



구현 가이드

AWS의 분산 부하 테스트



AWS에서의 분산 로드 테스트: 구현 안내서

저작권 © 2024 Amazon Web Services, Inc. 및/또는 그 계열사. 판권 소유.

Amazon의 상표 및 상품 외장은 Amazon의 것이 아닌 제품 또는 서비스와 관련하여 고객 사이에 혼란을 야기할 수 있는 방식으로 또는 Amazon을 폄하하거나 신용을 떨어뜨리는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 해당 소유자의 자산이며, 해당 소유자는 Amazon과 제휴, 연결 또는 후원을 받거나 받지 않을 수 있습니다.

목차

솔루션 개요	1	
비용	2	
아키텍처 개요	4	
구성품	6	
프론트엔드	6	
부하 테스트 API	6	
웹 콘솔	6	
백엔드	7	
컨테이너 이미지 파이프라인	7	
테스트 인프라	7	
부하 테스트 엔진	7	
보안	9	
IAM 역할	9	
Amazon CloudFront	9	
AWS Fargate 보안 그룹	9	
네트워크 스트레스 테스트	9	
공용 사용자 인터페이스에 대한 액세스 제한	10	
디자인 고려사항		11
지원되는 애플리케이션	11	
JMeter 스크립트 지원	11	
테스트 예약		12
부하 테스트 할당량	12	
동시 테스트	12	
Amazon EC2 테스트 정책	13	
Amazon CloudFront 로드 테스트 정책	13	
사용자 관리	13	
지역 배포	13	
AWS CloudFormation 템플릿	14	
자동 배포	15	
스택 시작	15	
다중 지역 배포	19	
자원	22	
기타 리소스	22	
서비스 카탈로그 AppRegistry를 사용하여 솔루션 모니터링	23	

CloudWatch Application Insights 활성화	24
AWS Cost Explorer 활성화	25
솔루션과 관련된 비용 할당 태그 활성화	25
솔루션과 관련된 비용 태그 확인	26
컨테이너 이미지 맞춤설정	27
테스트 작업 흐름	32
시험 결과	35
분산 부하 테스트 API	36
GET /시나리오	37
설명	37
응답	37
POST/시나리오	37
설명	37
요청 본문	37
응답	38
옵션/시나리오	39
설명	39
응답	39
GET /scenarios/{testId}	39
설명	39
요청 매개변수	39
응답	40
POST /scenarios/{testId}	41
설명	41
요청 매개변수	41
응답	41
/scenarios/{testId} 삭제	41
설명	41
요청 매개변수	42
응답	42
옵션 /scenarios/{testId}	42
설명	42
응답	42
GET /작업	43
설명	43
응답	43

옵션 /작업	44 설
명	44 응
답	44 GET /지
역	44 설
명	44 응
답	44 옵션 /지
역	45 설
명	45 응
답	45 테스트 예약 작업 흐
름	47 사용자 수 결
정	48
컨테이너 리소스 늘리기	49
새 작업 정의 개정 생성	49
DynamoDB 테이블 업데이트	49
실시간 데이터	51
테스트 취소 워크플로	52
문제 해결	53 알려진 문
제	53
문제: 기존 VPC를 사용하고 있으며 테스트가 실패 상태로 실패하여 다음과 같은 오류 메시지가 나타납니	
다.	53 문제: 테스트를 실행하는 데 너무 오랜 시간이 걸리거나 무한정 실행이 중
단됩니다.	53 v3.2.6 이전 DLT 버전에서 최신 버전으로 업데이트할 때 스택을 업데이트하면 실패합니다.
원	55 케이스 생
성	55
어떻게 도와드릴까요?	55 추가 정
보	55
귀하의 사건을 더 빨리 해결하도록 도와주세요	56 지금 해결하거나 당사에 문의하
십시오	56
솔루션 제거	57
AWS 관리 콘솔 사용	57 AWS 명령줄 인터페이스 사
용	57 Amazon S3 버킷 삭
제	57 익명화된 데이터 수
집	59 소스 코
드	61
기여자	62
개정	63

공지사항 67

AWS 용어집 68

대규모로 소프트웨어 애플리케이션 테스트 자동화

발행일: 2019년 11월 ([최종 업데이트: 2024년 4월](#))

AWS의 분산 로드 테스트는 대규모 및 로드 시 소프트웨어 애플리케이션 테스트를 자동화하여 애플리케이션을 출시하기 전에 병목 현상을 식별하는 데 도움이 됩니다. 이 솔루션은 서버를 프로비저닝할 필요 없이 일정한 속도로 트랜잭션 기록을 생성하는 수천 명의 연결된 사용자를 생성하고 시뮬레이션합니다.

이 솔루션은 [AWS Fargate에서 Amazon Elastic Container Service\(Amazon ECS\)](#)를 활용합니다. 모든 시뮬레이션을 실행할 수 있고 다음 기능을 제공하는 컨테이너를 배포합니다.

- 테스트 중인 소프트웨어의 로드 기능을 테스트하기 위해 독립적으로 실행할 수 있는 AWS Fargate 컨테이너에 Amazon ECS를 배포합니다.
- 여러 AWS 리전에서 수만 명의 연결된 사용자를 시뮬레이션하여 지속적인 속도로 트랜잭션 기록을 생성합니다.
- 사용자 정의 JMeter 스크립트를 생성하여 애플리케이션 테스트를 사용자 [정의합니다](#).
- 미래 날짜 또는 반복 날짜에 자동으로 시작되도록 부하 테스트를 예약합니다.
- 애플리케이션 로드 테스트를 동시에 실행하거나 여러 테스트를 동시에 실행합니다.

이 구현 가이드에서는 Amazon Web Services(AWS) 클라우드에서 AWS에 분산 로드 테스트를 배포하기 위한 아키텍처 고려 사항 및 구성 단계에 대해 설명합니다. [AWS CloudFormation](#)에 대한 링크가 포함되어 있습니다. 보안 및 가용성에 대한 AWS 모범 사례를 사용하여 이 솔루션을 배포하는 데 필요한 AWS 서비스를 시작하고 구성하는 템플릿입니다.

이 가이드는 AWS 클라우드에 대한 실무 경험이 있는 IT 인프라 설계자, 관리자 및 DevOps 전문가를 대상으로 작성되었습니다.

비용

이 솔루션을 실행하는 동안 사용되는 AWS 서비스 비용은 귀하의 책임입니다. 전체 이 솔루션을 실행하는 데 드는 비용은 실행된 로드 테스트 횟수, 해당 로드 기간에 따라 달라집니다. 테스트 및 테스트의 일부로 사용되는 데이터의 양. 이번 개정 기준으로 운영 비용은 미국 동부(버지니아 북부) 지역의 기본 설정을 사용하는 이 솔루션의 비용은 월 약 \$30.90입니다. 비용 견적은 다음 요소를 가정합니다.

AWS 서비스	치수	비용 [USD]
AWS 파게이트	주문형 작업 10개(vCPU 2개 및 4GB 메모리 사용) 30시간 동안 달리다	\$29.62
아마존 다이아모DB	주문형 쓰기 용량 단위 1,000개 주문형 읽기 1,000개 용량 단위	\$0.0015
AWS 램다	요청 1,000개 총 소요시간 10분	\$1.25
AWS 단계 함수	1,000개의 상태 전환	\$0.025
총액: 월 \$30.90		

[예산](#)을 편성하는 것이 좋습니다 [AWS 비용 탐색기](#)를 통해 비용 관리에 도움이 됩니다. 가격은 변경될 수 있습니다. 자세한 내용은 여기에서 사용되는 각 AWS 서비스의 요금 웹페이지를 참조하세요. 해결책.

중요한

버전 1.3.0부터 CPU는 vCPU 2개, 메모리는 2개로 증가합니다.
4GB. 이러한 변경으로 인해 이전 버전에 비해 예상 비용이 증가합니다.
해결책. 로드 테스트에서 AWS 리소스에 대한 이러한 증가가 필요하지 않은 경우 다음을 수행할 수 있습니다.

그것들을 줄이세요. 자세한 내용은 이 가이드의 [컨테이너 리소스 늘리기](#) 섹션을 참조하세요 .

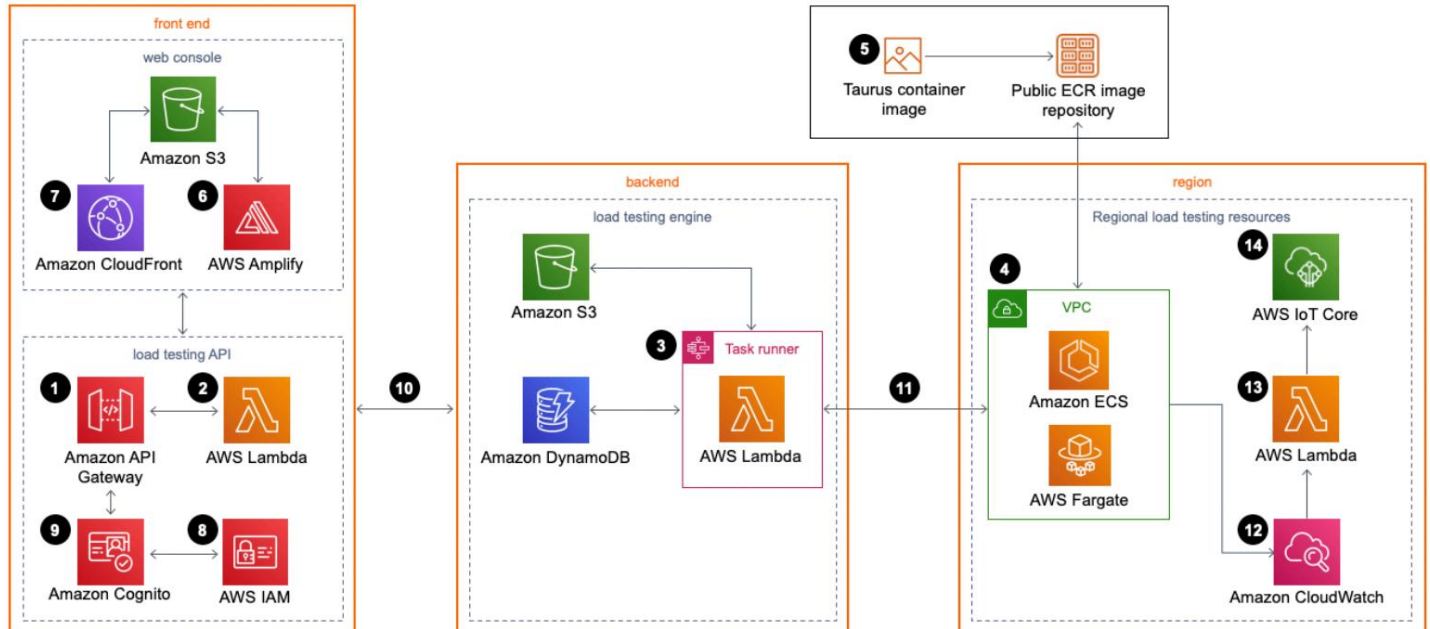
메모

이 솔루션은 테스트를 실행할 때 실시간 데이터를 포함하는 옵션을 제공합니다. 이 기능을 사용하려면 추가 비용이 발생하는 추가 AWS Lambda 함수와 AWS IoT Core 주제가 필요합니다.

가격은 변경될 수 있습니다. 자세한 내용은 이 솔루션에서 사용할 각 AWS 서비스에 대한 요금 웹페이지를 참조하세요.

아키텍처 개요

기본 매개변수를 사용하여 이 솔루션을 배포하면 AWS 클라우드에 다음 환경이 구축됩니다.



AWS 아키텍처의 분산 부하 테스트

AWS CloudFormation 템플릿은 다음 리소스를 배포합니다.

1. [Amazon API Gateway](#)를 활용하는 분산 로드 테스터 API 호출하는 솔루션의 마이크로서비스 ([AWS Lambda](#) 기능).
2. 마이크로서비스는 테스트 데이터를 관리하고 테스트를 실행하기 위한 비즈니스 로직을 제공합니다.
3. 이러한 마이크로서비스는 [Amazon Simple Storage Service](#)와 상호 작용합니다. (아마존 S3), 아마존 [다이나모DB](#), 및 [AWS 단계 함수](#) 테스트 시나리오 세부 정보 및 결과에 대한 저장소를 제공하고 테스트 시나리오를 실행합니다.
4. [Amazon Virtual Private Cloud](#) (Amazon VPC) 네트워크 토폴로지는 솔루션의 [Amazon Elastic Container Service](#)를 포함하여 배포됩니다. (Amazon ECS) 컨테이너는 [AWS Fargate](#)에서 실행됩니다.
5. 컨테이너에는 [항소자리](#)가 포함됩니다. 부하 테스트 개방형 컨테이너 [이니셔티브](#) (OCI) 규격 애플리케이션 성능을 테스트하기 위해 로드를 생성하는 데 사용되는 컨테이너 이미지입니다. Taurus는 오픈 소스 테스트 자동화 프레임워크입니다. 컨테이너 이미지는 [Amazon Elastic Container Registry](#)에서 AWS에 의해 호스팅됩니다. (Amazon ECR) 공개 저장소. ECR 이미지 저장소에 대한 자세한 내용은 [컨테이너 이미지 사용자 정의를 참조하세요](#).

6. [AWS Amplify](#) 로 구동되는 웹 콘솔 구성된 [Amazon S3 버킷](#)에 배포됩니다.

정적 웹 호스팅.

7. [아마존 클라우드프론트](#) 솔루션의 웹사이트 버킷 콘텐츠에 대한 안전한 공개 액세스를 제공합니다.

8. 초기 구성 중에 이 솔루션은 기본 솔루션 관리자 역할(IAM)도 생성합니다.

역할) 고객이 지정한 사용자 이메일 주소로 액세스 초대장을 보냅니다.

9. [아마존 코그니토](#) 사용자 풀은 콘솔에 대한 사용자 액세스와 분산 로드를 관리합니다.

테스터 API.

10. 이 솔루션을 배포한 후 웹 콘솔을 사용하여 정의하는 테스트 시나리오를 생성할 수 있습니다.

일련의 작업.

11. 마이크로서비스는 이 테스트 시나리오를 사용하여 리전의 AWS Fargate 작업에서 Amazon ECS를 실행합니다.

지정.

12. Amazon S3 및 DynamoDB에 결과를 저장하는 것 외에도 테스트가 완료되면

출력은 [Amazon CloudWatch](#) 에 기록됩니다.

13. 라이브 데이터 옵션을 선택하면 솔루션은 테스트가 실행된 각 리전에 대해 테스트 중에 AWS Fargate 작업에 대한 Amazon CloudWatch

로그를 Lambda 함수로 보냅니다.

14. 그런 다음 Lambda 함수는 [AWS IoT Core](#) 의 해당 주제에 데이터를 게시합니다. 메인 스택이 배포된 리전에서. [웹 콘솔은 주제를 구독하고 웹](#)

콘솔에서 테스트가 실행되는 동안 데이터를 볼 수 있습니다.

