

Guía de implementación

Pruebas de carga distribuidas en AWS



Pruebas de carga distribuidas en AWS: Guía de implementación

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Información general de la solución	1
Características	2
Ventajas	3
Casos de uso	4
Conceptos y definiciones	5
Información general de la arquitectura	7
Diagrama de arquitectura	7
Consideraciones sobre el diseño de AWS Well-Architected	8
Excelencia operativa	8
Seguridad	9
Fiabilidad	10
Eficiencia del rendimiento	10
Optimización de costos	10
Sostenibilidad	11
Detalles de la arquitectura	12
Parte delantera	12
API de pruebas de carga	12
Consola web	13
Backend	13
Canalización de imágenes de contenedores	13
Infraestructura de pruebas	14
Motor de pruebas de carga	14
Los servicios de AWS en esta solución	14
Cómo funcionan las pruebas de carga distribuidas en AWS	16
Consideraciones sobre el diseño	18
Aplicaciones compatibles	18
JMeter compatibilidad con scripts	18
Soporte para scripts K6	19
Compatible con el alfabeto Locust	19
Programar las pruebas	19
Pruebas simultáneas	20
Administración de usuarios	20
Implementación regional	21
Planificación de la implementación	22

Costo	22
Seguridad	23
Roles de IAM	24
Amazon CloudFront	24
Grupo de seguridad AWS Fargate	24
Prueba de stress de red	24
Restringir el acceso a la interfaz de usuario pública	25
Regiones de AWS admitidas	25
Cuotas	26
Cuotas para los servicios de AWS en esta solución	26
CloudFormation Cuotas de AWS	26
Cuotas de pruebas de carga	26
Pruebas simultáneas	20
Política de EC2 pruebas de Amazon	27
Política de pruebas CloudFront de carga de Amazon	27
Supervisión de la solución después de la implementación	28
Activación o configuración de CloudWatch alarmas	28
Implementación de la solución	29
Información general del proceso de implementación	29
CloudFormation Plantilla de AWS	29
Lanzar la pila	30
Despliegue multirregional	34
Actualización de la solución	37
Al actualizar desde versiones de DLT anteriores a la v3.2.6 a la v3.3.x y a la versión 3.3.x a la versión más reciente, se produce un error en la actualización de la pila	38
Solución de problemas	40
Resolución de problemas conocidos	40
Póngase en contacto con AWS Support.	42
Cree un caso	42
¿Cómo podemos ayudar?	43
Información adicional	43
Ayúdenos a resolver su caso más rápido	43
Resuelva ahora o póngase en contacto con nosotros	43
Desinstalar la solución	44
Uso de Consola de administración de AWS	44
Uso de la interfaz de línea de comandos de AWS	44

Eliminar los buckets de Amazon S3	44
Usa la solución	46
Resultados de la prueba	46
Flujo de trabajo para programar pruebas	47
Determine el número de usuarios	47
Datos en directo	48
Pruebe el flujo de trabajo de cancelación	49
Guía para desarrolladores	50
Código fuente	50
Mantenimiento	50
Versiones	50
Personalización de imágenes de contenedores	51
API de pruebas de carga distribuida	58
GET /scenarios	60
POST /escenarios	60
OPCIONES/ESCENARIOS	62
GET /scenarios/ {testID}	62
POST /escenarios/ {testID}	64
ELIMINAR /scenarios/ {testID}	65
OPCIONES /escenarios/ {testID}	65
GET /tasks	67
OPCIONES /tareas	67
GET /regions	67
OPCIONES/regiones	68
Aumente los recursos del contenedor	69
Cree una nueva revisión de la definición de tareas	69
Actualizar la tabla de DynamoDB	70
Referencia	71
Recopilación de datos anonimizados	71
Colaboradores	72
Revisiones	73
Avisos	74
.....	lxxv

Automatice las pruebas de sus aplicaciones de software a escala

Fecha de publicación: noviembre de 2019

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

Esta solución aprovecha [Amazon Elastic Container Service \(Amazon ECS\) en AWS](#) Fargate para implementar contenedores que puedan ejecutar todas sus simulaciones y ofrece las siguientes características:

- Implemente Amazon ECS en contenedores de AWS Fargate que se puedan ejecutar de forma independiente para probar las capacidades de carga del software que se está probando.
- Simule decenas de miles de usuarios conectados en varias regiones de AWS y genere registros transaccionales a un ritmo continuo.
- Personalice las pruebas de sus aplicaciones mediante la creación de [JMeter scripts](#) personalizados.
- Programe las pruebas de carga para que comiencen automáticamente en una fecha futura o en fechas recurrentes.
- Ejecute las pruebas de carga de las aplicaciones de forma simultánea o ejecute varias pruebas simultáneamente.

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.	Costo
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.	
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
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
Acceda al código fuente y, si lo desea, utilice el AWS Cloud Development Kit (AWS CDK) para implementar la solución.	GitHub repositorio

Características

La solución ofrece las siguientes características:

Out-of-the-Box Pruebas de rendimiento configurables

Incluye pruebas de rendimiento preconfiguradas disponibles para su uso inmediato.

Pruebas de aplicaciones personalizables

Permite personalizar las pruebas de forma flexible y precisa para identificar posibles problemas y adapta las pruebas a requisitos y escenarios específicos mediante JMeter scripts.

