Panduan Implementasi

Pengujian Beban Terdistribusi di AWS

Pengujian Beban Terdistribusi di AWS: Panduan Implementasi

Table of Contents

Otomatiskan pengujian aplikasi perangkat lunak Anda dalam skala besar

Tanggal publikasi: November 2019

Pengujian Beban Terdistribusi di AWS membantu Anda mengotomatiskan pengujian aplikasi perangkat lunak Anda dalam skala besar dan saat dimuat untuk mengidentifikasi kemacetan sebelum Anda merilis aplikasi. Solusi ini menciptakan dan mensimulasikan ribuan pengguna yang terhubung menghasilkan catatan transaksional dengan kecepatan konstan tanpa perlu menyediakan server.

Solusi ini memanfaatkan Amazon Elastic Container Service (Amazon ECS) di AWS Fargate untuk menerapkan container yang dapat menjalankan semua simulasi Anda dan menawarkan fitur berikut:

- Terapkan Amazon ECS pada kontainer AWS Fargate yang dapat berjalan secara independen untuk menguji kemampuan pemuatan perangkat lunak yang sedang diuji.
- Simulasikan puluhan ribu pengguna yang terhubung, di beberapa Wilayah AWS, menghasilkan catatan transaksional dengan kecepatan berkelanjutan.
- Sesuaikan pengujian aplikasi Anda dengan membuat JMeter skrip kustom.
- Jadwalkan tes beban untuk dimulai secara otomatis pada tanggal yang akan datang atau pada tanggal berulang.
- Jalankan pengujian pemuatan aplikasi Anda secara bersamaan atau jalankan beberapa pengujian secara bersamaan.

Panduan implementasi ini memberikan gambaran umum tentang solusi Distributed Load Testing on AWS, arsitektur referensi dan komponennya, pertimbangan untuk merencanakan penerapan, dan langkah-langkah konfigurasi untuk menerapkan solusi ke Amazon Web Services (AWS) Cloud. Ini mencakup tautan ke CloudFormation templat AWS yang meluncurkan dan mengonfigurasi layanan AWS yang diperlukan untuk menerapkan solusi ini menggunakan praktik terbaik AWS untuk keamanan dan ketersediaan.

Audiens yang dituju untuk menggunakan fitur dan kemampuan solusi ini di lingkungan mereka mencakup arsitek infrastruktur TI, administrator, dan DevOps profesional yang memiliki pengalaman praktis dalam merancang di AWS Cloud.

Gunakan tabel navigasi ini untuk menemukan jawaban atas pertanyaan-pertanyaan ini dengan cepat:

Jika kau mau.	Baca.
Ketahui biaya untuk menjalankan solusi ini. Perkiraan biaya untuk menjalankan solusi ini di Wilayah AS Timur (Virginia N.) adalah USD $30,90 per bulan untuk sumber daya AWS.	Biaya
Pahami pertimbangan keamanan untuk solusi ini.	Keamanan
Ketahui cara merencanakan kuota untuk solusi ini.	Kuota
Ketahui Wilayah AWS mana yang mendukung solusi ini.	Wilayah AWS yang Didukung
Lihat atau unduh CloudFormation templat AWS yang disertakan dalam solusi ini untuk secara otomatis menerapkan sumber daya infrastruktur (“tumpukan”) untuk solusi ini.	CloudFormation Templat AWS
Akses kode sumber dan secara opsional gunakan AWS Cloud Development Kit (AWS CDK) untuk menerapkan solusi.	GitHub repositori

Fitur

Solusinya menyediakan fitur-fitur berikut:

Out-of-the-Box Tes Kinerja yang Dapat Dikonfigurasi

Termasuk tes kinerja pra-konfigurasi yang tersedia untuk segera digunakan.

Tes Aplikasi yang Dapat Disesuaikan

Memungkinkan kustomisasi pengujian yang fleksibel dan tepat untuk mengidentifikasi potensi masalah dan menyesuaikan pengujian dengan persyaratan dan skenario tertentu menggunakan JMeter skrip.

Mensimulasikan Beban Pengguna Tinggi

Mampu mensimulasikan puluhan ribu pengguna yang terhubung untuk stress test aplikasi Anda.

Pembuatan Transaksi Berkelanjutan

Menghasilkan catatan transaksional secara terus menerus untuk mengevaluasi kinerja di bawah beban konstan.

Pemantauan Waktu Nyata

Menyediakan pemantauan real-time kemajuan tes dan hasil dan memungkinkan Anda untuk menjadwalkan tes untuk memulai secara otomatis pada tanggal yang ditentukan atau pada interval berulang.

Simulasi Permintaan Regional

Simulasikan permintaan pengguna dari wilayah mana pun untuk menilai kinerja global.

Fleksibilitas Titik Akhir

Uji titik akhir apa pun di seluruh wilayah AWS, lingkungan lokal, atau penyedia cloud lainnya.

Hasil Tes Terperinci

Lihat hasil pengujian komprehensif, termasuk waktu respons rata-rata, jumlah pengguna bersamaan, permintaan yang berhasil, dan permintaan yang gagal.

Konsol Web Intuitif

Menawarkan konsol easy-to-use web untuk mengelola dan memantau tes.

Mendukung Beberapa Protokol

Kompatibel dengan berbagai protokol seperti WebSocket, HTTP, HTTPS, JDBC, JMS, FTP, dan gRPC.

Manfaat

Solusinya memberikan manfaat sebagai berikut:

Mendukung Pengujian Kinerja Komprehensif

Memfasilitasi pengujian beban, stres, dan ketahanan untuk evaluasi aplikasi menyeluruh.

Deteksi Dini Masalah Kinerja

Mengidentifikasi masalah kinerja dan kemacetan sebelum rilis produksi.

Simulasi Penggunaan Dunia Nyata

Secara akurat mencerminkan pola penggunaan dunia nyata untuk menyoroti kemacetan dan area pengoptimalan.

Performance Insights Terperinci

Memberikan wawasan tentang kinerja dan ketahanan perangkat lunak di bawah beban yang signifikan.

Penilaian Kinerja Otomatis

Memungkinkan evaluasi kinerja reguler tanpa intervensi manual.

Pengujian Hemat Biaya

Menawarkan pay-as-you-go model, menghilangkan kebutuhan akan infrastruktur pengujian khusus dan biaya berlangganan.

Kasus penggunaan

Simulasikan Beban Produksi

Uji aplikasi web dan seluler dalam kondisi seperti produksi sebelum meluncurkan versi baru.

Validasi Kinerja Aplikasi

Pastikan aplikasi Anda dapat menangani lalu lintas pengguna yang diharapkan tanpa degradasi. Uji batas aplikasi menggunakan sumber daya default dan menilai skalabilitas infrastruktur.

Kelola Beban Puncak

Pastikan infrastruktur Anda dapat mengelola beban puncak atau lonjakan lalu lintas yang tidak terduga, memastikan stabilitas di bawah permintaan tinggi.

Optimalkan Kinerja

Pahami profil kinerja aplikasi Anda dan identifikasi kemacetan seperti eksekusi kode yang tidak efisien, kueri basis data, dan latensi jaringan.

Pengujian Cepat Start-Up

Mulailah pengujian dengan cepat dengan tes out-of-the-box kinerja.

Tes yang Dapat Disesuaikan

Sesuaikan pengujian dengan skenario dan persyaratan tertentu, sesuaikan jumlah pengguna dan tugas bersamaan yang diluncurkan.

Pengujian Terjadwal

Jadwalkan tes untuk pengujian regresi dan pemantauan kinerja berkelanjutan, memastikan kinerja aplikasi yang konsisten.

Evaluasi Kinerja Geografis

Mengevaluasi kinerja aplikasi di berbagai wilayah geografis untuk memastikan efisiensi global.

Integrasi Pipa CI/CD

Integrasikan pengujian kinerja ke dalam CI/CD pipeline Anda untuk pengujian yang mulus dan otomatis selama siklus pengembangan.

Konsep dan definisi

Bagian ini menjelaskan konsep-konsep kunci dan mendefinisikan terminologi khusus untuk solusi ini:

skenario

Definisi pengujian termasuk nama pengujian, deskripsi, jumlah tugas, konkurensi, Wilayah AWS, ramp-up, penahanan, jenis pengujian, tanggal jadwal, dan konfigurasi pengulangan.

jumlah tugas

Jumlah kontainer yang akan diluncurkan di cluster Fargate untuk menjalankan skenario pengujian. Tugas tambahan tidak akan dibuat setelah batas akun pada sumber daya Fargate tercapai. Namun, tugas yang sudah berjalan akan terus berlanjut.

konkurensi

Jumlah pengguna virtual bersamaan yang dihasilkan per tugas. Batas yang disarankan berdasarkan pengaturan default adalah 200 pengguna virtual. Konkurensi dibatasi oleh CPU dan memori. Untuk tes berdasarkan Apache JMeter, semakin tinggi jumlah pengguna virtual, semakin tinggi memori yang

digunakan oleh JVM pada tugas ECS. Definisi Tugas ECS default membuat tugas dengan memori 4 GB. Disarankan untuk memulai dengan nilai pengguna virtual yang lebih rendah untuk 1 tugas dan memantau CloudWatch metrik ECS untuk Task Cluster. Lihat metrik pemanfaatan klaster Amazon ECS.

ramp-up

Waktu untuk mencapai target konkurensi.

tahan-untuk

Saatnya menahan konkurensi target.

Untuk referensi umum istilah AWS, lihat Daftar Istilah AWS.

Gambaran umum arsitektur

Diagram arsitektur

Menerapkan solusi ini dengan parameter default akan menerapkan komponen berikut di akun AWS Anda.

Pengujian Beban Terdistribusi pada arsitektur AWS di AWS

> **Note**
>
> CloudFormation Sumber daya AWS dibuat dari konstruksi AWS Cloud Development Kit (AWS CDK).

Alur proses tingkat tinggi untuk komponen solusi yang digunakan dengan CloudFormation template AWS adalah sebagai berikut:

1. API penguji beban terdistribusi, yang memanfaatkan Amazon API Gateway untuk menjalankan layanan mikro solusi (fungsi AWS Lambda).
2. Layanan mikro menyediakan logika bisnis untuk mengelola data pengujian dan menjalankan tes.
3. Layanan mikro ini berinteraksi dengan Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, dan AWS Step Functions untuk menyediakan penyimpanan untuk detail skenario pengujian dan hasil serta menjalankan skenario pengujian.
4. Topologi jaringan Amazon Virtual Private Cloud (Amazon VPC) diterapkan yang berisi kontainer Amazon Elastic Container Service (Amazon ECS) solusi yang berjalan di AWS Fargate.
5. Container termasuk AmazonLinux(dengan kerangka pengujian beban blazemeter terpasang) Open Container Initiative (OCI) image kontainer yang sesuai, yang digunakan untuk menghasilkan beban untuk menguji kinerja aplikasi Anda. Taurus/Blazemeter adalah kerangka kerja otomatisasi pengujian sumber terbuka. Gambar kontainer di-host oleh AWS di repositori publik Amazon Elastic Container Registry (Amazon ECR). Untuk informasi selengkapnya tentang repositori gambar ECR, lihat kustomisasi gambar Container.
6. Konsol web yang didukung oleh AWS Amplify diterapkan ke bucket Amazon S3 yang dikonfigurasi untuk hosting web statis.

7. Amazon CloudFront menyediakan akses publik yang aman ke konten bucket situs web solusi.
8. Selama konfigurasi awal, solusi ini juga membuat peran administrator solusi default (peran IAM) dan mengirimkan undangan akses ke alamat email pengguna yang ditentukan pelanggan.
9. Kumpulan pengguna Amazon Cognito mengelola akses pengguna ke konsol dan API penguji beban terdistribusi.
10. Setelah menerapkan solusi ini, Anda dapat menggunakan konsol web untuk membuat skenario pengujian yang mendefinisikan serangkaian tugas.
11. Layanan mikro menggunakan skenario pengujian ini untuk menjalankan Amazon ECS pada tugas AWS Fargate di Wilayah yang ditentukan.
12. Selain menyimpan hasil di Amazon S3 dan DynamoDB, setelah pengujian selesai, output dicatat di Amazon. CloudWatch
13. Jika Anda memilih opsi data langsung, solusi akan mengirimkan CloudWatch log Amazon untuk tugas AWS Fargate ke fungsi Lambda selama pengujian, untuk setiap Wilayah tempat pengujian dijalankan.
14. Fungsi Lambda kemudian menerbitkan data ke topik yang sesuai di AWS IoT Core di Wilayah tempat tumpukan utama digunakan. Konsol web berlangganan topik, dan Anda dapat melihat data saat pengujian berjalan di konsol web.

Pertimbangan desain AWS Well-Architected

Solusi ini menggunakan praktik terbaik dari AWS Well-Architected Framework, yang membantu pelanggan merancang dan mengoperasikan beban kerja yang andal, aman, efisien, dan hemat biaya di cloud.

Bagian ini menjelaskan bagaimana prinsip-prinsip desain dan praktik terbaik dari Well-Architected Framework menguntungkan solusi ini.

Keunggulan operasional

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar keunggulan operasional.

- Sumber daya didefinisikan sebagai infrastruktur sebagai penggunaan kode CloudFormation.
- Solusi ini mendorong metrik ke Amazon CloudWatch pada berbagai tahap untuk memberikan pengamatan ke dalam infrastruktur: fungsi Lambda, tugas Amazon ECS, bucket Amazon S3, dan komponen solusi lainnya.