솔루션 구성요소

AWS의 분산 부하 테스트 솔루션은 프런트 엔드와 백엔드라는 두 가지 상위 수준 구성 요소로 구성됩니다.

프런트엔드

프런트 엔드는 솔루션의 백엔드와 상호 작용하는 데 사용하는 부하 테스트 API와 웹 콘솔로 구성됩니다.

부하 테스트 API

AWS의 분산 로드 테스트는 솔루션의 RESTful API를 호스팅하도록 Amazon API Gateway를 구성합니다. 사용자는 포함된 웹 콘솔과 RESTful API를 통해 테스트 데이터와 안전하게 상호 작용할 수 있습니다. API는 Amazon DynamoDB에 저장된 테스트 데이터에 액세스하기 위한 "현관" 역할을 합니다. 또한 API를 사용하여 솔루션에 구축한 확장 기능에 액세스할 수도 있습니다.

이 솔루션은 Amazon Cognito 사용자 풀의 사용자 인증 기능을 활용합니다.

사용자를 성공적으로 인증한 후 Amazon Cognito는 콘솔이 솔루션의 API(Amazon API Gateway 엔드포인트)에 요청을 제출할 수 있도록 하는 데 사용되는 JSON 웹 토큰을 발급합니다.

HTTPS 요청은 콘솔에서 토큰이 포함된 인증 헤더와 함께 API로 전송됩니다.

요청에 따라 API Gateway는 적절한 AWS Lambda 함수를 호출하여 DynamoDB 테이블에 저장된 데이터에 대해 필요한 작업을 수행하고, 테스트 시나리오를 Amazon S3에 JSON 객체로 저장하고, Amazon CloudWatch 지표 이미지를 검색하고, AWS에 테스트 시나리오를 제출합니다. Step Functions 상태 머신.

솔루션의 API에 대한 자세한 내용은 이 가이드의 [분산 부하 테스트 API](#) 섹션을 참조하세요.

웹 콘솔

이 솔루션에는 테스트를 구성 및 실행하고, 실행 중인 테스트를 모니터링하고, 자세한 테스트 결과를 보는 데 사용할 수 있는 웹 콘솔이 포함되어 있습니다. 콘솔은 Amazon S3에서 호스팅되고 Amazon CloudFront를 통해 액세스되는 ReactJS 애플리케이션입니다. 애플리케이션은 AWS Amplify를 활용하여 Amazon Cognito와 통합하여 사용자를 인증합니다. 웹 콘솔에는 실행 중인 테스트에 대한 실시간 데이터를 볼 수 있는 옵션도 포함되어 있으며, 여기에서 AWS IoT Core의 해당 주제를 구독합니다.

웹 콘솔은 이 부하 테스트 솔루션과 상호 작용하는 방법을 보여주기 위해 설계되었습니다.

프로덕션 환경에서는 특정 요구 사항에 맞게 웹 콘솔을 사용자 지정하거나 자체 콘솔을 구축하는 것이 좋습니다.

웹 콘솔 URL은 CloudFormation 출력에서 콘솔로 찾을 수 있는 CloudFront 배포 도메인 이름입니다. CloudFormation 템플릿을 실행하면 웹 콘솔 URL과 로그인에 필요한 일회용 비밀번호가 포함된 이메일도 받게 됩니다.

백엔드

백엔드는 테스트용 로드를 생성하는 데 사용하는 컨테이너 이미지 파이프라인과 로드 테스트 엔진으로 구성됩니다. 프론트엔드를 통해 백엔드와 상호 작용합니다. 또한 각 테스트에 대해 시작된 AWS Fargate 기반 Amazon ECS 작업에는 고유한 테스트 식별자(ID)가 태그로 지정됩니다. 이러한 테스트 ID 태그를 사용하여 이 솔루션의 비용을 모니터링할 수 있습니다. 자세한 내용은 [사용자 정의 비용 할당 태그](#)를 참조하세요. AWS Billing and Cost Management 사용 설명서 에 나와 있습니다 .

컨테이너 이미지 파이프라인

이 솔루션은 [Taurus](#)의 컨테이너 이미지를 활용합니다. 로드 테스트 프레임워크. 이 이미지는 Amazon Elastic Container Registry(Amazon ECR) 공용 리포지토리에서 호스팅됩니다. 이미지는 AWS Fargate 클러스터의 Amazon ECS에서 작업을 실행하는 데 사용됩니다.

자세한 내용은 이 가이드의 [컨테이너 이미지 사용자 지정](#) 섹션을 참조하세요.

테스트 인프라

솔루션은 기본 템플릿 외에도 여러 리전에서 테스트를 실행하는 데 필요한 리소스를 시작하는 보조 템플릿을 생성합니다. 템플릿은 Amazon S3에 저장되며, 템플릿에 대한 링크는 웹 콘솔에 제공됩니다. 보조 템플릿은 라이브 데이터 처리를 위한 VPC, AWS Fargate 클러스터 및 Lambda 함수를 생성합니다.

보조 리전을 시작하는 방법에 대한 자세한 내용은 이 가이드의 [다중 리전 배포](#) 섹션을 참조하세요.

부하 테스트 엔진

분산 로드 테스트 솔루션은 Amazon Elastic Container Service(Amazon ECS) 및 AWS Fargate를 사용하여 여러 지역에 걸쳐 수천 명의 연결된 사용자를 시뮬레이션하여 초당 선택된 수의 트랜잭션을 생성합니다.

포함된 웹 콘솔을 사용하여 테스트의 일부로 실행될 작업에 대한 매개변수를 정의합니다. 솔루션은 이러한 매개변수를 사용하여 JSON 테스트 시나리오를 생성하고 이를 Amazon S3에 저장합니다.

AWS Step Functions 상태 머신은 AWS Fargate 클러스터에서 Amazon ECS 작업을 실행하고 모니터링합니다. AWS Step Functions 상태 머신에는 ecr-checker AWS Lambda 함수, task-status-checker AWS Lambda 함수, 작업 실행자 AWS Lambda 함수, 작업 취소자 AWS Lambda 함수 및 결과 구문 분석기 AWS Lambda 함수가 포함되어 있습니다. 워크플로에 대한 자세한 내용은 이 가이드의 [테스트 워크플로](#) 섹션을 참조하세요. 테스트 결과에 대한 자세한 내용은 이 가이드의 [테스트 결과](#) 섹션을 참조하세요. 테스트 취소 워크플로에 대한 자세한 내용은 이 가이드의 [테스트 취소 워크플로](#) 섹션을 참조하세요.

라이브 데이터를 선택하면 솔루션은 해당 지역의 Fargate 작업에 해당하는 CloudWatch 로그를 통해 각 지역에서 실시간 데이터 게시자 Lambda 함수를 시작합니다. 그런 다음 솔루션은 메인 스택을 시작한 리전 내 AWS IoT Core의 주제에 데이터를 처리하고 게시합니다. 자세한 내용은 이 가이드의 [라이브 데이터](#) 섹션을 참조하세요.

보안

AWS 인프라에 시스템을 구축하면 보안 책임이 귀하와 AWS 간에 공유됩니다. 이 공유 모델은 AWS가 호스트 운영 체제 및 가상화 계층부터 서비스가 운영되는 시설의 물리적 보안에 이르기까지 구성 요소를 운영, 관리 및 제어하므로 운영 부담을 줄일 수 있습니다. AWS의 보안에 대한 자세한 내용을 보려면 [AWS 클라우드 보안을 방문하십시오](#).

IAM 역할

AWS Identity and Access Management(IAM) 역할을 통해 고객은 AWS의 서비스 및 사용자에게 세분화된 액세스 정책과 권한을 할당할 수 있습니다. 이 솔루션은 솔루션의 AWS Lambda 함수에 이 솔루션에 사용되는 다른 AWS 서비스에 대한 액세스 권한을 부여하는 역할을 포함하여 여러 IAM 역할을 생성합니다.

아마존 클라우드프론트

이 솔루션은 [호스팅되는](#) 정적 웹 사이트를 배포합니다. Amazon [Simple Storage Service](#)(Amazon S3) 버킷에 있습니다. 지연 시간을 줄이고 보안을 향상시키기 위해 이 솔루션에는 솔루션의 웹 사이트 버킷 콘텐츠에 대한 안전한 공개 액세스를 제공하는 데 도움이 되는 특수 CloudFront 사용자인 원본 액세스 ID가 있는 Amazon CloudFront 배포가 포함되어 있습니다. 자세한 내용은 [원본 액세스 ID를 사용하여 Amazon S3 콘텐츠에 대한 액세스 제한을 참조하십시오](#).

AWS Fargate 보안 그룹

기본적으로 이 솔루션은 AWS Fargate 보안 그룹의 아웃바운드 규칙을 공개합니다.

AWS Fargate가 모든 곳으로 트래픽을 전송하는 것을 차단하려면 아웃바운드 규칙을 특정 CIDR(Classless Inter-Domain Routing)로 변경하십시오.

이 보안 그룹에는 동일한 보안 그룹에 속한 모든 소스에 대한 포트 50,000의 로컬 트래픽을 허용하는 인바운드 규칙도 포함되어 있습니다. 이는 컨테이너가 서로 통신할 수 있도록 하는 데 사용됩니다.

네트워크 스트레스 테스트

[네트워크 스트레스 테스트 정책](#)에 따라 이 솔루션을 사용할 책임은 귀하에게 있습니다. 이 정책은 [대용량 네트워크 테스트를 직접 실행하려는 경우와 같은](#) 상황을 다룹니다.

Amazon EC2 인스턴스에서 다른 Amazon EC2 인스턴스, AWS 속성/서비스 또는 외부 엔드포인트와 같은 다른 위치로. 이러한 테스트를 스트레스 테스트, 부하 테스트 또는 게임데이 테스트라고도 합니다. 대부분의 고객 테스트에는 이 정책이 적용되지 않습니다. 그러나 전체적으로 1분 이상, 1Gbps(초당 10억 비트) 또는 1Gpps 이상을 유지하는 트래픽을 생성한다고 생각하는 경우 이 정책을 참조하세요. (초당 10억 패킷)

공용 사용자 인터페이스에 대한 액세스 제한

IAM 및 Amazon Cognito에서 제공하는 인증 및 권한 부여 메커니즘 이상으로 공용 사용자 인터페이스에 대한 액세스를 제한하려면 [AWS WAF\(웹 애플리케이션 방화벽\)](#)을 사용하십시오.

[보안 자동화 솔루션.](#)

이 솔루션은 일반적인 웹 기반 공격을 필터링하는 일련의 AWS WAF 규칙을 자동으로 배포합니다.

사용자는 AWS WAF 웹 액세스 제어 목록(ACL)에 포함된 규칙을 정의하는 사전 구성된 보호 기능 중에서 선택할 수 있습니다.

디자인 고려 사항

지원되는 애플리케이션

이 솔루션은 AWS 계정에서 애플리케이션까지 네트워크 연결이 있는 한 클라우드 기반 애플리케이션과 온프레미스 애플리케이션을 지원합니다. 이 솔루션은 HTTP 또는 HTTPS를 사용하는 API를 지원합니다. 또한 HTTP 요청 헤더를 제어할 수 있으므로 승인 또는 사용자 정의 헤더를 추가하여 토큰이나 API 키를 전달할 수 있습니다.

JMeter 스크립트 지원

이 솔루션의 사용자 인터페이스(UI)를 사용하여 테스트 시나리오를 생성할 때 JMeter 테스트 스크립트를 사용할 수 있습니다. JMeter 스크립트 파일을 선택하면 `<stack-name>`- 시나리오 버킷 Amazon Simple Storage Service(Amazon S3) 버킷에 업로드됩니다. Amazon Elastic Container Service(Amazon ECS) 작업이 실행 중이면 JMeter 스크립트가 `<stack-name>`-scenariosbucket Amazon S3 버킷에서 다운로드되고 테스트가 실행됩니다.

JMeter 입력 파일이 있는 경우 JMeter 스크립트와 함께 입력 파일을 압축할 수 있습니다. 테스트 시나리오를 생성할 때 zip 파일을 선택할 수 있습니다.

플러그인을 포함하려는 경우 번들 zip 파일의 `/plugins` 하위 디렉터리에 포함된 모든 `.jar` 파일이 JMeter 확장 디렉터리에 복사되어 로드 테스트에 사용할 수 있습니다.

메모

JMeter 스크립트 파일에 JMeter 입력 파일을 포함하는 경우 JMeter 스크립트 파일에 입력 파일의 상대 경로를 포함해야 합니다. 또한 입력 파일은 상대 경로에 있어야 합니다. 예를 들어, JMeter 입력 파일과 스크립트 파일이 `/home/` 사용자 디렉터리에 있고 JMeter 스크립트 파일의 입력 파일을 참조하는 경우 입력 파일의 경로는 `./INPUT_FILES`여야 합니다. 대신 `/home/user/INPUT_FILES`를 사용하면 입력 파일을 찾을 수 없기 때문에 테스트가 실패합니다.

JMeter 플러그인을 포함하는 경우 `.jar` 파일은 zip 파일 루트 내의 `/plugins`라는 하위 디렉터리에 번들로 포함되어야 합니다. zip 파일의 루트를 기준으로 `.jar` 파일의 경로는 `./plugins/BUNDLED_PLUGIN.jar`이어야 합니다.

JMeter 스크립트 사용 방법에 대한 자세한 내용은 [JMeter 사용자 매뉴얼](#)을 참조하세요.

테스트 예약

나중에 실행되도록 테스트를 예약하거나 지금 실행 옵션을 사용할 수 있습니다. 나중에 일회성 실행으로 테스트를 예약하거나 첫 번째 실행 날짜와 계획된 반복을 지정하는 반복 테스트를 설정할 수 있습니다. 반복 옵션에는 매일, 매주, 격주 및 매월이 포함됩니다. 예약 작동 방식에 대한 자세한 내용은 이 가이드의 [테스트 예약 워크플로](#) 섹션을 참조하세요.

부하 테스트 할당량

AWS Fargate 시작 유형을 사용하여 Amazon ECS에서 실행할 수 있는 최대 작업 수는 작업의 vCPU 크기를 기반으로 합니다. AWS의 분산 부하 테스트의 기본 작업 크기는 vCPU 2개입니다. 현재 기본 할당량을 보려면 [Amazon ECS 서비스 할당량을 참조하세요](#). 현재 계정 할당량은 나열된 할당량과 다를 수 있습니다. 계정별 할당량을 확인하려면 AWS Management Console에서 [Fargate 온디맨드 vCPU 리소스](#) 수에 대한 서비스 할당량을 확인하세요.

증가를 요청하는 방법에 대한 지침은 [AWS 서비스 할당량](#)을 참조하세요. AWS 일반 [참조 안내서](#)에서.

Taurus 부하 테스트 컨테이너 이미지는 작업당 동시 연결을 제한하지 않지만 이것이 무제한의 사용자를 지원할 수 있다는 의미는 아닙니다. 컨테이너가 테스트를 위해 생성할 수 있는 동시 사용자 수를 확인하려면 이 가이드의 [사용자 수 확인](#) 섹션을 참조하세요.