Simula una alta carga de usuarios

Capaz de simular decenas de miles de usuarios conectados para poner a prueba su aplicación.

Generación continua de transacciones

Genera registros transaccionales de forma continua para evaluar el rendimiento bajo una carga constante.

Supervisión en tiempo real

Proporciona un monitoreo en tiempo real del progreso y los resultados de las pruebas y le permite programar las pruebas para que comiencen automáticamente en fechas específicas o en intervalos periódicos.

Simulación de solicitudes regionales

Simule las solicitudes de los usuarios de cualquier región para evaluar el rendimiento global.

Flexibilidad de terminales

Pruebe cualquier punto final en todas las regiones de AWS, entornos locales u otros proveedores de nube.

Resultados detallados de las pruebas

Vea los resultados completos de las pruebas, incluidos el tiempo medio de respuesta, el número de usuarios simultáneos, las solicitudes satisfactorias y las solicitudes fallidas.

Consola web intuitiva

Ofrece una consola easy-to-use web para gestionar y supervisar las pruebas.

Soporta múltiples protocolos

Compatible con varios protocolos como HTTP WebSocket, HTTPS, JDBC, JMS, FTP y gRPC.

Ventajas

La solución ofrece las siguientes ventajas:

Soporta pruebas de rendimiento exhaustivas

Facilita las pruebas de carga, stress y resistencia para una evaluación exhaustiva de la aplicación.

Detección temprana de problemas de rendimiento

Identifica los problemas de rendimiento y los cuellos de botella antes del lanzamiento de la versión de producción.

Simulación de uso en el mundo real

Refleja con precisión los patrones de uso del mundo real para resaltar los cuellos de botella y las áreas de optimización.

Performance Insights detallados

Proporciona información sobre el rendimiento y la resiliencia del software bajo una carga significativa.

Evaluación del rendimiento automatizada

Permite realizar evaluaciones periódicas del rendimiento sin intervención manual.

Pruebas rentables

Ofrece un pay-as-you-go modelo que elimina la necesidad de una infraestructura de pruebas dedicada y de cuotas de suscripción.

Casos de uso

Simule la carga de producción

Pruebe las aplicaciones web y móviles en condiciones similares a las de producción antes de lanzar una nueva versión.

Valide el rendimiento de las aplicaciones

Asegúrese de que su aplicación pueda gestionar el tráfico de usuarios esperado sin degradarse. Pruebe los límites de las aplicaciones con los recursos predeterminados y evalúe la escalabilidad de la infraestructura.

Gestione los picos de carga

Compruebe que su infraestructura puede gestionar los picos de carga o los picos de tráfico inesperados, lo que garantiza la estabilidad en situaciones de alta demanda.

Optimice el rendimiento

Comprenda el perfil de rendimiento de su aplicación e identifique los cuellos de botella, como la ejecución ineficiente de código, las consultas a las bases de datos y la latencia de la red.

Inicio rápido de las pruebas

Comience a realizar pruebas rápidamente con pruebas de out-of-the-box rendimiento.

Pruebas personalizables

Adapte las pruebas a escenarios y requisitos específicos, ajustando la cantidad de usuarios simultáneos y las tareas lanzadas.

Pruebas programadas

Programe las pruebas para las pruebas de regresión y la supervisión continua del rendimiento, a fin de garantizar un rendimiento uniforme de las aplicaciones.

Evaluación del rendimiento geográfico

Evalúe el rendimiento de las aplicaciones en diferentes regiones geográficas para garantizar la eficiencia global.

Integración de canalizaciones de CI/CD

Integre las pruebas de rendimiento en su CI/CD proceso para realizar pruebas automatizadas y sin problemas durante los ciclos de desarrollo.

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

El número de usuarios virtuales simultáneos generados por tarea. El límite recomendado según la configuración predeterminada es de 200 usuarios virtuales. La simultaneidad está limitada por la CPU y la memoria. En el caso de las pruebas basadas en Apache JMeter, cuanto mayor sea el número de usuarios virtuales, mayor será 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 más bajos de usuarios virtuales 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 tiempo necesario para alcanzar el objetivo de simultaneidad.

espera

Es hora de mantener la simultaneidad objetivo.

Para obtener 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, que aprovecha 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 prueba y ejecutar escenarios de prueba.
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 incluyen una imagen de [contenedor compatible con la AmazonLinuxOpen Container Initiative](#) (OCI) (con el marco de pruebas de carga Blazemeter instalado), que se utiliza para generar carga y probar el rendimiento de la aplicación. Taurus/Blazemeter es un marco de automatización de pruebas de código abierto. AWS aloja la imagen del contenedor en un repositorio público de [Amazon Elastic Container Registry](#) (Amazon ECR). Para obtener más información sobre el repositorio de imágenes ECR, consulte Personalización de [imágenes de contenedores](#).