Keamanan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar keamanan.

- Amazon Cognito mengautentikasi dan mengotorisasi pengguna aplikasi UI web.
- Semua komunikasi antar layanan menggunakan peran AWS Identity and Access Management (IAM) yang berlaku.
- Semua peran yang digunakan oleh solusi mengikuti akses hak istimewa paling sedikit. Mereka hanya berisi izin minimum yang diperlukan untuk menyelesaikan tugas.
- Semua penyimpanan data, termasuk bucket S3, mengenkripsi data saat istirahat.
- Kumpulan pengguna Amazon Cognito mengelola akses pengguna ke konsol dan titik akhir API Gateway tester beban terdistribusi.
- Pencatatan, penelusuran, dan pembuatan versi diaktifkan jika berlaku.
- Akses jaringan bersifat pribadi secara default dengan titik akhir Amazon Virtual Private Cloud (Amazon VPC) diaktifkan jika tersedia.

> **Note**
>
> Pengujian Beban Terdistribusi di AWS membuat beberapa grup CloudWatch log berdasarkan layanan yang digunakan. Periode retensi log untuk log bervariasi berdasarkan penyimpanan dan biaya volume peristiwa log yang dihasilkan. Solusinya membuat log wawasan kontainer untuk tugas ECS; log ini memiliki retensi log yang dikonfigurasi hingga 1 hari. Log yang dibuat untuk layanan AWS Step Functions, log kustom ECS Load Testing, dan log API Gateway dikonfigurasi dengan periode retensi 1 tahun. Log runtime AWS Lambda memiliki retensi log yang dikonfigurasi hingga 2 tahun. Log eksekusi API Gateway memiliki retensi log yang dikonfigurasi agar tidak pernah kedaluwarsa. Anda dapat mengubah periode penyimpanan log berdasarkan kebutuhan Anda di CloudWatch konsol.

Keandalan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar keandalan.

- Solusinya menggunakan layanan tanpa server AWS sedapat mungkin (contoh: Lambda, API Gateway, Amazon S3, AWS Step Functions, Amazon DynamoDB, dan AWS Fargate) untuk memastikan ketersediaan dan pemulihan yang tinggi dari kegagalan layanan.
- Semua pemrosesan komputasi menggunakan fungsi Lambda atau Amazon ECS di AWS Fargate.
- Data disimpan di DynamoDB dan Amazon S3, sehingga tetap ada di beberapa Availability Zone secara default.

Efisiensi kinerja

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar efisiensi kinerja.

- Solusinya menggunakan arsitektur tanpa server dengan kemampuan untuk menskalakan secara horizontal sesuai kebutuhan.
- Solusi ini dapat diluncurkan di Wilayah mana pun yang mendukung layanan AWS dalam solusi ini, seperti: AWS Lambda, Amazon API Gateway, Amazon S3, AWS Step Functions, Amazon DynamoDB, Amazon ECS, AWS Fargate, dan Amazon Cognito.
- Solusi ini menggunakan layanan terkelola secara keseluruhan untuk mengurangi beban operasional penyediaan dan manajemen sumber daya.
- Solusi ini secara otomatis diuji dan diterapkan setiap hari untuk mencapai konsistensi seiring perubahan layanan AWS, serta ditinjau oleh arsitek solusi dan pakar materi pelajaran untuk area yang dapat dicoba dan ditingkatkan.

Optimalisasi biaya

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik dari pilar pengoptimalan biaya.

- Solusinya menggunakan arsitektur tanpa server; oleh karena itu, pelanggan hanya dikenakan biaya untuk apa yang mereka gunakan.
- Amazon DynamoDB menskalakan kapasitas sesuai permintaan, jadi Anda hanya membayar untuk kapasitas yang Anda gunakan.
- AWS ECS di AWS Fargate memungkinkan Anda membayar hanya untuk sumber daya komputasi yang Anda gunakan, tanpa biaya di muka.

Keberlanjutan

Bagian ini menjelaskan bagaimana kami merancang solusi ini menggunakan prinsip dan praktik terbaik pilar keberlanjutan.

- Solusi ini menggunakan layanan tanpa server terkelola untuk meminimalkan dampak lingkungan dari layanan backend dibandingkan dengan layanan lokal yang terus beroperasi.
- Layanan tanpa server memungkinkan Anda untuk meningkatkan atau menurunkan skala sesuai kebutuhan.

Detail arsitektur

Bagian ini menjelaskan komponen dan layanan AWS yang membentuk solusi ini dan detail arsitektur tentang cara komponen ini bekerja sama.

Solusi Pengujian Beban Terdistribusi pada AWS terdiri dari dua komponen tingkat tinggi: ujung depan dan backend.

Ujung depan

Bagian depan terdiri dari API pengujian beban dan konsol web yang Anda gunakan untuk berinteraksi dengan backend solusi.

Muat pengujian API

Pengujian Beban Terdistribusi di AWS mengonfigurasi Amazon API Gateway untuk meng-host RESTful API solusi. Pengguna dapat berinteraksi dengan data pengujian secara aman melalui konsol web dan RESTful API yang disertakan. API bertindak sebagai “pintu depan” untuk akses ke data pengujian yang disimpan di Amazon DynamoDB. Anda juga dapat menggunakan APIs untuk mengakses fungsionalitas tambahan apa pun yang Anda buat ke dalam solusi.

Solusi ini memanfaatkan fitur otentikasi pengguna kumpulan pengguna Amazon Cognito. Setelah berhasil mengautentikasi pengguna, Amazon Cognito mengeluarkan token web JSON yang digunakan untuk mengizinkan konsol mengirimkan permintaan ke solusi (titik akhir Amazon API APIs Gateway). Permintaan HTTPS dikirim oleh konsol ke APIs header otorisasi yang menyertakan token.

Berdasarkan permintaan tersebut, API Gateway memanggil fungsi AWS Lambda yang sesuai untuk melakukan tugas yang diperlukan pada data yang disimpan dalam tabel DynamoDB, menyimpan skenario pengujian sebagai objek JSON di Amazon S3, mengambil gambar metrik CloudWatch Amazon, dan mengirimkan skenario pengujian ke mesin status AWS Step Functions.

Untuk informasi selengkapnya tentang API solusi, lihat bagian API pengujian beban terdistribusi dari panduan ini.

Konsol web

Solusi ini mencakup konsol web yang dapat Anda gunakan untuk mengonfigurasi dan menjalankan pengujian, memantau pengujian yang sedang berjalan, dan melihat hasil pengujian terperinci. Konsol adalah aplikasi ReactJS yang dihosting di Amazon S3 dan diakses melalui Amazon.

CloudFront Aplikasi ini memanfaatkan AWS Amplify untuk berintegrasi dengan Amazon Cognito untuk mengautentikasi pengguna. Konsol web juga berisi opsi untuk melihat data langsung untuk pengujian yang sedang berjalan, di mana ia berlangganan topik yang sesuai di AWS IoT Core.

Konsol web dirancang untuk menunjukkan bagaimana Anda dapat berinteraksi dengan solusi pengujian beban ini. Dalam lingkungan produksi, kami sarankan untuk menyesuaikan konsol web untuk memenuhi kebutuhan spesifik Anda atau membuat konsol Anda sendiri.

URL konsol web adalah nama domain CloudFront distribusi yang dapat ditemukan di CloudFormation output sebagai Konsol. Setelah Anda meluncurkan CloudFormation template, Anda juga akan menerima email yang berisi URL konsol web dan kata sandi satu kali untuk masuk ke dalamnya.

Backend

Backend terdiri dari pipa gambar kontainer dan mesin pengujian beban yang Anda gunakan untuk menghasilkan beban untuk pengujian. Anda berinteraksi dengan backend melalui ujung depan. Selain itu, Amazon ECS pada tugas AWS Fargate yang diluncurkan untuk setiap pengujian ditandai dengan pengenal pengujian (ID) unik. Tag ID uji ini dapat digunakan untuk membantu Anda memantau biaya untuk solusi ini. Untuk informasi tambahan, lihat Tag Alokasi Biaya yang Ditentukan Pengguna di Panduan Pengguna AWS Billing and Cost Management.

Pipa gambar kontainer

Solusi ini memanfaatkan gambar kontainer yang dibangun AmazonLinuxsebagai gambar dasar dengan kerangka pengujian beban blazemeter terpasang. Gambar ini di-host di repositori publik Amazon Elastic Container Registry (Amazon ECR). Gambar digunakan untuk menjalankan tugas di Amazon ECS di klaster AWS Fargate.

Untuk informasi selengkapnya, lihat bagian kustomisasi gambar Container dari panduan ini.

Infrastruktur pengujian

Selain template utama, solusi membuat template sekunder untuk meluncurkan sumber daya yang diperlukan untuk menjalankan pengujian di beberapa Wilayah. Template disimpan di Amazon S3, dan tautan ke template disediakan di konsol web. Template sekunder membuat VPC, kluster AWS Fargate, dan fungsi Lambda untuk memproses data langsung.

Untuk informasi selengkapnya tentang cara meluncurkan Wilayah sekunder, lihat bagian penyebaran Multi-Region dari panduan ini.

Mesin uji beban

Solusi Pengujian Beban Terdistribusi menggunakan Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS) dan AWS Fargate untuk mensimulasikan ribuan pengguna yang terhubung, di beberapa Wilayah, menghasilkan sejumlah transaksi per detik tertentu.

Anda menentukan parameter untuk tugas yang akan dijalankan sebagai bagian dari pengujian menggunakan konsol web yang disertakan. Solusinya menggunakan parameter ini untuk menghasilkan skenario pengujian JSON dan menyimpannya di Amazon S3.

Mesin status AWS Step Functions menjalankan dan memantau tugas Amazon ECS di klaster AWS Fargate. Mesin status AWS Step Functions mencakup fungsi AWS Lambda ecr-checker, fungsi AWS Lambda, fungsi AWS Lambda pelari tugas, fungsi task-status-checker AWS Lambda pembatal tugas, dan fungsi AWS Lambda parser hasil. Untuk informasi selengkapnya tentang alur kerja, lihat bagian Alur kerja Uji panduan ini. Untuk informasi lebih lanjut tentang hasil tes, lihat bagian Hasil tes dari panduan ini. Untuk informasi selengkapnya tentang alur kerja pembatalan pengujian, lihat bagian Alur kerja pembatalan pengujian pada panduan ini.

Jika Anda memilih data langsung, solusi akan memulai fungsi real-time-data-publisher Lambda di setiap Wilayah dengan CloudWatch log yang sesuai dengan tugas Fargate di Wilayah tersebut. Solusinya kemudian memproses dan menerbitkan data ke topik di AWS IoT Core dalam Wilayah tempat Anda meluncurkan tumpukan utama. Untuk informasi selengkapnya, lihat bagian Data langsung dari panduan ini.

Layanan AWS dalam solusi ini

Layanan AWS berikut disertakan dalam solusi ini:

AWS service	Deskripsi
Amazon API Gateway	Inti. Menghosting titik akhir REST API dalam solusi.
AWS CloudFormation	Inti. Mengelola penyebaran untuk infrastruktur solusi.
Amazon CloudFront	Inti. Melayani konten web yang dihosting di Amazon S3.
Amazon CloudWatch	Inti. Menyimpan log dan metrik solusi.
Amazon Cognito	Inti. Menangani manajemen pengguna dan otentikasi untuk API.

AWS service	Deskripsi
Amazon DynamoDB	Inti. Menyimpan informasi penyebaran dan menguji detail skenario dan hasil.
Layanan Kontainer Elastis Amazon	Inti. Menerapkan dan mengelola tugas Amazon ECS independen di kontainer AWS Fargate.
AWS Fargate	Inti. Wadah Amazon ECS solusi host
AWS Identity and Access Management	Inti. Menangani peran pengguna dan manajemen izin.
AWS Lambda	Inti. Menyediakan logika untuk APIs implementasi, menguji hasil parsing, dan meluncurkan workers/leader tugas.
AWS Step Functions	Inti. Mengatur penyediaan kontainer Amazon ECS pada tugas AWS Fargate di wilayah yang ditentukan
AWS Amplify	Mendukung. Menyediakan konsol web yang didukung oleh AWS Amplify.
CloudWatch Acara Amazon	Mendukung. Menjadwalkan tes untuk secara otomatis dimulai pada tanggal tertentu atau pada tanggal berulang.
Amazon Elastic Container Registry	Mendukung. Menghosting gambar kontainer di repositori ECR publik.
AWS IoT Core	Mendukung. Memungkinkan melihat data langsung untuk pengujian yang sedang berjalan dengan berlangganan topik terkait di AWS IoT Core.
AWS Systems Manager	Mendukung. Menyediakan pemantauan sumber daya tingkat aplikasi dan visualisasi operasi sumber daya dan data biaya.
Amazon S3	Mendukung. Menghosting konten web statis, log, metrik, dan data pengujian.
Amazon Virtual Private Cloud	Mendukung. Berisi wadah Amazon ECS solusi yang berjalan di AWS Fargate.

Cara Kerja Pengujian Beban Terdistribusi di AWS

Rincian rinci berikut menunjukkan langkah-langkah yang terlibat dalam menjalankan skenario pengujian.