메모

기본 설정을 기준으로 권장되는 동시 사용자 제한은 200명입니다.

동시 테스트

이 솔루션에는 각 테스트에 대한 Amazon CloudWatch 대시보드가 포함되어 있으며 Amazon ECS 클러스터에서 해당 테스트에 대해 실행 중인 모든 작업의 결합된 출력을 실시간으로 표시합니다. CloudWatch 대시보드에는 평균 응답 시간, 동시 사용자 수, 성공한 요청 수, 실패한 요청 수가 표시됩니다. 각 지표는 초 단위로 집계되며 대시보드는 1분마다 업데이트됩니다.

Amazon EC2 테스트 정책

네트워크 트래픽이 1Gbps 미만으로 유지되는 한 이 솔루션을 사용하여 로드 테스트를 실행하는 데 AWS의 승인이 필요하지 않습니다. 테스트에서 1Gbps를 초과하는 속도가 생성되면 AWS에 문의하세요. 자세한 내용은 [Amazon EC2 테스트 정책을 참조하세요](#).

Amazon CloudFront 로드 테스트 정책

CloudFront 엔드포인트의 로드 테스트를 계획하는 경우 [로드 테스트 지침](#)을 참조하세요. [Amazon CloudFront](#) 개발자 안내서 에 있습니다. 또한 여러 작업과 지역에 트래픽을 분산시키는 것이 좋습니다. 부하 테스트를 위해 최소 30분의 준비 시간을 제공하십시오. 초당 500,000개 이상의 요청을 보내거나 300Gbps 이상의 데이터를 요구하는 부하 테스트의 경우 먼저 트래픽 전송에 대한 사전 승인을 얻는 것이 좋습니다. CloudFront는 CloudFront 서비스 가용성에 영향을 미치는 승인되지 않은 로드 테스트 트래픽을 제한할 수 있습니다.

사용자 관리

초기 구성 중에 Amazon Cognito가 솔루션의 웹 콘솔에 대한 액세스 권한을 부여하는 데 사용하는 사용자 이름과 이메일 주소를 제공합니다. 콘솔은 사용자 관리 기능을 제공하지 않습니다. 사용자를 더 추가하려면 Amazon Cognito 콘솔을 사용해야 합니다. 자세한 내용은 [사용자 풀에서 사용자 관리를 참조하세요](#). Amazon Cognito 개발자 안내서 에 있습니다.

지역 배포

이 솔루션은 특정 AWS 리전에서만 사용할 수 있는 Amazon Cognito를 사용합니다. 따라서 Amazon Cognito를 사용할 수 있는 지역에 이 솔루션을 배포해야 합니다. 지역별 최신 서비스 가용성은 [AWS 지역 서비스 목록을 참조하세요](#).

AWS CloudFormation 템플릿

이 솔루션은 AWS CloudFormation을 사용하여 분산 로드 테스트 배포를 자동화합니다.

AWS에서. 여기에는 다음과 같은 AWS CloudFormation 템플릿이 포함되어 있습니다. 이 템플릿은 이전에 다운로드할 수 있습니다.

전개:

View template

배포

load-testing-on-aws.template - 이 템플릿을 사용하여 솔루션 및 모든 관련 구성 요소를 시작합니다. 기본 구성은 Amazon Elastic Container Service(Amazon ECS), AWS Fargate, Amazon Virtual Private Cloud(Amazon VPC), AWS Lambda, Amazon Simple 스토리지 서비스(Amazon S3), AWS Step Functions, Amazon DynamoDB, Amazon CloudWatch Logs, Amazon API Gateway, Amazon Cognito, AWS Identity and Access Management(IAM) 및 Amazon CloudFront이지만 특정 네트워크 요구 사항에 따라 템플릿을 사용자 지정할 수도 있습니다.

자동화된 배포

중요한

v3.2.6 이전 버전에서 최신 버전으로 스택을 업데이트하는 경우 스택을 업데이트하기 전에 [이 섹션](#)을 읽으십시오.

자동화된 배포를 시작하기 전에 이 가이드에 설명된 아키텍처 및 기타 고려 사항을 검토하세요. 이 섹션의 단계별 지침에 따라 AWS에서 분산 로드 테스트를 구성하고 계정에 배포하십시오.

배포 시간: 약 15분

스택 시작

중요한

이 솔루션에는 익명화된 운영 지표를 AWS로 보내는 옵션이 포함되어 있습니다. 우리는 이 데이터를 사용하여 고객이 이 솔루션과 관련 서비스를 사용하는 방법을 더 잘 이해하고

제품. AWS는 이 설문조사를 통해 수집된 데이터를 소유합니다. 데이터 수집에는 AWS 개인정보 보호정책이 적용됩니다.

이 기능을 옵트아웃하려면 템플릿을 다운로드하고 AWS CloudFormation 매핑 섹션을 수정한 다음 AWS CloudFormation 콘솔을 사용하여 템플릿을 업로드하고 솔루션을 배포하십시오. 자세한 내용은 이 가이드의 [익명 데이터 수집](#) 섹션을 참조하세요.

이 자동화된 AWS CloudFormation 템플릿은 AWS에 분산 로드 테스트를 배포합니다.

메모

이 솔루션을 실행하는 동안 사용되는 AWS 서비스 비용은 귀하의 책임입니다. 자세한 내용은 이 가이드의 [비용](#) 섹션을 방문하고 이 솔루션에 사용된 각 AWS 서비스에 [대한](#) 요금 웹페이지를 참조하세요.

1. AWS Management Console에 로그인하고 아래 버튼을 클릭하여 AWS CloudFormation 템플릿에 대한 분산 부하 테스트.

**Launch
solution**

또는 [템플릿을 다운로드](#) 할 수도 있습니다. [자신의 출발점으로 구현.](#)

2. 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 이 솔루션을 시작하려면 다른 AWS 리전에서는 콘솔 탐색 모음의 리전 선택기를 사용합니다.

메모

이 솔루션은 현재 특정 AWS 리전에서 사용 가능한 Amazon Cognito를 사용합니다. 오직. 따라서 Amazon이 있는 AWS 리전에서 이 솔루션을 시작해야 합니다. 코그니토를 사용할 수 있습니다. 지역별 최신 서비스 가용성은 다음을 참조하세요. [AWS 지역 서비스 목록.](#)

3. 스택 생성 페이지에서 Amazon S3 URL 에 올바른 템플릿 URL이 표시되는지 확인합니다. 텍스트 상자를 클릭하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택에 이름을 할당합니다.
5. 매개변수 에서 템플릿의 매개변수를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음 기본값을 사용합니다.

매개변수	기본	설명
관리자 이름	<입력이 필요합니다>	초기 사용자 이름 솔루션 관리자.
관리자 이메일	<입력이 필요합니다>	이메일 주소 관리자 사용자. 후에 실행하면 이메일이 전송됩니다 콘솔을 사용하여 이 주소로 로그인 지침.

매개변수	기본	설명
기존 VPC ID	<선택입력>	VPC가 있는 경우 사용하고 싶는데 이미 사용 중입니다. 생성된 경우 해당 ID를 입력하세요. 스택이 있는 동일한 리전에 있는 기존 VPC 배치되었습니다. 예를 들어, vpc-1a2b3c4d5e6f .
첫 번째 기존 서브넷	<선택입력>	첫 번째 서브넷의 ID 기존 VPC 내에서. 이 서브넷에는 경로가 필요합니다. 컨테이너 이미지를 가져오기 위 해 인터넷에 접속 테스트를 실행 중입니다. 예를 들어, 서브넷-7h8i9j0k .
두 번째 기존 서브넷	<선택입력>	두 번째 서브넷의 ID 기존 VPC 내에서. 이 서브넷에는 경로가 필요합니다. 컨테이너 이미지를 가져오기 위 해 인터넷에 접속 테스트를 실행 중입니다. 예를 들어, 서브넷-1x2y3z .
AWS Fargate VPC CIDR 차단하다	192.168.0.0/16	값을 제공하지 않는 경우 기존 VPC의 경우 솔루션을 위한 CIDR 블록 - Amazon VPC 생성 에 대한 IP 주소가 포함되어 있습니다. AWS 파게이트.

매개변수	기본	설명
AWS Fargate 서브넷 A CIDR 블록	192.168.0.0/20	값을 제공하지 않는 경우 기존 VPC의 경우 CIDR 블록에 IP 가 포함됩니다. Amazon VPC 주소 서브넷 A.
AWS Fargate 서브넷 B CIDR 차단하다	192.168.16.0/20	값을 제공하지 않는 경우 기존 VPC의 경우 CIDR 블록에는 IP가 포함됩니다. Amazon VPC 주소 서브넷 B.
AWS Fargate 보안 그룹 CIDR 블록	0.0.0.0/0	제한하는 CIDR 블록 Amazon ECS 컨테이너 아웃바운드 액세스.

6. 다음을 선택합니다.

7. 스택 옵션 구성 페이지에서 다음을 선택합니다 .

8. 검토 페이지에서 설정을 검토하고 확인합니다. 다음 사항을 인정하는 확인란을 선택하세요.

템플릿은 AWS Identity and Access Management(IAM) 리소스를 생성합니다.

9. 스택 생성을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택 상태를 볼 수 있습니다 .

약 15분 후에 CREATE_COMPLETE 상태를 받게 됩니다 .

메모

기본 AWS Lambda 함수 외에도 이 솔루션에는 초기 구성 중에 또는 리소스가 필요할 때만 실행되는 사용자 지정 리소스 Lambda 함수가 포함되어 있습니다.

업데이트되거나 삭제됩니다.

이 솔루션을 실행할 때 사용자 지정 리소스 Lambda 함수는 비활성화됩니다. 하지만,
관련 리소스를 관리하기 위해 필요한 기능이므로 삭제하지 마세요.

다중 지역 배포

배포 시간: 약 5분

여러 리전에서 테스트를 실행할 수 있습니다. 분산 부하 테스트 솔루션을 배포할 때, 3개의 Amazon S3 버킷을 생성합니다. 솔루션은 보조 지역 스택을 생성하고 이를 Amazon S3 시나리오 버킷.

메모

버킷 명명 규칙은 `<stack-name>-`

`dltestrunnerstoragedltestscenariosbucket<[0-9][0-9]..<[0-9]>` 입니다.

`[0-9]..` 버킷 이름에 키워드 시나리오를 사용하여 찾을 수 있습니다.

S3 콘솔로 이동한 다음 버킷으로 이동합니다.

다중 지역 배포를 실행하려면 지역 CloudFormation 템플릿을 배포해야 합니다.

테스트를 실행하려는 지역의 Amazon S3 시나리오 버킷에 저장됩니다. 당신은 할 수 있습니다

다음을 수행하여 지역 템플릿을 설치합니다.

1. 솔루션 웹 콘솔의 상단 메뉴에서 지역 관리 로 이동합니다.
2. 클립보드 아이콘을 사용하여 Amazon S3에서 CloudFormation 템플릿 링크를 복사합니다.
3. [AWS CloudFormation 콘솔](#) 에 로그인 올바른 지역을 선택하세요.
4. 스택 생성 페이지에서 Amazon S3 URL 에 올바른 템플릿 URL이 표시되는지 확인합니다 .
텍스트 상자를 클릭하고 다음을 선택합니다.
5. 스택 세부 정보 지정 페이지에서 솔루션 스택에 이름을 할당합니다.
6. 매개변수 에서 템플릿의 매개변수를 검토하고 필요에 따라 수정합니다.
이 솔루션은 다음 기본값을 사용합니다.

매개변수	기본	설명
기존 VPC ID	<선택입력>	VPC가 있는 경우 사용하고 싶는데 이미 사용 중입니다. 생성된 경우 해당 ID를 입력하세요. 동일한 기존 VPC 스택이 있는 지역

매개변수	기본	설명
		배치되었습니다. 예를 들어, vpc-1a2b3c4d5e6f .
첫 번째 기존 서브넷	<선택입력>	<p>첫 번째 서브넷의 ID</p> <p>기존 VPC 내에서.</p> <p>이 서브넷에는 경로가 필요합니다.</p> <p>컨테이너 이미지를 가져오기 위</p> <p>해 인터넷에 접속</p> <p>테스트를 실행 중입니다. 예를 들어, 서브넷-7h8i9j0k .</p>
두 번째 기존 서브넷	<선택입력>	<p>두 번째 서브넷의 ID</p> <p>기존 VPC 내에서.</p> <p>이 서브넷에는 경로가 필요합니다.</p> <p>컨테이너 이미지를 가져오기 위</p> <p>해 인터넷에 접속</p> <p>테스트를 실행 중입니다. 예를 들어, 서브넷-1x2y3z .</p>
AWS Fargate VPC CIDR 차단하다	192.168.0.0/16	<p>값을 제공하지 않는 경우</p> <p>기존 VPC의 경우</p> <p>솔루션을 위한 CIDR 블록 -</p> <p>Amazon VPC 생성</p> <p>에 대한 IP 주소가 포함되어 있습니다.</p> <p>AWS 파게이트.</p>
AWS Fargate 서브넷 A CIDR 블록	192.168.0.0/20	<p>값을 제공하지 않는 경우</p> <p>기존 VPC의 경우</p> <p>CIDR 블록에는 IP가 포함됩니다.</p> <p>Amazon VPC 주소</p> <p>서브넷 A.</p>

매개변수	기본	설명
AWS Fargate 서브넷 B CIDR 차단하다	192.168.16.0/20	기존 VPC에 대한 값을 제공하지 않으면 CIDR 블록에 Amazon VPC 서브넷 B의 IP 주소가 포함됩니다.
AWS Fargate 보안 그룹 CIDR 블록	0.0.0.0/0	제한하는 CIDR 블록 Amazon ECS 컨테이너 아웃바운드 액세스.

7. 다음을 선택합니다.

8. 스택 옵션 구성 페이지에서 다음을 선택합니다 .

9. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management(IAM) 리소스를 생성한다는 것을 확인하는 확인란을 선택해야 합니다.

10. 스택 생성을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택 상태를 볼 수 있습니다 .

약 5분 후에 CREATE_COMPLETE 상태를 받게 됩니다 .

지역이 성공적으로 배포되면 웹 콘솔에 표시됩니다. 테스트를 생성하면 새 지역이 지역 관리 모달 에 나열됩니다 . 테스트 생성 시 선택하여 테스트에서 이 지역을 사용할 수 있습니다. 솔루션은 해당 지역의 테스트 리소스에 관한 필수 정보가 포함된 시나리오 테이블에서 시작된 각 지역에 대해 DynamoDB 항목을 생성합니다. 웹 콘솔에서 테스트 결과를 지역별로 정렬할 수 있습니다. API 제약으로 인해 다중 리전 테스트에서 모든 리전의 집계 결과를 Amazon CloudWatch 측정치에 그래프로 표시하여 볼 수만 있습니다. 테스트가 완료되면 테스트 결과에서 그래프의 소스 코드를 확인할 수 있습니다.

