구현 안내서

AWS의 가상 대기실

AWS의 가상 대기실: 구현 안내서

Table of Contents

의 가상 대기실을 사용하여 웹 사이트로 대량 트래픽 버스트를 흡수합니다. AWS

게시 날짜: 2021년 11월

의 가상 대기실 AWS 솔루션은 트래픽이 대량으로 급증하는 동안 웹 사이트로 들어오는 사용자 요청을 제어하는 데 도움이 됩니다. 웹 사이트로 들어오는 트래픽을 일시적으로 오프로드하도록 설계된 클라우드 인프라를 생성하고 가상 대기실을 사용자 지정하고 통합하는 옵션을 제공합니다. 이 솔루션을 신규 또는 기존 웹 사이트와 통합하여 갑작스러운 트래픽 급증을 처리하도록 원활하게 확장할 수 있습니다.

웹 사이트 트래픽이 급증할 수 있는 대규모 이벤트의 예는 다음과 같습니다.

- 콘서트 또는 스포츠 이벤트 티켓 판매 시작
- 블랙 프라이데이와 같은 소방 판매 또는 기타 대규모 소매 판매
- 광범위한 마케팅 발표가 포함된 신제품 출시
- 온라인 테스트 및 레슨을 위한 시험 액세스 및 클래스 참석
- 의료 예약 슬롯 릴리스
- 계정 생성 및 결제가 필요한 새로운 direct-to-customer 서비스 출시

이 솔루션은 웹 사이트 방문자의 대기 영역 역할을 하며 충분한 용량이 있을 때 트래픽이 통과할 수 있도록 합니다. 방문자가 사용하는 클라이언트 소프트웨어는 웹 사이트가 최대 용량이 될 때까지 대기실을 통한 트래픽을 투명하게 허용하도록 구성할 수 있습니다. 대기실은 방문자를 대기시킵니다. 웹 사이트에 더 많은 트래픽을 저장할 수 있는 용량이 있으면이 솔루션은 사용자가 웹 사이트에 액세스할 수 있는 JSON 웹 토큰(JWT)을 생성합니다. 예를 들어 2시간 동안 지속되는 이벤트가 있고 웹 사이트에서 초당 50명의 사용자를 처리할 수 있지만 초당 250명의 볼륨이 예상되는 경우이 솔루션을 사용하여 트래픽을 규제하는 동시에 사용자가 대기열에서 자신의 위치를 유지할 수 있도록 할 수 있습니다.

이 솔루션은 다음과 같은 주요 기능을 제공합니다.

- 웹 사이트에 대한 사용자의 구조화된 대기열
- 매우 큰 이벤트 크기로 트래픽을 제어하는 확장성
- 대상 사이트에 대한 입력을 허용하는 JSON 웹 토큰 생성
- 모든 기능은 REST APIs.
- 클라이언트 솔루션용 Turnkey API Gateway 권한 부여자

- 독립 실행형 통합 또는 OpenID와 함께 사용

이 구현 가이드에서는 Amazon Web Services(AWS) 클라우드 AWS 에서에 가상 대기실을 배포하기 위한 아키텍처 고려 사항 및 구성 단계를 설명합니다. 여기에는 보안 및 가용성 AWS 모범 사례를 사용하여이 솔루션을 배포하는 데 필요한 AWS 서비스를 시작하고 구성하는 AWS CloudFormation 템플릿에 대한 링크가 포함되어 있습니다.

이 가이드는 AWS 클라우드에서 실제 설계 경험이 있는 IT 아키텍트, 개발자, DevOps 직원, 데이터 분석가 및 마케팅 기술 전문가를 대상으로 합니다.

비용

이 솔루션을 실행하는 동안 사용되는 AWS 서비스의 비용은 사용자가 부담합니다. 이 개정부터 미국 동부(버지니아 북부) 리전의 기본 설정으로이 솔루션을 실행하는 데 드는 비용은 스택당 일일 약 10.00 USD이며 이벤트 크기를 기준으로 API 요청 및 데이터 트래픽에 대한 요금이 부과됩니다.

이벤트 없이 솔루션을 유지 관리하기 위한 일일 비용

AWS service	요청/시간	비용[USD]
Amazon API Gateway	0	0.00 USD
Amazon CloudFront	0	0.00 USD
Amazon CloudWatch	0	0.00 USD
Amazon DynamoDB	0	0.00 USD
Amazon ElastiCache	컴퓨팅 노드 시간(Redis)	~$6.00
AWS Lambda	프리 티어*	0.00 USD
AWS Secrets Manager	프리 티어*	0.00 USD
Amazon Simple Storage Service(S3)	프리 티어*	0.00 USD
Amazon Virtual Private Cloud(VPC)	VPC 엔드포인트 시간 NAT 게이트웨이 시간	~$5.00
합계:		~$11.00

*비용 추정은 깨끗한 환경을 기반으로 합니다. 이 솔루션 외부에서이 AWS 서비스를 사용하는 경우 프리 티어 할당량을 초과할 수 있습니다.

