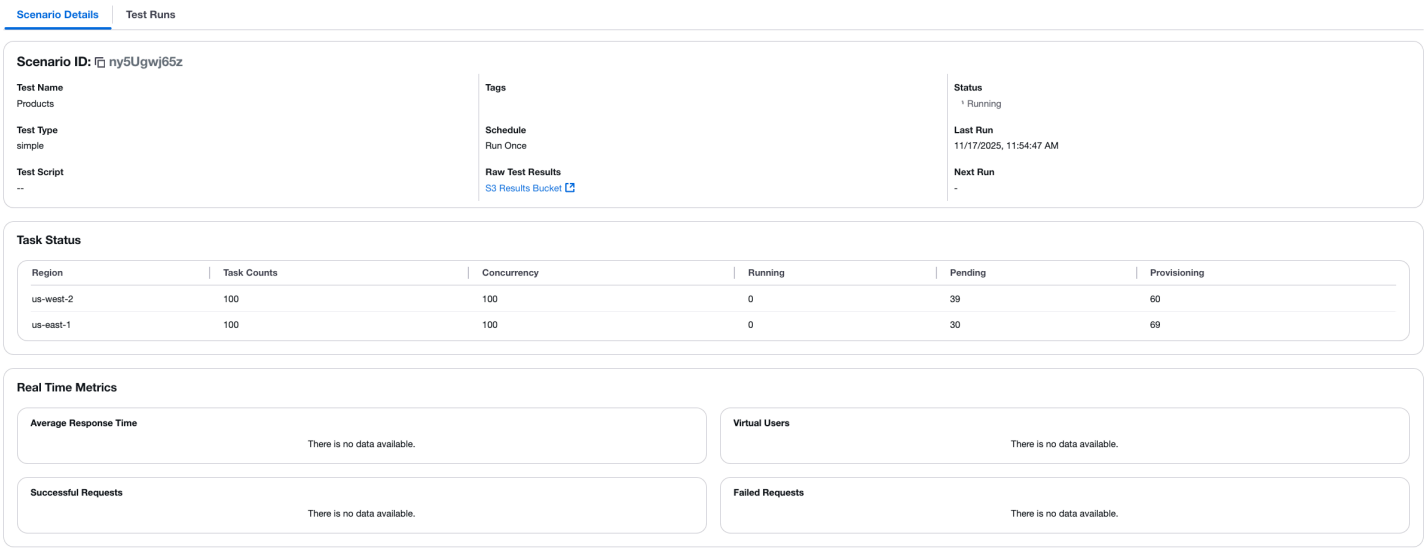
Administrar escenarios de prueba

Tras crear un escenario de prueba, puede:

- Editar: modificar la configuración de la prueba. Los casos de uso comunes incluyen:
  - Refinar la forma del tráfico para lograr la tasa de transacción deseada.
- Copiar: duplique un escenario de prueba existente para crear variantes. Los casos de uso comunes incluyen:
  - Actualizar los puntos finales o añadir headers/body parámetros.
  - Añadir o modificar scripts de prueba.
- Eliminar: elimine los escenarios de prueba que ya no necesite.

Ejecute un escenario de prueba

Tras crear un escenario de prueba, puede ejecutarlo inmediatamente o programarlo para que se ejecute en un momento específico en el futuro. Al navegar hasta una prueba en ejecución, la consola muestra la pestaña Detalles del escenario con el estado y las métricas de las tareas en tiempo real.



Vista de detalles del escenario

La pestaña Detalles del escenario muestra información clave sobre la prueba. La tabla de estado de la tarea incluye información en tiempo real para cada región.

Tabla de estado de las tareas

La tabla de estado de la tarea muestra información en tiempo real de cada región:

- Región: la región de AWS en la que se ejecutan las tareas
- Recuentos de tareas: el número total de tareas configuradas para la región
- Simultaneidad: el número de usuarios virtuales por tarea
- En ejecución: número de tareas que actualmente están ejecutando la prueba
- Pendiente: número de tareas en espera de iniciarse
- Aprovisionamiento: número de tareas que se están aprovisionando

flujo de trabajo de ejecución de pruebas

Cuando se inicia una prueba, se produce el siguiente flujo de trabajo:

1. Aprovisionamiento de tareas: la solución aprovisiona contenedores (tareas) en las regiones de AWS especificadas. Las tareas aparecen en la columna «Aprovisionamiento».

2. Inicio de tareas: la solución continúa aprovisionando tareas hasta que se alcance el número de tareas objetivo en cada región. Las tareas pasan de «Aprovisionamiento» a «Pendientes» y luego a «En ejecución».
3. Generación de tráfico: una vez que la solución aprovisiona todas las tareas de una región, comienza a enviar tráfico al punto final de destino.
4. Ejecución de la prueba: la prueba se ejecuta durante el tiempo configurado (aumento + tiempo de espera).
5. Análisis de resultados: cuando finaliza la prueba, un trabajo de análisis en segundo plano agrega y procesa los resultados de todas las regiones.

Estados de ejecución de la prueba

Las ejecuciones de prueba pueden tener los siguientes estados:

- Programado: la prueba está programada para ejecutarse en el futuro.
- En ejecución: la prueba se encuentra actualmente en curso.
- Cancelada: un usuario ha cancelado una prueba en curso.
- Error: la ejecución de la prueba detectó un error.
- Completa: la prueba se ha realizado correctamente y los resultados están listos.

Supervisión con datos en tiempo real

Si habilitó los datos en tiempo real al crear el escenario de prueba, puede ver las métricas en tiempo real mientras se ejecuta la prueba. La sección Métricas en tiempo real muestra cuatro gráficos que se actualizan continuamente a medida que avanza la prueba, con datos agregados en intervalos de un segundo.

Descripciones gráficas

Tiempo medio de respuesta

Muestra el tiempo medio de respuesta en segundos de las solicitudes procesadas por cada región. El eje Y muestra el tiempo de respuesta en segundos y el eje X muestra la hora del día. Cada región se representa con un color diferente en la leyenda.

Usuarios virtuales

Muestra el número de usuarios virtuales simultáneos que generan carga de forma activa en cada región. El gráfico muestra cómo aumentan los usuarios virtuales durante la prueba y mantiene el nivel de simultaneidad objetivo.

Solicitudes satisfactorias

Muestra el recuento acumulado de solicitudes satisfactorias a lo largo del tiempo para cada región. El gráfico muestra la velocidad a la que se procesan las solicitudes aprobadas.

Solicitudes fallidas

Muestra el recuento acumulado de solicitudes fallidas a lo largo del tiempo para cada región. Un recuento bajo o cero indica una ejecución correcta de la prueba.

Visualización multirregional

Al ejecutar pruebas en varias regiones, cada gráfico muestra los datos de todas las regiones simultáneamente. La leyenda en la parte inferior de cada gráfico identifica qué color representa cada región (por ejemplo, us-west-2 y us-east-1).

Implementación técnica

El grupo de CloudWatch registros de las tareas de Fargate contiene un filtro de suscripción que captura los resultados de las pruebas. Cuando se detecta el patrón, una función de Lambda estructura los datos y los publica en un tema de AWS IoT Core. La consola web se suscribe a este tema y muestra las métricas en tiempo real.

> **Note**
>
> Los datos en tiempo real son efímeros y solo están disponibles mientras se ejecuta la prueba. La consola web conserva un máximo de 5000 puntos de datos, tras lo cual los datos más antiguos se sustituyen por los más recientes. Si la página se actualiza, los gráficos estarán en blanco y comenzarán desde el siguiente punto de datos disponible. Una vez finalizada la prueba, la solución almacena los datos de los resultados en DynamoDB y Amazon S3. Si aún no hay datos disponibles, los gráficos muestran el mensaje «No hay datos disponibles».

Cancelar una prueba

Puede cancelar una prueba en ejecución desde la consola web. Al cancelar una prueba, se produce el siguiente flujo de trabajo:

1. La solicitud de cancelación se envía a la `microservices` API
2. La `microservices` API llama a la función `task-canceler` Lambda, que detiene todas las tareas actualmente iniciadas.
3. Si la función `task-runner` Lambda sigue ejecutándose después de la llamada de cancelación inicial, es posible que las tareas se sigan iniciando brevemente
4. Una vez `task-runner` finalizada la función Lambda, AWS Step Functions pasa al `Cancel Test` paso, en el que se vuelve a ejecutar la función `task-canceler` Lambda para detener las tareas restantes.

> **Note**
>
> Las pruebas canceladas tardan un tiempo en completar el proceso de cierre, ya que la solución cierra todos los contenedores. El estado de la prueba cambiará a «Cancelada» una vez que se hayan limpiado todos los recursos.

Explore los resultados de las pruebas

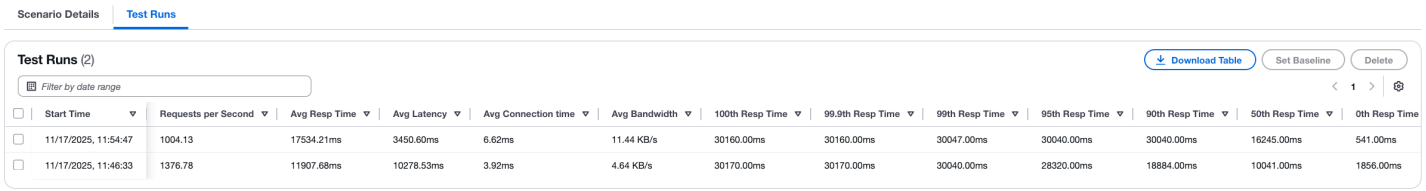
Una vez que se complete el trabajo de análisis, los resultados de las pruebas estarán disponibles para su análisis. La solución proporciona métricas y herramientas integrales para ayudarlo a comprender el rendimiento de su aplicación bajo carga.

Métricas resumidas de la ejecución de pruebas

Cuando se completa una prueba, la solución genera un resumen que incluye las siguientes métricas:

- Tiempo medio de respuesta: el tiempo medio de respuesta, en segundos, de todas las solicitudes generadas por la prueba.
- Latencia media: la latencia media, en segundos, de todas las solicitudes generadas por la prueba.
- Tiempo medio de conexión: el tiempo medio, en segundos, que se tarda en conectarse al host para todas las solicitudes.
- Ancho de banda promedio: el ancho de banda promedio de todas las solicitudes generadas por la prueba.
- Recuento total: el número total de solicitudes.
- Recuento de solicitudes satisfactorias: número total de solicitudes satisfactorias.
- Recuento de errores: el número total de errores.
- Solicitudes por segundo: el promedio de solicitudes por segundo de todas las solicitudes generadas por la prueba.
- Percentiles: los percentiles del tiempo de respuesta incluyen p50 (mediana), p90, p95 y p99, que muestran la distribución de los tiempos de respuesta entre todas las solicitudes.

Tabla de ejecuciones de pruebas



La tabla de ejecuciones de prueba muestra todas las ejecuciones de prueba históricas de un escenario. Puede:

- Ver las métricas resumidas de cada ejecución de la prueba.
- Establezca una ejecución de prueba de referencia para comparar el rendimiento.
- Descarga la tabla como un archivo CSV.
- Cambia las columnas para personalizar la vista.
- Seleccione una ejecución de prueba para ver los resultados detallados.