메모

웹 콘솔 없이 지역 스택을 시작할 수 있습니다. Amazon S3 시나리오 버킷에서 지역 템플릿에 대한 링크를 얻고 필요한 지역에서 지역 스택을 시작할 때 이를 소스로 제공합니다. 또는 템플릿을 다운로드하여 원하는 지역의 소스로 업로드할 수 있습니다.

추가 리소스

AWS 서비스	
Amazon 탄력적 컨테이너 서비스	아마존 코그니토
Amazon Elastic Container Registry	아마존 클라우드프론트
AWS 파게이트	아마존 가상 프라이빗 클라우드
AWS 람다	AWS 서비스 카탈로그
아마존 심플 스토리지 서비스	AWS 시스템 관리자
아마존 다이나모DB	AWS ID 및 액세스 관리
아마존 클라우드워치	AWS 증폭
Amazon CloudWatch 이벤트	AWS 클라우드포메이션
아마존 API 게이트웨이	AWS IoT 코어
AWS 단계 함수	

기타 리소스

- [황소자리](#)

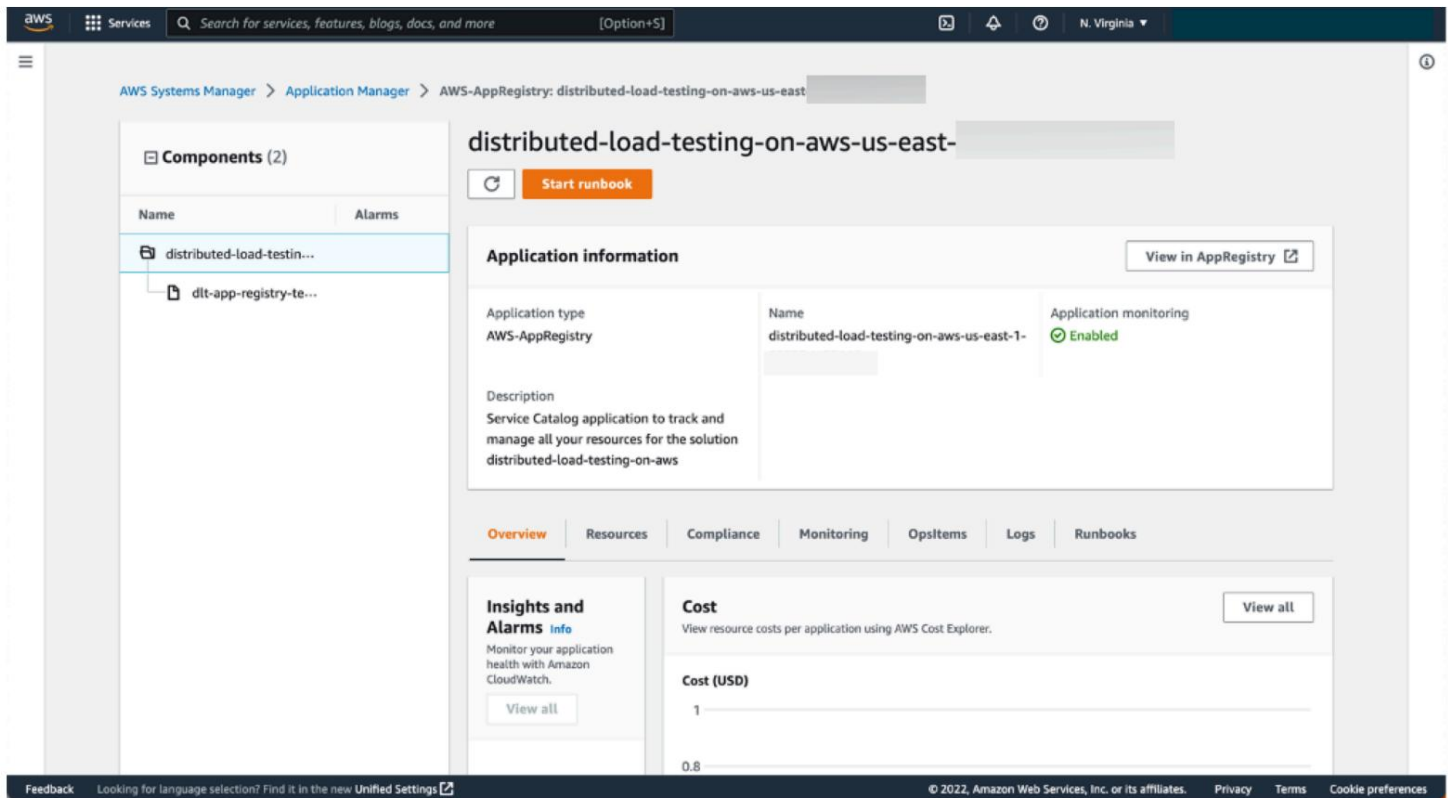
서비스 카탈로그로 솔루션 모니터링 앱등록

AWS의 분산 로드 테스트 솔루션에는 CloudFormation 템플릿과 기본 리소스를 [Service Catalog AppRegistry](#) 모두에 애플리케이션으로 등록하기 위한 Service Catalog AppRegistry 리소스가 포함되어 있습니다. 및 [AWS 시스템 관리자 애플리케이션 관리자](#).

AWS Systems Manager Application Manager는 이 솔루션과 해당 리소스에 대한 애플리케이션 수준 보기를 제공하므로 다음을 수행할 수 있습니다.

- 리소스, 스택 및 AWS 계정 전체에 배포된 리소스 비용 및 로그를 모니터링합니다.
중앙 위치에서 이 솔루션과 연결됩니다.
- 배포 상태, CloudWatch 경보, 리소스 구성, 운영 문제 등 애플리케이션 컨텍스트에서 이 솔루션의 리소스에 대한 운영 데이터를 봅니다.

다음 그림은 Application Manager의 AWS 스택에 대한 분산 로드 테스트에 대한 애플리케이션 보기의 예를 보여줍니다.



Application Manager의 AWS 스택에 대한 분산 로드 테스트

메모

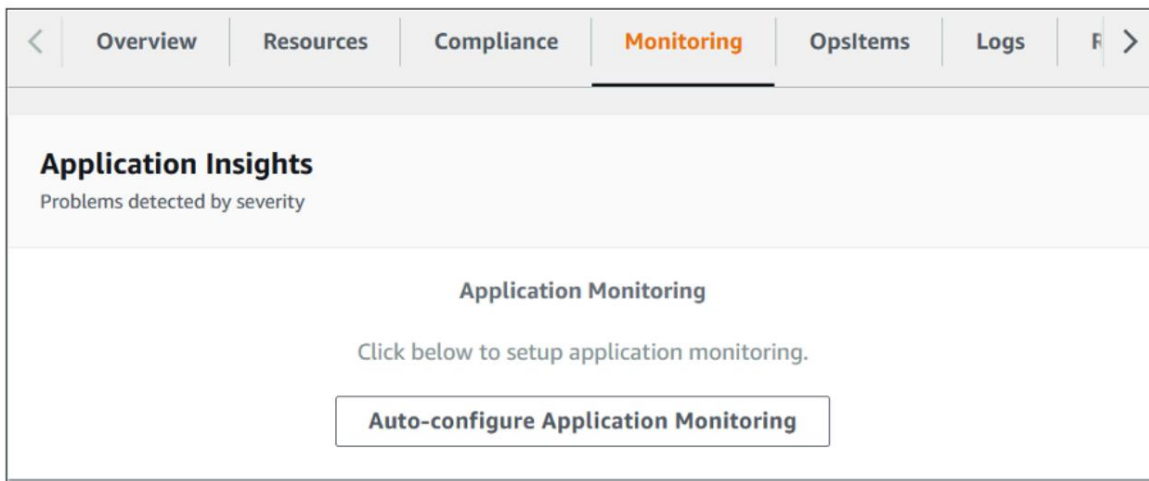
CloudWatch Application Insights, AWS Cost Explorer 및 이 솔루션과 연결된 비용 할당 태그를 활성화해야 합니다. 기본적으로 활성화되지 않습니다.

CloudWatch Application Insights 활성화

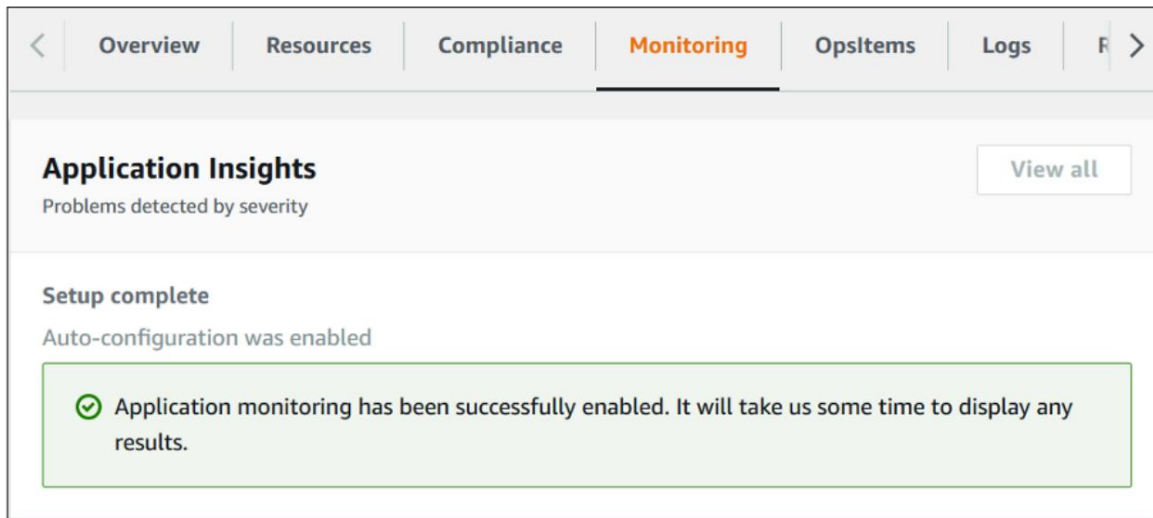
1. [시스템 관리자 콘솔을 엽니다](#). 탐색 창에서 [애플리케이션](#) 관리자를 선택합니다.
2. 애플리케이션 에서 AppRegistry 애플리케이션을 선택합니다 .
3. AppRegistry 애플리케이션 에서 이 솔루션의 애플리케이션 이름을 검색하고 선택합니다.

다음에 응용 프로그램 관리자를 열면 AppRegistry 응용 프로그램 범주에서 솔루션에 대한 새 응용 프로그램을 찾을 수 있습니다.

4. 구성 요소 트리에서 활성화하려는 애플리케이션 스택을 선택합니다.
5. 모니터링 탭 의 Application Insights 에서 애플리케이션 모니터링 자동 구성을 선택합니다 .



6. 이제 애플리케이션 모니터링이 활성화되고 다음 상태 상자가 나타납니다.



AWS 비용 탐색기 활성화

AWS Cost Explorer와의 통합을 통해 Application Manager 콘솔 내에서 애플리케이션 및 애플리케이션 구성 요소와 관련된 비용의 개요를 볼 수 있습니다. 비용 탐색기는 시간 경과에 따른 AWS 리소스 비용 및 사용량에 대한 보기를 제공하여 비용을 관리하는 데 도움이 됩니다. 솔루션에 대한 Cost Explorer를 활성화하려면 다음을 수행하십시오.

1. [AWS Cost Management 콘솔에 로그인합니다.](#)
2. 왼쪽 탐색 메뉴에서 Cost Explorer를 선택합니다.
3. 비용 탐색기 시작 페이지에서 비용 탐색기 시작을 선택합니다.

활성화 프로세스를 완료하는 데 최대 24시간이 걸릴 수 있습니다. 활성화되면 비용 탐색기 사용자 인터페이스를 열어 솔루션에 대한 비용 데이터를 추가로 분석할 수 있습니다.

솔루션과 관련된 비용 할당 태그 활성화

비용 탐색기를 활성화한 후 이 솔루션의 비용을 보려면 비용 할당 태그를 활성화해야 합니다. 비용 할당 태그는 조직의 마스터 계정에서만 활성화할 수 있습니다. 비용 할당 태그를 활성화하려면 다음을 수행하십시오.

1. [AWS Billing and Cost Management 콘솔](#)에 로그인합니다. 왼쪽 탐색 메뉴에서 비용 할당 태그를 선택합니다.
2. 비용 할당 태그 페이지에서 AppManagerCFNStackKey 태그를 필터링한 후 표시된 결과에서 태그를 선택합니다.

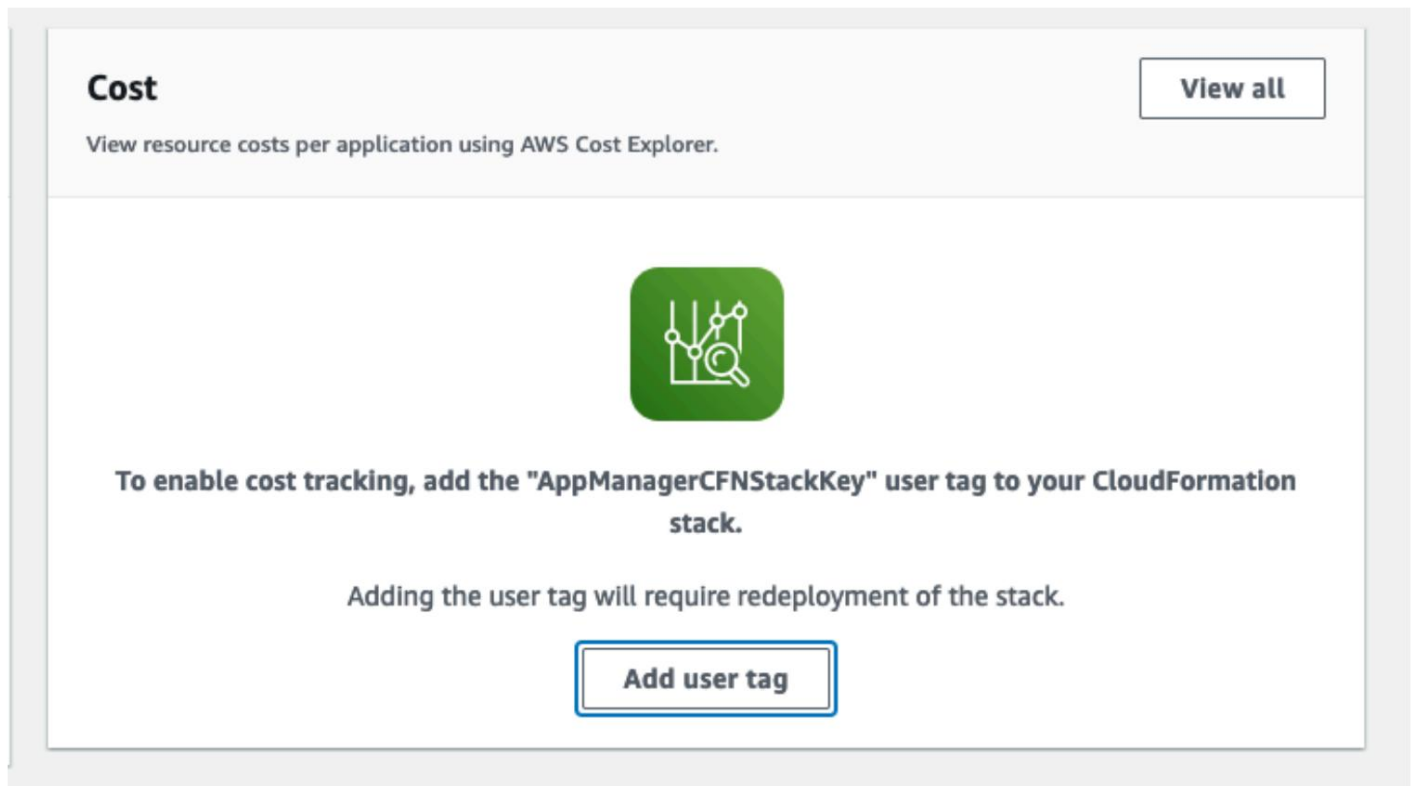
3. 활성화를 선택합니다.