6. Se implementa una consola web con tecnología [AWS Amplify](#) en un bucket de Amazon 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, esta solución también crea una función de administrador de la solución 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 y a la API del comprobador de carga distribuido.
10. Tras implementar esta solución, puede utilizar la consola web para crear un escenario de prueba que defina una serie de tareas.
11. Los microservicios utilizan este escenario de prueba para ejecutar Amazon ECS en las tareas de AWS Fargate en las regiones especificadas.
12. [Además de almacenar los resultados en Amazon S3 y DynamoDB, una vez finalizada la prueba, el resultado se registra en Amazon. CloudWatch](#)
13. Si selecciona la opción de datos en tiempo real, la solución envía los CloudWatch registros de Amazon de las tareas de AWS Fargate a una función de Lambda durante la prueba, para cada región en la que se ejecutó la prueba.
14. A continuación, la función Lambda publica los datos en el tema correspondiente de [AWS IoT Core](#) en la región en la que se implementó la pila principal. La consola web se suscribe al tema y usted puede ver los datos mientras se ejecuta la prueba en la consola web.

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](#).

- Los recursos se definen como infraestructura como el uso de código. CloudFormation
- La solución envía las métricas a Amazon CloudWatch en varias etapas para proporcionar observabilidad en la infraestructura: funciones Lambda, tareas de Amazon ECS, depósitos de Amazon S3 y el resto de los 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](#).

- Amazon Cognito autentica y autoriza a los usuarios de la aplicación de interfaz de usuario web.
- Todas las comunicaciones entre servicios utilizan las funciones correspondientes de [AWS Identity and Access Management](#) (IAM).
- Todos los roles que utiliza la solución se basan en el acceso con privilegios mínimos. Solo contienen los permisos mínimos necesarios para realizar la tarea.
- Todo el almacenamiento de datos, incluidos los depósitos S3, cifra los datos en reposo.
- Un grupo de usuarios de Amazon Cognito administra el acceso de los usuarios a la consola y a los puntos de enlace API Gateway del comprobador de carga distribuido.
- El registro, el rastreo y el control de versiones están activados cuando corresponde.
- El acceso a la red es privado de forma predeterminada y los puntos de conexión de [Amazon Virtual Private Cloud](#) (Amazon VPC) están activados cuando están disponibles.

Note

Las pruebas de carga distribuidas en AWS crean varios grupos de CloudWatch registros en función del servicio que se utilice. El período de retención de los registros varía en función del almacenamiento y el costo del volumen de eventos de registro generados. La solución crea registros de información de contenedores para las tareas de ECS; estos registros tienen una retención de registros configurada para un día. Los registros creados para los servicios AWS Step Functions, los registros personalizados de ECS Load Testing y los registros de API Gateway se configuran con un período de retención de 1 año. Los registros de tiempo de ejecución de AWS Lambda tienen una retención de registros configurada en 2 años. Los registros de ejecución de API Gateway tienen la retención de registros configurada para que

nunca caduquen. Puede cambiar los períodos de retención de los registros en función de sus requisitos en la CloudWatch consola.

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 todos los días 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 en busca de 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.

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 on AWS consta de dos componentes de alto nivel: un [front-end](#) y un [backend](#).

Parte delantera

La interfaz consiste en una API de pruebas de carga y una consola web que se utiliza para interactuar con el backend de la solución.

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 los datos de las pruebas de forma segura a través de la consola web y la RESTful API incluidas. La API actúa como una «puerta de entrada» 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 alojada en Amazon S3 y a la que se accede 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 consola web está diseñada para demostrar cómo puede interactuar con esta solución de pruebas de carga. En un entorno de producción, le recomendamos que personalice la consola web para que se adapte a sus necesidades específicas o que cree su propia consola.

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.

Backend

El backend consiste en un contenedor, una imagen, una canalización y un motor de pruebas de carga que se utiliza para generar la carga de las pruebas. Interactúa 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 aprovecha una imagen de contenedor creada [AmazonLinux](#) como imagen base con el marco de pruebas de carga Blazemeter instalado. Esta imagen está alojada en un repositorio público de Amazon Elastic Container Registry (Amazon ECR). La imagen se usa 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 plantilla principal, la solución crea una plantilla secundaria para lanzar los recursos necesarios para ejecutar las pruebas en varias regiones. La plantilla se almacena en Amazon S3 y se proporciona un enlace a la plantilla en la consola web. Las plantillas secundarias crean una VPC, un clúster de AWS Fargate y una función Lambda para procesar datos en tiempo real.

Para obtener más información sobre cómo lanzar una región secundaria, 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 conectados en varias regiones y generar un número selecto de transacciones por segundo.

Los parámetros de las tareas que se ejecutarán como parte de la 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.

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.

Los servicios de AWS en esta solución

Esta solución incluye los siguientes servicios de AWS:

Servicio de AWS	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.
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.

Servicio de AWS	Descripción
Núcleo de AWS IoT	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.
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.

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 ID 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 de forma simultánea.
 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.

- 13A continuación, la función Lambda estructura los datos recibidos y los publica en un tema de AWS IoT Core.
- 14La 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.
- 15Una 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.
- 16Cuando 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.
- 17La función AWS Lambda del analizador de resultados escribe el resultado agregado en una tabla de Amazon DynamoDB.

Consideraciones sobre el diseño

Aplicaciones compatibles

Esta solución es compatible con aplicaciones basadas en la nube y aplicaciones locales siempre que tenga una conexión de red desde su cuenta de AWS a su aplicación. La solución admite el uso de HTTP o HTTPS. También tienes control sobre los encabezados de las solicitudes HTTP, por lo que puedes añadir encabezados de autorización o personalizados para transferir tokens o claves de API.