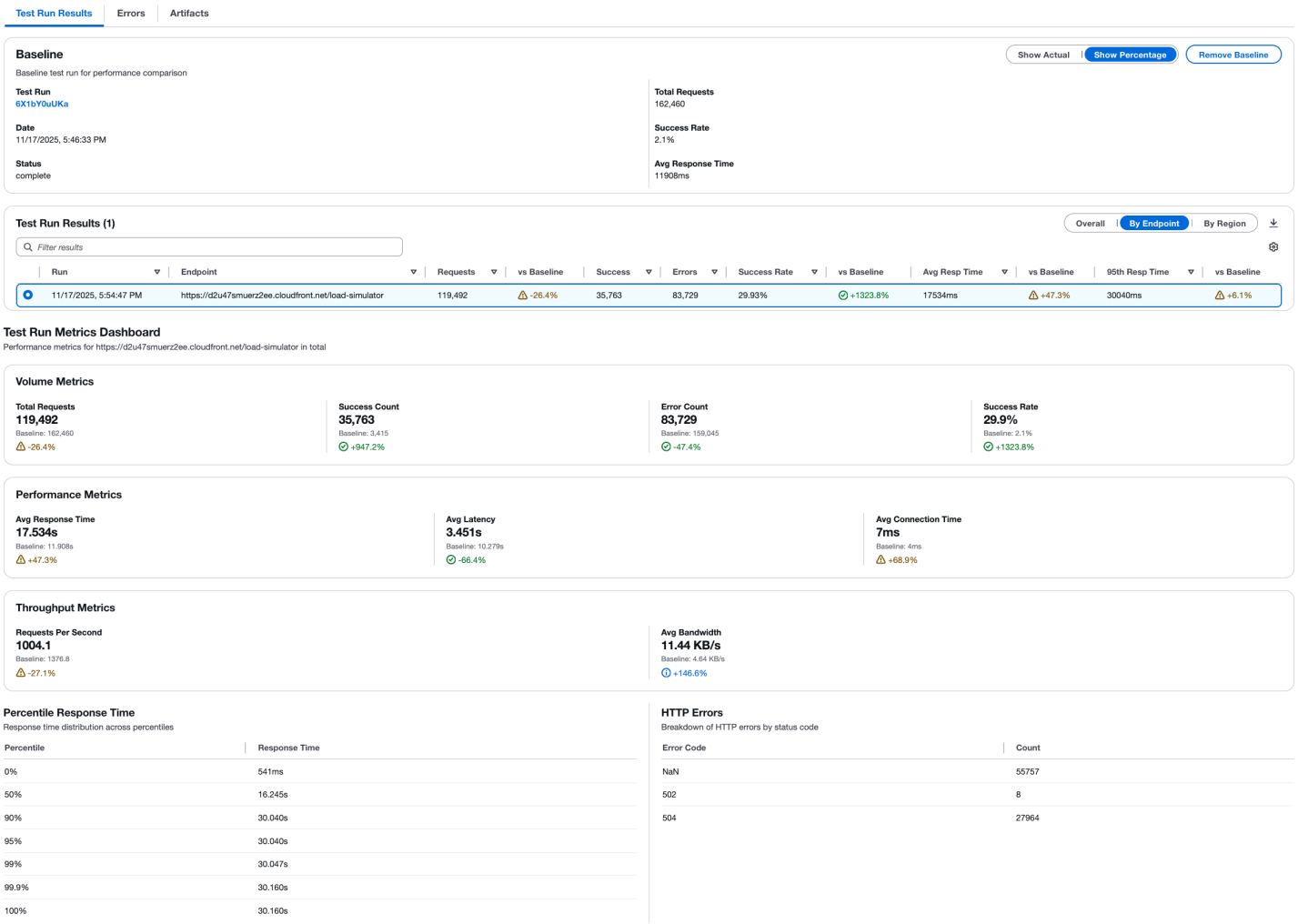
Comparación de referencia

Puede designar una ejecución de prueba como referencia para comparar futuras ejecuciones de prueba con ella. Cuando se establece una línea base:

- La tabla de pruebas muestra las diferencias porcentuales (+/-%) en comparación con la línea base de cada métrica.
- El indicador de referencia le ayuda a identificar rápidamente las mejoras o regresiones del rendimiento.
- Puede cambiar o borrar la línea base en cualquier momento.

Resultados detallados de las pruebas

Al seleccionar una ejecución de prueba, se abre la vista de resultados detallados con tres pestañas: Resultados de la prueba, Errores y Artefactos.



Información de referencia

Si se establece una ejecución de prueba de referencia, se muestra en la parte superior de la página. Puede elegir Mostrar valores reales, Mostrar porcentaje o Eliminar línea base para controlar cómo se muestran las comparaciones de referencia.

Tabla de resultados de la prueba

La tabla de resultados proporciona métricas detalladas con las siguientes características:

Vistas de dimensiones

Cambie entre tres vistas mediante los botones de dimensión:

- En general: resultados agregados en todos los puntos finales y regiones
- Por punto final: resultados desglosados por puntos finales individuales

- Por región: resultados desglosados por región de AWS

Botones de acción

- Mostrar valores métricos reales: muestra los valores métricos reales
- Mostrar porcentaje: muestra las diferencias porcentuales con respecto a la línea base
- Eliminar la línea base: borra la comparación de la línea base

Exportación y personalización de datos

- Descargue la tabla de resultados como un archivo CSV
- Cambia las columnas para personalizar la vista
- Filtra y ordena los datos para centrarte en métricas específicas
- Filtra y ordena los datos para centrarte en métricas específicas.

Pestaña de errores

La pestaña de errores proporciona un análisis detallado de los errores:

- Vea los recuentos de errores por tipo.
- Consulte los errores agregados por prueba general o por punto final.
- Identifique los patrones en las solicitudes fallidas.
- Solucione problemas con puntos finales o regiones específicos.

Pestaña de artefactos

La pestaña de artefactos le permite acceder a todos los archivos generados durante la prueba:

- Vea los artefactos individuales (registros, archivos de resultados).
- Descargue artefactos específicos para analizarlos sin conexión.
- Descargue todos los artefactos de las pruebas realizadas en un único archivo.

Estructura de resultados de S3

En la versión 4.0, la estructura de resultados de S3 ha cambiado para mejorar la organización:

- Nueva estructura `-scenario-id/test-run-id/results-files`.
- Estructura heredada: las pruebas realizadas antes de la versión 4.0 muestran todos los archivos de resultados en el nivel de ID del escenario.

> **Note**
>
> Los resultados de las pruebas se muestran en la consola. También puede acceder a los resultados de las pruebas sin procesar directamente en el depósito de Amazon S3 que se encuentra debajo de la `Results` carpeta. Para obtener más información sobre los resultados de las pruebas de Taurus, consulte Generación de informes de pruebas en el manual del usuario de Taurus.

Integración de servidores MCP

Si implementó el componente de servidor MCP opcional durante la implementación de la solución, puede integrar la solución de pruebas de carga distribuidas con herramientas de desarrollo de IA compatibles con el protocolo Model Context. El servidor MCP proporciona acceso programático para recuperar, gestionar y analizar las pruebas de carga mediante asistentes de IA.

Los clientes pueden conectarse al servidor MCP DLT mediante el cliente que elijan (Amazon Q, Claude, etc.), cada uno con instrucciones de configuración ligeramente diferentes. En esta sección se proporcionan instrucciones de configuración para MCP Inspector, Amazon Q CLI, Cline y Amazon Q Suite.

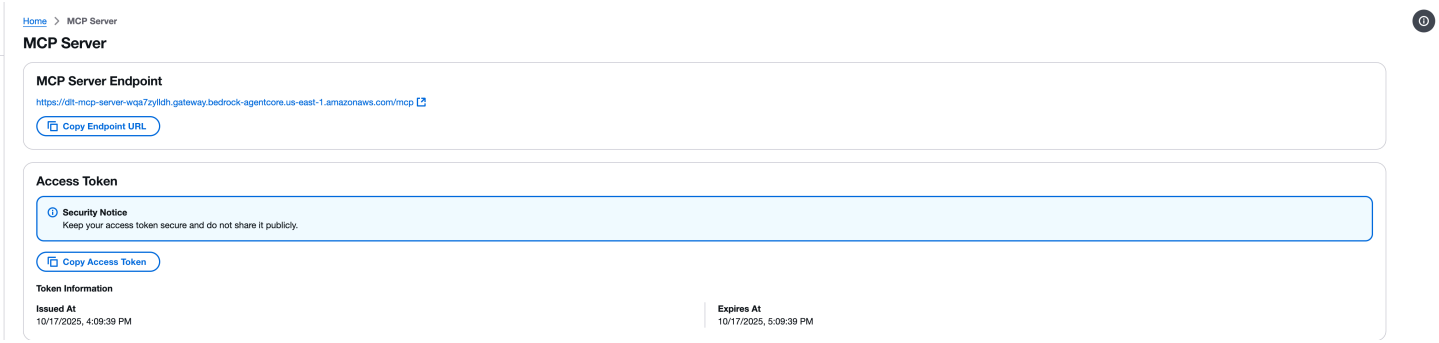
Paso 1: Obtenga el punto final y el token de acceso de MCP

Antes de configurar cualquier cliente MCP, debe recuperar el punto final del servidor MCP y el token de acceso de la consola web DLT.

1. Navegue a la página del servidor MCP en la consola web de pruebas de carga distribuidas.
2. Localice la sección Punto final del servidor MCP.
3. Copie la URL del punto final con el botón Copiar la URL del punto final. La URL del punto final tiene el siguiente formato: `https://{gateway-id}.gateway.bedrock-agentcore.{region}.amazonaws.com/mcp`
4. Localice la sección del token de acceso.

5. Copie el token de acceso con el botón Copiar el token de acceso.

> **Important**
>
> Mantén tu token de acceso seguro y no lo compartas públicamente. El token proporciona acceso de solo lectura a su solución de pruebas de carga distribuidas a través de la interfaz MCP.



Paso 2: Probar con el Inspector MCP

El Model Context Protocol ofrece el Inspector MCP, una herramienta para conectarse directamente a los servidores MCP e invocar herramientas. Esto proporciona una interfaz de usuario práctica y ejemplos de solicitudes de red para probar la conexión del servidor MCP antes de configurar los clientes de IA.

> **Note**
>
> El Inspector MCP requiere la versión 0.17 o posterior. Todas las solicitudes también se pueden realizar directamente con JSON RPC, pero MCP Inspector proporciona una interfaz más fácil de usar.

Instalar e iniciar MCP Inspector

1. Instale npm si es necesario.
2. Ejecute el siguiente comando para iniciar el Inspector MCP:

```
npx @modelcontextprotocol/inspector
```

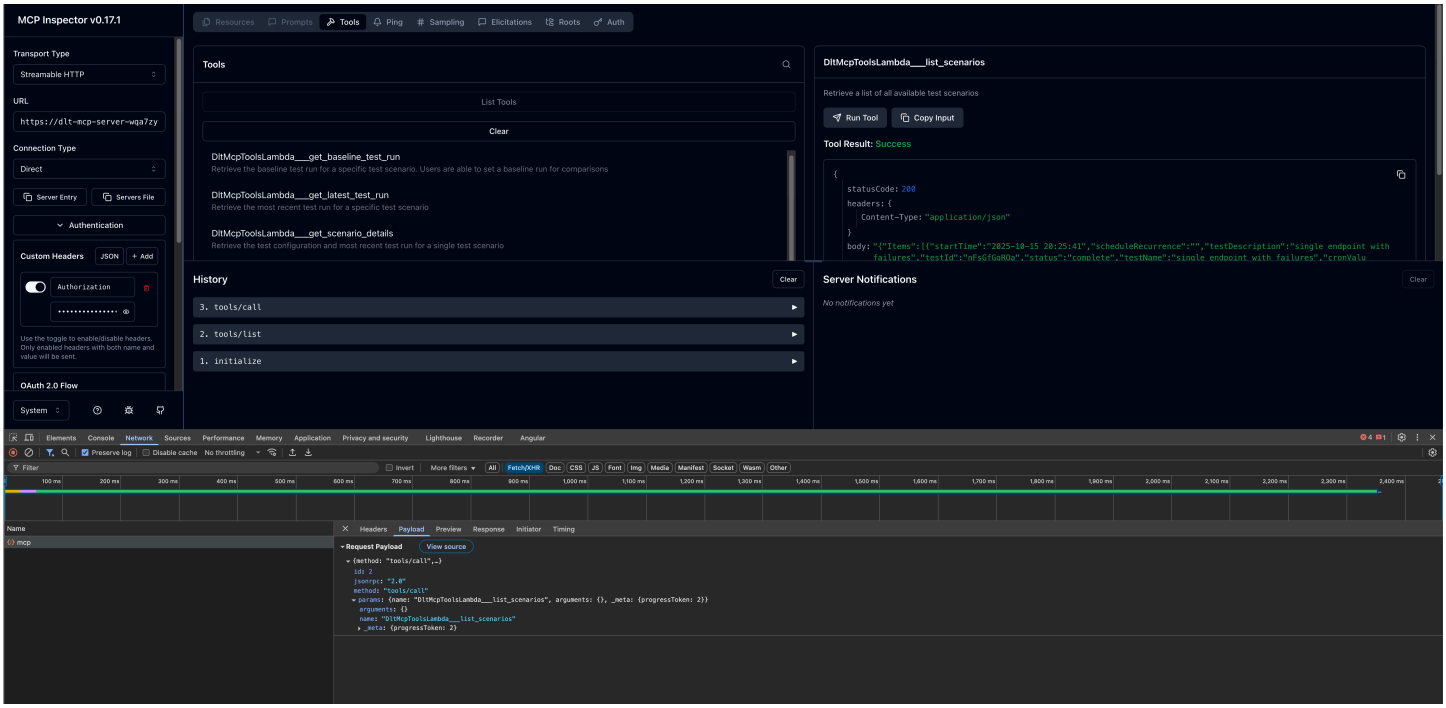
Configure la conexión

1. En la interfaz del Inspector MCP, introduzca la URL del punto final del servidor MCP.
2. Agregue un encabezado de autorización con su token de acceso.
3. Haga clic en Conectar para establecer la conexión.