활성화 프로세스가 완료되고 태그 데이터가 표시되는 데 최대 24시간이 걸릴 수 있습니다.

솔루션과 관련된 비용 태그 확인

솔루션과 연결된 비용 할당 태그를 활성화한 후 이 솔루션에 대한 비용을 보려면 비용 할당 태그를 확인해야 합니다. 비용 할당 태그를 확인하려면:

1. 시스템 관리자 콘솔에 로그인합니다.
2. 탐색 창에서 애플리케이션 관리자를 선택합니다.
3. 애플리케이션 에서 이 솔루션의 애플리케이션 이름을 선택하고 선택합니다.
4. 개요 탭 의 비용 에서 사용자 태그 추가를 선택합니다 .



5. 사용자 태그 추가 페이지에서 확인을 입력한 후 사용자 태그 추가를 선택합니다.

활성화 프로세스가 완료되고 태그 데이터가 표시되는 데 최대 24시간이 걸릴 수 있습니다.

컨테이너 이미지 맞춤설정

이 솔루션은 AWS에서 관리하는 공개 Amazon Elastic Container Registry(Amazon ECR) 이미지 리포지토리를 사용하여 구성된 테스트를 실행하는 데 사용되는 Taurus 이미지를 저장합니다. 컨테이너 이미지를 사용자 지정하려면 이미지를 다시 빌드하고 AWS 계정의 ECR 이미지 리포지토리에 푸시하면 됩니다.

이 솔루션을 사용자 지정하려면 기본 컨테이너 이미지를 사용하거나 필요에 맞게 이 컨테이너를 편집하면 됩니다. 솔루션을 사용자 지정하는 경우 사용자 지정된 솔루션을 빌드하기 전에 다음 코드 샘플을 사용하여 환경 변수를 선언하세요.

```
#!/bin/bash
```

```
내보내기 REGION=aws-region-code # 솔루션을 시작할 AWS 지역(예: us-east-1) 내보내기 BUCKET_PREFIX=my-bucket-name # 지역 코드
가 없는 버킷 이름의 접두사 내보내기 BUCKET_NAME=$BUCKET_PREFIX-$REGION # 코드가 상주할 전체 버킷 이름 내보내기
SOLUTION_NAME=my-solution-name 내보내기 VERSION=my-version # 사용자 정의된 코드의 버전 번호 내보내기
PUBLIC_ECR_REGISTRY=public.ecr.aws/awssolutions/
distributed-load-testing-on-aws -load-tester # 다른 컨테이너 이미지를 사용하려는 경우 컨테이너 레지스트리
및 이미지로 교체 내보내기 PUBLIC_ECR_TAG=v3.1.0 # 다른 컨테이너 이미지를 사용하려는 경우 컨테이너 이미지 태그로 교체
```

컨테이너 이미지를 사용자 지정하기로 선택한 경우 프라이빗 이미지 리포지토리 또는 AWS 계정의 퍼블릭 이미지 리포지토리에서 호스팅할 수 있습니다. 이미지 리소스는 코드 베이스에 있는 배포/ecr/distributed-load-testing-on-aws-load-tester 디렉터리에 있습니다.

이미지를 빌드하고 호스트 대상에 푸시할 수 있습니다.

- 프라이빗 Amazon ECR 리포지토리 및 이미지에 대해서는 [Amazon ECR 프라이빗 리포지토리를 참조하세요. 그리고 비공개 이미지 Amazon ECR 사용 설명서](#)에 나와 있습니다.
- 퍼블릭 Amazon ECR 리포지토리 및 이미지에 대해서는 [Amazon ECR 퍼블릭 리포지토리를 참조하세요. 그리고 공개 이미지 Amazon ECR 공개 사용 설명서](#)에 나와 있습니다.

자신만의 이미지를 생성한 후에는 맞춤형 솔루션을 구축하기 전에 다음 환경 변수를 선언할 수 있습니다.

```
#!/bin/bash
```

내보내기 PUBLIC_ECR_REGISTRY=YOUR_ECR_REGISTRY_URI # 예: YOUR_ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/
YOUR_IMAGE_NAME

내보내기 PUBLIC_ECR_TAG=YOUR_ECR_TAG # 예: 최신, v2.0.0

다음 예에서는 컨테이너 파일을 보여줍니다.

```
블레이즈미터/토라스에서
# 황소자리에는 Python과 pip가 포함됩니다.
RUN /usr/bin/python3 -m pip install --upgrade pip
RUN pip install --no-cache-dir awscli
RUN apt-get -y install xmlstarlet bc procs
# 황소자리 작업 디렉터리 = /bzt-configs
추가 ./load-test.sh /bzt-configs/
./*.jar /bzt-configs/ 추가
./*.py /bzt-configs/ 추가
chmod 755 /bzt-configs/load-test.sh 실행
chmod 755 /bzt-configs/ecslister.py 실행
chmod 755 /bzt-configs/ecscontroller.py 실행
진입점 ["/load-test.sh"]
```

컨테이너 파일 외에도 디렉터리에는 다음을 다운로드하는 bash 스크립트가 포함되어 있습니다.
Taurus 프로그램을 실행하기 전에 Amazon S3에서 구성을 테스트하십시오.

```
#!/bin/bash

# S3의 결과 xml 파일 이름에 대한 uuid를 설정합니다.
UUID=$(cat /proc/sys/kernel/random/uuid)
피피드=0
echo "S3_BUCKET:: ${S3_BUCKET}"
echo "TEST_ID:: ${TEST_ID}"
echo "TEST_TYPE:: ${TEST_TYPE}"
echo "FILE_TYPE:: ${FILE_TYPE}"
echo "프리픽스:: ${PREFIX}"
echo "UUID:: ${UUID}"
echo "LIVE_DATA_ENABLED:: ${LIVE_DATA_ENABLED}"

sigterm_handler() { if [ $pypid
-ne 0 ]; 그 다음에
echo "컨테이너가 SIGTERM을 받았습니다."
kill -15 $pypid $pypid 대기

143번 출구 #128 + 15
fi
```

```

}

트랩 'sigterm_handler' SIGTERM

echo "테스트 시나리오 다운로드"
AWS s3 cp s3://$S3_BUCKET/test-scenarios/$TEST_ID-$AWS_REGION.json test.json

# JMeter jmx 파일 다운로드
if [ "$TEST_TYPE" != "간단" ]; then # *.jar를 JMeter 라이
브러리 경로에 복사합니다. Taurus JMeter 경로를 참조하세요: https://
gettaurus.org/docs/JMeter/
JMETER_LIB_PATH=`find ~/.bzt/jmeter-aurus -type d -name "lib"` echo "cp $PWD/*.jar
$JMETER_LIB_PATH" cp $PWD/*.jar $JMETER_LIB_PATH

if [ "$FILE_TYPE" != "zip" ]; 그런 다음 aws s3 cp s3://$S3_BUCKET/
public/test-scenarios/$TEST_TYPE/$TEST_ID.jmx ./ else aws s3 cp s3://$S3_BUCKET/public/test-scenarios/$TEST_TYPE/$TEST_ID.zip ./ unzip

$TEST_ID.zip # 첫 번째 jmx 파일만 찾습니다.

JMETER_SCRIPT=`찾기 . -이름 "*.jmx" | head -n 1` if [ -z "$JMETER_SCRIPT" ]; then echo "zip 파일에 JMeter
스크립트가 없습니다." 1번 출구

fi

sed -i -e "s|$TEST_ID.jmx|$JMETER_SCRIPT|g" test.json

# 번들 플러그인 jar를 jmeter 확장 폴더에 복사하여 사용할 수 있도록 합니다.
jmeter
BUNDLED_PLUGIN_DIR=`$PWD -type d -name "플러그인" 찾기 | head -n 1` # 업로드에 /plugins 폴더가 있는 경우에만 복사를 시도합니다. if [ -z
"$BUNDLED_PLUGIN_DIR" ]; then echo "플러그인 설치 건너뛰는 중(업로드 시 /plugins 폴더 없음)" else

# jmeter 확장 폴더가 있는지 확인 JMETER_EXT_PATH=`find ~/.bzt/jmeter-aurus
-type d -name "ext"` if [ -z "$JMETER_EXT_PATH" ]; then # 빠르게 실패 - 플러그인이 번들로 제공되면 테스트에 필요할
것입니다. echo "jmeter 확장 경로(~/.bzt/jmeter-aurus/**/ext)를 찾
을 수 없습니다 - 찾을 수 없습니다.

번들 플러그인 설치" 출구 1

fi cp -v
$BUNDLED_PLUGIN_DIR/*.jar $JMETER_EXT_PATH

```

```

fi
fi
fi

#파이썬 스크립트 다운로드
if [ -z "$IPNETWORK" ]; 그런 다음 python3
-u $SCRIPT $TIMEOUT & pypid=$! $pypid
pypid=0 기다리세요
요

그렇지
않으면 python3 -u $SCRIPT $IPNETWORK $IPHOSTS
fi

echo "테스트 실행 중"
stdbuf -i0 -o0 -e0 bzt test.json -o module.console.disable=true | stdbuf -i0 -o0 -e0 tee -a result.tmp | sed -u -e "s/^|$TEST_ID
$LIVE_DATA_ENABLED|"
CALCULATED_DURATION=`cat 결과.tmp | grep -m1 "테스트 기간" | awk -F $5 }' | awk -F ':' '{ 인쇄 ($1 * 3600) + ($2
* 60) + $3 }'`

# 사용자 정의 결과가 있는 경우 S3에 업로드
# 모든 파일은 결과를 올바르게 구별하기 위해 $TEST_ID/$PREFIX/$UUID 아래에 위치합니다.
if [ "$TEST_TYPE" != "간단" ]; 그런 다음 if [ "$FILE_TYPE" != "zip" ]; 그런 다
음 $TEST_ID.jmx 고양이 | grep 파일 이름 > results.txt else cat
$JMeter_SCRIPT | grep 파일 이름 > results.txt fi

sed -i -e 's/<stringProp name="filename">///g' results.txt sed -i -e 's/<\/stringProp>///g'
results.txt sed -i -e 's / ///g' 결과.txt

echo "결과로 업로드할 파일" cat results.txt

"${files[@]}"의 f에 대한 files=(`cat results.txt`); do
p="s3://$S3_BUCKET/results/$TEST_ID/JMeter_Result/
$PREFIX/$UUID/$f" if [[ $f = /* ]]; 그런 다음 p="s3://$S3_BUCKET/results/$TEST_ID/JMeter_Result/$PREFIX/$UUID/$f" fi

echo "$p 업로드 중" aws s3 cp $f $p

```

완료

fi

if [-f /tmp/artifacts/results.xml]; 그런 다음 "테스트 기간 확인 중"을 에
코합니다.

TEST_DURATION=`xmlstarlet sel -t -v "/FinalStatus/TestDuration" /tmp/artifacts/
결과.xml`

if ((\$(echo "\$TEST_DURATION > \$CALCULATED_DURATION" | bc -l))); 그런 다음 echo "테스트 기간 업데이트 중:
\$CALCULATED_DURATION s" xmlstarlet ed -L -u /FinalStatus/TestDuration -v
\$CALCULATED_DURATION /tmp/
유물/results.xml 파일

echo "결과, bzt 로그, JMeter 로그, 출력 및 오류 파일 업로드 중" aws s3 cp /tmp/artifacts/results.xml s3://\$S3_BUCKET/
results/\${TEST_ID}/\${PREFIX}-
\${UUID}-\${AWS_REGION}.xml aws s3
cp /tmp/artifacts/bzt.log s3://\$S3_BUCKET/results/\${TEST_ID}/bzt-\${PREFIX}-
\${UUID}-\${AWS_REGION}.log aws s3 cp /
tmp/artifacts/jmeter.log s3://\$S3_BUCKET/results/\${TEST_ID}/jmeter-
\${PREFIX}-\${UUID}-\${AWS_REGION}.log aws s3 cp /tmp/
artifacts/jmeter.out s3://\$S3_BUCKET/results/\${TEST_ID}/jmeter-
\${PREFIX}-\${UUID}-\${AWS_REGION}.out aws s3 cp /tmp/
artifacts/jmeter.err s3://\$S3_BUCKET/results/\${TEST_ID}/jmeter-
\${PREFIX}-\${UUID}-\${AWS_REGION}.err
또 다른
echo "테스트를 실행하는 동안 오류가 발생했습니다."
fi

[Dockerfile](#) 외에도 [bash 스크립트](#)에는 두 개의 Python 스크립트도 포함되어 있습니다.

예배 규칙서. 각 작업은 bash 스크립트 내에서 Python 스크립트를 실행합니다. 작업자 작업이 실행됩니다.

ecslister.py 스크립트, 리더 작업은 ecscontroller.py 스크립트를 실행합니다.