Alur kerja uji

1. Anda menggunakan konsol web untuk mengirimkan skenario pengujian yang menyertakan detail konfigurasi ke API solusi.
2. Konfigurasi skenario pengujian diunggah ke Amazon Simple Storage Service (Amazon S3) sebagai file JSON (). `s3://<bucket-name>/test-scenarios/<$TEST_ID>/<$TEST_ID>.json`
3. Mesin status AWS Step Functions berjalan menggunakan ID pengujian, jumlah tugas, jenis pengujian, dan jenis file sebagai input mesin status AWS Step Functions. Jika pengujian dijadwalkan, pengujian akan membuat aturan CloudWatch Acara terlebih dahulu, yang memicu AWS Step Functions pada tanggal yang ditentukan. Untuk detail selengkapnya tentang alur kerja penjadwalan, lihat bagian Alur kerja penjadwalan pengujian pada panduan ini.
4. Detail konfigurasi disimpan dalam tabel skenario Amazon DynamoDB.
5. Dalam alur kerja runner tugas AWS Step Functions, fungsi AWS task-status-checker Lambda memeriksa apakah tugas Amazon Elastic Container Service (Amazon ECS) Container Service (Amazon ECS) sudah berjalan untuk ID pengujian yang sama. Jika tugas dengan ID pengujian yang sama ditemukan berjalan, itu menyebabkan kesalahan. Jika tidak ada tugas Amazon ECS yang berjalan di klaster AWS Fargate, fungsi akan mengembalikan ID pengujian, jumlah tugas, dan jenis pengujian.
6. Fungsi AWS Lambda pelari tugas mendapatkan detail tugas dari langkah sebelumnya dan menjalankan tugas pekerja Amazon ECS di klaster AWS Fargate. Amazon ECS API menggunakan RunTask tindakan untuk menjalankan tugas pekerja. Tugas pekerja ini diluncurkan dan kemudian menunggu pesan awal dari tugas pemimpin untuk memulai tes. RunTask Tindakan ini dibatasi hingga 10 tugas per definisi. Jika jumlah tugas Anda lebih dari 10, definisi tugas akan berjalan beberapa kali hingga semua tugas pekerja dimulai. Fungsi ini juga menghasilkan awalan untuk membedakan pengujian saat ini dalam fungsi AWS Lambda yang mengurai hasil.
7. Fungsi task-status-checker AWS Lambda memeriksa apakah semua tugas pekerja Amazon ECS berjalan dengan ID pengujian yang sama. Jika tugas masih disediakan, ia menunggu selama satu menit dan memeriksa lagi. Setelah semua tugas Amazon ECS berjalan, ia mengembalikan ID

pengujian, jumlah tugas, jenis pengujian, semua tugas IDs dan awalan dan meneruskannya ke fungsi task-runner.

8. Fungsi AWS Lambda pelari tugas berjalan lagi, kali ini meluncurkan satu tugas Amazon ECS untuk bertindak sebagai node pemimpin. Tugas ECS ini mengirimkan pesan uji awal ke setiap tugas pekerja untuk memulai pengujian secara bersamaan.
9. Fungsi task-status-checker AWS Lambda kembali memeriksa apakah tugas Amazon ECS berjalan dengan ID pengujian yang sama. Jika tugas masih berjalan, ia menunggu selama satu menit dan memeriksa lagi. Setelah tidak ada tugas Amazon ECS yang berjalan, ia mengembalikan ID pengujian, jumlah tugas, jenis pengujian, dan awalan.
10. Saat fungsi AWS Lambda pelari tugas menjalankan tugas Amazon ECS di klaster AWS Fargate, setiap tugas mengunduh konfigurasi pengujian dari Amazon S3 dan memulai pengujian.
11. Setelah pengujian berjalan, waktu respons rata-rata, jumlah pengguna bersamaan, jumlah permintaan yang berhasil, dan jumlah permintaan yang gagal untuk setiap tugas dicatat di Amazon CloudWatch dan dapat dilihat di CloudWatch dasbor.
12. Jika Anda menyertakan data langsung dalam pengujian, solusi akan memfilter hasil pengujian real-time CloudWatch menggunakan filter langganan. Kemudian solusinya meneruskan data ke fungsi Lambda.
13. Fungsi Lambda kemudian menyusun data yang diterima dan menerbitkannya ke topik AWS IoT Core.
14. Konsol web berlangganan topik AWS IoT Core untuk pengujian dan menerima data yang dipublikasikan ke topik untuk membuat grafik data waktu nyata saat pengujian sedang berjalan.
15. Saat pengujian selesai, gambar kontainer mengekspor laporan terperinci sebagai file XHTML ke Amazon S3. Setiap file diberi UUID untuk nama file. Misalnya, s3://dlte-bucket/test-scenarios/ <$TEST_ID> /results/ <$UUID> .json.
16. Saat file XHTML diunggah ke Amazon S3, fungsi AWS Lambda parser hasil membaca hasil dalam file XMLdimulai dengan awalan dan mem-parsing dan menggabungkan semua hasil menjadi satu hasil yang diringkas.
17. Fungsi AWS Lambda parser hasil menulis hasil agregat ke tabel Amazon DynamoDB.

Pertimbangan desain

Aplikasi-aplikasi yang didukung

Solusi ini mendukung aplikasi berbasis cloud, dan aplikasi lokal selama Anda memiliki koneksi jaringan dari akun AWS ke aplikasi Anda. Solusinya APIs mendukung penggunaan HTTP atau HTTPS. Anda juga memiliki kontrol atas header permintaan HTTP, sehingga Anda dapat menambahkan otorisasi atau header khusus untuk meneruskan token atau kunci API.

JMeter dukungan skrip

Saat membuat skenario pengujian menggunakan antarmuka pengguna (UI) solusi ini, Anda dapat menggunakan skrip JMeter pengujian. Setelah memilih file JMeter skrip, file tersebut diunggah ke bucket <stack-name>-scenariosbucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3). Saat tugas Amazon Elastic Container Service (Amazon ECS) berjalan, JMeter skrip akan diunduh dari <stack-name>bucket Amazon S3 -scenariosbucket dan pengujian berjalan.

Jika Anda memiliki file JMeter input, Anda dapat zip file input bersama dengan JMeter skrip. Anda dapat memilih file zip saat Anda membuat skenario pengujian.

Jika Anda ingin menyertakan plugin, file.jar apa pun yang disertakan dalam subdirektori /plugins dalam file zip yang dibundel akan disalin ke direktori JMeter ekstensi dan tersedia untuk pengujian beban.

> **Note**
>
> Jika Anda menyertakan file JMeter input dengan file JMeter skrip Anda, Anda harus menyertakan jalur relatif dari file input dalam file JMeter skrip Anda. Selain itu, file input harus berada di jalur relatif. Misalnya, ketika file JMeter input dan file skrip Anda berada di/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 sebagai gantinya, pengujian akan gagal karena tidak akan dapat menemukan file input.

Jika Anda menyertakan JMeter plugin, file.jar harus dibundel dalam subdirektori bernama /plugins dalam root file zip. Sehubungan dengan root file zip, jalur ke file jar harus. /plugins/bundled_plugin.jar.

Untuk informasi selengkapnya tentang cara menggunakan JMeter skrip, lihat Panduan JMeter Pengguna.

Dukungan skrip K6

Solusinya mendukung pengujian berbasis kerangka kerja K6. K6 dirilis dengan lisensi AGPL-3.0. Solusinya menampilkan pesan pengakuan lisensi saat membuat pengujian baru untuk K6. File uji K6 bersama dengan file input yang diperlukan dapat disertakan dalam file arsip dan diunggah untuk skenario pengujian menggunakan opsi unggah.

Dukungan skrip belalang

Solusinya mendukung pengujian berbasis kerangka kerja Locust. File pengujian Locust bersama dengan file input yang diperlukan dapat disertakan dalam file arsip dan diunggah untuk skenario pengujian menggunakan opsi unggah.

Tes penjadwalan

Anda dapat menjadwalkan tes untuk dijalankan di masa depan atau menggunakan opsi Run Now. Anda dapat menjadwalkan pengujian sebagai satu kali dijalankan di masa mendatang atau menyiapkan pengujian berulang di mana Anda menentukan tanggal jalankan pertama, dan pengulangan yang direncanakan. Opsi untuk kekambuhan meliputi: harian, mingguan, dua mingguan, dan bulanan. Untuk informasi selengkapnya tentang cara kerja penjadwalan, lihat bagian Alur kerja penjadwalan pengujian pada panduan ini.

Mulai versi 3.3.0, Pengujian Beban Terdistribusi di AWS memungkinkan pengguna menjadwalkan pengujian pemuatan menggunakan ekspresi cron. Pilih Run on Schedule dan kemudian tab CRON untuk memasukkan nilai cron secara manual atau menggunakan bidang drop-down. cronExpiryDate Harus sesuai dengan tanggal uji coba yang dijadwalkan. Tinjau Tanggal Jalankan Berikutnya (UTC) untuk mengonfirmasi jadwal Anda.

> **Note**
>
> - Durasi tes: Pertimbangkan total durasi tes saat menjadwalkan. Misalnya, tes dengan waktu ramp-up 10 menit dan waktu penahanan 40 menit akan memakan waktu sekitar 80 menit untuk menyelesaikannya.
> - Interval minimum: Pastikan interval antara tes terjadwal lebih lama dari perkiraan durasi tes. Misalnya, jika tes memakan waktu sekitar 80 menit, jadwalkan untuk berjalan tidak lebih sering dari setiap 3 jam.

- Batasan per jam: Sistem tidak mengizinkan tes dijadwalkan dengan perbedaan hanya satu jam meskipun perkiraan durasi tes kurang dari satu jam.

Tes bersamaan

Solusi ini mencakup CloudWatch dasbor Amazon untuk setiap pengujian dan menampilkan output gabungan dari semua tugas yang berjalan untuk pengujian tersebut di cluster Amazon ECS secara real-time. CloudWatch Dasbor menampilkan waktu respons rata-rata, jumlah pengguna bersamaan, jumlah permintaan yang berhasil, dan jumlah permintaan yang gagal. Setiap metrik dikumpulkan oleh yang kedua, dan dasbor diperbarui setiap menit.

Manajemen pengguna

Selama konfigurasi awal, Anda memberikan nama pengguna dan alamat email yang digunakan Amazon Cognito untuk memberi Anda akses ke konsol web solusi. Konsol tidak menyediakan administrasi pengguna. Untuk menambahkan pengguna tambahan, Anda harus menggunakan konsol Amazon Cognito. Untuk informasi selengkapnya, lihat Mengelola Pengguna di Kumpulan Pengguna di Panduan Pengembang Amazon Cognito.

Untuk memigrasikan pengguna yang ada ke kumpulan pengguna Amazon Cognito, lihat Pendekatan blog AWS untuk memigrasikan pengguna ke kumpulan pengguna Amazon Cognito.

Penyebaran regional

Solusi ini menggunakan Amazon Cognito yang hanya tersedia di Wilayah AWS tertentu. Oleh karena itu, Anda harus menerapkan solusi ini di wilayah tempat Amazon Cognito tersedia. Untuk ketersediaan layanan terbaru menurut Wilayah, lihat Daftar Layanan Regional AWS.

Rencanakan penyebaran Anda

Bagian ini menjelaskan biaya, keamanan, Wilayah, dan pertimbangan lain sebelum menerapkan solusi.

Biaya

Anda bertanggung jawab atas biaya layanan AWS yang digunakan saat menjalankan solusi ini. Total biaya untuk menjalankan solusi ini tergantung pada jumlah uji beban yang dijalankan, durasi uji beban tersebut, dan jumlah data yang digunakan sebagai bagian dari pengujian. Pada revisi ini, biaya untuk menjalankan solusi ini dengan pengaturan default di Wilayah AS Timur (Virginia N.) adalah sekitar $30,90 per bulan.

Tabel berikut memberikan rincian biaya sampel untuk menerapkan solusi ini dengan parameter default di Wilayah AS Timur (Virginia N.) selama satu bulan.

AWS service	Dimensi	Biaya [USD]
AWS Fargate	10 tugas sesuai permintaan (menggunakan dua memori v CPUs dan 4 GB) berjalan selama 30 jam	$29,62
Amazon DynamoDB	1.000 unit kapasitas tulis sesuai permintaan 1.000 unit kapasitas baca sesuai permintaan	$0.0015
AWS Lambda	1.000 permintaan Total durasi 10 menit	$1,25
AWS Step Functions	1.000 transisi negara	$0,025
Jumlah:		$30,90 per bulan

Sumber daya solusi ditandai dengan Key= SolutionId dan value=SO0062. Anda dapat mengaktifkan kunci tag SolutionId dengan mengikuti tag pengaktifan dokumentasi. Setelah tag diaktifkan, Anda dapat membuat aturan kategori biaya dengan mengikuti dokumentasi untuk membuat kategori biaya. Anda dapat melihat biaya yang dikeluarkan untuk solusi dengan memantau konsol kategori biaya dan memilih nama kategori biaya.

Sebaiknya buat anggaran melalui AWS Cost Explorer untuk membantu mengelola biaya. Harga dapat berubah sewaktu-waktu. Untuk detail selengkapnya, lihat halaman web harga untuk setiap layanan AWS yang digunakan dalam solusi ini.