Invoca herramientas

Una vez conectado, puede probar las herramientas MCP disponibles:

1. Examine la lista de herramientas disponibles en el panel izquierdo.
2. Seleccione una herramienta (por ejemplo,`list_scenarios`).
3. Proporcione los parámetros necesarios.
4. Haga clic en Invocar para ejecutar la herramienta y ver la respuesta.



Paso 3: Configurar los clientes de desarrollo de IA

Después de verificar la conexión de su servidor MCP con el Inspector MCP, puede configurar su cliente de desarrollo de IA preferido.

CLI de Amazon Q

Amazon Q CLI proporciona acceso desde la línea de comandos al desarrollo asistido por IA con integración de servidores MCP.

Pasos de configuración

1. Edite el archivo de configuración `mcp.json`. Para obtener más información sobre la ubicación del archivo de configuración, consulte Configuración de servidores MCP remotos en la Guía del usuario para desarrolladores de Amazon Q.
2. Añada la configuración de su servidor MCP DLT:

```
{
  "mcpServers": {
    "dlt-mcp": {
      "type": "http",
      "url": "https://[api-id].execute-api.[region].amazonaws.com/[stage]/gateway/
backend-agent/sse/mcp",
```

```
      "headers": {
        "Authorization": "your_access_token_here"
      }
    }
  }
}
```

Compruebe la configuración

1. En una terminal, escriba q para iniciar Amazon Q CLI.
2. Escriba `/mcp` para ver todos los servidores MCP disponibles.
3. Escriba `/tools` para ver las herramientas disponibles proporcionadas por `dlt-mcp` y otros servidores MCP configurados.
4. Compruebe que se inicializa `dlt-mcp` correctamente.

Cline

Cline es un asistente de codificación de IA que admite la integración del servidor MCP.

Pasos de configuración

1. En Cline, vaya a Administrar servidores MCP > Configurar > Configurar servidores MCP.
2. Actualice el archivo `cline_mcp_settings.json`:

```
{
  "mcpServers": {
    "dlt-mcp": {
      "type": "streamableHttp",
      "url": "https://[api-id].execute-api.[region].amazonaws.com/[stage]/gateway/
backend-agent/sse/mcp",
      "headers": {
        "Authorization": "your_access_token_here"
      }
    }
  }
}
```

3. Guarde el archivo de configuración.
4. Reinicie Cline para aplicar los cambios.

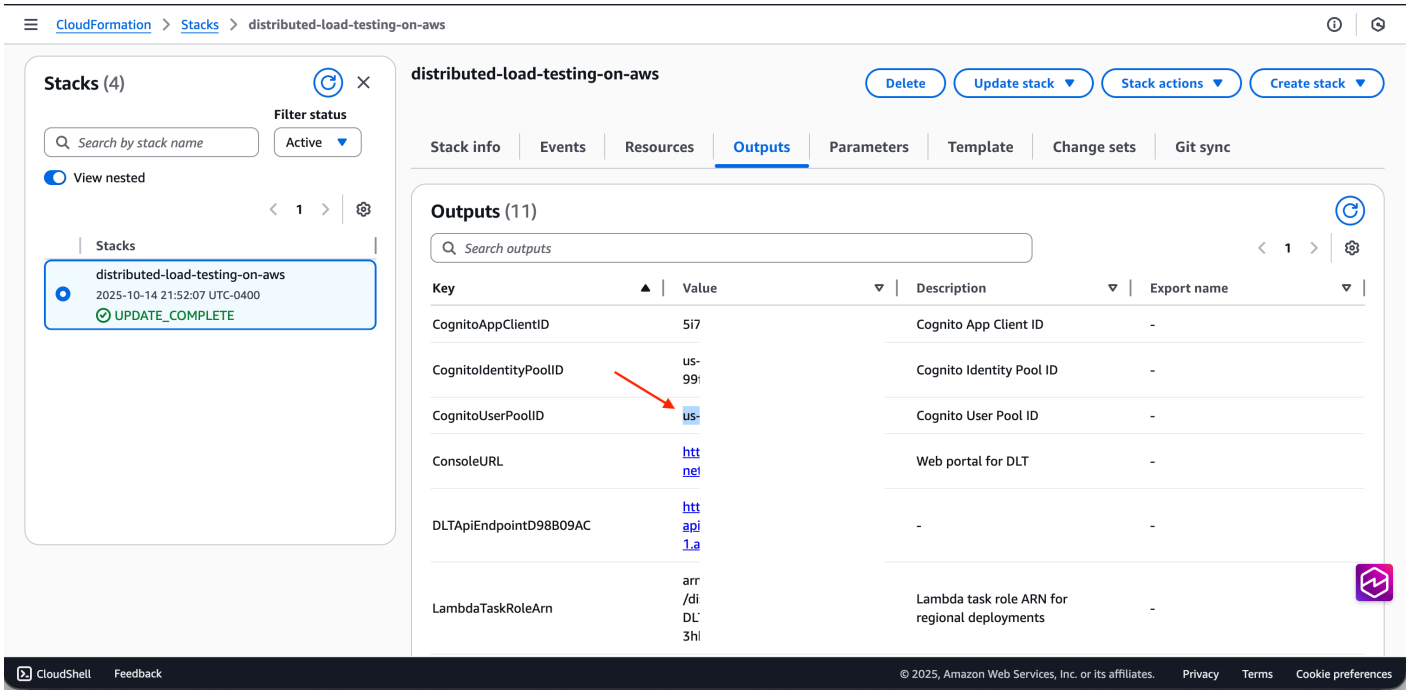
Amazon Q Suite

Amazon Q Suite proporciona una plataforma integral de asistente de IA compatible con las acciones del servidor MCP.

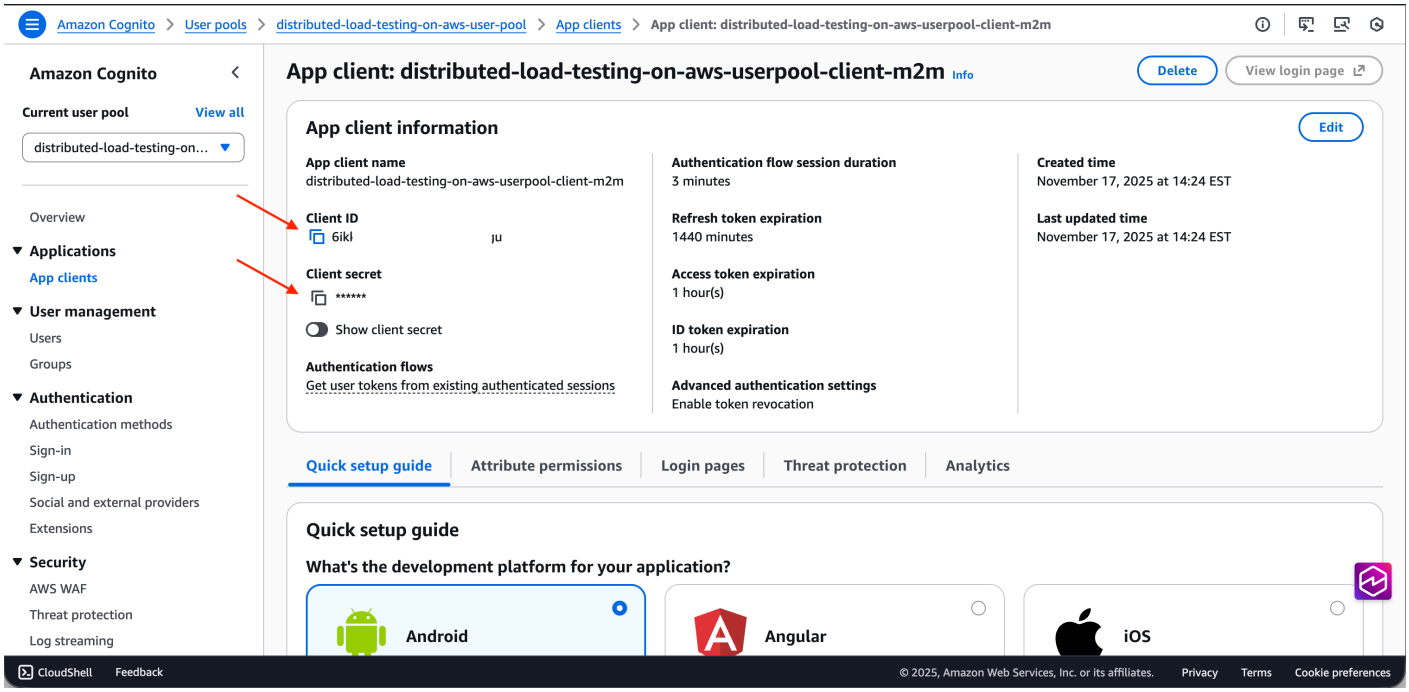
Requisitos previos

Antes de configurar el servidor MCP en Amazon Q Suite, debe recuperar OAuth las credenciales del grupo de usuarios de Cognito de la implementación de DLT:

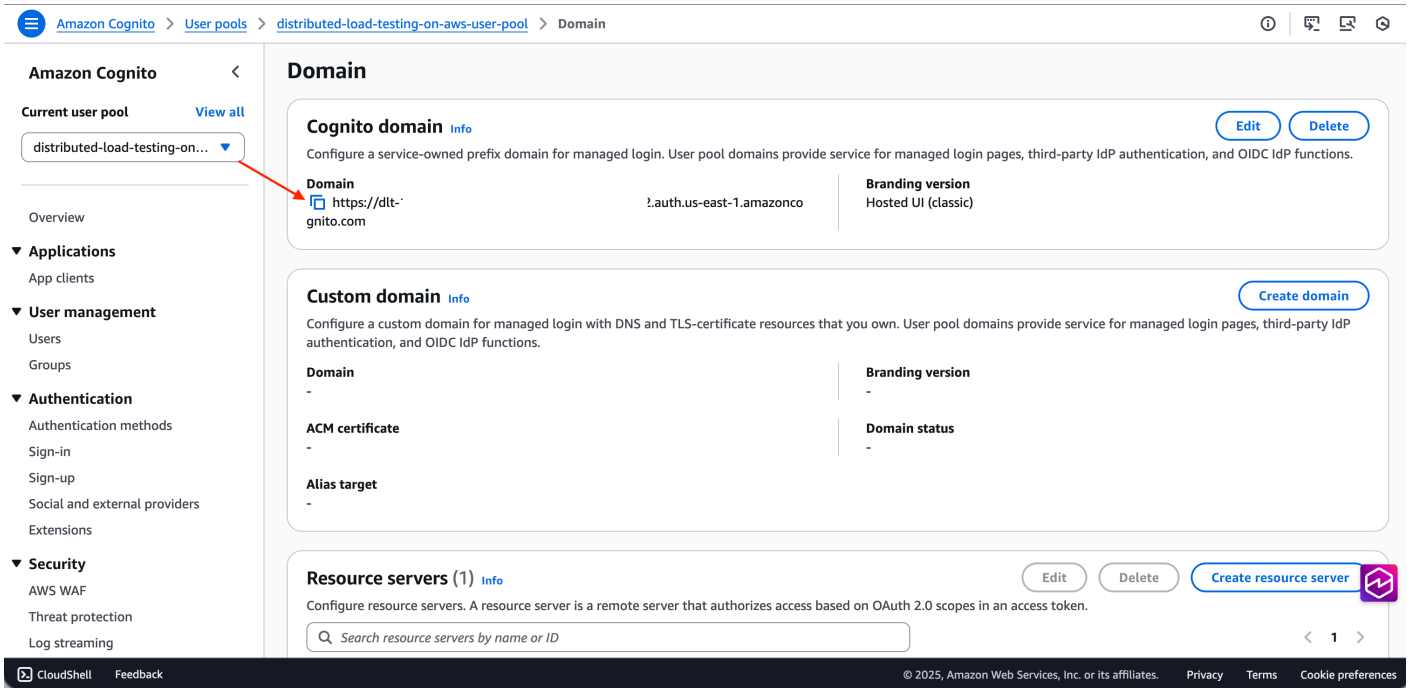
1. Navegue hasta la CloudFormation consola de AWS.
2. Seleccione la pila de pruebas de carga distribuidas.
3. En la pestaña Salidas, localice y copie el ID del grupo de usuarios de Cognito asociado a su implementación de DLT.