다음 표에는 50,000명의 사용자와 100,000명의 사용자 대기실에 대한 예상 비용이 나와 있으며, 이벤트 지속 시간은 초당 500명의 사용자와 발신 1,000명의 사용자로 2~4시간입니다. 요금은 변경될 수 있습니다. 자세한 내용은이 솔루션에 사용되는 각 AWS 서비스의 요금 웹 페이지를 참조하세요.

2시간 이벤트 중 대기실 사용자 50,000명에 대한 예상 비용

AWS service	Dimensions	비용[USD]
Amazon API Gateway	요청	2.00 USD
CloudFront	요청, 대역폭	75.00 USD
CloudWatch	지표, 경보, 스토리지	1.00 USD
Amazon CloudWatch Events	이벤트	1.00 USD
DynamoDB	읽기/쓰기 단위, 스토리지	1.00 USD
ElastiCache	노드 시간	8.00 USD
Lambda	요청, 컴퓨팅 시간	1.00 USD
AWS Secrets Manager	보안 암호, 요청	1.00 USD
Amazon S3	요청, 스토리지	1.00 USD
Amazon VPC	데이터 전송, 엔드포인트 시간	2.00 USD
합계		94.00 USD

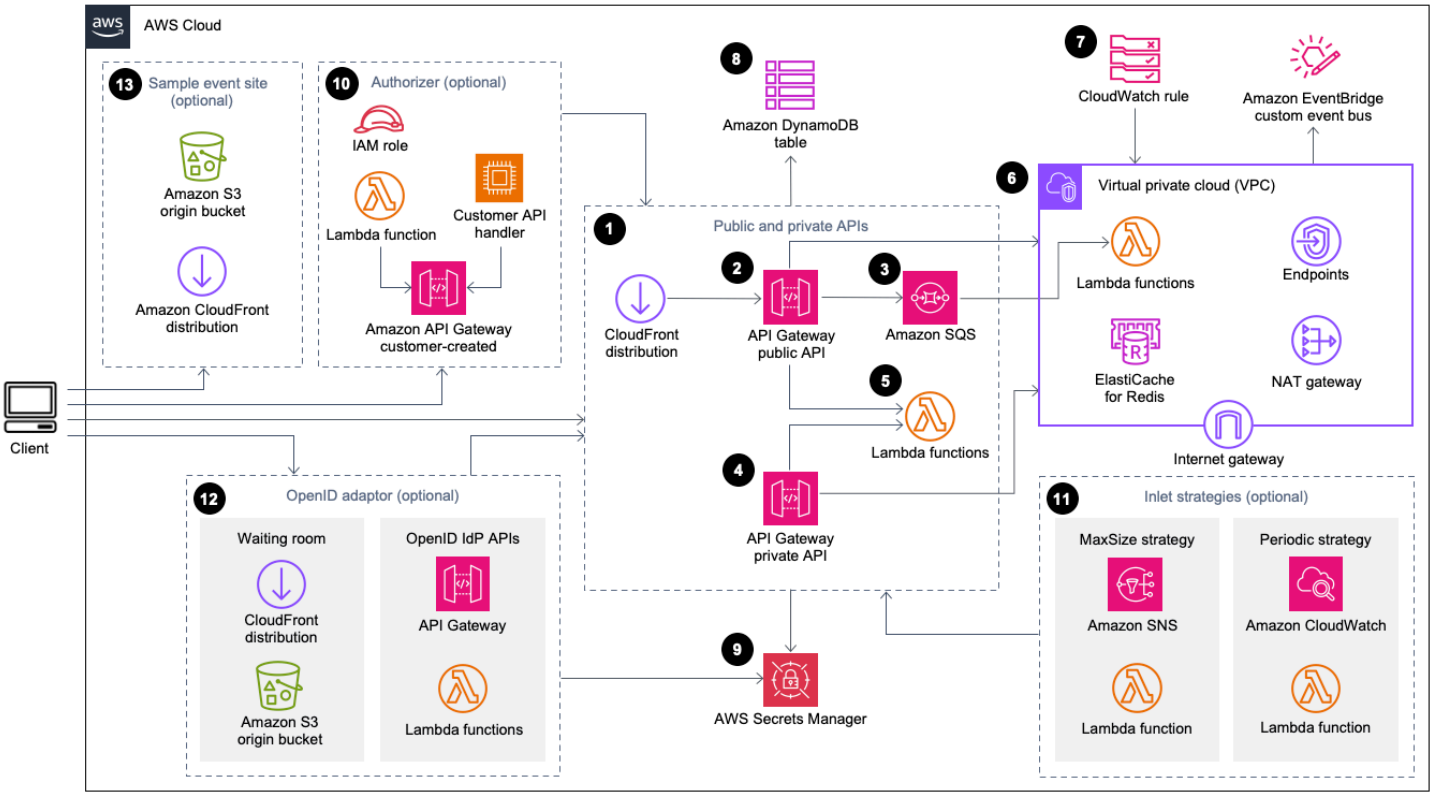
2시간 이벤트 중 대기실 사용자 100,000명에 대한 예상 비용

AWS service	Dimensions	비용[USD]
Amazon API Gateway	요청	4.00 USD
CloudFront	요청, 대역폭	296.00 USD
CloudWatch	지표, 경보, 스토리지	1.00 USD
CloudWatch Events	이벤트	1.00 USD
DynamoDB	읽기/쓰기 단위, 스토리지	4.00 USD