Important

Mulai versi 1.3.0, CPU ditingkatkan menjadi 2 vCPU dan memori ditingkatkan menjadi 4 GB. Perubahan ini meningkatkan perkiraan biaya dibandingkan dengan versi sebelumnya dari solusi ini. Jika pengujian beban Anda tidak memerlukan peningkatan ini ke sumber daya AWS Anda, Anda dapat menguranginya. Untuk informasi tambahan, lihat bagian Meningkatkan sumber daya kontainer dalam panduan ini.

Note

Solusi ini menyediakan opsi untuk menyertakan data langsung saat menjalankan pengujian. Fitur ini memerlukan fungsi AWS Lambda tambahan dan topik AWS IoT Core yang menimbulkan biaya tambahan.

Harga dapat berubah sewaktu-waktu. Untuk detail selengkapnya, lihat halaman web harga untuk setiap layanan AWS yang akan Anda gunakan dalam solusi ini.

Keamanan

Saat Anda membangun sistem pada infrastruktur AWS, tanggung jawab keamanan dibagi antara Anda dan AWS. Model tanggung jawab bersama ini mengurangi beban operasional Anda karena AWS mengoperasikan, mengelola, dan mengontrol komponen termasuk sistem operasi host, lapisan virtualisasi, dan keamanan fisik fasilitas tempat layanan beroperasi. Untuk informasi selengkapnya tentang keamanan AWS, kunjungi AWS Cloud Security.

Peran IAM

Peran AWS Identity and Access Management (IAM) memungkinkan pelanggan menetapkan kebijakan akses terperinci dan izin untuk layanan dan pengguna di AWS Cloud. Solusi ini menciptakan peran IAM yang memberikan akses fungsi AWS Lambda solusi untuk membuat sumber daya Regional.

Amazon CloudFront

Solusi ini menerapkan UI web yang dihosting di bucket Amazon S3, yang didistribusikan oleh Amazon. CloudFront Untuk membantu mengurangi latensi dan meningkatkan keamanan, solusi ini mencakup CloudFront distribusi dengan identitas akses asal, yaitu CloudFront pengguna yang menyediakan akses publik ke konten bucket situs web solusi. Secara default, CloudFront distribusi menggunakan TLS 1.2 untuk menegakkan protokol keamanan tingkat tertinggi. Untuk informasi selengkapnya, lihat Membatasi akses ke asal Amazon S3 di Panduan Pengembang CloudFront Amazon.

CloudFront mengaktifkan mitigasi keamanan tambahan untuk menambahkan header keamanan HTTP ke setiap respons penampil. Untuk informasi selengkapnya, lihat Menambahkan atau menghapus header HTTP dalam CloudFront tanggapan.

Solusi ini menggunakan CloudFront sertifikat default, yang memiliki protokol keamanan minimum yang didukung TLS v1.0. Untuk menegakkan penggunaan TLS v1.2 atau TLS v1.3, Anda harus menggunakan sertifikat SSL kustom alih-alih sertifikat default. CloudFront Untuk informasi selengkapnya, lihat Bagaimana cara mengonfigurasi CloudFront distribusi saya untuk menggunakan SSL/TLS sertifikat.

Grup keamanan AWS Fargate

Secara default, solusi ini membuka aturan keluar grup keamanan AWS Fargate kepada publik. Jika Anda ingin memblokir AWS Fargate agar tidak mengirim lalu lintas ke mana pun, ubah aturan keluar ke Perutean Antar-Domain Tanpa Kelas (CIDR) tertentu.

Grup keamanan ini juga menyertakan aturan masuk yang memungkinkan lalu lintas lokal di port 50.000 ke sumber apa pun yang termasuk dalam grup keamanan yang sama. Ini digunakan untuk memungkinkan wadah berkomunikasi satu sama lain.

Tes stress jaringan

Anda bertanggung jawab untuk menggunakan solusi ini di bawah kebijakan Uji Stres Jaringan. Kebijakan ini mencakup situasi seperti ketika Anda berencana untuk menjalankan pengujian jaringan volume tinggi langsung dari instans Amazon Anda ke lokasi lain seperti EC2 instans Amazon lainnya, properti/layanan AWS EC2 , atau titik akhir eksternal. Tes ini kadang-kadang disebut stress test, load test, atau gameday test. Sebagian besar pengujian pelanggan tidak akan termasuk dalam kebijakan ini; namun, lihat kebijakan ini jika Anda yakin Anda akan menghasilkan lalu lintas yang menopang, secara agregat, selama lebih dari 1 menit, lebih dari 1 Gbps (1 miliar bit per detik) atau lebih dari 1 Gpps (1 miliar paket per detik).

Membatasi akses ke antarmuka pengguna publik

Untuk membatasi akses ke antarmuka pengguna yang menghadap publik di luar mekanisme autentikasi dan otorisasi yang disediakan oleh IAM dan Amazon Cognito, gunakan solusi Otomasi Keamanan AWS WAF (firewall aplikasi web).

Solusi ini secara otomatis menerapkan seperangkat aturan AWS WAF yang memfilter serangan berbasis web umum. Pengguna dapat memilih dari fitur pelindung yang telah dikonfigurasi sebelumnya yang menentukan aturan yang disertakan dalam daftar kontrol akses web AWS WAF (web ACL).

Wilayah AWS yang Didukung

Solusi ini menggunakan layanan Amazon Cognito, yang saat ini tidak tersedia di semua Wilayah AWS. Untuk ketersediaan terbaru layanan AWS menurut Wilayah, lihat Daftar Layanan Regional AWS.

Pengujian Beban Terdistribusi di AWS tersedia di Wilayah AWS berikut:

Nama wilayah	
AS Timur (Ohio)	Asia Pasifik (Tokyo)
AS Timur (Virginia Utara)	Kanada (Pusat)
AS Barat (California Utara)	Eropa (Frankfurt)
AS Barat (Oregon)	Eropa (Irlandia)

Nama wilayah	
Asia Pasifik (Mumbai)	Eropa (London)
Asia Pasifik (Seoul)	Eropa (Paris)
Asia Pasifik (Singapura)	Eropa (Stockholm)
Asia Pasifik (Sydney)	Amerika Selatan (Sao Paulo)

Kuota

Service quotas, juga disebut batasan, adalah jumlah maksimum sumber daya layanan atau operasi untuk akun AWS Anda.

Kuota untuk layanan AWS dalam solusi ini

Pastikan Anda memiliki kuota yang cukup untuk setiap layanan yang diterapkan dalam solusi ini. Untuk informasi lebih lanjut, lihat Service quotas AWS.

Gunakan tautan berikut untuk membuka halaman untuk layanan itu. Untuk melihat kuota layanan untuk semua layanan AWS dalam dokumentasi tanpa berpindah halaman, lihat informasi di titik akhir Layanan dan halaman kuota di PDF sebagai gantinya.

CloudFormation Kuota AWS

Akun AWS Anda memiliki CloudFormation kuota AWS yang harus Anda ketahui saat meluncurkan tumpukan dalam solusi ini. Dengan memahami kuota ini, Anda dapat menghindari kesalahan pembatasan yang akan mencegah Anda menerapkan solusi ini dengan sukses. Untuk informasi selengkapnya, lihat CloudFormation kuota AWS di Panduan CloudFormation Pengguna AWS.

Kuota pengujian beban

Jumlah maksimum tugas yang dapat dijalankan di Amazon ECS menggunakan jenis peluncuran AWS Fargate didasarkan pada ukuran tugas vCPU. Ukuran tugas default dalam Pengujian Beban Terdistribusi di AWS adalah 2 vCPU. Untuk melihat kuota default saat ini, lihat kuota layanan Amazon ECS. Kuota akun saat ini mungkin berbeda dari kuota yang tercantum. Untuk memeriksa kuota khusus untuk akun, periksa kuota layanan untuk jumlah sumber daya vCPU sesuai permintaan

Fargate di AWS Management Console. Untuk petunjuk tentang cara meminta peningkatan, lihat kuota layanan AWS di Panduan Referensi Umum AWS.

AmazonLinux Gambar kontainer gambar (dengan Blazemeter diinstal) tidak membatasi koneksi bersamaan per tugas, tetapi itu tidak berarti bahwa itu dapat mendukung jumlah pengguna yang tidak terbatas. Untuk menentukan jumlah pengguna bersamaan yang dapat dihasilkan kontainer untuk pengujian, lihat bagian Tentukan jumlah pengguna dari panduan ini.

> **Note**
>
> Batas yang disarankan untuk pengguna bersamaan berdasarkan pengaturan default adalah 200 pengguna.

Tes bersamaan

Solusi ini mencakup CloudWatch dasbor Amazon untuk setiap pengujian dan menampilkan output gabungan dari semua tugas yang berjalan untuk pengujian tersebut di cluster Amazon ECS secara real-time. CloudWatch Dasbor menampilkan waktu respons rata-rata, jumlah pengguna bersamaan, jumlah permintaan yang berhasil, dan jumlah permintaan yang gagal. Setiap metrik dikumpulkan oleh yang kedua, dan dasbor diperbarui setiap menit.

Kebijakan EC2 pengujian Amazon

Anda tidak memerlukan persetujuan dari AWS untuk menjalankan pengujian beban menggunakan solusi ini selama lalu lintas jaringan Anda tetap di bawah 1 Gbps. Jika pengujian Anda akan menghasilkan lebih dari 1 Gbps, hubungi AWS. Untuk informasi selengkapnya, lihat Kebijakan EC2 Pengujian Amazon.

Kebijakan pengujian CloudFront beban Amazon

Jika Anda berencana untuk menguji beban CloudFront titik akhir, lihat pedoman pengujian beban di Panduan CloudFront Pengembang Amazon. Kami juga merekomendasikan penyebaran lalu lintas di beberapa tugas dan Wilayah. Berikan setidaknya 30 menit waktu ramp-up untuk uji beban. Untuk tes beban yang mengirimkan lebih dari 500.000 permintaan per detik atau menuntut lebih dari 300 Gbps data, kami sarankan terlebih dahulu mendapatkan pra-persetujuan untuk mengirim lalu lintas. CloudFront dapat membatasi lalu lintas uji beban yang tidak disetujui yang memengaruhi CloudFront ketersediaan layanan.

Memantau solusi pasca penerapan

Mengaktifkan atau mengatur CloudWatch Alarm

Disarankan untuk terus memantau sumber daya solusi pasca-penerapan. Anda dapat mempertimbangkan untuk menyiapkan CloudWatch Metrik untuk sumber daya yang dibuat oleh solusi.

Metrik CloudFront Distribusi Amazon

Tinjau metrik CloudFront Distribusi dan atur alarm untuk memantau dan menerima notifikasi.

Metrik Amazon API Gateway

Tinjau dimensi dan metrik Amazon API Gateway dan atur alarm untuk memantau dan menerima notifikasi.

Terapkan solusinya

Solusi ini menggunakan CloudFormation templat dan tumpukan AWS untuk mengotomatiskan penerapannya. CloudFormation Template menentukan sumber daya AWS yang disertakan dalam solusi ini dan propertinya. CloudFormation Tumpukan menyediakan sumber daya yang dijelaskan dalam template.

Ikhtisar proses penyebaran

Ikuti step-by-step petunjuk di bagian ini untuk mengonfigurasi dan menyebarkan solusi ke akun Anda.

Sebelum Anda meluncurkan solusi, tinjau biaya, arsitektur, keamanan jaringan, dan pertimbangan lain yang dibahas sebelumnya dalam panduan ini.

Waktu untuk menyebarkan: Sekitar 15 menit

CloudFormation Templat AWS

Anda dapat mengunduh CloudFormation template untuk solusi ini sebelum menerapkannya. Solusi ini menggunakan AWS CloudFormation untuk mengotomatiskan penerapan Pengujian Beban Terdistribusi di AWS. Ini mencakup CloudFormation template AWS berikut, yang dapat Anda unduh sebelum penerapan:

distribu

load-testing-on-aws.template - Gunakan template ini untuk meluncurkan solusi dan semua komponen terkait. Konfigurasi default menerapkan layanan inti dan pendukung yang ditemukan di layanan AWS di bagian solusi ini, tetapi Anda dapat menyesuaikan template untuk memenuhi kebutuhan spesifik Anda.

> **Note**
>
> CloudFormation Sumber daya AWS dibuat dari konstruksi AWS Cloud Development Kit (AWS CDK). Jika sebelumnya Anda telah menerapkan solusi ini, lihat Memperbarui solusi untuk petunjuk pemutakhiran.

Luncurkan tumpukan

> **Important**
>
> Jika Anda memperbarui tumpukan dari versi sebelum v3.2.6 ke versi terbaru, baca bagian ini sebelum memperbarui tumpukan.

Sebelum Anda meluncurkan penerapan otomatis, tinjau arsitektur dan pertimbangan lain yang dibahas dalam panduan ini. Ikuti step-by-step petunjuk di bagian ini untuk mengonfigurasi dan menerapkan Pengujian Beban Terdistribusi di AWS ke akun Anda.

Waktu untuk menyebarkan: Sekitar 15 menit

> **Important**
>
> Solusi ini mencakup opsi untuk mengirim metrik operasional anonim ke AWS. Kami menggunakan data ini untuk lebih memahami bagaimana pelanggan menggunakan solusi ini dan layanan serta produk terkait. AWS memiliki data yang dikumpulkan melalui survei ini. Pengumpulan data tunduk pada Pemberitahuan Privasi AWS.
> Untuk memilih keluar dari fitur ini, unduh templat, ubah bagian CloudFormation pemetaan AWS, lalu gunakan CloudFormation konsol AWS untuk mengunggah templat Anda yang diperbarui dan menerapkan solusinya. Untuk informasi selengkapnya, lihat bagian Pengumpulan data anonim dari panduan ini.