JMeter compatibilidad con scripts

Al crear un escenario de prueba mediante la interfaz de usuario (UI) de esta solución, puede utilizar un script de JMeter prueba. Tras seleccionar el archivo de JMeter script, se carga en el bucket <stack-name>-scenariosbucket de Amazon Simple Storage Service (Amazon S3). Cuando se ejecutan las tareas de Amazon Elastic Container Service (Amazon ECS), JMeter el script se descarga del <stack-name>bucket Amazon S3 -scenariosbucket y se ejecuta 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](#)

Soporte para scripts K6

La solución es compatible con las pruebas basadas en el marco K6. [K6 se publica con la licencia AGPL-3.0](#). La solución muestra un mensaje de confirmación de licencia al crear una nueva prueba para K6. El archivo de prueba K6, junto con los archivos de entrada necesarios, se pueden incluir en un archivo y cargarse para el escenario de prueba mediante la opción de carga.

Compatible con el alfabeto Locust

La solución es compatible con las pruebas basadas en el marco Locust. El archivo de prueba de Locust, junto con los archivos de entrada necesarios, pueden incluirse en un archivo y cargarse para el escenario de prueba mediante la opción de carga.

Programar las pruebas

Puede programar las pruebas para que se ejecuten en una fecha futura o usar la opción Ejecutar ahora. Puede programar una prueba como una ejecución única en el futuro o configurar una prueba periódica en la que especifique una fecha de primera ejecución y una periodicidad planificada. Las opciones de recurrencia incluyen: diaria, semanal, quincenal y mensual. Para obtener más información sobre cómo funciona la programación, consulte la sección [Flujo de trabajo de programación de pruebas](#) de esta guía.

A partir de la versión 3.3.0, las pruebas de carga distribuidas en AWS permiten a los usuarios programar las pruebas de carga mediante expresiones cron. Seleccione Ejecutar según lo programado y, a continuación, la pestaña CRON para introducir manualmente un valor cron o utilizar los campos desplegables. cronExpiryDate Debe coincidir con la fecha de ejecución de la prueba programada. Revisa las fechas de la próxima ejecución (UTC) para confirmar tu programación.

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 incluye un CloudWatch panel de Amazon para cada prueba y muestra el resultado combinado de todas las tareas que se ejecutan para esa prueba 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 aceptadas y el número de solicitudes fallidas. Cada métrica se agrega por segundo y el panel se actualiza 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 el [costo](#), la [seguridad](#), [las regiones](#) y otras consideraciones antes de implementar la solución.

Costo

Usted es responsable del coste de los servicios de AWS utilizados durante la ejecución de esta solución. El coste total de ejecutar esta solución depende del número de pruebas de carga ejecutadas, su duración y la cantidad de datos utilizados como parte de las pruebas. A partir de esta revisión, el coste 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 la clave = y el valor = SO0062SolutionId . [Puede activar la clave SolutionId de etiqueta 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](#).

Important

A partir de la versión 1.3.0, la CPU se incrementa a 2 vCPU y la memoria a 4 GB. Estos cambios aumentan el costo estimado en comparación con las versiones anteriores de esta solución. Si sus pruebas de carga no requieren estos aumentos en sus recursos de AWS, puede reducirlos. Para obtener información adicional, consulte 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.

Los precios están sujetos a cambios. Para obtener más información, consulte la página web de precios de cada servicio de AWS que vaya a utilizar en esta solución.

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 la carga operativa, ya que AWS opera, administra y controla los componentes, incluidos el sistema operativo anfitrión, la capa de virtualización y la seguridad física de 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 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

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.

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 a otras ubicaciones, como otras EC2 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.

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 (Bombay)	Europa (Londres)
Asia-Pacífico (Seúl)	Europa (París)

Nombre de la región	
Asia-Pacífico (Singapur)	Europa (Estocolmo)
Asia-Pacífico (Sídney)	América del Sur (São Paulo)

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 los 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](#).

Utilice los siguientes enlaces para ir a la página de ese servicio. Para ver las cuotas de servicio de todos los servicios de AWS en la documentación sin cambiar de página, consulte la información en la página de [puntos finales y cuotas del servicio](#) en el 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 AmazonLinux imagen del contenedor (con Blazemeter 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 incluye un CloudWatch panel de Amazon para cada prueba y muestra el resultado combinado de todas las tareas que se ejecutan para esa prueba 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 aceptadas y el número de solicitudes fallidas. Cada métrica se agrega por segundo y el panel se actualiza cada minuto.

Política de EC2 pruebas de Amazon

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, consulta la [Política de EC2 pruebas de Amazon](#).

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

Activación o configuración de CloudWatch alarmas

Se recomienda monitorear continuamente los recursos de la solución después de la implementación. Puede considerar configurar [CloudWatch métricas](#) para los recursos creados por la solución.

Métricas CloudFront de distribución de Amazon

Revisa las [métricas CloudFront de distribución](#) y configura las alarmas para monitorear y recibir notificaciones.

Métricas de Amazon API Gateway

Revise las [dimensiones y las métricas de Amazon API Gateway](#) y configure las alarmas para supervisar y recibir notificaciones.

Implementación de la solución

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.