ElastiCache	노드 시간	32.00 USD
Lambda	요청, 컴퓨팅 시간	1.00 USD
AWS Secrets Manager	보안 암호, 요청	1.00 USD
Amazon Simple Queue Service(Amazon SQS)	요청	1.00 USD
Amazon S3	요청, 스토리지	1.00 USD
Amazon VPC	데이터 전송, 엔드포인트 시간	6.00 USD
합계		348.00 USD

아키텍처 개요

기본 파라미터를 사용하여 필수 및 선택적 템플릿으로이 솔루션을 배포하면 AWS 클라우드에 다음 환경이 빌드됩니다.



AWS 아키텍처의 가상 대기실

AWS CloudFormation 템플릿은 다음 인프라를 배포합니다.

1. 클라이언트에 대한 퍼블릭 API 호출을 전달하는 Amazon CloudFront 배포입니다.
2. 가상 대기실에서 대기열 요청을 처리하고, 대기열 위치를 추적하고, 대상 웹 사이트에 대한 액세스를 허용하는 토큰 검증을 지원하는 Amazon API Gateway 퍼블릭 API 리소스입니다.
3. 대기열 메시지를 처리하는 AWS Lambda 함수에 대한 트래픽을 규제하는 Amazon Simple Queue Service(Amazon SQS) 대기열입니다. 각 요청에 대해 Lambda 함수를 호출하는 대신 SQS 대기열은 수신되는 요청 버스트를 일괄 처리합니다.
4. 관리 함수를 지원하는 API Gateway 프라이빗 API 리소스입니다.
5. Lambda는 퍼블릭 및 프라이빗 API 요청을 검증 및 처리하고 적절한 응답을 반환하는 기능을 합니다.

6. Amazon Virtual Private Cloud(VPC)는 Elasticache(Redis OSS) 클러스터와 직접 상호 작용하는 Lambda 함수를 호스팅합니다. VPC 엔드포인트를 사용하면 VPC의 Lambda 함수가 솔루션 내의 서비스와 통신할 수 있습니다. 또한 NAT 게이트웨이를 사용하면 VPC의 Lambda 함수가 CloudFront 엔드포인트를 연결하고 필요에 따라 캐시를 무효화할 수 있습니다.
7. 사용자 지정 Amazon EventBridge 버스와 함께 작동하여 상태 업데이트를 주기적으로 브로드캐스트하는 Lambda 함수를 호출하는 Amazon CloudWatch 규칙입니다. EventBridge
8. 토큰, 대기열 위치 및 서비스 카운터 데이터를 저장하는 Amazon DynamoDB 테이블입니다.
9. 토큰 작업 및 기타 민감한 데이터를 위한 키를 저장하는 AWS Secrets Manager입니다.
10. (선택 사항) API Gateway와 함께 사용할 AWS Identity and Access Management (IAM) 역할과 Lambda 권한 부여자 함수로 구성된 권한 부여자 구성 요소입니다.
11. (선택 사항) Amazon Simple Notification Service(Amazon SNS), CloudWatch 및 Lambda 함수는 두 가지 인렛 전략을 지원합니다.
12. (선택 사항) OpenID 공급자가 웹 사이트에 사용자를 인증할 수 있도록 API Gateway 및 Lambda 함수가 포함된 OpenID 어댑터 구성 요소입니다. 이 구성 요소의 대기실 페이지에 대한 Amazon Simple Storage Service(Amazon S3) 버킷이 있는 CloudFront 배포입니다.
13. (선택 사항) 샘플 대기실 웹 애플리케이션을 위한 Amazon S3 오리진 버킷이 있는 CloudFront 배포입니다.

솔루션 작동 방식

이 섹션에서는 AWS 가상 대기실 워크플로의 단계를 상위 수준에서 설명합니다. 웹 사이트의 대기실 구축, 사용자 지정 및 통합에 대한 자세한 내용은 GitHub의 개발자 안내서를 참조하세요.

대기실의 퍼블릭 API는 사이트 경계 보안 뒤에 위치하거나 승인 없이 사용할 수 있습니다. 대기실을 웹 사이트와 통합하는 데 사용하는 접근 방식에 따라 사용자는 대기실로 이동하여 대기열의 위치를 얻기 전에 먼저 웹 사이트에 인증해야 할 수 있습니다.

대기실에 들어가서 다른 요청을 하려면 클라이언트 소프트웨어에 이벤트 ID가 있어야 합니다. 이벤트 ID는 퍼블릭 및 프라이빗 APIs. 이벤트 ID는 코어 API 스택을 설치하는 동안 설정됩니다. 작업 중에 이벤트 ID는 대기실 페이지를 통해 URL 파라미터 또는 쿠키로 제공할 수 있습니다. 인증 토큰 클레임의 일부로 제공하거나 다른 데이터 경로를 통해 클라이언트에 배포할 수 있습니다.