4. Vaya a la consola de Amazon Cognito.
5. Seleccione el grupo de usuarios mediante el ID del grupo de usuarios de los CloudFormation resultados.
6. En el menú de navegación de la izquierda, selecciona Integración de aplicaciones > Clientes de aplicaciones.



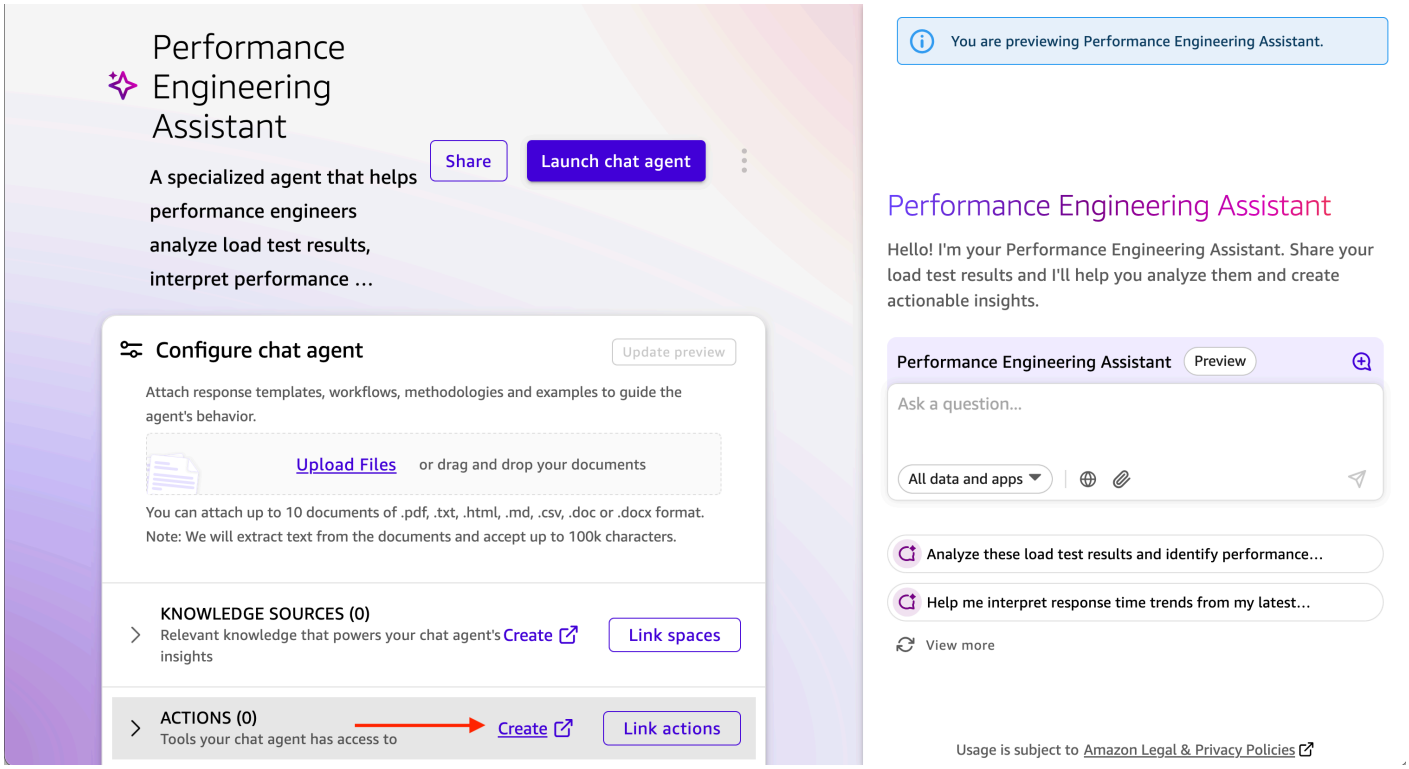
7. Localice el cliente de la aplicación cuyo nombre termine en `m2m` (machine-to-machine).
8. Copie el ID del cliente y el secreto del cliente.
9. Obtenga el dominio del grupo de usuarios en la pestaña Dominio.

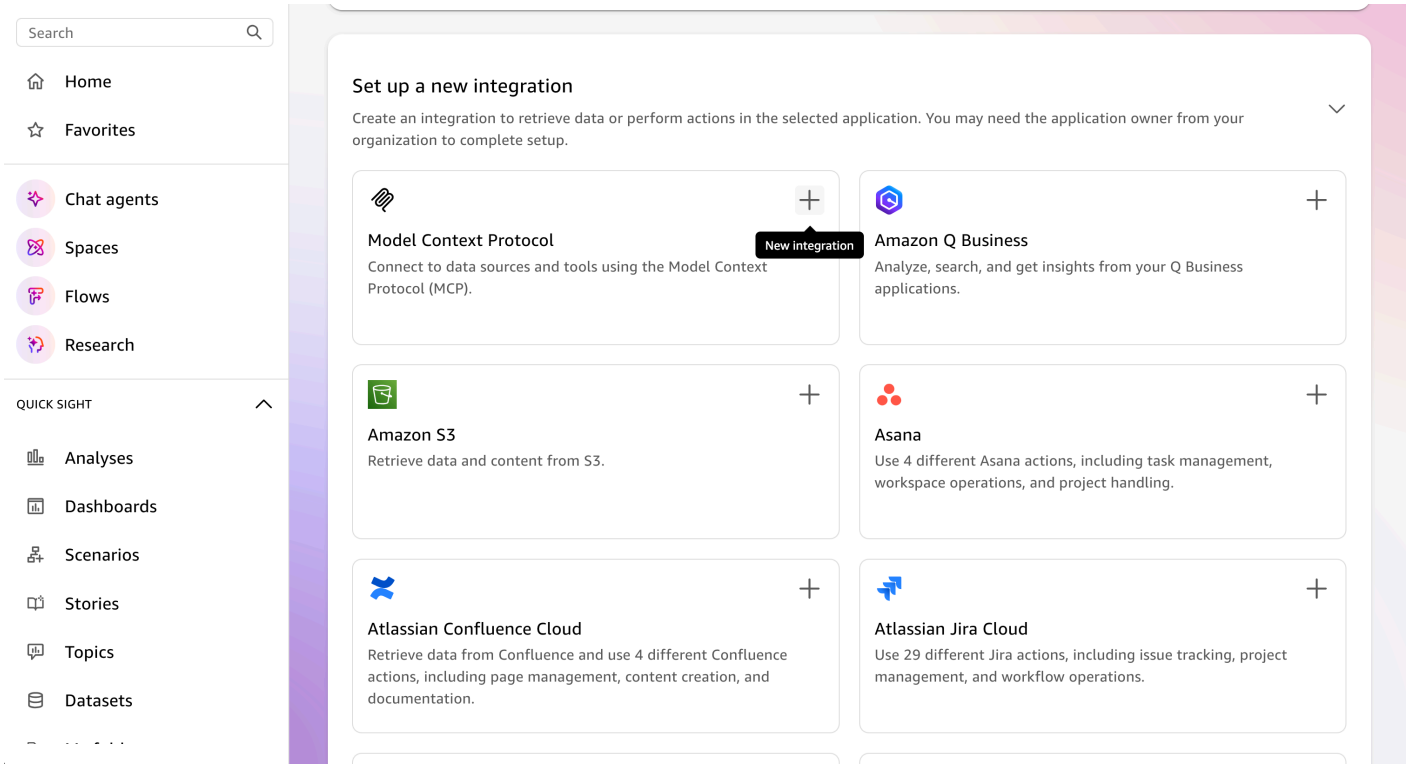


10. Cree la URL del punto final del token `/oauth2/token` agregándola al final del dominio.

Pasos de configuración

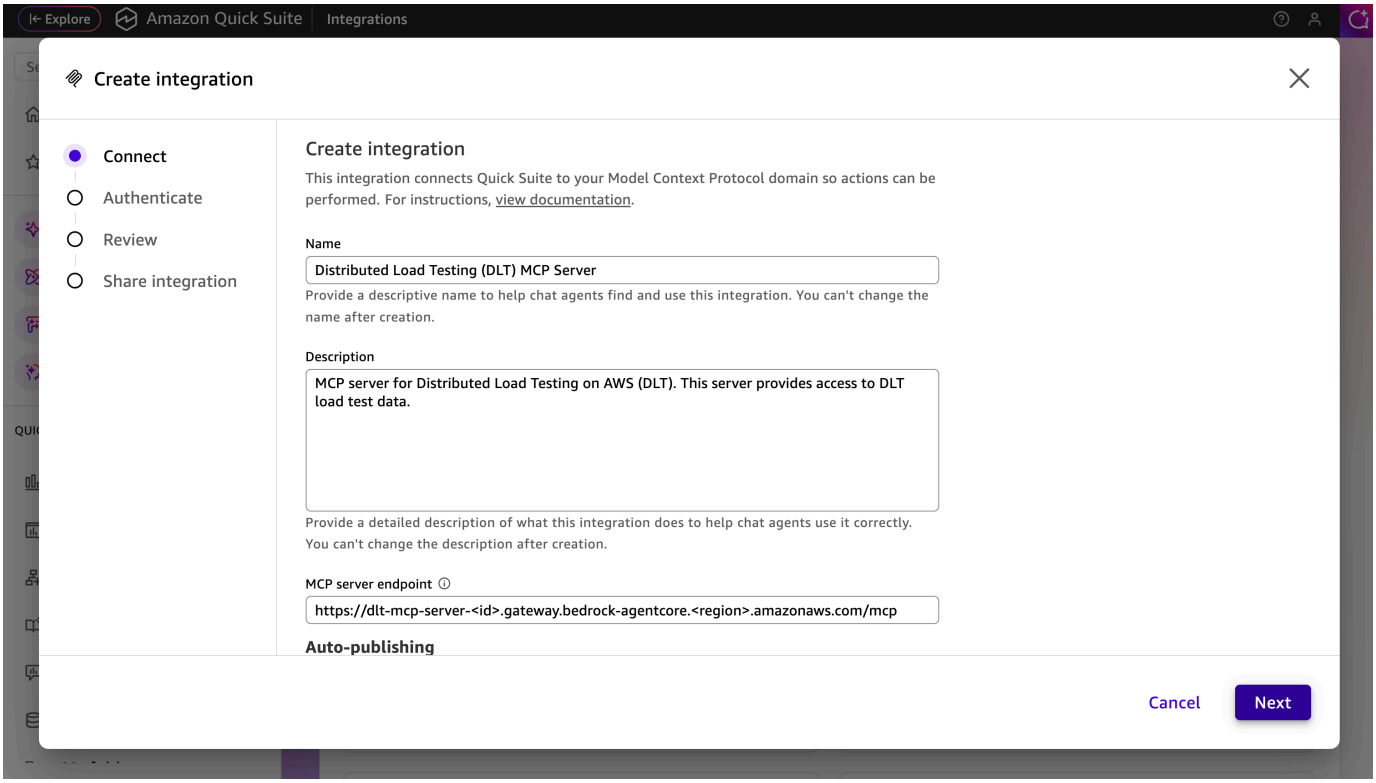
1. En Amazon Q Suite, cree un agente nuevo o seleccione uno existente.
2. Agregue un mensaje de agente que describa cómo interactuar con el servidor MCP de DLT.
3. Agregue una nueva acción y seleccione la acción del servidor MCP.