Información general del proceso de implementación

Siga las step-by-step instrucciones de esta sección para configurar e implementar la solución en su cuenta.

Antes de lanzar la solución, revise el [costo](#), la [arquitectura](#), la [seguridad de la red](#) y otras consideraciones descritas anteriormente en esta guía.

Tiempo de implementación: aproximadamente 15 minutos

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:

[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

Important

Si va a actualizar la pila desde una versión anterior a la versión 3.2.6 a la versión más reciente, lea [esta sección](#) antes de actualizar la pila.

Antes de lanzar la implementación automatizada, revise la arquitectura y otras consideraciones que se describen en esta guía. Siga las step-by-step instrucciones de esta sección para configurar e implementar las pruebas de carga distribuidas en AWS en su cuenta.

Tiempo de implementación: aproximadamente 15 minutos

Important

Esta solución incluye una opción para enviar métricas operativas anonimizadas a AWS. Utilizamos estos datos para comprender mejor cómo utilizan los clientes esta solución, así como los servicios y productos relacionados. AWS es propietario de los datos recopilados a través de esta encuesta. La recopilación de datos está sujeta al [Aviso de privacidad de AWS](#). Para excluirse de esta función, descargue la plantilla, modifique la sección de CloudFormation mapeo de AWS y, a continuación, utilice la CloudFormation consola de AWS para cargar la plantilla actualizada e implementar la solución. Para obtener más información, consulte la sección [Recopilación de datos anónimos](#) de esta guía.

Esta CloudFormation plantilla de AWS automatizada implementa las pruebas de carga distribuidas en 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.

1. Inicie sesión en la consola de administración de AWS y seleccione el botón de abajo para lanzar la CloudFormation plantilla distributed-load-testing-on -aws AWS.

Como alternativa, también puede [descargar la plantilla](#) como punto de partida para su propia implementación.

- 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](#).

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

Parámetro	Predeterminado/a	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,

Parámetro	Predeterminado/a	Descripción
		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 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

Parámetro	Predeterminado/a	Descripción
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	Yes	Utilice automáticamente la imagen más actualizada y segura hasta la próxima versión secundaria. Al seleccionar No, se mostrará la imagen tal y como se publicó originalmente, sin ninguna actualización de seguridad.

6. Elija Siguiente.
7. En la página Configurar opciones de pila, elija Siguiente.
8. En la página Revisar, revise y confirme la configuración. Marque la casilla para confirmar que la plantilla creará recursos de AWS Identity and Access Management (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 cinco minutos

Puede realizar pruebas en varias regiones. Al implementar la solución Distributed Load Testing, se crean tres buckets de Amazon S3. La solución crea una pila regional secundaria y la almacena en el depósito de escenarios de Amazon S3.

Note

La convención de nomenclatura de los cubos es `<stack-name> - dlttestrunnerstoragedltscenariosbucket <_[0-9][0-9]..-<[0-9][0-9].._` con la palabra clave escenarios en el nombre del bucket, que puede localizar navegando a la consola S3 y luego a Buckets.

Para ejecutar una implementación en varias regiones, debe implementar la CloudFormation plantilla regional, que se almacena en el depósito de escenarios de Amazon S3, en las regiones en las que desee ejecutar la prueba. Puede instalar la plantilla regional de la siguiente manera:

1. En la consola web de la solución, vaya a Administrar regiones en el menú superior.
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/a	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

Parámetro	Predeterminado/a	Descripción
		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 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, la nueva región aparecerá en el modal Administrar regiones. Para usar esta región en una prueba, selecciónela al crear la prueba. La solución crea un elemento de DynamoDB para cada región lanzada 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. Debido a las restricciones de la API, solo puedes ver los resultados agregados de todas las regiones en una prueba multirregional si los graficas en las métricas de Amazon CloudWatch . 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

Si ya implementó la solución anteriormente, siga este procedimiento para actualizar la CloudFormation pila de soluciones y obtener la versión más reciente del marco de la solución.

1. Inicie sesión en la [CloudFormation consola](#), seleccione la CloudFormation pila existente y seleccione Actualizar pila.
2. Selecciona Realizar una actualización directa.
3. Selecciona Reemplazar plantilla existente.
4. En Especificar plantilla:
 - a. Seleccione URL de Amazon S3.
 - b. Copia 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 confirmar que la plantilla podría 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.

Al actualizar desde versiones de DLT anteriores a la v3.2.6 a la v3.3.x y a la versión 3.3.x a la versión más reciente, se produce un error en la actualización de la pila

1. Descargue la plantilla [distributed-load-testing-on-aws.template](https://github.com/aws-samples/distributed-load-testing-on-aws-template).
2. Abra la plantilla y vaya a Condiciones: y busque DLTKCommonResourcesAppRegistryCondition
3. Debería ver algo similar a lo siguiente:

```
Conditions:
DLTKCommonResourcesAppRegistryConditionCCEF54F8:
Fn::Equals:
- "true"
- "true"
```

4. Cambie el segundo valor verdadero a falso:

```
Conditions:
DLTKCommonResourcesAppRegistryConditionCCEF54F8:
Fn::Equals:
- "true"
- "false"
```

5. Usa la plantilla personalizada para actualizar tu pila.
6. Esta pila elimina de la pila los recursos relacionados con el registro de aplicaciones. Por lo tanto, la actualización debería completarse.
7. Realiza otra actualización de la pila utilizando la URL de la plantilla más reciente para volver a añadir los recursos de la aplicación de registro de aplicaciones a tu pila.