CloudFormation Template AWS otomatis ini menerapkan Pengujian Beban Terdistribusi di AWS.

> **Note**
>
> Anda bertanggung jawab atas biaya layanan AWS yang digunakan saat menjalankan solusi ini. Untuk detail selengkapnya, kunjungi bagian Biaya dalam panduan ini dan lihat halaman web harga untuk setiap layanan AWS yang digunakan dalam solusi ini.

1. Masuk ke AWS Management Console dan pilih tombol di bawah ini untuk meluncurkan CloudFormation template distributed-load-testing-on AWS -aws.

Atau, Anda juga dapat mengunduh template sebagai titik awal untuk implementasi Anda sendiri.

2. Template diluncurkan di Wilayah AS Timur (Virginia N.) secara default. Untuk meluncurkan solusi ini di Wilayah AWS yang berbeda, gunakan pemilih wilayah di bilah navigasi konsol.

> **Note**
>
> Solusi ini menggunakan Amazon Cognito, yang saat ini hanya tersedia di Wilayah AWS tertentu. Oleh karena itu, Anda harus meluncurkan solusi ini di Wilayah AWS tempat Amazon Cognito tersedia. Untuk ketersediaan layanan terbaru menurut Wilayah, lihat Daftar Layanan Regional AWS.

3. Pada halaman Buat tumpukan, verifikasi bahwa URL templat yang benar ditampilkan di kotak teks URL Amazon S3 dan pilih Berikutnya.
4. Pada halaman Tentukan detail tumpukan, tetapkan nama ke tumpukan solusi Anda.
5. Di bawah Parameter, tinjau parameter untuk templat dan modifikasi seperlunya. Solusi ini menggunakan nilai default berikut.

Parameter	Default	Deskripsi
Nama Administrator	<Requires input>	Nama pengguna untuk administrator solusi awal.
Email Administrator	*<Requires input>*	Alamat email pengguna administrator. Setelah diluncurkan, email akan dikirim ke alamat ini dengan instruksi login konsol.
ID VPC yang ada	<Optional input>	Jika Anda memiliki VPC yang ingin Anda gunakan dan sudah dibuat, masukkan ID VPC yang ada di Wilayah yang sama tempat tumpukan digunakan. Misalnya, vpc-1a2b3c4d5e6f.

Parameter	Default	Deskripsi
Subnet pertama yang ada	<Optional input>	ID subnet pertama dalam VPC Anda yang ada. Subnet ini membutuhkan rute ke internet untuk menarik gambar kontainer untuk menjalankan pengujian. Misalnya, subnet-7h8i9j0k.
Subnet kedua yang ada	<Optional input>	ID subnet kedua dalam VPC yang ada. Subnet ini membutuhkan rute ke internet untuk menarik gambar kontainer untuk menjalank an pengujian. Misalnya, subnet-1x2y3z.
Berikan blok CIDR yang valid untuk solusi membuat VPC	192.168.0.0/16	Anda dapat membiarkan parameter ini kosong jika Anda menggunakan VPC yang ada
Berikan blok CIDR yang valid untuk subnet A untuk solusi membuat VPC	192.168.0.0/20	Blok CIDR untuk subnet A dari AWS Fargate VPC
Berikan blok CIDR yang valid untuk subnet B untuk solusi membuat VPC	192.168.16.0/20	Blok CIDR untuk subnet B dari AWS Fargate VPC
Menyediakan blok CIDR untuk memungkinkan lalu lintas keluar dari tugas Fargate	0.0.0.0/0	Blok CIDR yang membatasi akses keluar kontainer Amazon ECS.

Parameter	Default	Deskripsi
Perbarui Gambar Kontainer Otomatis	`Yes`	Secara otomatis menggunakan gambar yang paling mutakhir dan aman hingga rilis minor berikutnya. Memilih No akan menarik gambar seperti yang dirilis semula, tanpa pembaruan keamanan apa pun.

6. Pilih Berikutnya.
7. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan. Centang kotak yang menyatakan bahwa template akan membuat sumber daya AWS Identity and Access Management (IAM).
9. Pilih Membuat tumpukan untuk menerapkannya.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status CREATE_COMPLETE dalam waktu sekitar 15 menit.

> **Note**
>
> Selain fungsi AWS Lambda utama, solusi ini mencakup fungsi Lambda sumber daya khusus, yang hanya berjalan selama konfigurasi awal atau saat sumber daya diperbarui atau dihapus.

Saat menjalankan solusi ini, fungsi Lambda sumber daya khusus tidak aktif. Namun, jangan hapus fungsi ini karena perlu untuk mengelola sumber daya terkait.

Penyebaran Multi-Wilayah

Waktu untuk menyebarkan: Sekitar lima menit

Anda dapat menjalankan pengujian di beberapa Wilayah. Saat Anda menerapkan solusi Pengujian Beban Terdistribusi, solusi ini akan membuat tiga bucket Amazon S3. Solusinya membuat tumpukan regional sekunder dan menyimpannya di bucket skenario Amazon S3.

> **Note**
>
> Konvensi penamaan bucket adalah ***<stack-name>*** -` dlttestrunnerstoragedltscenariosbucket ***<_[0-9][0-9]..-<[0-9][0-9].._*** dengan skenario kata kunci dalam nama bucket, yang dapat Anda temukan dengan menavigasi ke konsol S3, lalu Bucket.

Untuk menjalankan penerapan Multi-wilayah, Anda harus menerapkan CloudFormation template regional, yang disimpan di bucket skenario Amazon S3, di Wilayah tempat Anda ingin menjalankan pengujian. Anda dapat menginstal template regional dengan melakukan hal berikut:

1. Di konsol web solusi, navigasikan ke Kelola Wilayah di menu atas.
2. Gunakan ikon clipboard untuk menyalin tautan CloudFormation templat di Amazon S3.
3. Masuk ke CloudFormation konsol AWS dan pilih Wilayah yang benar.
4. Pada halaman Buat tumpukan, verifikasi bahwa URL templat yang benar ditampilkan di kotak teks URL Amazon S3 dan pilih Berikutnya.
5. Pada halaman Tentukan detail tumpukan, tetapkan nama ke tumpukan solusi Anda.
6. Di bawah Parameter, tinjau parameter untuk templat dan modifikasi seperlunya. Solusi ini menggunakan nilai default berikut.

Parameter	Default	Deskripsi
ID VPC yang ada	<Optional input>	Jika Anda memiliki VPC yang ingin Anda gunakan dan sudah dibuat, masukkan ID VPC yang ada di Wilayah yang sama tempat tumpukan digunakan. Misalnya, vpc-1a2b3c4d5e6f.
Subnet pertama yang ada	<Optional input>	ID subnet pertama dalam VPC Anda yang ada. Subnet ini membutuhkan rute ke internet untuk menarik gambar kontainer untuk

Parameter	Default	Deskripsi
		menjalankan pengujian. Misalnya, subnet-7h8i9j0k.
Subnet kedua yang ada	<Optional input>	ID subnet kedua dalam VPC yang ada. Subnet ini membutuhkan rute ke internet untuk menarik gambar kontainer untuk menjalank an pengujian. Misalnya, subnet-1x2y3z.
Berikan blok CIDR yang valid untuk solusi membuat VPC	192.168.0.0/16	Jika Anda tidak memberika n nilai untuk VPC yang ada, blok CIDR untuk Amazon VPC yang dibuat solusi berisi alamat IP untuk AWS Fargate.
Menyediakan blok CIDR untuk memungkinkan lalu lintas keluar dari tugas Fargate	0.0.0.0/0	Blok CIDR yang membatasi akses keluar kontainer Amazon ECS.

7. Pilih Berikutnya.
8. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
9. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan. Pastikan untuk mencentang kotak yang mengakui bahwa template akan membuat sumber daya AWS Identity and Access Management (IAM).
10. Pilih Membuat tumpukan untuk menerapkannya.

    Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima status CREATE_COMPLETE dalam waktu kurang lebih lima menit.

Ketika Wilayah telah berhasil digunakan, mereka akan muncul di konsol web. Saat Anda membuat pengujian, Wilayah baru akan terdaftar di modal Kelola Wilayah. Anda dapat menggunakan Wilayah ini dalam pengujian dengan memilihnya pada pembuatan pengujian. Solusinya membuat item

DynamoDB untuk setiap Wilayah yang diluncurkan dalam tabel skenario, yang berisi informasi yang diperlukan mengenai sumber daya pengujian di Wilayah tersebut. Anda dapat mengurutkan hasil pengujian di konsol web berdasarkan Wilayah. Karena kendala API, Anda dapat melihat hasil agregat semua Wilayah dalam pengujian Multi-wilayah hanya dengan membuat grafik dalam metrik Amazon. CloudWatch Anda dapat menemukan kode sumber untuk grafik dalam hasil tes setelah tes selesai.

> **Note**
>
> Anda dapat meluncurkan tumpukan regional tanpa konsol web. Dapatkan tautan ke templat regional di bucket skenario Amazon S3 dan berikan sebagai sumber saat meluncurkan tumpukan regional di Wilayah yang diperlukan. Atau, Anda dapat mengunduh templat dan mengunggahnya sebagai sumber untuk Wilayah yang Anda inginkan.

Perbarui solusinya

Jika sebelumnya Anda telah menerapkan solusi, ikuti prosedur ini untuk memperbarui CloudFormation tumpukan solusi untuk mendapatkan versi terbaru dari kerangka kerja solusi.

1. Masuk ke CloudFormation konsol, pilih CloudFormation tumpukan yang ada, dan pilih Perbarui tumpukan.
2. Pilih Buat pembaruan langsung.
3. Pilih Ganti template yang ada.
4. Di bawah Tentukan template:
   a. Pilih URL Amazon S3.
   b. Salin tautan templat terbaru.
   c. Tempel tautan di kotak URL Amazon S3.
   d. Verifikasi bahwa URL templat yang benar ditampilkan di kotak teks URL Amazon S3.
   e. Pilih Berikutnya.
   f. Pilih Selanjutnya sekali lagi.
5. Di bawah Parameter, tinjau parameter untuk templat dan modifikasi seperlunya. Lihat Luncurkan tumpukan untuk detail tentang parameter.
6. Pilih Berikutnya.
7. Pada halaman Konfigurasikan opsi tumpukan, pilih Berikutnya.
8. Pada halaman Ulasan, tinjau dan konfirmasikan pengaturan.
9. Pilih kotak yang mengakui bahwa template mungkin membuat sumber daya IAM.
10. Pilih Lihat set perubahan dan verifikasi perubahan.
11. Pilih Perbarui tumpukan untuk menyebarkan tumpukan.

Anda dapat melihat status tumpukan di CloudFormation konsol AWS di kolom Status. Anda akan menerima `UPDATE_COMPLETE` status dalam waktu sekitar 15 menit.

Saat memperbarui dari versi DLT yang lebih lama dari v3.2.6 ke v3.3.x dan v3.3.x ke yang terbaru, memperbarui tumpukan gagal

1. Unduh distributed-load-testing-on-aws.template.

2. Buka template dan navigasikan ke Kondisi: dan cari DLTCommon ResourcesAppRegistryCondition
3. Anda akan melihat sesuatu yang serupa dengan yang berikut:

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

4. Ubah nilai true kedua menjadi false:

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

5. Gunakan template yang disesuaikan untuk memperbarui tumpukan Anda.
6. Tumpukan ini menghapus sumber daya terkait registri aplikasi dari tumpukan. Oleh karena itu, pembaruan harus diselesaikan.
7. Lakukan pembaruan tumpukan lain menggunakan URL template terbaru untuk menambahkan kembali sumber daya aplikasi registri aplikasi ke tumpukan Anda.

> **Note**
>
> AWS Systems Manager Application Manager memberi Anda tampilan tingkat aplikasi ke dalam solusi ini dan sumber dayanya sehingga Anda dapat:
>
> 1. Pantau sumber dayanya, biaya untuk sumber daya yang diterapkan di seluruh tumpukan dan akun AWS, dan log yang terkait dengan solusi ini dari lokasi pusat.
> 2. Lihat data operasi untuk sumber daya solusi ini dalam konteks aplikasi, seperti status penerapan, CloudWatch alarm, konfigurasi sumber daya, dan masalah operasional.

Pemecahan Masalah

Resolusi masalah yang diketahui memberikan instruksi untuk mengurangi kesalahan yang diketahui. Jika petunjuk ini tidak mengatasi masalah Anda, Hubungi AWS Support memberikan petunjuk untuk membuka kasus AWS Support untuk solusi ini.

Resolusi masalah yang diketahui

Masalah: Anda menggunakan VPC yang ada dan pengujian Anda gagal dengan status Gagal, menghasilkan pesan galat berikut:

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

- Penyelesaian:

Pastikan bahwa subnet ada di VPC yang ditentukan dan bahwa mereka memiliki rute ke internet dengan gateway internet atau gateway NAT. AWS Fargate memerlukan akses untuk menarik image container dari repositori publik agar berhasil menjalankan pengujian.