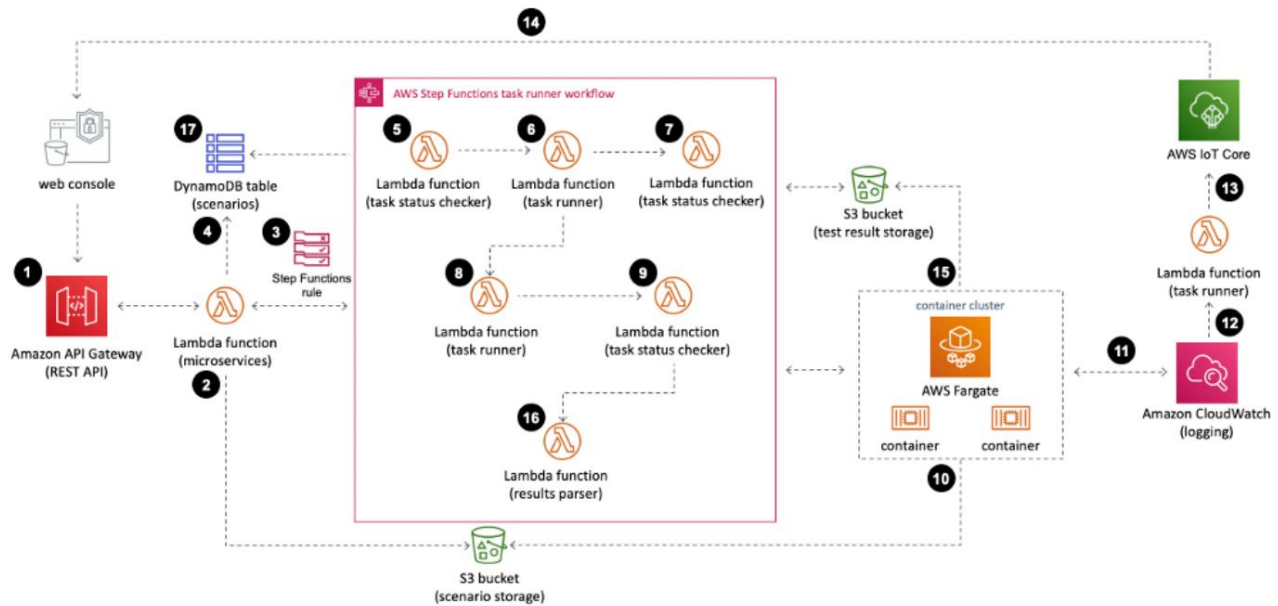
ecslister.py 스크립트는 포트 50000에 소켓을 생성하고 메시지를 기다립니다. 그만큼

ecscontroller.py 스크립트는 소켓에 연결하고 시작 테스트 메시지를 작업자에게 보냅니다.

동시에 시작할 수 있는 작업입니다.

테스트 작업흐름

다음 세부 분석에서는 테스트 시나리오 실행과 관련된 단계를 보여줍니다.



테스트 작업흐름

1. 웹 콘솔을 사용하여 구성 세부 정보가 포함된 테스트 시나리오를 제출합니다.
솔루션의 API.
2. 테스트 시나리오 구성은 Amazon Simple Storage Service(Amazon S3)에 JSON 파일 (s3://<bucket-name>/test-scenarios/<\$TEST_ID>/<\$TEST_ID>.json)로 업로드됩니다.
3. AWS Step Functions 상태 시스템은 테스트 ID, 작업 수, 테스트 유형 및 파일 유형을 AWS Step Functions 상태 시스템 입력으로 사용하여 실행됩니다. 테스트가 예약되면 먼저 지정된 날짜에 AWS Step Functions를 트리거하는 CloudWatch 이벤트 규칙을 생성합니다. 예약 워크플로에 대한 자세한 내용은 이 가이드의 [테스트 예약 워크플로](#) 섹션을 참조하세요.
4. 구성 세부 정보는 시나리오 Amazon DynamoDB 테이블에 저장됩니다.
5. AWS Step Functions 작업 실행기 워크플로에서 task-status-checker AWS Lambda 함수는 Amazon Elastic Container Service(Amazon ECS) 작업이 동일한 테스트 ID에 대해 이미 실행 중인지 확인합니다. 동일한 테스트 ID를 가진 작업이 실행 중이면 오류가 발생합니다. AWS Fargate 클러스터에서 실행 중인 Amazon ECS 작업이 없는 경우 함수는 테스트 ID, 작업 수 및 테스트 유형을 반환합니다.
6. 작업 실행자 AWS Lambda 함수는 이전 단계에서 작업 세부 정보를 가져오고
AWS Fargate 클러스터에서 Amazon ECS 작업자 작업을 실행합니다. Amazon ECS API는 다음을 사용합니다.

작업자 작업을 실행하는 RunTask 작업입니다. 이러한 작업자 작업은 시작된 다음

테스트를 시작하려면 리더 태스크에서 메시지를 시작하세요. RunTask 작업은 정의당 10개의 작업으로 제한됩니다. 작업 수가 10개를 초과하면 모든 작업자 작업이 시작될 때까지 작업 정의가 여러 번 실행됩니다. 또한 이 함수는 결과 구문 분석기 AWS Lambda 함수에서 현재 테스트를 구별하기 위해 접두사를 생성합니다.

7. task-status-checker AWS Lambda 함수는 모든 Amazon ECS 작업자 작업이 완료되었는지 확인합니다.

동일한 테스트 ID로 실행 중입니다. 작업이 아직 프로비저닝 중인 경우 1분 동안 기다렸다가 다시 확인합니다. 모든 Amazon ECS 작업이 실행되면 테스트 ID, 작업 수, 테스트 유형, 모든 작업 ID 및 접두사를 반환하고 이를 작업 실행기 함수에 전달합니다.

8. 작업 실행자 AWS Lambda 함수가 다시 실행되며, 이번에는 리더 노드 역할을 하는 단일 Amazon ECS 작업을 시작합니다. 이 ECS 작업은 테스트를 동시에 시작하기 위해 각 작업자 작업에 테스트 시작 메시지를 보냅니다.

9. task-status-checker AWS Lambda 함수는 Amazon ECS 작업이 다음과 같은지 다시 확인합니다.

동일한 테스트 ID로 실행 중입니다. 작업이 계속 실행 중인 경우 1분 동안 기다렸다가 다시 확인합니다. 실행 중인 Amazon ECS 작업이 없으면 테스트 ID, 작업 수, 테스트 유형 및 접두사를 반환합니다.

10. 작업 실행자 AWS Lambda 함수가 AWS Fargate 클러스터에서 Amazon ECS 작업을 실행할 때 각 작업은 Amazon S3에서 테스트 구성을 다운로드하고 테스트를 시작합니다.

11. 테스트가 실행되면 각 작업에 대한 평균 응답 시간, 동시 사용자 수, 성공한 요청 수 및 실패한 요청 수가 Amazon CloudWatch에 기록되고 CloudWatch 대시보드에서 볼 수 있습니다.

12. 테스트에 라이브 데이터를 포함한 경우 솔루션은 다음을 사용하여 CloudWatch에서 실시간 테스트 결과를 필터링합니다. 구독 필터. 그런 다음 솔루션은 데이터를 Lambda 함수에 전달합니다.

13. 그런 다음 Lambda 함수는 수신된 데이터를 구조화하여 AWS IoT Core에 게시합니다. 주제.

14. 웹 콘솔은 테스트를 위해 AWS IoT Core 주제를 구독하고 게시된 데이터를 수신합니다. 테스트가 실행되는 동안 실시간 데이터를 그래프로 표시하려면 주제로 이동하세요.

15. 테스트가 완료되면 컨테이너 이미지는 세부 보고서를 XML 파일로 Amazon S3에 내보냅니다. 각 파일에는 파일 이름에 대한 UUID가 제공됩니다. 예를 들어 s3://dlte-bucket/test-scenes/<\$TEST_ID>/results/<\$UUID>.json입니다.

16. XML 파일이 Amazon S3에 업로드되면 결과 파서 AWS Lambda 함수는 접두사로 시작하는 XML 파일의 결과를 읽고 모든 결과를 구문 분석하고 집계하여 하나의 요약 결과로 만듭니다.

17.결과 파서 AWS Lambda 함수는 집계 결과를 Amazon에 기록합니다.
DynamoDB 테이블.

시험 결과

AWS의 분산 로드 테스트는 로드 테스트 프레임워크를 활용하여 대규모로 애플리케이션 테스트를 실행합니다. 테스트가 완료되면 다음 결과가 포함된 세부 보고서가 생성됩니다.

- 평균 응답 시간 - 생성된 모든 요청에 대한 평균 응답 시간(초)입니다.
테스트로.
- 평균 대기 시간 - 테스트에서 생성된 모든 요청에 대한 평균 대기 시간(초)입니다.
- 평균 연결 시간 - 모든 호스트에 연결하는 데 걸리는 평균 시간(초)입니다.
테스트에 의해 생성된 요청입니다.
- 평균 대역폭 - 테스트에서 생성된 모든 요청에 대한 평균 대역폭입니다.
- 총 개수 - 총 요청 개수입니다. • 성공 횟수 - 성공한 요청의 총 횟수
입니다.
- 오류 개수 - 총 오류 개수입니다.
- 초당 요청 - 생성된 모든 요청에 대한 초당 평균 요청입니다.
시험.
- 백분위수 - 테스트에 대한 응답 시간의 백분위수입니다. 최대 응답 시간은 100%입니다. 최소 응답 시간은 0%입니다.

메모

테스트 결과가 콘솔에 표시됩니다. 시나리오 Amazon S3 버킷의 결과 폴더에서 원시 테스트 결과에 대한 XML 파일을 볼 수 있습니다.

항소자리 테스트 결과에 대한 자세한 내용은 [테스트 보고서 생성을 참조하세요.](#) [항소자리 사용자 매뉴얼](#)에서 .

분산 부하 테스트 API

이 부하 테스트 솔루션은 테스트 결과 데이터를 안전한 방식으로 노출하는 데 도움이 됩니다. API는 Amazon DynamoDB에 저장된 테스트 데이터에 액세스하기 위한 "현관" 역할을 합니다. 또한 API를 사용하여 솔루션에 구축한 확장 기능에 액세스할 수도 있습니다.

이 솔루션은 식별 및 승인을 위해 Amazon API Gateway와 통합된 Amazon Cognito 사용자 풀을 사용합니다. API와 함께 사용자 풀을 사용하는 경우 클라이언트는 유효한 자격 증명 토큰을 제공한 후에만 사용자 풀 활성화 메서드를 호출할 수 있습니다.

API를 통해 직접 테스트를 실행하는 방법에 대한 자세한 내용은 [요청 서명을](#) 참조하세요. Amazon [API Gateway REST API](#) 참조 문서에 있습니다.

솔루션의 API에서 다음 작업을 사용할 수 있습니다.

메모

testScenario 및 기타 매개변수에 대한 자세한 내용은 [시나리오](#)를 참조하세요. 및 [페이로드 예시](#) [GitHub 저장소에](#) 있습니다.

시나리오

- [GET /시나리오](#)
- [POST/시나리오](#)
- [옵션/시나리오](#)
- [GET /scenarios/{testId}](#)
- [POST /scenarios/{testId}](#)
- [/scenarios/{testId} 삭제](#)
- [옵션 /scenarios/{testId}](#)

작업

- [GET /작업](#)
- [OPTIONS /작업](#)

지역

- [GET /지역](#)
- [OPTIONS /지역](#)

GET /시나리오

설명

GET /scenarios 작업을 사용하면 테스트 시나리오 목록을 검색할 수 있습니다.

응답

이름	설명
데이터	ID, 이름, 각 테스트에 대한 설명, 상태 및 실행 시간

POST /시나리오

설명

POST /scenarios 작업을 사용하면 테스트 시나리오를 생성하거나 예약할 수 있습니다.

요청 본문

이름	설명
테스트 이름	테스트 이름
테스트 설명	테스트에 대한 설명
testTaskConfig	동시성(병렬 실행 횟수), taskCount(테스트를 실행하는 데 필요한 작업 수) 및 시나리오 지역

이름	설명
테스트시나리오	동시성, 테스트를 포함한 테스트 정의 테스트 시간, 호스트 및 방법
테스트 유형	테스트 유형(예: 단순, jmeter)
파일 유형	업로드 파일 형식(예: 없음, 스크립트, zip)
일정날짜	테스트를 실행할 날짜입니다. 경우에만 제공 테스트 예약(예: 2021-02-28)
일정 시간	테스트를 실행할 시간입니다. 경우에만 제공 테스트 예약(예: 21:07)
일정단계	일정 프로세스의 단계입니다. 오직 반복 테스트를 예약하는 경우 제공됩니다. (사용 가능한 단계에는 생성 및 시작이 포함됩니다)
회귀	예약된 테스트의 반복입니다. 오직 반복 테스트를 예약하는 경우 제공됩니다(예를 들어 매일, 매주, 격주 또는 월간 간행물)

응답

이름	설명
테스트 ID	테스트의 고유 ID
테스트 이름	테스트 이름
상태	테스트 상태

옵션/시나리오

설명

OPTIONS /scenarios 작업은 올바른 CORS가 포함된 요청에 대한 응답을 제공합니다.

응답 헤더.

응답

이름	설명
테스트 ID	테스트의 고유 ID
테스트 이름	테스트 이름
상태	테스트 상태

GET /scenarios/{testId}

설명

GET /scenarios/{testId} 작업을 사용하면 특정 테스트의 세부 정보를 검색할 수 있습니다.

대본.

요청 매개변수

테스트 ID

테스트의 고유 ID

유형: 문자열

필수: 예

응답

이름	설명
테스트 ID	테스트의 고유 ID
테스트 이름	테스트 이름
테스트 설명	테스트에 대한 설명
테스트 유형	실행되는 테스트 유형(예: 단순, jmeter)
파일 유형	업로드된 파일 유형(예: 없음, 스크립트, zip)
상태	테스트 상태
시작 시간	마지막 테스트가 시작된 시간과 날짜
종료 시간	마지막 테스트가 종료된 시간과 날짜
테스트 시나리오	동시성, 테스트를 포함한 테스트 정의 테스트 시간, 호스트 및 방법
작업 개수	테스트를 실행하는 데 필요한 작업 수
taskId	테스트 실행을 위한 작업 ID 목록
결과	테스트의 최종 결과
역사	과거 테스트의 최종 결과 목록
오류 이유	오류가 발생하면 생성되는 오류 메시지 발생하다
다음 실행	다음 예약된 실행(예: 2017-04-22 17:18:00)

이름	설명
일정반복	테스트의 재발(예: 매일, 매주, 격주, 매월)

POST /scenarios/{testId}

설명

POST /scenarios/{testId} 작업을 사용하면 특정 테스트 시나리오를 취소할 수 있습니다.

요청 매개변수

테스트 ID

테스트의 고유 ID

유형: 문자열

필수: 예

응답

이름	설명
상태	테스트 상태

/scenarios/{testId} 삭제

설명

DELETE /scenarios/{testId} 작업을 사용하면 특정 항목과 관련된 모든 데이터를 삭제할 수 있습니다.
테스트 시나리오.

요청 매개변수

테스트 ID

테스트의 고유 ID

유형: 문자열

필수: 예

응답

이름	설명
상태	테스트 상태

옵션 /scenarios/{testId}

설명

OPTIONS /scenarios/{testId} 작업은 다음과 같은 요청에 대한 응답을 제공합니다.
CORS 응답 헤더를 수정하세요.

응답

이름	설명
테스트 ID	테스트의 고유 ID
테스트 이름	테스트 이름
테스트설명	테스트에 대한 설명
테스트 유형	실행되는 테스트 유형(예: 단순, jmeter)
파일 유형	업로드된 파일 유형(예: 없음, 스크립트, zip)

이름	설명
상태	테스트 상태
시작 시간	마지막 테스트가 시작된 시간과 날짜
종료시간	마지막 테스트가 종료된 시간과 날짜
테스트시나리오	동시성, 테스트를 포함한 테스트 정의 테스트 시간, 호스트 및 방법
작업 개수	테스트를 실행하는 데 필요한 작업 수
taskId	테스트 실행을 위한 작업 ID 목록
결과	테스트의 최종 결과
역사	과거 테스트의 최종 결과 목록
오류 이유	오류가 발생하면 생성되는 오류 메시지 발생하다

GET /작업

설명

GET /tasks 작업을 사용하면 실행 중인 Amazon Elastic Container 목록을 검색할 수 있습니다.
서비스(Amazon ECS) 작업.