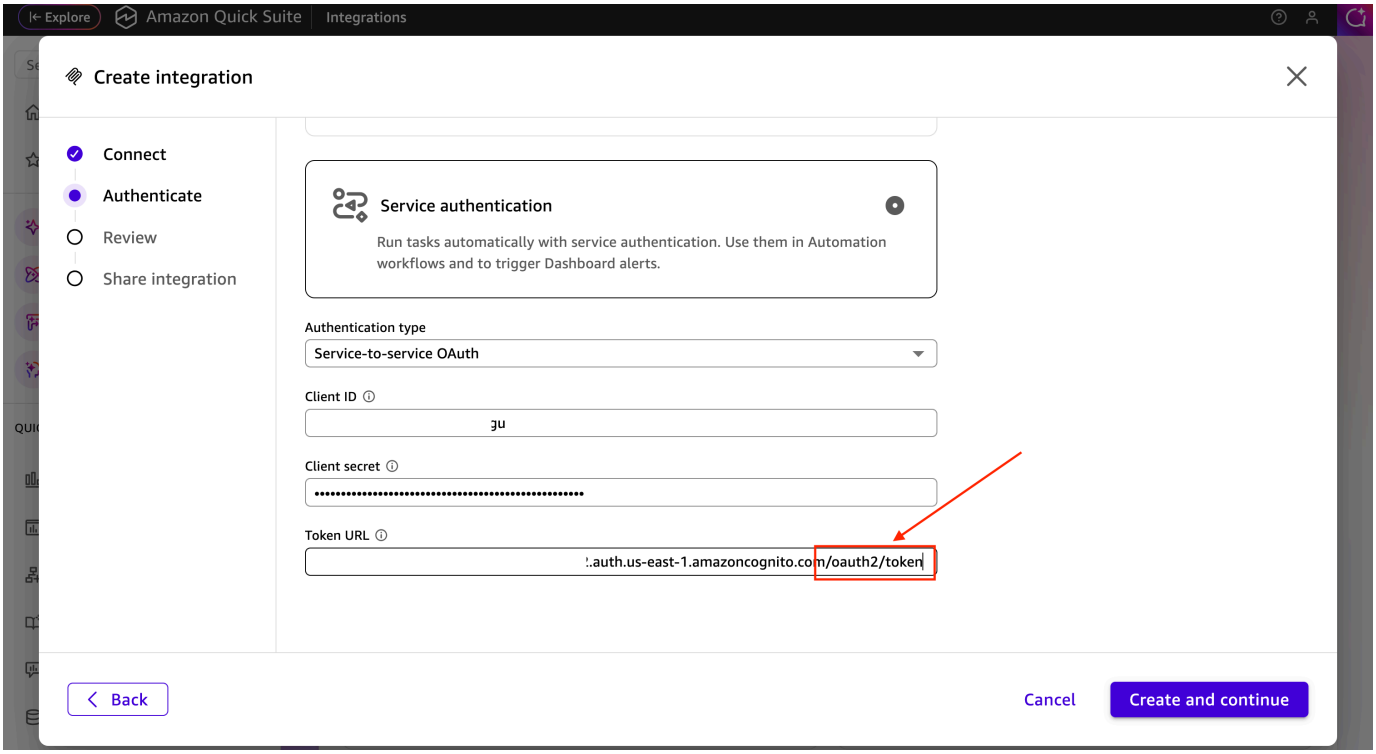


4. Configure los detalles del servidor MCP:

- URL del servidor MCP: su punto final MCP de DLT



- Tipo de autenticación: autenticación basada en servicios
- Token Endpoint: la URL de su punto final de token de Cognito
- ID de cliente: el ID de cliente del cliente de la aplicación m2m
- Secreto de cliente: el secreto de cliente del cliente de la aplicación m2m



5. Guarde la configuración de acciones del servidor MCP.
6. Añada la nueva acción del servidor MCP a su agente.

Inicie y pruebe el agente

1. Inicie el agente en Amazon Q Suite.
2. Inicie una conversación con el agente mediante instrucciones en lenguaje natural.
3. El agente utilizará las herramientas del MCP para recuperar y analizar los datos de las pruebas de carga.

Ejemplos de peticiones

Los siguientes ejemplos muestran cómo interactuar con su asistente de IA para analizar los datos de las pruebas de carga a través de la interfaz MCP. Personalice la prueba IDs, los intervalos de fechas y los criterios para que se ajusten a sus necesidades específicas de prueba.

Para obtener información detallada sobre las herramientas MCP disponibles y sus parámetros, consulte las especificaciones de las herramientas MCP en la Guía para desarrolladores.

Consulta sencilla de resultados de pruebas

La interacción en lenguaje natural con el servidor MCP puede ser tan simple `Show me the load tests that have completed in the last 24 hours with their associated completion status` o más descriptiva, como

```
Use list_scenarios to find my load tests. Then use get_latest_test_run to show me the
 basic execution data and performance metrics for the most recent test. If the results
 look concerning, also get the detailed performance metrics using get_test_run.
```

Análisis de rendimiento interactivo con divulgación progresiva

```
I need to analyze my load test performance, but I'm not sure which specific tests to
 focus on. Please help me by:

1. First, use list_scenarios to show me available test scenarios
2. Ask me which tests I want to analyze based on the list you show me
3. For my selected tests, use list_test_runs to get the test run history
4. Then use get_test_run with the test_run_id to get detailed response times,
 throughput, and error rates
5. If I want to compare tests, use get_baseline_test_run to compare against the
 baseline
6. If there are any issues, use get_test_run_artifacts to help me understand what went
 wrong

Please guide me through this step by step, asking for clarification whenever you need
 more specific information.
```

Validación de la disponibilidad de producción

```
Help me validate if my API is ready for production deployment:

1. Use list_scenarios to find recent test scenarios
2. For the most recent test scenario, use get_latest_test_run to get basic execution
 data
3. Use get_test_run with that test_run_id to get detailed response times, error rates,
 and throughput
4. Use get_scenario_details with the test_id to show me what load patterns and
 endpoints were tested
5. If I have a baseline, use get_baseline_test_run to compare current results with the
 baseline
```

```
6. Provide a clear go/no-go recommendation based on the performance data
7. If there are any concerns, use get_test_run_artifacts to help identify potential
 issues

My SLA requirements are: response time under [X]ms, error rate under [Y]%.
```

Análisis de tendencias de rendimiento

```
Analyze the performance trend for my load tests over the past [TIME_PERIOD]:

1. Use list_scenarios to get all test scenarios
2. For each scenario, use list_test_runs with start_date and end_date to get tests from
 that period
3. Use get_test_run for the key test runs to get detailed metrics
4. Use get_baseline_test_run to compare against the baseline
5. Identify any significant changes in response times, error rates, or throughput
6. If you detect performance degradation, use get_test_run_artifacts on the problematic
 tests to help identify causes
7. Present the trend analysis in a clear format showing whether performance is
 improving, stable, or degrading

Focus on completed tests and limit results to [N] tests if there are too many.
```

Solución de problemas de pruebas falli

```
Help me troubleshoot my failed load tests:

1. Use list_scenarios to find test scenarios
2. For each scenario, use list_test_runs to find recent test runs
3. Use get_test_run with the test_run_id to get the basic execution data and failure
 information
4. Use get_test_run_artifacts to get detailed error messages and logs
5. Use get_scenario_details to understand what was being tested when it failed
6. If I have a similar test that passed, use get_baseline_test_run to identify
 differences
7. Summarize the causes of failure and suggest next steps for resolution

Show me the most recent [N] failed tests from the past [TIME_PERIOD].
```

Guía para desarrolladores

En esta sección se proporciona el código fuente de la solución y otras personalizaciones.

Código fuente

Visite nuestro GitHub repositorio para descargar las plantillas y los scripts de esta solución y compartir sus personalizaciones con otras personas.

Mantenimiento

Esta solución utiliza imágenes de Docker con versiones fijas que corresponden a cada versión de la solución. De forma predeterminada, las actualizaciones automáticas están deshabilitadas, lo que le da un control total sobre cuándo y qué actualizaciones de versión se aplican a su implementación. El equipo de soluciones de AWS utiliza el escaneo mejorado de Amazon ECR para detectar vulnerabilidades y exposiciones comunes (CVEs) en la imagen base y en los paquetes instalados. Cuando es posible, el equipo publica las imágenes parcheadas con la misma etiqueta de versión para resolverlas CVEs sin interrumpir la compatibilidad con la versión publicada de la solución.

Cuando las imágenes se parchean en la misma versión secundaria, la etiqueta estable se actualiza automáticamente y se crea una etiqueta de imagen adicional en ese formato. `<solution-version>_<date-of-fix>` Si se publica una versión principal o secundaria, debe realizar una actualización completa para obtener la última versión de la imagen, ya que la etiqueta estable se incrementa para que coincida con la versión de la solución.

Si opta por las actualizaciones automáticas durante la implementación, los cambios en la imagen, incluidos los parches de CVE y las correcciones de errores menores, se aplicarán automáticamente hasta la última versión secundaria correspondiente.

Versiones

De forma predeterminada, esta solución se implementa con las actualizaciones automáticas deshabilitadas. Esto significa que la versión de la imagen del contenedor está bloqueada en la versión específica que coincida con la versión de la solución implementada, lo que le proporciona un control total sobre las actualizaciones de la versión.

Si decide habilitar las actualizaciones automáticas seleccionando Sí durante la CloudFormation implementación, la solución recibirá automáticamente los parches de seguridad y las correcciones de errores menores que no afecten a la última versión secundaria coincidente. Por ejemplo, si implementa la versión 4.0.0 con las actualizaciones automáticas habilitadas, recibirá las actualizaciones hasta la versión 4.0.x, pero no la 4.1.0 o superior.

Para controlar manualmente la versión de la imagen del contenedor, puede editar la definición de la tarea para especificar una versión de imagen concreta utilizando el formato de versión etiquetada. Esto le permite fijar una versión de imagen específica independientemente de la configuración de actualizaciones automáticas.

Personalización de imágenes de contenedores

Esta solución utiliza un repositorio de imágenes público del Amazon Elastic Container Registry (Amazon ECR) administrado por AWS para almacenar la imagen que se utiliza para ejecutar las pruebas configuradas. Si desea personalizar la imagen del contenedor, puede reconstruirla e insertarla en un repositorio de imágenes ECR en su propia cuenta de AWS.

Si desea personalizar esta solución, puede usar la imagen de contenedor predeterminada o editar este contenedor para adaptarlo a sus necesidades. Si personaliza la solución, utilice el siguiente ejemplo de código para declarar las variables de entorno antes de crear la solución personalizada.

```
#!/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
```

Si decide personalizar la imagen del contenedor, puede alojarla en un repositorio de imágenes privado o en un repositorio de imágenes público en su cuenta de AWS. Los recursos de imagen se encuentran en el `deployment/ecr/distributed-load-testing-on-aws-load-tester` directorio, ubicado en la base de código.

Puede crear y enviar la imagen al destino del host.

- Para los repositorios e imágenes privados de Amazon ECR, consulte los repositorios privados e imágenes privadas de Amazon ECR en la Guía del usuario de Amazon ECR.
- Para ver los repositorios e imágenes públicos de Amazon ECR, consulte los repositorios públicos e imágenes públicas de Amazon ECR en la Guía del usuario público de Amazon ECR.

Una vez que haya creado su propia imagen, podrá declarar las siguientes variables de entorno antes de crear su solución personalizada.

```
#!/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, v3.4.0
```

El siguiente ejemplo muestra el archivo contenedor.

```
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==78.1.1 h11 urllib3==2.2.2 && \
    $PIP_INSTALL --upgrade bzt
COPY ./.bzt-rc /root/.bzt-rc
RUN chmod 755 /root/.bzt-rc

# 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,robot,newman
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/ecslistener.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* && \
    rm -rf /usr/local/lib/python3.11/site-packages/setuptools-65.5.0.dist-info && \
    rm -rf /usr/local/lib/python3.11/site-packages/urllib3-1.26.17.dist-info

# 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"]
```

Además de un archivo contenedor, el directorio contiene el siguiente script bash que descarga la configuración de prueba de Amazon S3 antes de ejecutar el Taurus/Blazemeter programa.

```
#!/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=$(grep -oE '[a-f0-9]{32}' /proc/self/cgroup | 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 $JMETER_LIB_PATH
  elif [ "$TEST_TYPE" == "k6" ]; then
    curl --output /tmp/artifacts/k6.rpm https://dl.k6.io/rpm/x86_64/k6-v0.58.0-
amd64.rpm
    rpm -ivh /tmp/artifacts/k6.rpm
    dnf install -y k6
    rm -rf /tmp/artifacts/k6.rpm
    EXT="js"
    KPI_EXT="csv"
    TYPE_NAME="K6"
  elif [ "$TEST_TYPE" == "locust" ]; then
    EXT="py"
    TYPE_NAME="Locust"
```

```
  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

    # If zip and locust, make sure to pick locustfile
    if [ "$TEST_TYPE" != "locust" ]; then
      TEST_SCRIPT=$(find . -name "*.${EXT}" | head -n 1)
    else
      TEST_SCRIPT=$(find . -name "locustfile.py" | head -n 1)
    fi
    # 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

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
```

Además del Dockerfile y el script bash, en el directorio también se incluyen dos scripts de Python. Cada tarea ejecuta un script de Python desde el script bash. Las tareas de trabajo ejecutan el `ecslistener.py` script, mientras que la tarea principal ejecutará el `ecscontroller.py` script. El `ecslistener.py` script crea un conector en el puerto 50000 y espera un mensaje. El

`ecscontroller.py` script se conecta al socket y envía el mensaje de inicio de la prueba a las tareas del trabajador, lo que permite que se inicien simultáneamente.