Masalah: Pengujian memakan waktu terlalu lama untuk dijalankan atau macet berjalan tanpa batas waktu

- Penyelesaian:

Batalkan pengujian dan periksa AWS Fargate untuk memastikan bahwa semua tugas telah berhenti. Jika mereka belum berhenti, hentikan semua tugas Fargate secara manual. Periksa batas tugas Fargate sesuai permintaan di akun Anda untuk memastikan bahwa Anda dapat meluncurkan jumlah tugas yang diinginkan. Anda juga dapat memeriksa CloudWatch log untuk fungsi runner tugas Lambda untuk wawasan lebih lanjut tentang kegagalan saat meluncurkan tugas Fargate. Periksa log CloudWatch ECS untuk detail tentang apa yang terjadi di kontainer Fargate yang sedang berjalan.

Masalah: Tes dimulai tetapi gagal diselesaikan atau status tugas ECS tidak diketahui

- Penyelesaian:

Jika Anda memilih opsi untuk menyediakan VPC yang ada di akun tempat solusi telah digunakan, pastikan bahwa VPC yang digunakan oleh Tugas ECS memiliki alamat IP gratis yang cukup untuk

memulai jumlah tugas yang disediakan dalam input pengujian. Definisi tugas ECS menggunakan gambar ECR yang membutuhkan gateway internet atau rute ke internet sehingga layanan ECS dapat menyediakan tugas dengan mengunduh gambar solusi ECR dari aws-solutions/ -. distributed-load-testing-on aws-load-tester Jika Anda tidak dapat memberikan rute ke internet karena semua subnet di VPC bersifat pribadi, Anda dapat meng-host gambar ECR di akun Anda menggunakan ECR pull through cache. Perbarui definisi tugas dengan URI gambar ECR baru dan buat revisi baru. Setelah definisi tugas diperbarui, konfigurasi solusi dalam tabel DynamoDB perlu diperbarui untuk menggunakan revisi baru. Nama tabel DynamoDB dapat ditemukan di tab output tumpukan CloudFormation di bawah kunci. ScenariosTable Perbarui atribut TaskDefinition untuk item dengan kunci TestID dan nilai region- [SOLUTION-DEPLOYED-REGION].

Masalah: Pengujian perlu menggunakan titik akhir yang bersifat pribadi atau tidak tersedia melalui gateway internet

- Penyelesaian:

Saat menguji titik akhir API pribadi yang tidak dapat diakses melalui gateway internet, pertimbangkan pendekatan berikut:

1. Konfigurasi Jaringan: Pastikan tabel rute subnet yang digunakan oleh tugas ECS diperbarui dengan rute ke rentang alamat IP dari titik akhir pribadi yang sedang diuji. Ini memungkinkan lalu lintas pengujian mencapai titik akhir pribadi dalam VPC Anda.
2. Resolusi DNS: Untuk domain kustom, konfigurasikan pengaturan DNS di VPC Anda untuk menyelesaikan nama domain titik akhir pribadi. Lihat dokumentasi DNS VPC untuk petunjuk terperinci.
3. Titik Akhir VPC: Jika menguji layanan AWS, pertimbangkan untuk menggunakan titik akhir VPC (PrivateLinkAWS) untuk membuat konektivitas pribadi. Misalnya, untuk menguji API Gateway pribadi, Anda dapat membuat titik akhir VPC untuk API Gateway. Lihat dokumentasi API Gateway Pribadi.
4. Pengintip VPC: Jika titik akhir pribadi berada di VPC yang berbeda, buat peering VPC antara VPC tempat solusi diterapkan dan VPC yang berisi titik akhir pribadi. Konfigurasikan tabel rute yang sesuai di keduanya VPCs. Lihat dokumentasi VPC Peering.
5. Transit Gateway: Untuk skenario jaringan yang lebih kompleks yang melibatkan beberapa VPCs, pertimbangkan untuk menggunakan AWS Transit Gateway untuk merutekan lalu lintas antara VPC solusi dan VPC yang berisi titik akhir pribadi. Lihat dokumentasi Transit Gateway.

6. Grup Keamanan: Pastikan bahwa grup keamanan yang terkait dengan tugas ECS Anda memungkinkan lalu lintas keluar ke titik akhir pribadi, dan grup keamanan titik akhir pribadi mengizinkan lalu lintas masuk dari tugas ECS.

Untuk menguji Application Load Balancer atau EC2 instance internal, pastikan bahwa rentang VPC CIDR tidak tumpang tindih dan rute yang diperlukan dikonfigurasi dalam tabel rute.

Masalah: Pengujian selesai tetapi hasilnya tidak tersedia di UI

- Penyelesaian:

Jika pengujian telah selesai tetapi hasilnya tidak tersedia di UI, file hasil harus tetap tersedia di Bucket S3 dari tugas ECS yang menjalankan pengujian. Ini adalah batasan yang diketahui dalam solusinya. Dalam arsitektur saat ini, solusinya menggunakan fungsi penguraian hasil Lambda untuk meringkas hasil dari beberapa tugas ECS, yang kemudian disimpan sebagai item dalam tabel DynamoDB. Tabel DynamoDB memiliki batas ukuran item maksimum 400 KB. Keterbatasan ini tercapai tergantung pada kompleksitas skrip pengujian, konkurensi, dan jumlah tugas yang digunakan. Kesalahan tidak berarti tes gagal; ini menunjukkan bahwa proses untuk meringkas hasil dan menyimpannya dalam tabel DynamoDB untuk operasi CRUD telah gagal. Hasilnya masih tersedia di bucket S3 untuk skenario pengujian.

Hubungi AWS Support

Jika Anda memiliki AWS Developer Support, AWS Business Support, atau AWS Enterprise Support, Anda dapat menggunakan Support Center untuk mendapatkan bantuan ahli terkait solusi ini. Bagian berikut memberikan petunjuk.

Buat kasus

1. Masuk ke Support Center.
2. Pilih Buat kasus.

Bagaimana kami bisa membantu?

1. Pilih Teknis
2. Untuk Layanan, pilih Solusi.

3. Untuk Kategori, pilih Pengujian Beban Terdistribusi di AWS.
4. Untuk Keparahan, pilih opsi yang paling cocok dengan kasus penggunaan Anda.
5. Saat Anda memasukkan Layanan, Kategori, dan Tingkat Keparahan, antarmuka akan mengisi tautan ke pertanyaan pemecahan masalah umum. Jika Anda tidak dapat menyelesaikan pertanyaan Anda dengan tautan ini, pilih Langkah selanjutnya: Informasi tambahan.

Informasi tambahan

1. Untuk Subjek, masukkan teks yang merangkum pertanyaan atau masalah Anda.
2. Untuk Deskripsi, jelaskan masalah ini secara rinci.
3. Pilih Lampirkan file.
4. Lampirkan informasi yang dibutuhkan AWS Support untuk memproses permintaan.

Bantu kami menyelesaikan kasus Anda lebih cepat

1. Masukkan informasi yang diminta.
2. Pilih Langkah selanjutnya: Selesaikan sekarang atau hubungi kami.

Selesaikan sekarang atau hubungi kami

1. Tinjau solusi Selesaikan sekarang.
2. Jika Anda tidak dapat menyelesaikan masalah Anda dengan solusi ini, pilih Hubungi kami, masukkan informasi yang diminta, dan pilih Kirim.

Copot pemasangan solusinya

Anda dapat menghapus instalasi Pengujian Beban Terdistribusi pada solusi AWS dari AWS Management Console atau dengan menggunakan AWS Command Line Interface. Anda harus secara manual menghapus konsol, skenario, dan mencatat bucket Amazon Simple Storage Service (Amazon S3) yang dibuat oleh solusi ini. Implementasi AWS Solutions tidak secara otomatis menghapusnya jika Anda memiliki data untuk disimpan.

> **Note**
>
> Jika Anda telah menerapkan tumpukan regional, Anda harus menghapus tumpukan di Wilayah tersebut sebelum menghapus tumpukan utama.

Menggunakan Konsol Manajemen AWS

1. Masuk ke CloudFormation konsol AWS.
2. Pada halaman Stacks, pilih tumpukan instalasi solusi ini.
3. Pilih Hapus.

Menggunakan AWS Command Line Interface

Tentukan apakah AWS Command Line Interface (AWS CLI) tersedia di lingkungan Anda. Untuk petunjuk penginstalan, lihat Apa itu Antarmuka Baris Perintah AWS di Panduan Pengguna AWS CLI. Setelah mengonfirmasi bahwa AWS CLI tersedia, jalankan perintah berikut.

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

Menghapus bucket Amazon S3

Solusi ini dikonfigurasi untuk mempertahankan bucket Amazon S3 yang dibuat solusi (untuk diterapkan di Wilayah keikutsertaan) jika Anda memutuskan untuk menghapus tumpukan AWS untuk mencegah kehilangan data yang tidak disengaja. CloudFormation Setelah menghapus instalan solusi, Anda dapat menghapus bucket S3 ini secara manual jika Anda tidak perlu menyimpan data. Ikuti langkah-langkah ini untuk menghapus bucket Amazon S3.

1. Masuk ke konsol Amazon S3.
2. Pilih Bucket dari panel navigasi kiri.
3. Di bidang Temukan ember berdasarkan nama, masukkan nama tumpukan solusi ini.
4. Pilih salah satu bucket S3 solusi dan pilih Kosong.
5. Masukkan hapus secara permanen di bidang verifikasi dan pilih Kosong.
6. Pilih bucket S3 yang baru saja Anda kosongkan dan pilih Delete.
7. Masukkan nama bucket S3 di kolom verifikasi dan pilih Delete bucket.

Ulangi langkah 4 hingga 7 hingga Anda menghapus semua ember S3.

Untuk menghapus bucket S3 menggunakan AWS CLI, jalankan perintah berikut:

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

Gunakan solusinya

Bagian ini mencakup informasi tentang cara menggunakan solusi Pengujian Beban Terdistribusi pada AWS, termasuk hasil Pengujian, alur kerja penjadwalan pengujian, dan data Langsung.

Hasil tes

Pengujian Beban Terdistribusi di AWS memanfaatkan kerangka kerja Pengujian Beban untuk menjalankan pengujian aplikasi dalam skala besar. Ketika tes selesai, laporan rinci dihasilkan berisi hasil berikut.

- Waktu respons rata-rata - Waktu respons rata-rata, dalam hitungan detik, untuk semua permintaan yang dihasilkan oleh tes.
- Latensi rata-rata - Latensi rata-rata, dalam hitungan detik, untuk semua permintaan yang dihasilkan oleh pengujian.
- Waktu koneksi rata-rata - Waktu rata-rata, dalam hitungan detik, diperlukan untuk terhubung ke host untuk semua permintaan yang dihasilkan oleh pengujian.
- Bandwidth rata-rata - Bandwidth rata-rata untuk semua permintaan yang dihasilkan oleh tes.
- Jumlah total - Jumlah total permintaan.
- Hitungan keberhasilan - Jumlah total permintaan yang berhasil.
- Jumlah kesalahan - Jumlah total kesalahan.
- Permintaan per detik - Permintaan rata-rata per detik untuk semua permintaan yang dihasilkan oleh pengujian.
- Persentil - Persentil waktu respons untuk tes. Waktu respons maksimum adalah 100%; waktu respons minimum adalah 0%.

> **Note**
>
> Hasil pengujian ditampilkan di konsol. Anda dapat melihat file XMLuntuk hasil pengujian mentah di `Results` folder bucket `Scenarios` Amazon S3.

Untuk informasi selengkapnya tentang hasil tes Taurus, lihat Menghasilkan Laporan Uji di Panduan Pengguna Taurus.

Alur kerja penjadwalan uji

Gunakan konsol web untuk menjadwalkan uji beban. Saat menjadwalkan pengujian, alur kerja berikut berjalan:

- Ketika pengujian beban dibuat dengan opsi untuk menjadwalkan, parameter jadwal dikirim ke API solusi melalui Amazon API Gateway.
- API kemudian meneruskan parameter ke fungsi Lambda yang membuat aturan CloudWatch Acara, yang akan dijadwalkan untuk berjalan pada tanggal yang ditentukan.
- Jika pengujian adalah tes satu kali, aturan CloudWatch Peristiwa berjalan pada tanggal yang ditentukan. Fungsi `api-services` Lambda menjalankan pengujian baru melalui alur kerja pengujian.
- Jika tes adalah tes berulang, aturan CloudWatch Peristiwa diaktifkan pada tanggal yang ditentukan. Fungsi `api-services` Lambda berjalan, yang menghapus aturan CloudWatch Peristiwa saat ini dan membuat aturan lain yang berjalan segera saat dibuat dan berulang setelahnya berdasarkan frekuensi pengulangan yang ditentukan.