클라이언트가 특정 API를 호출하기 위해 이벤트 ID와 요청 ID가 모두 필요한 경우가 있습니다. 요청 ID는 대기실에서 발급된 고유 ID로, 특정 클라이언트를 일렬로 나타냅니다.

다음 단계에서는 대기열 항목에 대한 API 요청 흐름, 대기열 진행 대기, 웹 사이트의 액세스 토큰을 사용하여 대기실 종료를 설명합니다.

사용자가 대기실에 들어옵니다.

1. 사용자에게 대기실 진입점을 나타내는 화면 또는 페이지가 표시됩니다. 대기열에 들어가도록 선택하고 클라이언트 소프트웨어(브라우저, 모바일, 디바이스)가 `assign_queue_num` 퍼블릭 API를 호출하여 대기열 위치를 요청합니다.
2. API Gateway는 API 요청을 Amazon SQS 대기열로 즉시 전송합니다.
3. 요청이 대기열에 배치되면 `assign_queue_num` API 호출이를 반환합니다. 클라이언트는 나중에 대기열 위치, 요청 시간 및 액세스 토큰을 검색하는 데 사용할 수 있는 고유한 요청 ID를 수신합니다.
4. `AssignQueueNum` Lambda 함수는 SQS 대기열에서 최대 10개의 요청 배치를 수신합니다. Lambda 서비스는 호출을 팬아웃하여 여러 배치의 요청을 처리합니다.
5. `AssignQueueNum` Lambda 함수는 배치의 각 메시지를 검증하고, Elasticache(Redis OSS)의 대기열 카운터를 늘리고, 연결된 대기열 위치와 함께 각 요청을 Elasticache(Redis OSS)에 저장합니다.
6. 각 메시지는 성공적으로 처리되면 삭제됩니다. 오류 조건과 관련된 메시지는 이후 배치에서 한 번 재처리됩니다. 두 번째 실패 후 CloudWatch 경보에 연결된 dead-letter-queue로 전송됩니다.
7. 클라이언트는 `assign_queue_num` 호출에서 요청 ID를 수신한 후 `queue_num` API 폴링을 시작할 수 있습니다. 클라이언트는 이벤트 ID와 요청 ID를 `queue_num` API로 전송하고 숫자 대기열 위치 또는 요청이 아직 처리되지 않았음을 나타내는 응답을 수신합니다. 클라이언트는 대규모 이벤트 중

에이 호출을 두 번 이상 수행해야 할 수 있습니다. GetQueueNum Lambda 함수는 API Gateway에서 호출되며 DynamoDB에서 대기열의 클라이언트 숫자 위치를 반환합니다.

사용자가 대기실에서 대기합니다.

8. 클라이언트가 대기열에 위치하면 정기적으로 serving_num API 폴링을 시작할 수 있습니다. serving_num API는 이벤트 ID로 호출되고 대기열의 현재 제공 위치를 반환합니다. serving_num API의 응답은 대기실에서 최종 트랜잭션이 발생할 수 있는 실제 대상 사이트로 이동할 수 있는 시기를 클라이언트에게 알려줍니다. GetServingNum Lambda 함수는 대기실의 현재 서빙 위치를 반환합니다.
9. 제공 위치가 클라이언트의 대기열(요청) 위치와 같거나 크면 클라이언트는 퍼블릭 API에서 JSON 웹 토큰(JWT)을 요청할 수 있습니다. 토큰은 대상 사이트와 함께 사용하여 트랜잭션을 완료할 수 있습니다. generate_token API는 이벤트 ID 및 요청 ID와 함께 호출됩니다. API Gateway는 파라미터와 함께 GenerateToken Lambda 함수를 호출합니다.
10. GenerateToken Lambda 함수는 요청을 검증하고이 토큰이 이전에 생성되었는지 확인합니다. Lambda 함수는 DynamoDB 테이블에서 일치하는 토큰을 쿼리합니다. 이 토큰이 발견되면 호출자에게 반환되고 다시 생성되지 않습니다. 이 프로세스는 단일 요청 ID를 사용하여 새 만료 시간으로 여러 개의 서로 다른 토큰을 생성하는 것을 방지합니다.
11. DynamoDB에서 토큰을 찾을 수 없는 경우 Lambda 함수는 키를 검색하여 토큰을 생성하고 이벤트 ID 및 클라이언트의 요청 ID와 함께 DynamoDB에 토큰을 저장합니다. Lambda 함수는 EventBridge에 이벤트를 작성하여 새 토큰이 생성되었음을 알립니다. Lambda 함수는 이벤트에 대해 생성된 토큰 수를 추적하는 Elasticache(Redis OSS) 카운터를 증가시킵니다.
12. queue_pos_expiry이 켜져 있으면 클라이언트는 GetQueuePositionExpiryTime Lambda 함수를 호출하는 queue_pos_expiry API를 호출하여 만료 전 남은 시간을 쿼리할 수 있습니다.