API de pruebas de carga distribuida

Esta solución de pruebas de carga le ayuda a exponer los datos de los resultados de las pruebas de forma segura. La API actúa como una «puerta principal» para acceder a los datos de prueba almacenados en Amazon DynamoDB. También puede utilizarla APIs para acceder a cualquier funcionalidad ampliada que incorpore a la solución.

Esta solución utiliza un grupo de usuarios de Amazon Cognito integrado con Amazon API Gateway para su identificación y autorización. Cuando se utiliza un grupo de usuarios con la API, los clientes solo pueden llamar a los métodos activados por el grupo de usuarios después de proporcionar un token de identidad válido.

Para obtener más información sobre la ejecución de pruebas directamente a través de la API, consulte Firmar solicitudes en la documentación de referencia de la API REST de Amazon API Gateway.

Las siguientes operaciones están disponibles en la API de la solución.

> **Note**
>
> Para obtener más información `testScenario` y otros parámetros, consulte los escenarios y el ejemplo de carga útil en el GitHub repositorio.

Información de pila

- OBTENGA /stack-info

Escenarios

- GET /scenarios
- POST/escenarios
- OPCIONES/escenarios
- GET /scenarios/ {testID}
- POST /escenarios/ {testID}

- ELIMINAR /scenarios/ {testID}
- OPCIONES /scenarios/ {testID}

Ejecuciones de prueba

- GET /scenarios/ {testID} /testruns
- OBTENGA /scenarios/ {testID} /testruns/ {} testRunId
- ELIMINAR /scenarios/ {testID} /testruns/ {} testRunId

Referencia

- OBTENGA /scenarios/ {testID} /baseline
- PUT /scenarios/ {testID} /baseline
- ELIMINE /scenarios/ {testID} /baseline

Tareas

- GET /tasks
- OPCIONES /tareas

Regiones

- GET /regiones
- OPCIONES/regiones

OBTENGA /stack-info

Description (Descripción)

La `GET /stack-info` operación recupera información sobre la pila implementada, incluida la hora de creación, la región y la versión. El front-end utiliza este punto final.

Respuesta

200 - ¡Éxito

Name	Description (Descripción)
`created_time`	Marca de tiempo ISO 8601 cuando se creó la pila (por ejemplo,) `2025-09-09T19:40:22Z`
`region`	Región de AWS en la que se implementa la pila (por ejemplo,`us-east-1`)
`version`	Versión de la solución implementada (por ejemplo,`v4.0.0`)

Respuestas de error

- `403`- Prohibido: permisos insuficientes para acceder a la información de la pila
- `404`- No se ha encontrado: la información de la pila no está disponible
- `500`- Error interno del servidor

GET/scenarios

Description (Descripción)

La `GET /scenarios` operación le permite recuperar una lista de escenarios de prueba.

Respuesta

Name	Description (Descripción)
`data`	Una lista de escenarios que incluye el ID, el nombre, la descripción, el estado, el tiempo de ejecución, las etiquetas, el total de ejecuciones y la última ejecución de cada prueba

POST/escenarios

Description (Descripción)

La `POST /scenarios` operación le permite crear o programar un escenario de prueba.

Cuerpo de la solicitud

Name	Description (Descripción)
`testName`	El nombre de la prueba
`testDescription`	La descripción de la prueba
`testTaskConfigs`	Un objeto que especifica `concurrency` (el número de ejecuciones paralelas), `taskCount` (el número de tareas necesarias para ejecutar una prueba) y `region` para el escenario
`testScenario`	La definición de la prueba incluye la simultaneidad, la hora de la prueba, el anfitrión y el método de la prueba
`testType`	El tipo de prueba (por ejemplo,`simple`,`jmeter`)
`fileType`	El tipo de archivo de carga (por ejemplo,`none`,`script`,`zip`)
`tags`	Un conjunto de cadenas para categorizar las pruebas. Campo opcional con una longitud máxima de 5 (por ejemplo,`["blue", "3.0", "critical"]`)
`scheduleDate`	La fecha en la que se realizará una prueba. Solo se proporciona si se programa una prueba (por ejemplo,`2021-02-28`)

Name	Description (Descripción)
`scheduleTime`	El tiempo necesario para ejecutar una prueba. Solo se proporciona si se programa una prueba (por ejemplo,`21:07`)
`scheduleStep`	El paso del proceso de programación. Solo se proporciona si se programa una prueba periódica. (Los pasos disponibles incluyen `create` `ystart`)
`cronvalue`	El valor cron para personalizar la programac ión periódica. Si se usa, omita ScheduleDate y ScheduleTime.
`cronExpiryDate`	Fecha obligatoria para que el cron caduque y no se ejecute indefinidamente.
`recurrence`	La recurrencia de una prueba programada. Solo se proporciona si se programa una prueba periódica (por ejemplo`daily`,,`weekly`,`biweekly`, o`monthly`)

Respuesta

Name	Description (Descripción)
`testId`	El identificador único de la prueba
`testName`	El nombre de la prueba
`status`	El estado de la prueba

OPCIONES/ESCENARIOS

Description (Descripción)

La `OPTIONS /scenarios` operación proporciona una respuesta a la solicitud con los encabezados de respuesta CORS correctos.

Respuesta

Name	Description (Descripción)
`testId`	El identificador único de la prueba
`testName`	El nombre de la prueba
`status`	El estado de la prueba

GET /scenarios/ {testID}

Description (Descripción)

La `GET /scenarios/{testId}` operación le permite recuperar los detalles de un escenario de prueba específico.

Parámetros de solicitud

`testId`

- El identificador único de la prueba

  Tipo: cadena

  Obligatorio: sí

`latest`

- Parámetro de consulta para devolver solo la última ejecución de la prueba. El valor predeterminado es `true`

  Tipo: Booleano

  Obligatorio: no

`history`

- Parámetro de consulta para incluir el historial de ejecución de las pruebas en la respuesta. El valor predeterminado es `true`. Configúrelo `false` en para excluir el historial

  Tipo: Booleano

  Obligatorio: no

Respuesta

Name	Description (Descripción)
`testId`	El identificador único de la prueba
`testName`	El nombre de la prueba
`testDescription`	La descripción de la prueba
`testType`	El tipo de prueba que se ejecuta (por ejemplo`simple`,`jmeter`)
`fileType`	El tipo de archivo que se carga (por ejemplo,`none`,`script`,`zip`)
`tags`	Un conjunto de cadenas para categorizar las pruebas
`status`	El estado de la prueba
`startTime`	La hora y la fecha en que se inició la última prueba
`endTime`	La hora y la fecha en que finalizó la última prueba
`testScenario`	La definición de la prueba incluye la simultane idad, la hora de la prueba, el anfitrión y el método de la prueba

Name	Description (Descripción)
`taskCount`	El número de tareas necesarias para ejecutar la prueba
`taskIds`	Una lista de tareas IDs para ejecutar las pruebas
`results`	Los resultados finales de la prueba
`history`	Una lista de los resultados finales de las pruebas anteriores (se excluyen cuando`history=false`)
`totalRuns`	El número total de pruebas realizadas en este escenario
`lastRun`	La marca de tiempo de la última ejecución de la prueba
`errorReason`	Un mensaje de error que se genera cuando se produce un error
`nextRun`	La próxima ejecución programada (por ejemplo,`2017-04-22 17:18:00`)
`scheduleRecurrence`	La recurrencia de la prueba (por ejemplo,`daily`,`weekly`,`biweekly`,`monthly`)

POST /escenarios/ {testID}

Description (Descripción)

La `POST /scenarios/{testId}` operación le permite cancelar un escenario de prueba específico.

Parámetro de solicitud

`testId`

- El identificador único de la prueba

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`status`	El estado de la prueba

ELIMINAR /scenarios/ {testID}

Description (Descripción)

La `DELETE /scenarios/{testId}` operación le permite eliminar todos los datos relacionados con un escenario de prueba específico.

Parámetro de solicitud

`testId`

- El identificador único de la prueba

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`status`	El estado de la prueba

OPCIONES /escenarios/ {testID}

Description (Descripción)

La `OPTIONS /scenarios/{testId}` operación proporciona una respuesta a la solicitud con los encabezados de respuesta CORS correctos.

Respuesta

Name	Description (Descripción)
`testId`	El identificador único de la prueba
`testName`	El nombre de la prueba
`testDescription`	La descripción de la prueba
`testType`	El tipo de prueba que se ejecuta (por ejemplo`simple`,`jmeter`)
`fileType`	El tipo de archivo que se carga (por ejemplo,`none`,`script`,`zip`)
`status`	El estado de la prueba
`startTime`	La hora y la fecha en que se inició la última prueba
`endTime`	La hora y la fecha en que finalizó la última prueba
`testScenario`	La definición de la prueba incluye la simultaneidad, la hora de la prueba, el anfitrión y el método de la prueba
`taskCount`	El número de tareas necesarias para ejecutar la prueba
`taskIds`	Una lista de tareas IDs para ejecutar las pruebas

Name	Description (Descripción)
`results`	Los resultados finales de la prueba
`history`	Una lista de los resultados finales de las pruebas anteriores
`errorReason`	Un mensaje de error que se genera cuando se produce un error

GET /scenarios/ {testID} /testruns

Description (Descripción)

La `GET /scenarios/{testId}/testruns` operación recupera la ejecución de la prueba para un escenario de prueba específico, IDs filtrada opcionalmente por rango de tiempo. Cuando`latest=true`, devuelve solo la ejecución de prueba más reciente.

Parámetros de solicitud

`testId`

- El ID del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

`latest`

- Devuelve solo el ID de ejecución de la prueba más reciente

  Tipo: Booleano

  Valor predeterminado: `false`

  Obligatorio: no

`start_timestamp`

- Marca de tiempo ISO 8601 desde la que filtrar las pruebas (incluida). Por ejemplo, `2024-01-01T00:00:00Z`

Tipo: cadena (formato de fecha y hora)

Obligatorio: no

`end_timestamp`

- Marca de tiempo ISO 8601 para filtrar las pruebas realizadas (inclusive). Por ejemplo, `2024-12-31T23:59:59Z`

  Tipo: cadena (formato de fecha y hora)

  Obligatorio: no

`limit`

- Número máximo de ejecuciones de prueba que se devolverán (se omite cuando) `latest=true`

  Tipo: entero (mínimo: 1, máximo: 100)

  Valor predeterminado: `20`

  Obligatorio: no

`next_token`

- Símbolo de paginación de la respuesta anterior para obtener la página siguiente

  Tipo: cadena

  Requerido: no

Respuesta

200 - ¡Éxito

Name	Description (Descripción)
`testRuns`	Matriz de objetos de prueba, cada uno de los cuales contiene `testRunId` (cadena) y `startTime` (fecha y hora ISO 8601)

Name	Description (Descripción)
`pagination`	Objeto que contiene `limit` (entero) y `next_token` (cadena o nulo). El token es nulo si no hay más resultados

Respuestas de error

- `400`- El formato o los parámetros de la marca de tiempo no son válidos
- `404`- No se encontró el escenario de prueba
- `500`- Error interno del servidor

Ejemplo de uso

- Solo la última prueba realizada: `GET /scenarios/test123/testruns?latest=true`
- Lo último dentro del intervalo de tiempo: `GET /scenarios/test123/testruns?latest=true&start_timestamp=2024-01-01T00:00:00Z`
- Solicitud de página siguiente: `GET /scenarios/test123/testruns?limit=20&next_token=eyJ0ZXN0SWQiOiJzZVFVeTEyTEtMIiwic3RhcnRUaW1lIjoiMjAyNC0wMS`

GET /scenarios/ {testID} /testruns/ {} testRunId

Description (Descripción)

La `GET /scenarios/{testId}/testruns/{testRunId}` operación recupera los resultados y las métricas completos de una ejecución de prueba específica. Si lo desea, omita los resultados del historial `history=false` para obtener una respuesta más rápida.