Note

AWS Systems Manager Application Manager le ofrece una visión a nivel de aplicación de esta solución y sus recursos para que pueda:

1. Supervise sus recursos, los costes de los recursos implementados en todas las pilas y cuentas de AWS y los registros asociados a esta solución desde una ubicación central.

2. Consulte los datos operativos de los recursos de esta 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.

Solución 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 las 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. Compruebe 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 EC2 instancias o balanceadores de carga de aplicaciones internos, asegúrate 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 cuenta con [AWS Developer Support](#), [AWS Business Support](#) o [AWS Enterprise Support](#), puede utilizar el Centro de soporte para obtener asistencia de expertos con esta solución. En las siguientes secciones, encontrará instrucciones.

Cree un caso

1. Inicie sesión en [Support Center](#).
2. Seleccione Crear caso.

¿Cómo podemos ayudar?

1. Elija Técnico
2. Para el 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 introducir el servicio, la categoría y la gravedad, la interfaz rellena los enlaces a las preguntas de solución de problemas más frecuentes. Si no puede resolver sus dudas con estos enlaces, seleccione Siguiente paso: información adicional.

Información adicional

1. En Asunto, introduce un texto que resuma tu pregunta o problema.
2. En Descripción, describe el problema en detalle.
3. Selecciona Adjuntar archivos.
4. Adjunte la información que AWS Support necesita para procesar la solicitud.

Ayúdenos a resolver su caso más rápido

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

Resuelva ahora o póngase en contacto con nosotros

1. Revise las soluciones Solve now.
2. Si no puede resolver su problema con estas soluciones, elija Contactar con nosotros, introduzca la información solicitada y pulse 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

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.

Uso de la interfaz de línea de comandos de AWS

Determine si la interfaz de línea de comandos de AWS (AWS CLI) está disponible en su entorno. Para obtener instrucciones de instalación, consulte [Qué es la interfaz de línea de comandos de AWS](#) en la Guía del usuario de la CLI de AWS. Tras confirmar que la AWS CLI 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
```

Usa la solución

En esta sección se incluye información sobre cómo utilizar la solución Distributed Load Testing en AWS, incluidos [los resultados de las pruebas](#), el [flujo de trabajo de programación de las pruebas](#) y [los datos en tiempo real](#).

Resultados de la prueba

Las pruebas de carga distribuidas en AWS utilizan el marco de pruebas de carga para ejecutar pruebas de aplicaciones a escala. Cuando se completa una prueba, se genera un informe detallado con los siguientes resultados.

- 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 generadas por la prueba.
- Ancho de banda medio: ancho de banda medio 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.
- Percentil: el percentil del tiempo de respuesta de la prueba. El tiempo de respuesta máximo es del 100%; el tiempo de respuesta mínimo es del 0%.

Note

Los resultados de las pruebas se muestran en la consola. Puede ver los archivos XML de los resultados de las pruebas sin procesar en la Results carpeta del bucket de Scenarios Amazon S3.

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.

Flujo de trabajo para programar pruebas

Utilice la consola web para programar una prueba de carga. Al programar una prueba, se ejecuta el siguiente flujo de trabajo:

- Cuando se crea una prueba de carga con la opción de programarla, los parámetros de programación se envían a la API de la solución a través de Amazon API Gateway.
- A continuación, la API pasa los parámetros a una función de Lambda que crea una regla de CloudWatch eventos, que se programará para ejecutarse en la fecha especificada.
- Si la prueba es de una sola vez, la regla de CloudWatch eventos se ejecuta en la fecha especificada. La función `api-services` Lambda ejecuta una nueva prueba a través del flujo de trabajo de la prueba.
- Si la prueba es periódica, la regla de CloudWatch eventos se activa en la fecha especificada. Se ejecuta la función `api-services` Lambda, que elimina la regla de CloudWatch eventos actual y crea otra regla que se ejecuta inmediatamente cuando se crea y de forma recurrente a partir de entonces en función de la frecuencia de recurrencia especificada.