사용자가 대기실을 떠납니다.

13. 클라이언트는 토큰을 수신하면 대상 사이트에 들어가 트랜잭션을 시작합니다. 인프라가 JWT와의 통합을 지원하는 방식에 따라 클라이언트는 요청 헤더, 쿠키 또는 다른 방법으로 토큰을 제시해야 할 수 있습니다. API Gateway에 대한 권한 부여자를 사용하여 클라이언트 요청에 포함된 토큰을 검증할 수 있습니다. JWTs를 검증하고 관리하기 위한 모든 상용 또는 오픈 소스 라이브러리는 AWS 토큰의 가상 대기실과 함께 사용할 수 있습니다. 토큰이 유효한 경우 클라이언트는 트랜잭션을 계속할 수 있습니다.
14. 클라이언트가 트랜잭션을 완료하면 프라이빗 API가 호출되어 클라이언트 토큰의 상태를 업데이트하고 DynamoDB에서 완료됩니다.

대기열 위치 만료:

15.이 기능이 활성화되면 특정 대기열 위치에 해당하는 요청 ID는 지정된 시간 간격 동안만 토큰을 생성할 수 있습니다.

대기열 위치 만료 시 증분 서빙 카운터:

16.이 기능이 활성화되면 토큰을 생성할 수 없었던 만료된 대기열 위치에 따라 서빙 카운터가 자동으로 증가합니다.