응답

이름	설명
작업	테스트 실행을 위한 작업 ID 목록

옵션 /작업

설명

OPTIONS /tasks 작업 작업은 올바른 CORS가 포함된 요청에 대한 응답을 제공합니다.

응답 헤더.

응답

이름	설명
taskId	테스트 실행을 위한 작업 ID 목록

GET /지역

설명

GET /regions 작업을 사용하면 필요한 지역 리소스 정보를 검색할 수 있습니다.

해당 지역에서 테스트를 실행합니다.

응답

이름	설명
테스트 ID	지역 ID
ecsCloudWatchLog그룹	Amazon CloudWatch 로그의 이름 Amazon Fargate 작업을 위한 그룹 지역
지역	테이블의 리소스가 있는 리전 존재하다
서브넷A	리전에 있는 서브넷 중 하나의 ID
서브넷B	리전에 있는 서브넷 중 하나의 ID

이름	설명
taskCluster	AWS Fargate 클러스터의 이름은 다음과 같습니다. 지역
작업 정의	리전에 있는 작업 정의의 ARN
작업이미지	리전의 작업 이미지 이름
태스크보안그룹	리전의 보안 그룹 ID

옵션 /지역

설명

OPTIONS /regions 작업은 올바른 CORS가 포함된 요청에 대한 응답을 제공합니다.

응답 헤더.

응답

이름	설명
테스트 ID	지역 ID
ecsCloudWatchLog그룹	Amazon CloudWatch 로그의 이름 Amazon Fargate 작업을 위한 그룹 지역
지역	테이블의 리소스가 있는 리전 존재하다
서브넷A	리전에 있는 서브넷 중 하나의 ID
서브넷B	리전에 있는 서브넷 중 하나의 ID
taskCluster	AWS Fargate 클러스터의 이름은 다음과 같습니다. 지역

이름	설명
작업 정의	리전에 있는 작업 정의의 ARN
작업이미지	리전의 작업 이미지 이름
태스크보안그룹	리전의 보안 그룹 ID

테스트 예약 작업 흐름

웹 콘솔을 사용하여 부하 테스트를 예약합니다. 테스트를 예약할 때 다음 워크플로가 실행됩니다.

- 예약 옵션을 사용하여 부하 테스트를 생성하면 일정 매개 변수가 Amazon API Gateway를 통한 솔루션의 API.
- 그런 다음 API는 CloudWatch 이벤트를 생성하는 Lambda 함수에 매개변수를 전달합니다. 지정된 날짜에 실행되도록 예약된 규칙입니다.
- 테스트가 일회성 테스트인 경우 CloudWatch 이벤트 규칙은 지정된 날짜에 실행됩니다. API는- 서비스 Lambda 함수는 이 가이드의 [테스트 워크플로](#) 섹션 에 지정된 워크플로를 통해 새로운 테스트를 실행합니다 .
- 테스트가 반복 테스트인 경우 CloudWatch 이벤트 규칙은 지정된 날짜에 활성화됩니다. 그만큼 api-services Lambda 함수가 실행되어 현재 CloudWatch 이벤트 규칙을 삭제하고 생성 시 즉시 실행되고 이후에는 지정된 반복 빈도에 따라 반복적으로 실행되는 또 다른 규칙을 생성합니다.

사용자 수 결정

컨테이너가 테스트를 위해 지원할 수 있는 사용자 수는 사용자 수를 점진적으로 늘리고 Amazon CloudWatch에서 성능을 모니터링하여 결정할 수 있습니다. CPU 및 메모리 성능이 한계에 도달한 것을 관찰하면 컨테이너가 기본 구성(vCPU 2개 및 메모리 4GB)에서 해당 테스트를 지원할 수 있는 최대 사용자 수에 도달한 것입니다. 다음 예제를 사용하여 테스트에 대한 동시 사용자 제한 결정을 시작할 수 있습니다.

1. 200명 이하의 사용자로 테스트를 만듭니다.
2. 테스트가 실행되는 동안 CloudWatch 콘솔을 사용하여 CPU와 메모리를 모니터링합니다.
 - ㄱ. 왼쪽 탐색 창의 Container Insights 에서 성능 모니터링을 선택합니다.
 - 비. 성능 모니터링 페이지 의 왼쪽 드롭다운 메뉴에서 ECS 클러스터를 선택합니다.
 - 씨. 오른쪽 드롭다운 메뉴에서 Amazon Elastic Container Service(Amazon ECS)를 선택합니다.
 - 무리.
3. 모니터링하면서 CPU와 메모리를 관찰해 보세요. CPU가 75%를 초과하지 않거나 메모리가 85%를 초과하지 않는 경우(일회성 최대치 무시) 더 많은 수의 사용자를 대상으로 또 다른 테스트를 실행할 수 있습니다.

테스트가 리소스 제한을 초과하지 않은 경우 1~3단계를 반복합니다. 선택적으로 더 많은 수의 동시 사용자를 허용하기 위해 컨테이너 리소스를 늘릴 수 있습니다. 그러나 이로 인해 비용이 더 높아집니다. 자세한 내용은 이 가이드의 [컨테이너 리소스 늘리기](#) 섹션을 참조하세요.

메모

정확한 결과를 얻으려면 동시 사용자 제한을 결정할 때 한 번에 하나의 테스트만 실행하십시오. 모든 테스트는 동일한 클러스터를 사용하며 CloudWatch 컨테이너 통찰력은 클러스터를 기반으로 성능 데이터를 집계합니다. 이로 인해 두 테스트가 동시에 CloudWatch 컨테이너 통찰력에 보고되어 단일 테스트에 대한 리소스 사용률 지표가 부정확해집니다.

엔진별 사용자 보정에 대한 자세한 내용은 [황소자리 테스트 보정](#)을 참조하세요. BlazeMeter 문서에서.

컨테이너 리소스 늘리기

현재 지원되는 사용자 수를 늘리려면 컨테이너 리소스를 늘리세요. 이를 통해 CPU와 메모리를 늘려 동시 사용자 증가를 처리할 수 있습니다.

새 작업 정의 개정 생성

1. [Amazon Elastic Container Service 콘솔](#)에 로그인합니다.
2. 왼쪽 탐색 메뉴에서 작업 정의를 선택합니다.
3. 이 솔루션에 해당하는 작업 정의 옆의 확인란을 선택합니다. 예를 들어,
 <스택 이름>-EcsTaskDefinition-<시스템 생성-랜덤-해시>.
4. 새 개정 생성을 선택합니다.
5. 새 개정 생성 페이지에서 다음 작업을 수행합니다.
 - ㅏ. 작업 크기에서 작업 메모리 및 작업 CPU를 수정합니다.
 - 비. 컨테이너 정의에서 하드/소프트 메모리 제한을 검토합니다. 이 한도가 원하는 메모리보다 낮으면 컨테이너를 선택하세요.
 - 씨. 컨테이너 편집 대화 상자에서 메모리 제한으로 이동하여 하드 제한을 업데이트합니다.
 원하는 기억.
 - 디. 업데이트를 선택합니다.
6. 새 개정 생성 페이지에서 생성을 선택합니다.
7. 작업 정의가 성공적으로 생성된 후 새 작업 정의의 이름을 기록합니다. 이 이름에는 버전 번호가 포함됩니다(예: <stackName>-EcsTaskDefinition-<system-generated-random-Hash>:<system-generated-versionNumber>).

DynamoDB 테이블 업데이트

1. [DynamoDB 콘솔](#)로 이동합니다.
2. 왼쪽 탐색 창의 테이블 아래 항목 탐색을 선택합니다.
3. 이 솔루션과 연결된 시나리오 테이블 DynamoDB 테이블을 선택합니다. 예를 들어 <stackName>-DLTTestRunnerStorageDLTScenariosTable-<system-generated-random-Hash>입니다.

4. 작업 정의를 수정한 지역에 해당하는 항목을 선택합니다.
예를 들어 **지역-<지역-이름>**입니다.
5. taskDefinition 속성을 새 작업 정의로 업데이트합니다.

실시간 데이터

테스트를 실행할 때 선택적으로 실시간 데이터를 포함하여 현재 상황에 대한 실시간 통찰력을 얻을 수 있습니다. Fargate 작업에 대한 CloudWatch 로그 그룹에는 라이브 데이터 옵션이 포함된 테스트 결과에 대한 구독 필터가 포함되어 있습니다. 솔루션이 패턴을 찾으려면 데이터를 구조화하고 AWS IoT Core 주제에 게시하는 Lambda 함수를 시작합니다. 웹 콘솔은 주제를 구독하고, 들어오는 데이터를 수신하며, 1초 간격으로 집계된 데이터를 그래프로 표시합니다. 웹 콘솔에는 평균 응답 시간, 가상 사용자, 성공 및 실패의 네 가지 그래프가 포함되어 있습니다.

메모

데이터는 일시적이며 테스트가 실행되는 동안 무슨 일이 일어나고 있는지 확인하는 데만 사용됩니다. 테스트가 완료되면 솔루션은 결과 데이터를 DynamoDB 및 Amazon S3에 저장합니다. 웹 콘솔은 최대 5,000개의 데이터 포인트를 유지하며 그 이후에는 가장 오래된 데이터가 최신 데이터로 대체됩니다. 페이지를 새로 고치면 그래프가 공백이 되고 사용 가능한 다음 데이터 포인트부터 시작됩니다.

테스트 취소 워크플로

웹 콘솔에서 부하 테스트를 취소하면 솔루션은 다음 테스트 취소를 실행합니다.
작업 흐름.

1. 취소 요청이 마이크로서비스 API로 전송됩니다.
2. 마이크로서비스 API는 다음이 될 때까지 작업을 취소하는 작업 취소자 Lambda 함수를 호출합니다.
현재 시작된 모든 작업이 중지됩니다.
3. 작업 실행자 Lambda 함수가 작업에 대한 초기 호출 이후에도 계속 실행되는 경우
canceler Lambda 함수를 사용하면 작업이 계속 시작됩니다. 일단 작업 실행자
Lambda 함수가 완료되면 AWS Step Functions는 작업 취소 Lambda 함수를 다시 실행하여 남은 작업을 중지하는 테스트 취소
단계를 계속 진행합니다.

문제 해결

이 섹션에서는 솔루션 배포 및 사용에 대한 문제 해결 지침을 제공합니다.

[알려진 문제 해결 방법](#)은 알려진 오류를 완화하기 위한 지침을 제공합니다. 이러한 지침으로 문제가 해결되지 않으면 [AWS 지원 문의](#) 섹션에서 이 솔루션에 대한 [AWS 지원 사례를 개설하는 방법](#)에 대한 지침을 참조하십시오.

알려진 문제

문제: 기존 VPC를 사용하고 있으며 테스트가 다음 상태로 실패합니다.

실패하여 다음 오류 메시지가 표시됩니다.

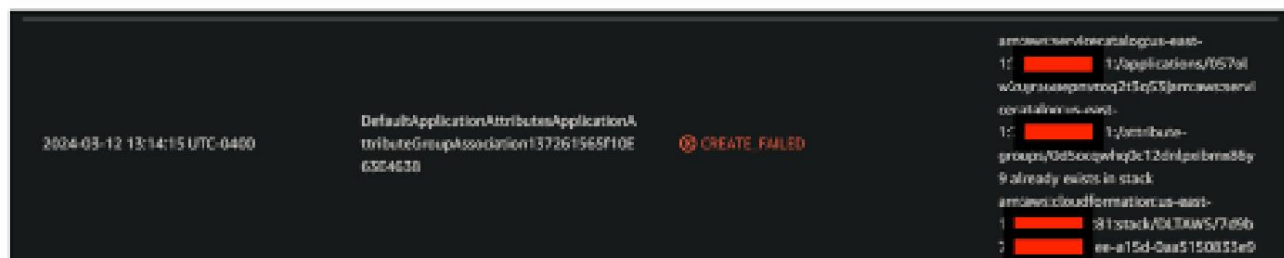
테스트 실행에 실패했을 수 있습니다.

해결 방법: 지정된 VPC에 서브넷이 존재하고 [인터넷 게이트웨이](#)를 통해 인터넷에 대한 경로가 있는지 확인하십시오. 또는 [NAT 게이트웨이](#). 테스트를 성공적으로 실행하려면 [AWS Fargate가 퍼블릭 리포지토리에서 컨테이너 이미지를 가져올 수 있는 액세스 권한이 필요합니다.](#)

문제: 테스트를 실행하는 데 너무 오랜 시간이 걸리거나 무기한 실행이 중단됩니다.

해결 방법: 테스트를 취소하고 AWS Fargate를 확인하여 모든 작업이 중지되었는지 확인하십시오. 중지되지 않은 경우 모든 Fargate 작업을 수동으로 중지하십시오. 원하는 작업 수를 시작할 수 있는지 확인하려면 계정의 주문형 Fargate 작업 제한을 확인하세요. Fargate 작업을 시작할 때 발생하는 오류에 대해 더 자세히 알아보려면 Lambda 작업 실행기 함수에 대한 CloudWatch 로그를 확인하세요. 실행 중인 Fargate 컨테이너에서 무슨 일이 일어나고 있는지에 대한 자세한 내용은 CloudWatch ECS 로그를 확인하세요.

v3.2.6 이전의 DLT 버전에서 최신 버전으로 업데이트할 때 스택 업데이트가 실패합니다.



해결 방법: 이는 v3.2.6 이전 버전에서 최신 버전으로 DLT를 업데이트할 때 알려진 문제입니다. 이 경우 아래 이미지와 같이 앱 레지스트리 애플리케이션 속성 그룹 연결 생성 실패로 인해 스택이 실패합니다. 제안된 솔루션에 따라 문제를 해결하고 스택을 계속 업데이트할 수 있습니다.

1. 다음 [링크](#)를 이용하세요 분산 부하 테스트-on-aws.템플릿을 다운로드하려면
2. 템플릿을 열고 조건:으로 이동하여 DLTCCommonResourcesAppRegistryCondition을 찾습니다.
3. 다음과 비슷한 내용이 표시되어야 합니다.

정황:

DLTCCommonResourcesAppRegistryConditionCCEF54F8:

Fn::같음:

- "진실"

- "진실"

4. 두 번째 참값을 false로 변경합니다.

정황:

DLTCCommonResourcesAppRegistryConditionCCEF54F8:

Fn::Equals: - "참"

- "거짓"

5. 사용자 지정된 템플릿을 사용하여 스택을 업데이트합니다.
6. 이 스택은 스택에서 앱 레지스트리 관련 리소스를 제거합니다. 따라서 업데이트는 다음과 같아야 합니다
완전한.
7. 최신 템플릿 URL을 사용하여 또 다른 스택 업데이트를 수행하여 앱 레지스트리 애플리케이션을 다시 추가합니다.
스택에 리소스를 추가합니다.