Determine el número de usuarios

La cantidad de usuarios que puede admitir un contenedor para una prueba se puede determinar aumentando gradualmente la cantidad de usuarios y monitoreando 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). 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. Opcionalmente, los recursos del contenedor se pueden aumentar para permitir un mayor número de usuarios simultáneos. Sin embargo, esto se traduce en un coste mayor. Para obtener más información, consulte la sección [Aumentar los recursos de contenedores](#) de esta guía.

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

Datos en directo

Si lo desea, puede incluir datos en tiempo real al ejecutar una prueba para obtener información en tiempo real sobre lo que está ocurriendo. El grupo de CloudWatch registros de las tareas de Fargate contiene un filtro de suscripción para los resultados de las pruebas que incluyen la opción de datos en tiempo real. Si la solución encuentra el patrón, inicia una función Lambda que estructura los datos y los publica en un tema de AWS IoT Core. La consola web se suscribe al tema, recibe los datos entrantes y grafica los datos agregados en intervalos de un segundo. La consola web contiene cuatro gráficos: tiempo de respuesta promedio, usuarios virtuales, éxitos y fracasos.

Note

Los datos son efímeros y solo se utilizan para ver lo que sucede mientras se ejecuta la prueba. Una vez finalizada la prueba, la solución almacena los datos de los resultados en DynamoDB y Amazon S3. 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.

Pruebe el flujo de trabajo de cancelación

Al cancelar una prueba de carga desde la consola web, la solución ejecuta el siguiente flujo de trabajo de cancelación de pruebas.

1. La solicitud de cancelación se envía a la `microservices API`.
2. La `microservices API` llama a la función `task-canceller Lambda`, que cancela las tareas hasta que se detengan todas las tareas actualmente iniciadas.
3. Si la función `task-runner Lambda` continúa ejecutándose después de la llamada inicial a la función `task-canceller Lambda`, las tareas seguirán lanzándose. Una vez `task-runner` finalizada la función `Lambda`, `AWS Step Functions` continúa con el `Cancel Test` paso, en el que se vuelve a ejecutar la función `task-canceller Lambda` para detener las tareas pendientes.

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 coinciden con cada versión de la solución si no se selecciona la opción de actualizaciones automáticas. El equipo de soluciones de AWS utiliza el escaneo mejorado con ECR para detectar vulnerabilidades y exposiciones comunes (CVEs) en la imagen base y en los paquetes instalados. Cuando sea posible, el equipo publicará las imágenes parcheadas con la misma etiqueta de versión para resolverlas CVEs, sin interrumpir la compatibilidad con la versión de la solución publicada. Cuando se parcheen las imágenes, si se encuentran en la misma versión secundaria, la etiqueta estable se actualizará automáticamente y se creará una etiqueta de imagen adicional en ese formato. `<solution-version>_<date-of-fix>` Si se publica una versión principal o secundaria, será necesaria una actualización completa para obtener la última versión de la imagen, ya que la etiqueta estable se incrementará para que su versión coincida con la versión de la solución. Si opta por las actualizaciones automáticas, los cambios en la imagen, incluidas las correcciones de errores menores CVEs y las correcciones de errores menores, se aplicarán automáticamente a la imagen hasta la última versión secundaria correspondiente.

Versiones

Los clientes que utilicen la versión más reciente de la solución recibirán automáticamente parches de seguridad y correcciones de errores menores que no afecten a ningún problema si optan por las actualizaciones automáticas de las imágenes. La imagen mostrará automáticamente la imagen más reciente hasta la última versión secundaria coincidente. Para bloquear el contenedor en una versión específica, se puede editar la definición de la tarea para especificar que el contenedor utilice una versión de imagen específica utilizando la versión etiquetada de la imagen. Las actualizaciones automáticas también se pueden desactivar seleccionando No para activar las actualizaciones

automáticas al lanzar la pila. CloudFormation Esto abrirá la versión de la imagen que coincida con la versión de la solución.

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

# Remove jar files from /tmp
RUN rm -rf /tmp/jmeter-plugins-manager-1* && \
    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
    fi