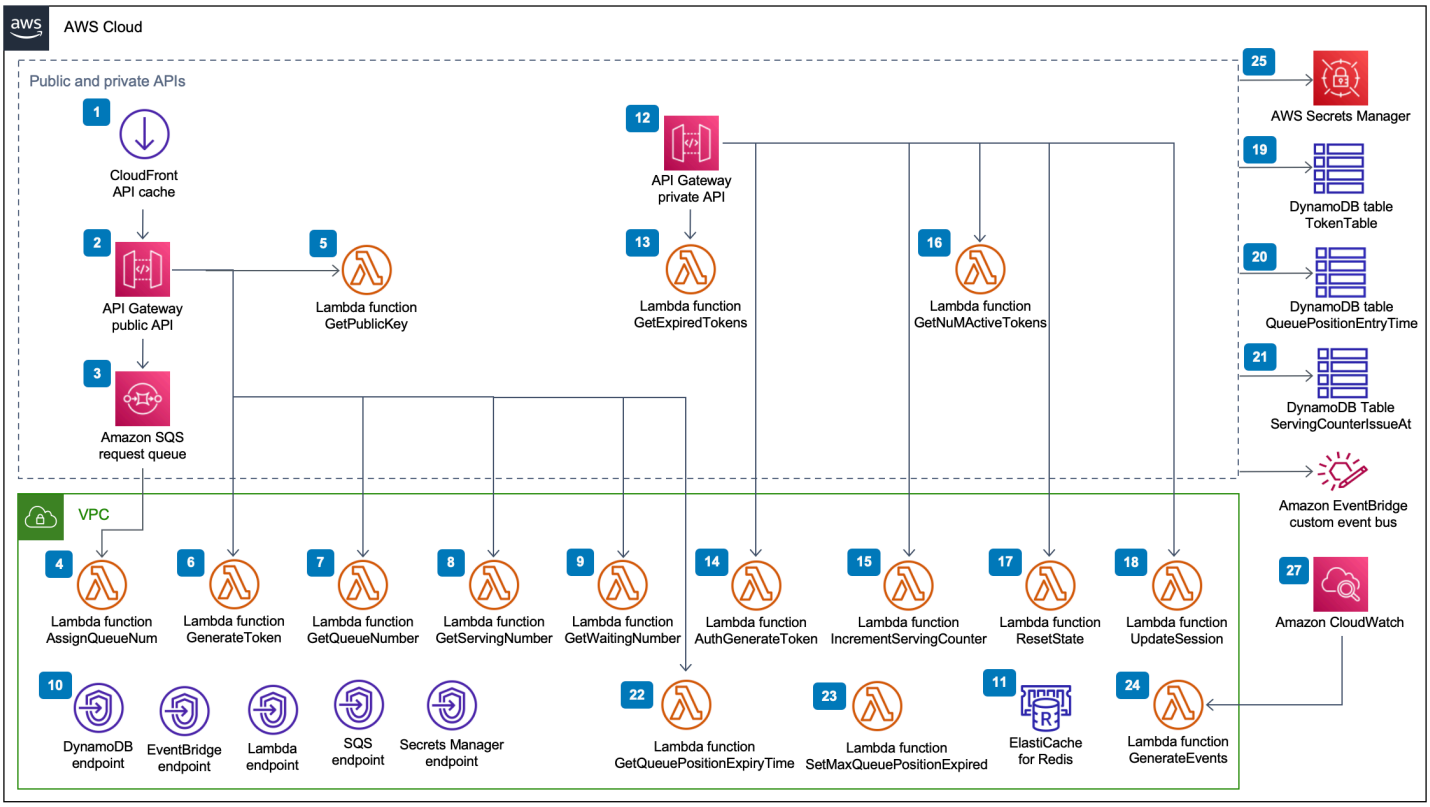
솔루션 구성 요소

대기실 퍼블릭 및 프라이빗 APIs

AWS 솔루션의 기본 목적은 대상 웹 사이트를 압도할 수 있는 새 사용자의 버스트를 방지하기 위해 클라이언트용 JSON 웹 토큰(JWT) 생성을 제어하기 위한 것입니다. JWTs는 사이트 보호, 대기실 토큰을 획득할 때까지 웹 페이지에 대한 액세스 방지 및 API 액세스 권한 부여에도 사용할 수 있습니다.

코어 템플릿은 AWS 작업 시 대부분의 가상 대기실에 사용되는 퍼블릭 API 및 프라이빗(IAM 인증) API를 설치합니다. 퍼블릭 API는 API 경로를 기반으로 여러 캐싱 정책이 있는 CloudFront 배포로 구성됩니다. DynamoDB 테이블과 EventBridge 이벤트 버스가 생성됩니다. 템플릿은 두 개의 가용 영역(AZs), 두 AZ의 Elasticache(Redis OSS) 클러스터 및 여러 Lambda 함수가 있는 새 VPCAZs 추가합니다. Elasticache(Redis OSS)와 상호 작용하는 Lambda 함수는 VPC 내에 네트워크 인터페이스가 있고 다른 모든 Lambda 함수는 기본 네트워크 연결이 있습니다. 코어 APIs는 솔루션과의 가장 낮은 상호 작용 계층입니다. 다른 Lambda 함수, Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 및 컨테이너는 확장 역할을 하고 코어 APIs를 호출하여 대기실을 구축하고, 인렛 트래픽을 제어하고, 솔루션에서 생성된 이벤트에 대응할 수 있습니다.

또한 코어 스택은 모든 Lambda 함수 오류 및 스로틀 조건에 대한 경보와 4XX 및 5XX 상태 코드에 대한 각 API Gateway 배포에 대한 경보를 생성합니다.



AWS 퍼블릭 및 프라이빗 APIs 구성 요소의 가상 대기실

1. CloudFront 배포는 클라이언트에 대한 퍼블릭 API 호출을 전송하고 적절한 경우 결과를 캐시합니다.
2. Amazon API Gateway 퍼블릭 API는 가상 대기실의 대기열 요청을 처리하고, 대기열 위치를 추적하고, 대상 웹 사이트에 대한 액세스를 허용하는 토큰 검증을 지원합니다.
3. SQS 대기열은 대기열 메시지를 처리하는 AWS Lambda 함수에 대한 트래픽을 조절합니다.
4. `AssignQueueNum` Lambda 함수는 수신된 배치의 각 메시지를 검증하고, Elasticache(Redis OSS)의 대기열 카운터를 늘리고, 연결된 대기열 위치와 함께 각 요청을 Elasticache(Redis OSS)에 저장합니다.
5. `GetPublicKey` Lambda 함수는 Secrets Manager에서 퍼블릭 키 값을 검색합니다.
6. `GenerateToken` Lambda 함수는 대상 사이트에서 트랜잭션을 완료할 수 있도록 허용된 유효한 요청에 대해 JWT를 생성합니다. 토큰이 생성되었다는 이벤트를 대기실의 사용자 지정 이벤트 버스에 씁니다. 이 요청에 대해 토큰이 이전에 생성된 경우 새 토큰이 생성되지 않습니다.
7. `GetQueueNumber` Lambda 함수는 Elasticache(Redis OSS)에서 대기열의 클라이언트 숫자 위치를 검색하고 반환합니다.

8. GetServingNumber Lambda 함수는 Elasticache(Redis OSS)에서 대기실에서 현재 처리 중인 번호를 검색하고 반환합니다.

9. GetWaitingNum Lambda 함수는 현재 대기실에서 대기 중이고 아직 토큰이 발급되지 않은 번호를 반환합니다.

10. VPC 엔드포인트를 사용하면 VPC의 Lambda 함수가 솔루션 내의 서비스와 통신할 수 있습니다.

11. Elasticache(Redis OSS) 클러스터는 유효한 이벤트 ID로 대기실에 들어가기 위한 모든 요청을 저장합니다. 또한 대기열에 추가된 요청 수, 현재 처리 중인 요청 수, 생성된 토큰 수, 완료된 세션 수, 중단된 세션 수와 같은 여러 카운터를 저장합니다.

12. 관리 함수를 지원하는 API Gateway 프라이빗 API 리소스입니다. 프라이빗 APIs는 AWS IAM 인증을 받습니다.