Tentukan jumlah pengguna

Jumlah pengguna yang dapat didukung penampung untuk pengujian dapat ditentukan dengan meningkatkan jumlah pengguna secara bertahap, dan memantau kinerja di Amazon CloudWatch. Setelah Anda mengamati bahwa kinerja CPU dan memori mendekati batasnya, Anda telah mencapai jumlah maksimum pengguna yang dapat didukung oleh wadah untuk pengujian tersebut dalam konfigurasi defaultnya (2 vCPU dan memori 4 GB). Anda dapat mulai menentukan batas pengguna bersamaan untuk pengujian Anda dengan menggunakan contoh berikut:

1. Buat tes dengan tidak lebih dari 200 pengguna.
2. Saat pengujian berjalan, pantau CPU dan Memori menggunakan CloudWatch konsol:
   a. Dari panel navigasi kiri, di bawah Container Insights, pilih Performance Monitoring.
   b. Pada halaman pemantauan kinerja, dari menu drop-down kiri, pilih ECS Clusters.
   c. Dari menu tarik-turun kanan, pilih klaster Amazon Elastic Container Service (Amazon ECS) Anda.
3. Saat memantau, perhatikan CPU dan Memori. Jika CPU tidak melampaui 75% atau Memori tidak melampaui 85% (abaikan puncak satu kali), Anda dapat menjalankan tes lain dengan jumlah pengguna yang lebih tinggi.

Ulangi langkah 1-3 jika tes tidak melebihi batas sumber daya. Secara opsional, sumber daya kontainer dapat ditingkatkan untuk memungkinkan jumlah pengguna bersamaan yang lebih tinggi. Namun, ini menghasilkan biaya yang lebih tinggi. Untuk detailnya, lihat bagian Meningkatkan sumber daya kontainer dari panduan ini.

> **Note**
>
> Untuk hasil yang akurat, jalankan hanya satu pengujian pada satu waktu saat menentukan batas pengguna bersamaan. Semua pengujian menggunakan cluster yang sama, dan wawasan CloudWatch kontainer mengumpulkan data kinerja berdasarkan cluster. Hal ini menyebabkan kedua pengujian dilaporkan ke CloudWatch Container Insights secara bersamaan, yang menghasilkan metrik pemanfaatan sumber daya yang tidak akurat untuk satu pengujian.

Untuk informasi lebih lanjut tentang kalibrasi pengguna per mesin, lihat Mengkalibrasi Tes Taurus dalam dokumentasi. BlazeMeter

Data langsung

Anda dapat secara opsional menyertakan data langsung saat menjalankan tes untuk mendapatkan wawasan waktu nyata tentang apa yang terjadi. Grup CloudWatch log untuk tugas Fargate berisi filter langganan untuk hasil dari pengujian yang menyertakan opsi data langsung. Jika solusi menemukan pola, itu memulai fungsi Lambda yang menyusun data dan menerbitkannya ke topik AWS IoT Core. Konsol web berlangganan topik, menerima data yang masuk, dan membuat grafik data yang dikumpulkan pada interval satu detik. Konsol web berisi empat grafik: waktu respons rata-rata, pengguna virtual, keberhasilan, dan kegagalan.

> **Note**
>
> Data bersifat fana dan hanya untuk digunakan untuk melihat apa yang terjadi saat tes sedang berjalan. Setelah tes selesai, solusi menyimpan data hasil di DynamoDB dan Amazon S3. Konsol web bertahan maksimal 5.000 titik data, setelah itu data tertua diganti dengan yang terbaru. Jika halaman diperbarui, grafik akan kosong dan mulai dari titik data berikutnya yang tersedia.

Uji alur kerja pembatalan

Saat Anda membatalkan uji pemuatan dari konsol web, solusi akan menjalankan alur kerja pembatalan pengujian berikut.

1. Permintaan pembatalan dikirim ke `microservices` API.
2. `microservices`API memanggil fungsi `task-canceler` Lambda yang membatalkan tugas hingga semua tugas yang diluncurkan saat ini dihentikan.
3. Jika fungsi `task-runner` Lambda terus berjalan setelah panggilan awal ke fungsi `task-canceler` Lambda, maka tugas akan terus diluncurkan. Setelah fungsi `task-runner` Lambda selesai, AWS Step Functions melanjutkan ke `Cancel Test` langkah, yang menjalankan fungsi `task-canceler` Lambda lagi untuk menghentikan tugas yang tersisa.

Panduan pengembang

Bagian ini menyediakan kode sumber untuk solusi dan penyesuaian tambahan.

Kode sumber

Kunjungi GitHub repositori kami untuk mengunduh templat dan skrip untuk solusi ini, dan untuk berbagi penyesuaian Anda dengan orang lain.

Maintenance

Solusi ini menggunakan gambar Docker dengan versi tetap yang cocok dengan setiap rilis solusi jika pembaruan otomatis tidak dipilih. Tim AWS Solutions menggunakan ECR Enhanced Scanning untuk mendeteksi Common Vulnerabilities and Exposures (CVEs) pada image dasar dan paket yang diinstal. Jika memungkinkan, tim akan mempublikasikan gambar yang ditambal dengan tag versi yang sama untuk diselesaikan CVEs, tanpa merusak kompatibilitas dengan versi solusi yang dirilis. Ketika gambar ditambal, jika mereka berada pada versi minor yang sama, tag stabil akan diperbarui secara otomatis, dan tag gambar tambahan akan dibuat dalam format`<solution-version>_<date-of-fix>`. Jika versi mayor atau minor dirilis, pembaruan tumpukan penuh akan diperlukan untuk mendapatkan versi gambar terbaru karena tag stabil akan bertambah sehingga versinya cocok dengan versi solusi. Jika memilih untuk memperbarui otomatis perubahan pada gambar, termasuk perbaikan bug CVEs dan minor, akan secara otomatis diterapkan ke gambar hingga rilis minor terbaru yang cocok.

Versi

Pelanggan pada versi solusi terbaru akan menerima patch keamanan dan perbaikan bug minor, tidak rusak, secara otomatis jika mereka ikut serta dalam pembaruan gambar otomatis. Gambar akan secara otomatis menarik gambar terbaru ke versi minor pencocokan terbaru. Untuk mengunci kontainer ke versi tertentu, definisi tugas dapat diedit untuk menentukan wadah untuk menggunakan versi gambar tertentu dengan menggunakan versi gambar yang ditandai. Pembaruan otomatis juga dapat dimatikan dengan memilih Tidak untuk pembaruan otomatis CloudFormation saat meluncurkan tumpukan. Ini akan meluncurkan versi gambar yang cocok dengan versi solusi.

Kustomisasi gambar kontainer

Solusi ini menggunakan repositori image Amazon Elastic Container Registry (Amazon ECR) publik yang dikelola oleh AWS untuk menyimpan gambar yang digunakan untuk menjalankan pengujian yang dikonfigurasi. Jika Anda ingin menyesuaikan gambar kontainer, Anda dapat membangun kembali dan mendorong gambar ke dalam repositori gambar ECR di akun AWS Anda sendiri.

Jika Anda ingin menyesuaikan solusi ini, Anda dapat menggunakan gambar kontainer default atau, edit wadah ini agar sesuai dengan kebutuhan Anda. Jika Anda menyesuaikan solusinya, gunakan contoh kode berikut untuk mendeklarasikan variabel lingkungan sebelum membuat solusi khusus Anda.

```
#!/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
```

Jika Anda memilih untuk menyesuaikan gambar kontainer, Anda dapat menghostingnya di repositori gambar pribadi, atau repositori gambar publik di akun AWS Anda. Sumber daya gambar ada di `deployment/ecr/distributed-load-testing-on-aws-load-tester` direktori, yang terletak di basis kode.

Anda dapat membangun dan mendorong gambar ke tujuan host.

- Untuk repositori dan gambar Amazon ECR pribadi, lihat repositori pribadi Amazon ECR dan gambar pribadi di Panduan Pengguna Amazon ECR.
- Untuk repositori dan gambar ECR Amazon publik, lihat repositori publik Amazon ECR dan gambar publik di Panduan Pengguna Publik Amazon ECR.

Setelah Anda membuat gambar Anda sendiri, Anda dapat mendeklarasikan variabel lingkungan berikut sebelum membangun solusi khusus Anda.

```
#!/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
```

Contoh berikut menunjukkan file kontainer.

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

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

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


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

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

RUN chmod 755 /bzt-configs/load-test.sh
RUN chmod 755 /bzt-configs/ecslistener.py
RUN chmod 755 /bzt-configs/ecscontroller.py
RUN chmod 755 /bzt-configs/jar_updater.py
RUN python3.11 /bzt-configs/jar_updater.py

# Remove jar files from /tmp
RUN rm -rf /tmp/jmeter-plugins-manager-1* && \
    rm -rf /usr/local/lib/python3.11/site-packages/setuptools-65.5.0.dist-info && \
    rm -rf /usr/local/lib/python3.11/site-packages/urllib3-1.26.17.dist-info
```

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

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

Selain file kontainer, direktori berisi skrip bash berikut yang mengunduh konfigurasi pengujian dari Amazon S3 sebelum menjalankan Taurus/Blazemeter program.

```
#!/bin/bash

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

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

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

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

# Set the default log file values to jmeter
```

```
LOG_FILE="jmeter.log"
OUT_FILE="jmeter.out"
ERR_FILE="jmeter.err"
KPI_EXT="jtl"

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

  # set variables based on TEST_TYPE
  if [ "$TEST_TYPE" == "jmeter" ]; then
    EXT="jmx"
    TYPE_NAME="JMeter"
    # Copy *.jar to JMeter library path. See the Taurus JMeter path: https://
gettaurus.org/docs/JMeter/
    JMETER_LIB_PATH=`find ~/.bzt/jmeter-taurus -type d -name "lib"`
    echo "cp $PWD/*.jar $JMETER_LIB_PATH"
    cp $PWD/*.jar $JMETER_LIB_PATH
  elif [ "$TEST_TYPE" == "k6" ]; then
    curl --output /tmp/artifacts/k6.rpm https://dl.k6.io/rpm/x86_64/k6-v0.58.0-
amd64.rpm
    rpm -ivh /tmp/artifacts/k6.rpm
    dnf install -y k6
    rm -rf /tmp/artifacts/k6.rpm
    EXT="js"
    KPI_EXT="csv"
    TYPE_NAME="K6"
  elif [ "$TEST_TYPE" == "locust" ]; then
    EXT="py"
    TYPE_NAME="Locust"

  fi

  if [ "$FILE_TYPE" != "zip" ]; then
    aws s3 cp s3://$S3_BUCKET/public/test-scenarios/$TEST_TYPE/$TEST_ID.$EXT ./ --
region $MAIN_STACK_REGION
  else
    aws s3 cp s3://$S3_BUCKET/public/test-scenarios/$TEST_TYPE/$TEST_ID.zip ./ --region
 $MAIN_STACK_REGION
    unzip $TEST_ID.zip
    echo "UNZIPPED"
```

```
    ls -l

    # If zip and locust, make sure to pick locustfile
    if [ "$TEST_TYPE" != "locust" ]; then
      TEST_SCRIPT=$(find . -name "*.${EXT}" | head -n 1)
    else
      TEST_SCRIPT=$(find . -name "locustfile.py" | head -n 1)
    fi
    # only looks for the first test script file.
    TEST_SCRIPT=`find . -name "*.${EXT}" | head -n 1`
    echo $TEST_SCRIPT
    if [ -z "$TEST_SCRIPT" ]; then
      echo "There is no test script (.${EXT}) in the zip file."
      exit 1
    fi

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

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

#Download python script
if [ -z "$IPNETWORK" ]; then
    python3.11 -u $SCRIPT  $TIMEOUT &
    pypid=$!
    wait $pypid
    pypid=0
```

```
else
    aws s3 cp s3://$S3_BUCKET/Container_IPs/${TEST_ID}_IPHOSTS_${AWS_REGION}.txt ./ --
region $MAIN_STACK_REGION
    export IPHOSTS=$(cat ${TEST_ID}_IPHOSTS_${AWS_REGION}.txt)
    python3.11 -u $SCRIPT $IPNETWORK $IPHOSTS
fi

echo "Running test"