Parámetros de solicitud

`testId`

- El ID del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

`testRunId`

- El ID específico de la ejecución de la prueba

  Tipo: cadena

  Obligatorio: sí

`history`

- Incluya una matriz de historial en la respuesta. Configúrelo `false` en para omitir el historial para una respuesta más rápida

  Tipo: Booleano

  Valor predeterminado: `true`

  Obligatorio: no

Respuesta

200 - ¡Éxito

Name	Description (Descripción)
`testId`	El identificador único de la prueba (por ejemplo,`seQUy12LKL`)
`testRunId`	El ID específico de la ejecución de la prueba (por ejemplo,`2DEwHItEne`)
`testDescription`	Descripción de la prueba de carga
`testType`	El tipo de prueba (por ejemplo,`simple`,`jmeter`)
`status`	El estado de la ejecución de la prueba: `completerunning`,`failed`, o `cancelled`
`startTime`	La hora y la fecha en que se inició la prueba (por ejemplo,`2025-09-09 21:01:00`)

Name	Description (Descripción)
`endTime`	La hora y la fecha en que finalizó la prueba (por ejemplo,`2025-09-09 21:18:29`)
`succPercent`	Porcentaje de éxito (por ejemplo,`100.00`)
`testTaskConfigs`	Matriz de objetos de configuración de tareas que contiene `regiontaskCount` , y `concurrency`
`completeTasks`	Asignación de regiones de objetos a recuentos de tareas completadas
`results`	Objeto que contiene métricas detalladas, como `avg_lt` (latencia media), percentiles (`p0_0`,`p50_0`,`p90_0`,,`p95_0`,`p99_0`,`p100_0`)`p99_9`, `avg_rt` (tiempo medio de respuesta), `avg_ct` (tiempo medio de conexión), `stdev_rt` (tiempo de respuesta a la desviación estándar) ,, `concurrency throughput` , `succ` (recuento de éxitos), `fail` (recuento de fallos),,`bytes`, `testDuration metricS3L ocation` , `rc` (matriz de códigos de respuesta) y matriz `labels`
`testScenario`	Objeto que contiene la configuración de prueba con `execution reporting` , y propiedades `scenarios`
`history`	Matriz de resultados históricos de las pruebas (se excluyen cuando`history=false`)

Respuestas de error

- `400`- ID de prueba no válido o testRunId
- `404`- No se encontró la ejecución de la prueba

- `500`- Error interno del servidor

ELIMINAR /scenarios/ {testID} /testruns/ {} testRunId

Description (Descripción)

La `DELETE /scenarios/{testId}/testruns/{testRunId}` operación elimina todos los datos y artefactos relacionados con una ejecución de prueba específica. Los datos de la ejecución de la prueba se eliminan de DynamoDB, mientras que los datos de prueba reales de S3 permanecen inalterados.

Parámetros de solicitud

`testId`

- El ID del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

`testRunId`

- El ID específico de la ejecución de la prueba que se va a eliminar

  Tipo: cadena

  Obligatorio: sí

Respuesta

204 - Éxito

La ejecución de la prueba se ha eliminado correctamente (no se ha devuelto ningún contenido)

Respuestas de error

- `400`- ID de prueba no válido o testRunId
- `403`- Prohibido: permisos insuficientes para eliminar la ejecución de la prueba
- `404`- No se encontró la ejecución de la prueba
- `409`- Conflicto: la ejecución de la prueba se está ejecutando actualmente y no se puede eliminar

- `500`- Error interno del servidor

GET /scenarios/ {testID} /baseline

Description (Descripción)

La `GET /scenarios/{testId}/baseline` operación recupera el resultado de la prueba de referencia designado para un escenario. Devuelve el identificador de ejecución de la prueba de referencia o los resultados de referencia completos en función del `data` parámetro.

Parámetros de solicitud

`testId`

- El ID del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

`data`

- Devuelve los datos completos de la ejecución de la prueba de referencia si`true`, de lo contrario, solo testRunId

  Tipo: Booleano

  Valor predeterminado: `false`

  Obligatorio: no

Respuesta

200 - Éxito

Cuándo `data=false` (predeterminado):

Name	Description (Descripción)
`testId`	El ID del escenario de prueba (por ejemplo,`seQUy12LKL`)

Name	Description (Descripción)
`baselineTestRunId`	El ID de ejecución de la prueba de referencia (por ejemplo,`2DEwHItEne`)

Cuándo`data=true`:

Name	Description (Descripción)
`testId`	El ID del escenario de prueba (por ejemplo,`seQUy12LKL`)
`baselineTestRunId`	El ID de ejecución de la prueba de referencia (por ejemplo,`2DEwHItEne`)
`baselineData`	Objeto completo con los resultados de la ejecución de la prueba (con la misma estructur a que`GET /scenarios/{testId}/ testruns/{testRunId}`)

Respuestas de error

- `400`- Parámetro TestID no válido
- `404`- No se encontró el escenario de prueba o no se estableció una línea base
- `500`- Error interno del servidor

PUT /scenarios/ {testID} /baseline

Description (Descripción)

La `PUT /scenarios/{testId}/baseline` operación designa una ejecución de prueba específica como referencia para la comparación del rendimiento. Solo se puede establecer una línea base por escenario.

Parámetros de solicitud

`testId`

- El ID del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

Cuerpo de la solicitud

Name	Description (Descripción)
`testRunId`	El ID de ejecución de la prueba que se va a establecer como referencia (por ejemplo,`2DEwHItEne`)

Respuesta

200 - ¿Éxito

Name	Description (Descripción)
`message`	Mensaje de confirmación (por ejemplo,`Baseline set successfully`)
`testId`	El ID del escenario de prueba (por ejemplo,`seQUy12LKL`)
`baselineTestRunId`	El ID de ejecución de la prueba de referencia que se estableció (por ejemplo,`2DEwHItEne`)

Respuestas de error

- `400`- ID de prueba no válido o testRunId
- `404`- No se encontró el escenario de prueba o la ejecución de la prueba

- `409`- Conflicto: la ejecución de la prueba no se puede establecer como línea base (por ejemplo, una prueba fallida)
- `500`- Error interno del servidor

ELIMINAR /scenarios/ {testID} /baseline

Description (Descripción)

La `DELETE /scenarios/{testId}/baseline` operación borra el valor de referencia de un escenario configurándolo en una cadena vacía.

Parámetros de solicitud

`testId`

- El ID del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

Respuesta

204 - ¿Éxito

La línea base se borró correctamente (no se devolvió contenido)

Respuestas de error

- `400`- ID de prueba no válido
- `500`- Error interno del servidor

GET /tasks

Description (Descripción)

La `GET /tasks` operación le permite recuperar una lista de las tareas en ejecución de Amazon Elastic Container Service (Amazon ECS).

Respuesta

Name	Description (Descripción)
`tasks`	Una lista de tareas IDs para ejecutar las pruebas

OPCIONES /tareas

Description (Descripción)

La operación de `OPTIONS /tasks` tareas proporciona una respuesta a la solicitud con los encabezados de respuesta CORS correctos.

Respuesta

Name	Description (Descripción)
`taskIds`	Una lista de tareas IDs para ejecutar las pruebas

GET /regions

Description (Descripción)

La `GET /regions` operación le permite recuperar la información de recursos regionales necesaria para ejecutar una prueba en esa región.

Respuesta

Name	Description (Descripción)
`testId`	El ID de la región
`ecsCloudWatchLogGroup`	El nombre del grupo de CloudWatch registros de Amazon para las tareas de Amazon Fargate en la región

Name	Description (Descripción)
`region`	La región en la que se encuentran los recursos de la tabla
`subnetA`	El ID de una de las subredes de la región
`subnetB`	El ID de una de las subredes de la región
`taskCluster`	El nombre del clúster de AWS Fargate en la región
`taskDefinition`	El ARN de la definición de tareas en la Región
`taskImage`	El nombre de la imagen de la tarea en la región
`taskSecurityGroup`	El ID del grupo de seguridad de la región

OPCIONES/regiones

Description (Descripción)

La `OPTIONS /regions` operación proporciona una respuesta a la solicitud con los encabezados de respuesta CORS correctos.

Respuesta

Name	Description (Descripción)
`testId`	El ID de la región
`ecsCloudWatchLogGroup`	El nombre del grupo de CloudWatch registros de Amazon para las tareas de Amazon Fargate en la región
`region`	La región en la que se encuentran los recursos de la tabla
`subnetA`	El ID de una de las subredes de la región

Name	Description (Descripción)
`subnetB`	El ID de una de las subredes de la región
`taskCluster`	El nombre del clúster de AWS Fargate en la región
`taskDefinition`	El ARN de la definición de tareas en la Región
`taskImage`	El nombre de la imagen de la tarea en la región
`taskSecurityGroup`	El ID del grupo de seguridad de la región

Aumente los recursos del contenedor

Para aumentar el número de usuarios virtuales simultáneos (simultaneidad) que sus pruebas de carga pueden simular, debe aumentar los recursos de CPU y memoria asignados a cada tarea de Amazon ECS. Esto implica crear una nueva revisión de la definición de tareas con límites de recursos más altos y, a continuación, actualizar la configuración de DynamoDB de la solución para usar la nueva definición de tarea en futuras ejecuciones de prueba.

Cree una nueva revisión de la definición de tareas

Siga estos pasos para crear una nueva definición de tarea con más recursos de CPU y memoria:

1. Inicie sesión en la consola de Amazon Elastic Container Service.
2. En el menú de navegación de la izquierda, seleccione Definiciones de tareas.
3. Seleccione la casilla de verificación situada junto a la definición de tarea que corresponda a esta solución. Por ejemplo, `[replaceable]` <stackName>`- EcsTaskDefinition -*<system-generated-random-Hash>*.
4. Elija Create new revision (Crear nueva revisión).
5. En la página Crear nueva revisión, lleve a cabo las siguientes acciones:
   a. En Tamaño de la tarea, modifique la memoria de la tarea y la CPU de la tarea a los valores que desee. Los valores más altos permiten más usuarios virtuales simultáneos por tarea.
   b. En Definiciones de contenedores, revise los límites de memoria dura y blanda. Si este límite es inferior a la memoria deseada, elija el contenedor.

c. En el cuadro de diálogo Editar contenedor, vaya a Límites de memoria y actualice el límite estricto para que coincida o sea inferior a la asignación de memoria de la tarea.

d. Elija Actualizar.

6. En la página Crear nueva revisión, selecciona Crear.
7. Una vez que la definición de tarea se haya creado correctamente, registre el ARN completo de la definición de tarea, incluido el número de versión. Por ejemplo: `[replaceable]` <stackName>`- EcsTaskDefinition -*<system-generated-random-Hash>*: [reemplazable]`<system-generated-versionNumber>`.

Actualizar la tabla de DynamoDB

Tras crear la nueva revisión de la definición de tareas, debe actualizar la tabla de DynamoDB de la solución para que en futuras ejecuciones de pruebas se utilice la nueva definición de tarea. Repita estos pasos para cada región de AWS en la que desee utilizar la definición de tarea actualizada:

1. Navegue hasta la consola de DynamoDB.
2. En el panel de navegación izquierdo, seleccione Explorar los elementos de las tablas.
3. Seleccione la tabla de `scenarios-table` DynamoDB asociada a esta solución. Por ejemplo, `[replaceable]` <stackName>`- DLTTest RunnerStorage DLTScenarios Tabla-. *<system-generated-random-Hash>*
4. Seleccione el elemento que corresponda a la región en la que creó la nueva revisión de la definición de tareas. Por ejemplo, `region-[replaceable]` <region-name>`.
5. En el editor de elementos, localice el atributo taskDefinition y actualice su valor con el ARN de definición de tarea completo que registró en la sección anterior (incluido el número de versión).
6. Seleccione Save changes (Guardar cambios).

> **Note**
>
> La definición de tarea actualizada solo se utilizará para nuevas ejecuciones de prueba. Todas las pruebas que se estén ejecutando o estén programadas en ese momento seguirán utilizando la definición de tarea anterior.