13. GetExpiredTokens Lambda 함수는 토큰이 만료된 요청 IDs 목록을 반환합니다.

14. AuthGenerateToken Lambda 함수는 대상 사이트에서 트랜잭션을 완료할 수 있도록 허용된 유효한 요청에 대한 토큰을 생성합니다. 코어 스택 배포 중에 처음 설정된 토큰의 발급자 및 유효 기간을 재정의할 수 있습니다. 토큰이 생성되었다는 이벤트를 대기실의 사용자 지정 이벤트 버스에 씁니다. 이 요청에 대해 토큰이 이전에 생성된 경우 새 토큰이 생성되지 않습니다.

15. IncrementServingCounter Lambda 함수는 값별로 증분하면 Elasticache(Redis OSS)에 저장된 대기실의 서빙 카운터를 증가시킵니다.

16. GetNumActiveTokens Lambda 함수는 아직 만료되지 않았고, 트랜잭션을 완료하는 데 사용되지 않았으며, 중단됨으로 표시되지 않은 토큰 수에 대해 DynamoDB를 쿼리합니다.

17. ResetState Lambda 함수는 Elasticache(Redis OSS)에 저장된 모든 카운터를 재설정합니다. 또한 TokenTable, 및 ServingCounterIssuedAt DynamoDB 테이블을 삭제QueuePositionEntryTime하고 다시 생성합니다. 또한 CloudFront 캐시 무효화를 수행합니다.

18. UpdateSession Lambda 함수는 TokenTable DynamoDB 테이블에 저장된 세션(토큰)의 상태를 업데이트합니다. 세션 상태는 정수로 표시됩니다. 로 설정된 세션은 완료를 1 나타내고 중단됨을 -1 나타냅니다. 세션이 업데이트되었다는 이벤트를 대기실의 사용자 지정 이벤트 버스에 씁니다.

19. TokenTable DynamoDB 테이블은 토큰 데이터를 저장합니다.

20. QueuePositionEntryTime DynamoDB 테이블은 대기열 위치 및 진입 시간 데이터를 저장합니다.

21. ServingCounterIssuedAt DynamoDB 테이블은 서빙 카운터에 대한 업데이트를 저장합니다.

22. 클라이언트가 나머지 대기열 위치 만료 시간을 요청하면 GetQueuePositionExpireTime Lambda 함수가 호출됩니다.

23 SetMaxQueuePositionExpired Lambda 함수는 ServingCounterIssuedAt 테이블 값에 해당하는 만료된 최대 대기열 위치를 설정합니다. 코어 스택 배포 true 중에 IncrSvcOnQueuePositionExpiry 파라미터가 로 설정된 경우 1분마다 실행됩니다.

24 GenerateEvents Lambda 함수는 다양한 대기실 지표를 대기실의 사용자 지정 이벤트 버스에 씁니다. 코어 스택 배포 true 중에 이벤트 생성 활성화 파라미터가 로 설정된 경우 1분마다 실행됩니다.

25 AWS Secrets Manager는 토큰 작업 및 기타 민감한 데이터를 위한 키를 저장합니다.

26 Amazon EventBridge 사용자 지정 이벤트 버스는 토큰이 생성되고 TokenTable DynamoDB 테이블에서 세션이 업데이트될 때마다 이벤트를 수신합니다. 또한 SetMaxQueuePositionExpired Lambda에서 서빙 카운터가 이동될 때 이벤트를 수신합니다. 코어 스택 배포 중에 활성화된 경우 다양한 대기실 지표로에 기록됩니다.

27 코어 스택 배포 중에 이벤트 생성 활성화 파라미터가 true로 설정된 경우 Amazon CloudWatch 이벤트 규칙이 생성됩니다. 이 이벤트 규칙은 1분마다 GenerateEvents Lambda 함수를 시작합니다.