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

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

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

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

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

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

```
        all_files+=("$f")
      done
    done

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

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

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

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


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

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

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

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

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

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

Selain Dockerfile dan skrip bash, dua skrip Python juga disertakan dalam direktori. Setiap tugas menjalankan skrip Python dari dalam skrip bash. Tugas pekerja menjalankan `ecslistener.py` skrip, sedangkan tugas pemimpin akan menjalankan `ecscontroller.py` skrip. `ecslistener.py`Skrip membuat soket pada port 50000 dan menunggu pesan. `ecscontroller.py`Skrip terhubung ke soket dan mengirimkan pesan uji awal ke tugas pekerja, yang memungkinkan mereka untuk memulai secara bersamaan.

API pengujian beban terdistribusi

Solusi pengujian beban ini membantu Anda mengekspos data hasil pengujian dengan cara yang aman. API bertindak sebagai “pintu depan” untuk akses ke data pengujian yang disimpan di Amazon DynamoDB. Anda juga dapat menggunakan APIs untuk mengakses fungsionalitas tambahan apa pun yang Anda buat ke dalam solusi.

Solusi ini menggunakan kumpulan pengguna Amazon Cognito yang terintegrasi dengan Amazon API Gateway untuk identifikasi dan otorisasi. Ketika kumpulan pengguna digunakan dengan API, klien hanya diperbolehkan memanggil metode yang diaktifkan kumpulan pengguna setelah mereka memberikan token identitas yang valid.

Untuk informasi selengkapnya tentang menjalankan pengujian secara langsung melalui API, lihat Permintaan Penandatanganan di dokumentasi Referensi API REST API Amazon API Gateway.

Operasi berikut tersedia di API solusi.

> **Note**
>
> Untuk informasi lebih lanjut tentang `testScenario` dan parameter lainnya, lihat skenario dan contoh payload di GitHub repositori.

Skenario

- DAPATKAN /skenario
- POST/skenario
- OPSI/skenario
- DAPATKAN /scenarios/ {TESid}
- POSTING /scenarios/ {TESid}
- HAPUS /scenarios/ {TESid}
- PILIHAN /scenarios/ {TESid}

Tugas

- DAPATKAN /tugas
- OPSI/tugas

Daerah

- DAPATKAN /wilayah
- OPSI/wilayah

GET /scenarios

Deskripsi

`GET /scenarios`Operasi ini memungkinkan Anda untuk mengambil daftar skenario pengujian.

Respons

Nama	Penjelasan
`data`	Daftar skenario termasuk ID, nama, deskripsi , status, dan waktu berjalan untuk setiap pengujian

POST/skenario

Deskripsi

`POST /scenarios`Operasi ini memungkinkan Anda untuk membuat atau menjadwalkan skenario pengujian.

Isi permintaan

Nama	Penjelasan
`testName`	Nama tes
`testDescription`	Deskripsi tes
`testTaskConfigs`	Objek yang menentukan `concurrency` (jumlah paralel berjalan), `taskCount` (jumlah tugas yang diperlukan untuk menjalankan tes), dan `region` untuk skenario
`testScenario`	Definisi tes termasuk konkurensi, waktu pengujian, host, dan metode untuk pengujian
`testType`	Jenis tes (misalnya,`simple`,`jmeter`)

Nama	Penjelasan
`fileType`	Jenis file upload (misalnya,`none`,`script`,`zip`)
`scheduleDate`	Tanggal untuk menjalankan tes. Hanya disediakan jika menjadwalkan tes (misalnya ,`2021-02-28`)
`scheduleTime`	Waktu untuk menjalankan tes. Hanya disediaka n jika menjadwalkan tes (misalnya,`21:07`)
`scheduleStep`	Langkah dalam proses jadwal. Hanya disediaka n jika menjadwalkan tes berulang. (Langkah-langkah yang tersedia termasuk `create` dan`start`)
`cronvalue`	Nilai cron untuk menyesuaikan penjadwalan berulang. Jika digunakan, hilangkan scheduleD ate dan ScheduleTime.
`cronExpiryDate`	Tanggal yang diperlukan sehingga cron kedaluwarsa dan tidak berjalan tanpa batas waktu.
`recurrence`	Terulangnya tes terjadwal. Hanya disediaka n jika menjadwalkan tes berulang (misalnya,,, `daily` `weekly`biweekly, atau) `monthly`

Respons

Nama	Penjelasan
`testId`	ID unik dari tes
`testName`	Nama tes
`status`	Status tes

OPSI/skenario

Deskripsi

`OPTIONS /scenarios`Operasi ini memberikan respons untuk permintaan dengan header respons CORS yang benar.

Respons

Nama	Penjelasan
`testId`	ID unik dari tes
`testName`	Nama tes
`status`	Status tes

DAPATKAN /scenarios/ {TESid}

Deskripsi

`GET /scenarios/{testId}`Operasi ini memungkinkan Anda untuk mengambil rincian skenario pengujian tertentu.

Parameter permintaan

`testId`

- ID unik dari tes

  Tipe: String

  Diperlukan: Ya

Respons

Nama	Penjelasan
`testId`	ID unik dari tes

Nama	Penjelasan
`testName`	Nama tes
`testDescription`	Deskripsi tes
`testType`	Jenis pengujian yang dijalankan (misalnya ,`simple`,`jmeter`)
`fileType`	Jenis file yang diunggah (misalnya,, `nonescript`,`zip`)
`status`	Status tes
`startTime`	Waktu dan tanggal ketika tes terakhir dimulai
`endTime`	Waktu dan tanggal ketika tes terakhir berakhir
`testScenario`	Definisi tes termasuk konkurensi, waktu pengujian, host, dan metode untuk pengujian
`taskCount`	Jumlah tugas yang dibutuhkan untuk menjalankan tes
`taskIds`	Daftar tugas IDs untuk menjalankan tes
`results`	Hasil akhir dari tes
`history`	Daftar hasil akhir dari tes sebelumnya
`errorReason`	Pesan galat yang dihasilkan saat terjadi kesalahan
`nextRun`	Jalankan terjadwal berikutnya (misalnya ,`2017-04-22 17:18:00`)
`scheduleRecurrence`	Pengulangan tes (misalnya,,`daily`, `weeklybiweekly`,`monthly`)

POSTING /scenarios/ {TESid}

Deskripsi

`POST /scenarios/{testId}`Operasi ini memungkinkan Anda untuk membatalkan skenario pengujian tertentu.

Parameter permintaan

`testId`

- ID unik dari tes

  Tipe: String

  Diperlukan: Ya

Respons

Nama	Penjelasan
`status`	Status tes

HAPUS /scenarios/ {TESid}

Deskripsi

`DELETE /scenarios/{testId}`Operasi ini memungkinkan Anda untuk menghapus semua data yang terkait dengan skenario pengujian tertentu.

Parameter permintaan

`testId`

- ID unik dari tes

  Tipe: String

  Diperlukan: Ya

Respons

Nama	Penjelasan
`status`	Status tes

PILIHAN /scenarios/ {TESid}

Deskripsi

`OPTIONS /scenarios/{testId}`Operasi ini memberikan respons untuk permintaan dengan header respons CORS yang benar.

Respons

Nama	Penjelasan
`testId`	ID unik dari tes
`testName`	Nama tes
`testDescription`	Deskripsi tes
`testType`	Jenis pengujian yang dijalankan (misalnya ,`simple`,`jmeter`)
`fileType`	Jenis file yang diunggah (misalnya,, `nonescript`,`zip`)
`status`	Status tes
`startTime`	Waktu dan tanggal ketika tes terakhir dimulai
`endTime`	Waktu dan tanggal ketika tes terakhir berakhir
`testScenario`	Definisi tes termasuk konkurensi, waktu pengujian, host, dan metode untuk pengujian

Nama	Penjelasan
`taskCount`	Jumlah tugas yang dibutuhkan untuk menjalankan tes
`taskIds`	Daftar tugas IDs untuk menjalankan tes
`results`	Hasil akhir dari tes
`history`	Daftar hasil akhir dari tes sebelumnya
`errorReason`	Pesan galat yang dihasilkan saat terjadi kesalahan

DAPATKAN /tugas

Deskripsi

`GET /tasks`Operasi ini memungkinkan Anda mengambil daftar tugas Amazon Elastic Container Service (Amazon ECS) yang sedang berjalan.

Respons

Nama	Penjelasan
`tasks`	Daftar tugas IDs untuk menjalankan tes

OPSI/tugas

Deskripsi

Operasi `OPTIONS /tasks` tugas memberikan respons untuk permintaan dengan header respons CORS yang benar.

Respons

Nama	Penjelasan
`taskIds`	Daftar tugas IDs untuk menjalankan tes

DAPATKAN /wilayah

Deskripsi

`GET /regions`Operasi ini memungkinkan Anda untuk mengambil informasi sumber daya regional yang diperlukan untuk menjalankan tes di Wilayah tersebut.

Respons

Nama	Penjelasan
`testId`	ID Wilayah
`ecsCloudWatchLogGroup`	Nama grup CloudWatch log Amazon untuk tugas Amazon Fargate di Wilayah
`region`	Wilayah di mana sumber daya dalam tabel ada
`subnetA`	ID salah satu subnet di Wilayah
`subnetB`	ID salah satu subnet di Wilayah
`taskCluster`	Nama klaster AWS Fargate di Wilayah
`taskDefinition`	ARN definisi tugas di Wilayah
`taskImage`	Nama gambar tugas di Wilayah
`taskSecurityGroup`	ID grup keamanan di Wilayah

OPSI/wilayah

Deskripsi

`OPTIONS /regions`Operasi ini memberikan respons untuk permintaan dengan header respons CORS yang benar.

Respons

Nama	Penjelasan
`testId`	ID Wilayah
`ecsCloudWatchLogGroup`	Nama grup CloudWatch log Amazon untuk tugas Amazon Fargate di Wilayah
`region`	Wilayah di mana sumber daya dalam tabel ada
`subnetA`	ID salah satu subnet di Wilayah
`subnetB`	ID salah satu subnet di Wilayah
`taskCluster`	Nama klaster AWS Fargate di Wilayah
`taskDefinition`	ARN definisi tugas di Wilayah
`taskImage`	Nama gambar tugas di Wilayah
`taskSecurityGroup`	ID grup keamanan di Wilayah

Meningkatkan sumber daya kontainer

Untuk meningkatkan jumlah pengguna yang saat ini didukung, tingkatkan sumber daya penampung. Ini memungkinkan Anda untuk meningkatkan memori CPUs dan untuk menangani peningkatan pengguna bersamaan.

Buat revisi definisi tugas baru

1. Masuk ke konsol Amazon Elastic Container Service.

2. Di menu navigasi kiri, pilih Definisi Tugas.
3. Pilih kotak centang di sebelah definisi tugas yang sesuai dengan solusi ini. Misalnya, `[replaceable]` <stackName>`- EcsTaskDefinition -*<system-generated-random-Hash>*.
4. Pilih Buat revisi baru.
5. Pada halaman Buat revisi baru, lakukan tindakan berikut:
   a. Di bawah Ukuran tugas, ubah memori Tugas dan CPU Tugas.
   b. Di bawah Container Definitions, tinjau batas memori Hard/Soft. Jika batas ini lebih rendah dari memori yang Anda inginkan, pilih wadahnya.
   c. Di kotak dialog Edit container, buka Memory Limits dan perbarui Hard Limit ke memori yang Anda inginkan.
   d. Pilih Perbarui.
6. Pada halaman Buat revisi baru, pilih Buat.
7. Setelah definisi tugas berhasil dibuat, catat nama definisi tugas baru. Nama ini termasuk nomor versi, misalnya: `[replaceable]` <stackName>`- EcsTaskDefinition -*<system-generated-random-Hash>*: [replaceable]`<system-generated-versionNumber>`.

Perbarui tabel DynamoDB

1. Arahkan ke konsol DynamoDB.
2. Dari panel navigasi kiri, pilih Jelajahi Item di bawah Tabel.
3. Pilih tabel `scenarios-table` DynamoDB yang terkait dengan solusi ini. Misalnya, `[replaceable]` <stackName>`- DLTTest RunnerStorage DLTScenarios Tabel-*<system-generated-random-Hash>*.
4. Pilih item yang sesuai dengan Wilayah tempat Anda telah memodifikasi definisi tugas. Misalnya, wilayah-*<region-name>*.
5. Perbarui atribut TaskDefinition dengan definisi tugas baru.

Referensi

Bagian ini mencakup informasi tentang fitur opsional untuk mengumpulkan metrik unik untuk solusi ini, petunjuk ke sumber daya terkait, dan daftar pembangun yang berkontribusi pada solusi ini.

Pengumpulan data anonim

Solusi ini mencakup opsi untuk mengirim metrik operasional anonim ke AWS. Kami menggunakan data ini untuk lebih memahami bagaimana pelanggan menggunakan solusi ini dan layanan serta produk terkait. Saat dipanggil, informasi berikut dikumpulkan dan dikirim ke AWS:

- ID Solusi - Pengidentifikasi solusi AWS
- Unique ID (UUID) - Pengidentifikasi unik yang dibuat secara acak untuk setiap penerapan solusi
- Stempel waktu - Stempel waktu pengumpulan data
- Test Type - Jenis tes yang dijalankan
- Jenis File - Jenis file yang diunggah
- Task Count - Jumlah tugas untuk setiap pengujian yang dikirimkan melalui API solusi
- Durasi Tugas - Total waktu berjalan untuk semua tugas yang diperlukan untuk menjalankan tes
- Hasil Tes - Hasil tes yang dijalankan

AWS memiliki data yang dikumpulkan melalui survei ini. Pengumpulan data tunduk pada Kebijakan Privasi AWS. Untuk memilih keluar dari fitur ini, selesaikan langkah-langkah berikut sebelum meluncurkan CloudFormation template AWS.

1. Unduh CloudFormation template AWS ke hard drive lokal Anda.
2. Buka CloudFormation template AWS dengan editor teks.
3. Ubah bagian pemetaan CloudFormation template AWS dari:

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

ke:

```
Solution:
```

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

4. Masuk ke CloudFormation konsol AWS.
5. Pilih Buat tumpukan.
6. Pada halaman Buat tumpukan, Tentukan templat bagian, pilih Unggah file templat.
7. Di bawah Unggah file templat, pilih Pilih file dan pilih templat yang diedit dari drive lokal Anda.
8. Pilih Berikutnya dan ikuti langkah-langkah dalam Luncurkan tumpukan di bagian Deploy the solution dari panduan ini.

Kontributor

- Tom Burung Bulbul
- Fernando Dingler
- Beomseok Lee
- George Lenz
- Erin McGill
- Dimitri Lopez
- Kamyar Ziabari
- Bassem Wani
- Garvit Singh
- Nikhil Reddy
- Simon Kroll

Revisi

Kunjungi ChangelOG.md di GitHub repositori kami untuk melacak peningkatan dan perbaikan khusus versi.

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik produk AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, pernyataan, atau ketentuan dalam bentuk apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

Pengujian Beban Terdistribusi di AWS dilisensikan berdasarkan ketentuan Lisensi Apache Versi 2.0 yang tersedia di The Apache Software Foundation.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.