Especificación de herramientas MCP

La solución de pruebas de carga distribuidas presenta un conjunto de herramientas de MCP que permiten a los agentes de IA interactuar con los escenarios y los resultados de las pruebas. Estas herramientas proporcionan capacidades abstractas de alto nivel que se adaptan a la forma en que los agentes de IA procesan la información, lo que les permite centrarse en el análisis y la información en lugar de centrarse en los contratos de API detallados.

> **Note**
>
> Todas las herramientas de MCP proporcionan acceso de solo lectura a los datos de la solución. No se admiten modificaciones en los escenarios o configuraciones de prueba a través de la interfaz MCP.

list_scenarios

Description (Descripción)

La `list_scenarios` herramienta recupera una lista de todos los escenarios de prueba disponibles con metadatos básicos.

Punto de conexión

`GET /scenarios`

Parameters

Ninguno

Respuesta

Name	Description (Descripción)
`testId`	Identificador único para el escenario de prueba
`testName`	Nombre del escenario de prueba
`status`	Estado actual del escenario de prueba

Name	Description (Descripción)
`startTime`	Cuándo se creó la prueba o se ejecutó por última vez
`testDescription`	Descripción del escenario de prueba

get_scenario_details

Description (Descripción)

La `get_scenario_details` herramienta recupera la configuración de la prueba y la ejecución más reciente de la prueba para un único escenario de prueba.

Punto de conexión

`GET /scenarios/<test_id>?history=false&results=false`

Parámetro de solicitud

`test_id`

- El identificador único del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`testTaskConfigs`	Configuración de tareas para cada región
`testScenario`	Pruebe la definición y los parámetros
`status`	Estado actual de la prueba
`startTime`	Fecha y hora de inicio de la prueba

Name	Description (Descripción)
`endTime`	Marca de tiempo de finalización de la prueba (si se ha completado)

list_test_runs

Description (Descripción)

La `list_test_runs` herramienta recupera una lista de las pruebas ejecutadas para un escenario de prueba específico, ordenadas de las más recientes a las más antiguas. Devuelve un máximo de 30 resultados.

Punto de conexión

```
GET /scenarios/<testid>/testruns/?limit=<limit>
```

o

```
GET /scenarios/<testid>/testruns/?
limit=30&start_date=<start_date>&end_date=<end_date>
```

Parámetros de solicitud

`test_id`

- El identificador único del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

`limit`

- Número máximo de ejecuciones de prueba que se devolverán

  Tipo: entero

  Predeterminado: 20

  Máximo: 30

  Obligatorio: no

`start_date`

- La marca de tiempo ISO 8601 para filtrar las ejecuciones a partir de una fecha específica

  Tipo: cadena (formato de fecha y hora)

  Obligatorio: no

`end_date`

- La marca de tiempo ISO 8601 para filtrar se ejecuta hasta una fecha específica

  Tipo: cadena (formato de fecha y hora)

  Obligatorio: no

Respuesta

Name	Description (Descripción)
`testRuns`	Matriz de resúmenes de las pruebas con métricas de rendimiento y percentiles para cada ejecución

get_test_run

Description (Descripción)

La `get_test_run` herramienta recupera los resultados detallados de una sola prueba con desgloses regionales y de puntos finales.

Punto de conexión

`GET /scenarios/<testid>/testruns/<testrunid>`

Parámetros de solicitud

`test_id`

- El identificador único del escenario de prueba

  Tipo: cadena

Obligatorio: sí

`test_run_id`

- El identificador único de la ejecución de la prueba específica

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`results`	Datos completos de la ejecución de la prueba, que incluyen el desglose de los resultados regionales, las métricas específicas del punto final, los percentiles de rendimiento (p50, p90, p95, p99), los recuentos de éxitos y fracasos, los tiempos de respuesta y la latencia, y la configuración de la prueba utilizada para la ejecución

get_latest_test_run

Description (Descripción)

La `get_latest_test_run` herramienta recupera la ejecución de prueba más reciente para un escenario de prueba específico.

Punto de conexión

`GET /scenarios/<testid>/testruns/?limit=1`

> **Note**
>
> Los resultados se ordenan por tiempo mediante un índice secundario global (GSI), lo que garantiza que se devuelva la prueba más reciente.

Parámetro de solicitud

`test_id`

- El identificador único del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`results`	Los datos más recientes de la ejecución de la prueba tienen el mismo formato que `get_test_run`

get_baseline_test_run

Description (Descripción)

La `get_baseline_test_run` herramienta recupera la ejecución de la prueba de referencia para un escenario de prueba específico. La línea base se utiliza para comparar el rendimiento.

Punto de conexión

`GET /scenarios/<test_id>/baseline`

Parámetro de solicitud

`test_id`

- El identificador único del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`baselineData`	Datos de la ejecución de la prueba de referenci a con fines de comparación, incluidas todas las métricas y la configuración de la ejecución de referencia designada

get_test_run_artifacts

Description (Descripción)

La `get_test_run_artifacts` herramienta recupera la información del bucket de Amazon S3 para acceder a los artefactos de las pruebas, incluidos los registros, los archivos de errores y los resultados.

Punto de conexión

```
GET /scenarios/<testid>/testruns/<testrunid>
```

Parámetros de solicitud

`test_id`

- El identificador único del escenario de prueba

  Tipo: cadena

  Obligatorio: sí

`test_run_id`

- El identificador único de la ejecución de la prueba específica

  Tipo: cadena

  Obligatorio: sí

Respuesta

Name	Description (Descripción)
`bucketName`	Nombre del depósito de S3 donde se almacenan los artefactos
`testRunPath`	Prefijo de ruta para el almacenamiento de artefactos actual (versión 4.0+)
`testScenarioPath`	Prefijo de ruta para el almacenamiento de artefactos antiguos (anterior a la versión 4.0)

> **Note**
>
> Todas las herramientas de MCP aprovechan los puntos finales de las API existentes. No es necesario modificar el componente subyacente APIs para admitir la funcionalidad del MCP.

Referencia

Esta sección incluye información sobre la recopilación de datos, consejos sobre recursos relacionados y una lista de los desarrolladores que han contribuido a esta solución.

Recopilación de datos

Esta solución envía métricas operativas a AWS (los «datos») sobre el uso de esta solución. Utilizamos estos datos para comprender mejor cómo utilizan los clientes esta solución y los servicios y productos relacionados. La recopilación de estos datos por parte de AWS está sujeta al Aviso de privacidad de AWS.

Colaboradores

- Tom Nightingale
- Fernando Dingler
- Beomseok Lee
- George Lenz
- Erin McGill
- Dimitri López
- Kamyar Ziabari
- Bassem Wanis
- Garvit Singh
- Nikhil Reddy
- Simón Kroll
- Ahern Knox
- Ian Downard
- Owen Brady
- Jim Thario
- Thyag Ramachandran
- Yang Qin
- James Wang

Glosario

En este glosario se definen los acrónimos y abreviaturas que se utilizan en la Guía de implementación de Distributed Load Testing en AWS.

Protocolos y formatos técnicos

AGPL

Licencia pública general de Affero. Una licencia de software de código abierto utilizada por K6.

API

Interfaz de programación de aplicaciones. Un conjunto de protocolos y herramientas para crear aplicaciones de software y permitir la comunicación entre diferentes sistemas.

CLI

Interfaz de línea de comandos. Una interfaz basada en texto para interactuar con el software y los sistemas operativos.

NÚCLEOS

Intercambio de recursos entre orígenes. Función de seguridad que permite o restringe el acceso de las aplicaciones web que se ejecutan en un origen a los recursos de otro origen.

CSV

Valores separados por comas. Formato de archivo que se utiliza para almacenar datos tabulares en texto sin formato y que se suele utilizar para la exportación de datos.

gRPC

Llamada de procedimiento remoto gRPC. Un marco de código abierto de alto rendimiento para llamadas a procedimientos remotos.

HTTP

Protocolo de transferencia de hipertexto. El protocolo básico utilizado para transmitir datos en la World Wide Web.

HTTPS

HTTP seguro. Extensión de HTTP que utiliza el cifrado para una comunicación segura a través de una red.

JSON

JavaScript Notación de objetos. Un formato ligero de intercambio de datos que es fácil de leer y escribir para los humanos y fácil de analizar y generar para las máquinas.

JWT

Token web JSON. Un medio compacto y seguro para representar las reclamaciones que se van a transferir entre dos partes para su autenticación y autorización.

OAuth

Autorización abierta. Estándar abierto para la delegación de acceso que se suele utilizar para la autenticación y la autorización basadas en tokens.

REST

Transferencia de estado representativa. Un estilo arquitectónico para diseñar aplicaciones en red mediante comunicación sin estado y métodos HTTP estándar.

SSE

Eventos enviados por el servidor. Tecnología push de servidor que permite a un cliente recibir actualizaciones automáticas de un servidor a través de una conexión HTTP.

IU

Interfaz de usuario. Los elementos visuales y los controles a través de los cuales los usuarios interactúan con las aplicaciones de software.

URL

Localizador uniforme de recursos. La dirección utilizada para acceder a los recursos de Internet.

XML

Lenguaje de marcado extensible. Lenguaje de marcado que define las reglas para codificar documentos en un formato legible tanto por humanos como por máquinas.

Términos de pruebas y bases de datos

FTP

Protocolo de transferencia de archivos. Protocolo de red estándar que se utiliza para transferir archivos entre un cliente y un servidor.

GSI

Índice secundario global. Función de DynamoDB que permite consultar datos mediante una clave alternativa.

JDBC

Conectividad a bases de datos Java. Una API de Java para conectar y ejecutar consultas con bases de datos.

JMS

Servicio de mensajes de Java. Una API de Java para enviar mensajes entre dos o más clientes.

TPS

Transacciones por segundo. Medida del número de transacciones que un sistema puede procesar en un segundo.

Términos de AWS y del sistema

ARN

Nombre de recurso de Amazon. Un identificador único de los recursos de AWS que se utiliza para especificar los recursos de los servicios de AWS.

ISO

Organización Internacional de Normalización. Una organización no gubernamental independiente que desarrolla estándares internacionales. Se hace referencia en esta guía al formato de fecha y hora de la norma ISO 8601.

SLA

Acuerdo de nivel de servicio. Un compromiso entre un proveedor de servicios y un cliente que define el nivel de servicio esperado.

UUID

Identificador universal único. Número de 128 bits que se utiliza para identificar de forma exclusiva la información en los sistemas informáticos.

vCPU

Unidad central de procesamiento virtual. Procesador virtual asignado a una máquina o contenedor virtual, que representa una parte de la potencia de procesamiento de la CPU física.

Términos de las pruebas de carga

concurrency

El número de usuarios virtuales simultáneos por tarea. Este parámetro controla cuántos usuarios simulados genera cada tarea de Fargate durante una prueba de carga.

pila regional

Una CloudFormation pila implementada en una región de AWS para proporcionar una infraestructura de pruebas para las pruebas de carga multirregionales.

recuento de tareas

El número de contenedores Fargate (tareas) lanzados para ejecutar un escenario de prueba. La carga total generada es igual al recuento de tareas multiplicado por la simultaneidad.

escenario de prueba

Una prueba de carga configurada que incluye el tipo de prueba, los puntos finales de destino, el recuento de tareas, la simultaneidad, la duración y otros parámetros.

Revisiones

Visite ChangeLog.md en nuestro GitHub repositorio para realizar un seguimiento de las mejoras y correcciones específicas de cada versión.

Avisos

Es responsabilidad de los clientes realizar su propia evaluación independiente de la información que contiene este documento. Este documento: (a) tiene únicamente fines informativos, (b) representa las ofertas y prácticas de productos actuales de AWS, que están sujetas a cambios sin previo aviso, y (c) no crea ningún compromiso ni garantía por parte de AWS y sus filiales, proveedores o licenciantes. Los productos o servicios de AWS se proporcionan «tal cual» sin garantías, declaraciones ni condiciones de ningún tipo, ya sean expresas o implícitas. Las responsabilidades y obligaciones de AWS con sus clientes están reguladas por los acuerdos de AWS, y este documento no forma parte de ningún acuerdo entre AWS y sus clientes ni lo modifica.

Las pruebas de carga distribuidas en AWS se licencian según los términos de la versión 2.0 de la licencia Apache, disponible en The Apache Software Foundation.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.