```

```

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

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 `ecslister.py` script, mientras que la tarea principal ejecutará el `ecscontroller.py` script. El `ecslister.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.

Escenarios

- [GET /scenarios](#)
- [POST/escenarios](#)
- [OPCIONES/escenarios](#)
- [GET /scenarios/ {testID}](#)
- [POST /escenarios/ {testID}](#)
- [ELIMINAR /scenarios/ {testID}](#)
- [OPCIONES /scenarios/ {testID}](#)

Tareas

- [GET /tasks](#)
- [OPCIONES /tareas](#)

Regiones

- [GET /regiones](#)
- [OPCIONES/regiones](#)

GET /scenarios

Descripción

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

Respuesta

Nombre	Descripción
data	Una lista de escenarios que incluye el ID, el nombre, la descripción, el estado y el tiempo de ejecución de cada prueba

POST /escenarios

Descripción

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

Cuerpo de la solicitud

Nombre	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

Nombre	Descripción
<code>testScenario</code>	La definición de la prueba incluye la simultaneidad, la hora de la prueba, el anfitrión y el método de la prueba
<code>testType</code>	El tipo de prueba (por ejemplo, <code>simple</code> , <code>jmeter</code>)
<code>fileType</code>	El tipo de archivo de carga (por ejemplo, <code>none</code> , <code>script</code> , <code>zip</code>)
<code>scheduleDate</code>	La fecha en la que se realizará una prueba. Solo se proporciona si se programa una prueba (por ejemplo, <code>2021-02-28</code>)
<code>scheduleTime</code>	El tiempo necesario para ejecutar una prueba. Solo se proporciona si se programa una prueba (por ejemplo, <code>21:07</code>)
<code>scheduleStep</code>	El paso del proceso de programación. Solo se proporciona si se programa una prueba periódica. (Los pasos disponibles incluyen <code>create</code> y <code>start</code>)
<code>cronvalue</code>	El valor cron para personalizar la programación periódica. Si se usa, omita <code>ScheduleDate</code> y <code>ScheduleTime</code> .
<code>cronExpiryDate</code>	Fecha obligatoria para que el cron caduque y no se ejecute indefinidamente.
<code>recurrence</code>	La recurrencia de una prueba programada. Solo se proporciona si se programa una prueba periódica (por ejemplo <code>daily</code> , <code>weekly</code> , <code>biweekly</code> , <code>monthly</code>)

Respuesta

Nombre	Descripción
testId	El identificador único de la prueba
testName	El nombre de la prueba
status	El estado de la prueba

OPCIONES/ESCENARIOS

Descripción

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

Respuesta

Nombre	Descripción
testId	El identificador único de la prueba
testName	El nombre de la prueba
status	El estado de la prueba

GET /scenarios/ {testID}

Descripción

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

Parámetro de solicitud

testId

- El identificador único de la prueba

Tipo: cadena

Obligatorio: sí

Respuesta

Nombre	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, <code>simple</code> , <code>jmeter</code>)
fileType	El tipo de archivo que se carga (por ejemplo, <code>none</code> , <code>script</code> , <code>zip</code>)
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
results	Los resultados finales de la prueba

Nombre	Descripción
history	Una lista de los resultados finales de las pruebas anteriores
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}

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

Nombre	Descripción
status	El estado de la prueba

ELIMINAR /scenarios/ {testID}

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

Nombre	Descripción
status	El estado de la prueba

OPCIONES /escenarios/ {testID}

Descripción

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

Respuesta

Nombre	Descripción
testId	El identificador único de la prueba
testName	El nombre de la prueba
testDescription	La descripción de la prueba

Nombre	Descripción
testType	El tipo de prueba que se ejecuta (por ejemplo, <code>simple</code> , <code>jmeter</code>)
fileType	El tipo de archivo que se carga (por ejemplo, <code>none</code> , <code>script</code> , <code>zip</code>)
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
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 /tasks

Descripción

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

Respuesta

Nombre	Descripción
tasks	Una lista de tareas IDs para ejecutar las pruebas

OPCIONES /tareass

Descripción

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

Respuesta

Nombre	Descripción
taskIds	Una lista de tareas IDs para ejecutar las pruebas

GET /regions

Descripción

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

Respuesta

Nombre	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
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

Descripción

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

Respuesta

Nombre	Descripción
testId	El ID de la región

Nombre	Descripción
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
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 compatibles actualmente, aumente los recursos de contenedores. Esto le permite aumentar la memoria CPUs y gestionar el aumento de usuarios simultáneos.

Cree una nueva revisión de la definición de tareas

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.
 - 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 duro a la memoria que desee.
 - d. Elija Actualizar.
6. En la página Crear nueva revisión, seleccione Crear.
 7. Una vez que la definición de tarea se haya creado correctamente, registre el nombre de la nueva definición de tarea. Este nombre incluye el número de versión, por ejemplo:
[replaceable] <stackName>`- EcsTaskDefinition -<system-generated-random-Hash>:
[replaceable]<system-generated-versionNumber>.

Actualizar la tabla de DynamoDB

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 escenarios-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 ha modificado la definición de la tarea. Por ejemplo, region-<region-name>.
5. Actualice el atributo taskDefinition con la nueva definición de tarea.

Referencia

Esta sección incluye información sobre una función opcional para recopilar métricas únicas para esta solución, sugerencias sobre los recursos relacionados y una lista de los desarrolladores que han contribuido a esta solución.

Recopilación de datos anonimizados

Esta solución incluye una opción para enviar métricas operativas anonimizadas a AWS. Utilizamos estos datos para comprender mejor cómo utilizan los clientes esta solución, así como los servicios y productos relacionados. Cuando se invoca, se recopila la siguiente información y se envía a AWS:

- ID de solución: el identificador de la solución de AWS
- ID único (UUID): identificador único generado aleatoriamente para cada implementación de solución
- Marca de tiempo: marca de tiempo de recopilación de datos
- Tipo de prueba: el tipo de prueba que se ejecuta
- Tipo de archivo: el tipo de archivo que se carga
- Recuento de tareas: el recuento de tareas de cada prueba enviada a través de la API de la solución
- Duración de la tarea: el tiempo total de ejecución de todas las tareas necesarias para ejecutar una prueba
- Resultado de la prueba: el resultado de la prueba que se ejecutó

AWS es propietario de los datos recopilados mediante esta encuesta. La recopilación de datos está sujeta a la [Política de privacidad de AWS](#). Para excluirse de esta función, complete los siguientes pasos antes de lanzar la CloudFormation plantilla de AWS.

1. Descargue la [CloudFormation plantilla de AWS](#) en su disco duro local.
2. Abra la CloudFormation plantilla de AWS con un editor de texto.
3. Modifique la sección de mapeo de CloudFormation plantillas de AWS desde:

```
Solution:  
Config:
```

```
SendAnonymousData: "Yes"
```

a:

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

4. Inicie sesión en la [CloudFormation consola de AWS](#).
5. Elija Crear pila.
6. En la página Crear pila, en la sección Especificar plantilla, seleccione Cargar un archivo de plantilla.
7. En Cargar un archivo de plantilla, seleccione Elegir archivo y después seleccione la plantilla editada de su unidad local.
8. Seleccione Siguiente y siga los pasos que se indican en [Lanzar la pila](#), en la sección Implementar la solución de esta guía.

Colaboradores

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

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](https://www.apache.org/licenses/LICENSE-2.0).

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.