메모

AWS Systems Manager Application Manager는 이 솔루션과 해당 리소스에 대한 애플리케이션 수준 보기를 제공하므로 다음을 수행할 수 있습니다. 1. 중앙 위치에서 해당 리소스, 스택 및 AWS 계정 전체에 배포된 리소스 비용, 이 솔루션과 관련된 로그를 모니터링합니다. .

2. 배포 상태, CloudWatch 경보, 리소스 구성, 운영 문제 등 애플리케이션 컨텍스트에서 이 솔루션의 리소스에 대한 운영 데이터를 봅니다.

AWS 지원에 문의

[AWS 개발자 지원](#)이 있는 경우, [AWS 비즈니스 지원](#), 또는 [AWS 엔터프라이즈 지원](#), 지원 센터를 사용하여 이 솔루션에 대한 전문가의 도움을 받을 수 있습니다. 다음 섹션에서는 지침을 제공합니다.

케이스 만들기

1. [지원 센터](#)에 로그인합니다.
2. 사례 생성을 선택합니다.

어떻게 도와드릴까요?

1. 기술을 선택하세요
2. 서비스에서 솔루션을 선택합니다.
3. 카테고리에서 AWS의 분산 로드 테스트를 선택합니다.
4. 심각도에서 사용 사례에 가장 적합한 옵션을 선택합니다.
5. 서비스, 범주 및 심각도를 입력하면 인터페이스에 일반적인 문제 해결 질문에 대한 링크가 채워집니다. 이러한 링크로 질문을 해결할 수 없는 경우 다음 단계: 추가 정보를 선택하세요.

추가 정보

1. 제목에 질문이나 문제를 요약하는 텍스트를 입력합니다.
2. 설명에는 문제를 자세히 설명합니다.
3. 파일 첨부 선택합니다.
4. AWS Support가 요청을 처리하는 데 필요한 정보를 첨부합니다.

귀하의 사건을 더 빨리 해결하도록 도와주세요

1. 요청된 정보를 입력하세요.
2. 다음 단계: 지금 해결을 선택하거나 당사에 문의하세요.

지금 해결하거나 당사에 문의하세요

1. 지금 해결 솔루션을 검토하세요.
2. 이러한 해결 방법으로 문제를 해결할 수 없는 경우 문의처를 선택하고 요청된 정보를 입력하세요.
정보를 입력하고 제출을 선택합니다.

솔루션 제거

AWS Management Console에서 또는 AWS 명령줄 인터페이스를 사용하여 AWS 분산 부하 테스트 솔루션을 제거할 수 있습니다. 이 솔루션으로 생성된 콘솔, 시나리오 및 로깅 Amazon Simple Storage Service(Amazon S3) 버킷을 수동으로 삭제해야 합니다. AWS 솔루션 구현에서는 보관할 데이터를 저장한 경우 이를 자동으로 삭제하지 않습니다.

메모

지역 스택을 배포한 경우 기본 스택을 삭제하기 전에 해당 지역의 스택을 삭제해야 합니다.

AWS 관리 콘솔 사용

1. AWS CloudFormation 콘솔에 로그인합니다.
2. 스택 페이지에서 이 솔루션의 설치 스택을 선택합니다.
3. 삭제를 선택합니다.

AWS 명령줄 인터페이스 사용

사용자 환경에서 AWS 명령줄 인터페이스(AWS CLI)를 사용할 수 있는지 확인합니다.

설치 지침은 [AWS 명령줄 인터페이스란 무엇입니까?](#) 를 참조하십시오. [AWS CLI 사용 설명서](#). AWS CLI를 사용할 수 있는지 확인한 후 다음 명령을 실행합니다.

```
$ aws cloudformation delete-stack --stack-name <설치-스택-이름>
```

Amazon S3 버킷 삭제

이 솔루션은 우발적인 데이터 손실을 방지하기 위해 AWS CloudFormation 스택을 삭제하기로 결정한 경우 솔루션에서 생성된 Amazon S3 버킷(선택 지역에 배포용)을 유지하도록 구성됩니다. 솔루션을 제거한 후 데이터를 보관할 필요가 없으면 이 S3 버킷을 수동으로 삭제할 수 있습니다. Amazon S3 버킷을 삭제하려면 다음 단계를 따르세요.

1. [Amazon S3 콘솔](#)에 로그인합니다.
2. 왼쪽 탐색 창에서 버킷을 선택합니다.
3. 이름으로 버킷 찾기 필드에 이 솔루션의 스택 이름을 입력합니다.
4. 솔루션의 S3 버킷 중 하나를 선택하고 비어 있음을 선택합니다.
5. 확인 필드에 영구 삭제를 입력하고 비어 있음을 선택합니다.
6. 방금 비운 S3 버킷 이름을 선택하고 삭제를 선택합니다.
7. 확인 필드에 S3 버킷 이름을 입력하고 버킷 삭제를 선택합니다.

S3 버킷을 모두 삭제할 때까지 3~7단계를 반복합니다.

AWS CLI를 사용하여 S3 버킷을 삭제하려면 다음 명령을 실행합니다.

```
$ aws s3 rb s3://<버킷 이름> --force
```


익명화된 데이터 수집

이 솔루션에는 익명화된 운영 지표를 AWS로 보내는 옵션이 포함되어 있습니다. 우리는 고객이 이 솔루션과 관련 서비스 및 제품을 사용하는 방법을 더 잘 이해하기 위해 이 데이터를 사용합니다. 활성화되면 다음 정보가 수집되어 AWS로 전송됩니다.

- 솔루션 ID - AWS 솔루션 식별자
- 고유 ID(UUID) - 각 솔루션 배포에 대해 무작위로 생성된 고유 식별자
- 타임스탬프 - 데이터 수집 타임스탬프
- 테스트 유형 - 실행되는 테스트 유형입니다.
- 파일 유형 - 업로드되는 파일 유형
- 작업 수 - 솔루션의 API를 통해 제출된 각 테스트의 작업 수
- 작업 기간 - 테스트를 실행하는 데 필요한 모든 작업의 총 실행 시간
- 테스트 결과 - 실행된 테스트의 결과

AWS는 이 설문조사를 통해 수집된 데이터를 소유합니다. 데이터 수집에는 AWS 개인정보 보호정책이 적용됩니다.

이 기능을 옵트아웃하려면 AWS CloudFormation 템플릿을 시작하기 전에 다음 단계를 완료하십시오.

1. [AWS CloudFormation 템플릿 다운로드 로컬 하드 드라이브에](#).
2. 텍스트 편집기를 사용하여 AWS CloudFormation 템플릿을 엽니다.
3. 다음에서 AWS CloudFormation 템플릿 매핑 섹션을 수정합니다.

해결책:

구성:

SendAnonymousData: "예"

에게:

해결책:

구성:

SendAnonymousData: "아니요"

4. [AWS CloudFormation 콘솔에 로그인합니다](#).

5. 스택 생성을 선택합니다.

6. 스택 생성 페이지의 템플릿 지정 섹션에서 템플릿 파일 업로드를 선택합니다.
7. 템플릿 파일 업로드에서 파일 선택을 선택하고 편집된 템플릿을 선택합니다.
로컬 드라이브.
8. 다음을 선택하고 자동화된 배포 섹션의 [스택 시작](#) 단계를 따릅니다.
이 가이드.

소스 코드

[GitHub 저장소를 방문하세요](#) 이 솔루션에 대한 템플릿과 스크립트를 다운로드하고 사용자 정의 내용을 다른 사람들과 공유하십시오.

기여자

다음 개인이 이 문서에 기여했습니다.

- 톰 나이팅게일
- 페르난도 덩글러
- 이범석
- 조지 렌츠
- 에린 맥길
- 디미트리 로페즈
- 카미아르 지아바리

개정

날짜	변화
2019년 11월	초판
2020년 9월	릴리스 버전 1.1.0: Amazon SQS를 AWS Step Functions로 대체하고 변경된 AWS 서비스를 자세히 설명하기 위해 아키텍처 다이어그램과 구성 요소 정보를 업데이트했습니다. JMeter 스크립트에 대한 자원이 추가되었습니다. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2020년 12월	릴리스 버전 1.2.0: AWS에 Amazon ECR 검사기가 추가되었습니다. 단계 기능; JMeter용 zip 파일 업로드에 대한 자원이 추가되어 JMeter 플러그인을 사용할 수 있습니다. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2021년 4월	릴리스 버전 1.3.0: 동시 테스트 실행에 대한 자원이 추가되었습니다. 동일한 테스트에 속한 작업 간에 동시에 테스트를 시작하기 위한 자원이 추가되었습니다. 테스트 예약에 대한 자원이 추가되었습니다. 작업 제한을 1,000개로 늘렸습니다. 동시 사용자 제한이 제거되었습니다. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2021년 9월	릴리스 버전 2.0.0: 이전 테스트 실행에서 전체 테스트 구성, 테스트 데이터 및 Amazon CloudWatch 대시보드를 볼 수 있는 자원이 추가되었습니다. 이제 솔루션 컨테이너 이미지가 AWS에서 관리되므로 고객 계정에서 AWS CodePipeline, AWS CodeBuild 및 Amazon ECR 이미지 리포지토리를 생성해야 하는 요구 사항이 제거됩니다. 최대 데이터 포인트를 표시하도록 CloudWatch 대시보드를 업데이트했습니다. 기존 Amazon VPC에 대한 자원이 추가되었습니다. CloudFormation 태그를 Fargate 작업에 전파했습니다. 테스트를 위한 Fargate 작업은 여러 가용성 영역에서 시작됩니다. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.

날짜	변화
2021년 12월	릴리스 버전 2.0.1: developmentme에서 업데이트된 AWS SDK 버전 AWS Lambda 함수에 대한 종속성; 해결된 문제 많은 수의 테스트를 표시합니다. 해결된 검증 n DynamoDB에 예외 오류가 발생했습니다. 자세한 내용은 다음을 참조하세요. CHANGELOG.md 에 GitHub 저장소에 있는 파일입니다.
2022년 8월	릴리스 버전 3.0.0: AWS CDK V2로 업데이트, 다중 리전 기능 추가, 라이브 날짜 기 능 추가, 추가 JMeter 확장 호환성 및 추가 개선 사항 그리고 버그 수정. 자세한 내용은 CHANGELOG 를 참조하세요. .md GitHub 저장소에 있는 파일입니다.
2022년 11월	릴리스 버전 3.1.0: 다음에 대한 AppRegistry 지원이 추가되었습니다. 해결책. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2022년 11월	릴리스 버전 3.1.1: 버그 수정. 자세한 내용은 다음을 참조하세요. CHANGELOG.md GitHub 저장소에 있는 파일입니다.
2023년 3월	릴리스 버전 3.2.0: 다음을 기반으로 작업 제한이 업데이트되었습니다. 계정 할당량, 자동 새로고침 기능 추가, Linting 추가 일관된 형식화, 부트스트랩 5로 업그레이드 사소한 버그 수정을 구현했습니다. 자세한 내용은 다음을 참조하세요. CHANGELOG.md GitHub 저장소에 있는 파일입니다.
2023년 4월	릴리스 버전 3.2.1: 새로운 기본값으로 인한 영향 완화 모든 새로운 S3에 대한 S3 객체 소유권(ACL 비활성화됨) 설정 버킷. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2023년 7월	릴리스 버전 3.2.2: 반응 스크립트 v5.0.1로 업데이트되었습니다. 기타 버그 수정 및 사소한 업데이트. 자세한 내용은 다음을 참조하세요. CHANGELOG.md 에 GitHub 저장소에 있는 파일입니다.
2023년 8월	설명서 업데이트: 다음에 대한 AWS 자원 섹션이 추가되었습니다. 자원팀에 문의 중입니다.

날짜	변화
2023년 10월	릴리스 버전 3.2.3: CDK 라이브러리 및 Docker 이미지가 업데이트되었습니다. 기타 버그 수정 및 보안 업데이트. 자세한 내용은, CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2023년 11월	설명서 업데이트: AWS 개발자 지원 추가 및 AWS 지원팀에 문의를 문제 해결에 병합했습니다. 부분.
2023년 11월	릴리스 버전 3.2.4: 사용하도록 업데이트된 Lambda 런타임 Node.js 18, Taurus 이미지를 버전 1.16.26으로 업데이트, 업데이트된 Docker 이미지 내의 JMeter 및 Taurus 종속성은 Docker 이미지의 보안 강화 및 추가 버그 수정. 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2023년 11월	문서 업데이트: "확인"이라는 섹션을 추가했습니다. 솔루션과 관련된 비용 태그 를 AWS Service Catalog AppRegistry를 사용하여 솔루션 모니터링 섹션에 나와 있습니다.
2024년 1월	릴리스 버전 3.2.5: 자동 문제 해결을 위한 버그 수정 플러그인 설치. 자세한 내용은 다음을 참조하세요. 변경 로그.md GitHub 저장소에 있는 파일입니다.
2024년 3월	릴리스 버전 3.2.6: 활성화된 문제를 해결하기 위한 버그 수정 ECS의 리소스 인증에 태그를 지정합니다. 황소자리 버전 업데이트 중, 반응, Chart.js, 증폭, 증폭/ui-반응. 자세한 내용은, CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.
2024년 3월	릴리스 버전 3.2.7: 정보를 포함하도록 변경 로그를 업데이트했습니다. 이전 릴리스에서는 켜져 있습니다. package-lock.json을 추가했습니다. 이상 자세한 내용은 CHANGELOG.md 를 참조하세요. GitHub 저장소에 있는 파일입니다.

날짜	변화
2024년 4월	<p>릴리스 버전 3.2.8: 취약점을 해결하기 위해 패키지 업데이트 npm 패키지와 docker 이미지에 있습니다. 해결해야 할 버그 수정 일정 문제. 자세한 내용은 다음을 참조하세요.</p> <p>변경 로그.md GitHub 저장소에 있는 파일입니다.</p>

공지사항

고객은 이 문서의 정보를 독립적으로 평가할 책임이 있습니다. 이 문서는 (a) 정보 제공의 목적으로만 제공되며, (b) 사전 통지 없이 변경될 수 있는 AWS의 현재 제품 제공 및 관행을 나타내며, (c) AWS 및 그 계열사, 공급업체 또는 라이선스 제공자. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 "있는 그대로" 제공됩니다. 고객에 대한 AWS의 책임과 책임은 AWS 계약에 의해 통제되며, 이 문서는 AWS와 고객 간의 계약의 일부가 아니며 계약을 수정하지도 않습니다.

AWS의 분산 로드 테스트는 [Apache Software Foundation](#) 에서 제공하는 Apache 라이선스 버전 2.0의 조건에 따라 라이선스가 부여됩니다 .

AWS 용어집

최신 AWS 용어는 [AWS 용어집](#) 참조하세요. [AWS 용어집](#) 참조 에서 .