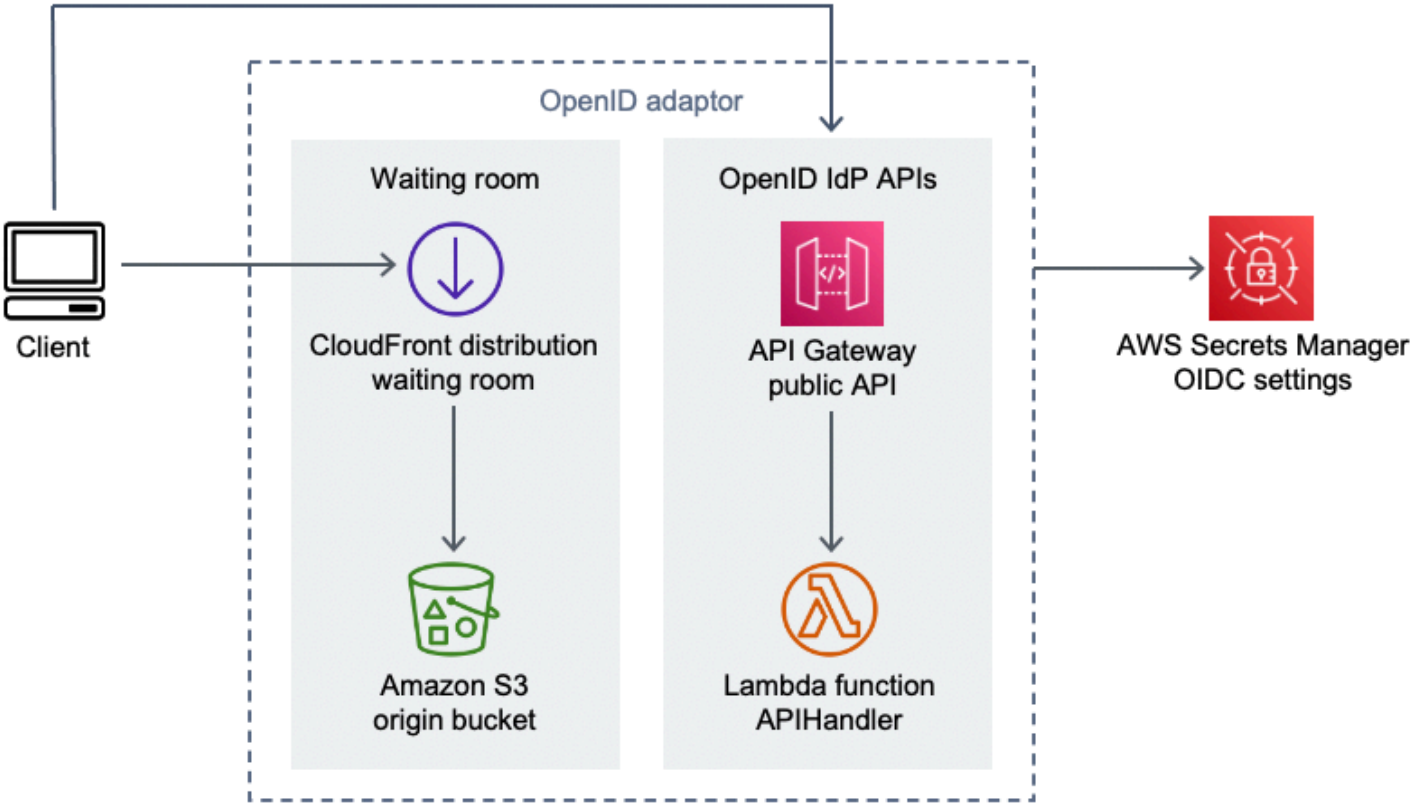
권한 부여자

솔루션에는 API Gateway Lambda 권한 부여자 스택이 포함되어 있습니다. 스택은 하나의 IAM 역할과 Lambda 함수로 구성됩니다. APIGatewayAuthorizer Lambda 함수는 API의 가상 대기실에서 발급한 토큰의 서명 및 클레임을 검증할 수 있는 AWS API Gateway의 권한 부여자입니다. 스택과 함께 제공되는 Lambda 함수는 사용자가 대기실을 통과하여 액세스 토큰을 받을 때까지 클라우드 APIs를 보호하는 데 사용할 수 있습니다. 권한 부여자는 토큰 확인을 위해 코어 API에서 퍼블릭 키와 구성을 자동으로 검색하고 캐싱합니다. 수정 없이 사용할 수 있으며가 지원하는 모든 AWS 리전에 설치할 수 있습니다 AWS Lambda.

OpenID 어댑터

OpenID 어댑터 스택은 OpenID 자격 증명 공급자 역할을 하는 API Gateway 및 Lambda 함수를 배포합니다. OpenID 어댑터는 AWS Elastic Load Balancer, WordPress와 같은 OIDC 자격 증명 공급자를 지원하는 기존 웹 호스팅 소프트웨어와 함께 사용하거나 Amazon Cognito 또는 유사한 서비스에 대한 페더레이션 자격 증명 공급자로 사용할 수 있는 OIDC 호환 APIs 세트를 제공합니다. 어댑터를 사용하면 통합 옵션이 제한된 off-the-shelf 웹 호스팅 소프트웨어를 사용할 때 고객이 AuthN/AuthZ 흐름의 대기실을 사용할 수 있습니다. 또한 스택은 Amazon S3 버킷 하나를 오리진으로 사용하고 다른 S3 버킷을 사용하여 요청을 로깅하는 CloudFront 배포를 설치합니다. OpenID 어댑터는 샘플 대기실 스택에 제공되지만 OpenID 인증 흐름을 위해 설계된 것과 유사한 샘플 대기실 페이지를 제공합니다. 인증 프로세스에는 대기실 대기열에서 위치를 얻고 서빙 위치가 클라이언트의 대기열 위치와 같거나 클 때까지

기다리는 작업이 포함됩니다. OpenID 대기실 페이지는 OpenID API를 사용하여 클라이언트의 토큰 획득 및 세션 구성을 완료하는 대상 사이트로 다시 리디렉션됩니다. 이 솔루션의 API 엔드포인트는 공식 OpenID Connect 1.0 흐름 사양인 name-for-name에 직접 매핑됩니다. 자세한 내용은 OpenID Connect Core 1.0 인증을 참조하세요.



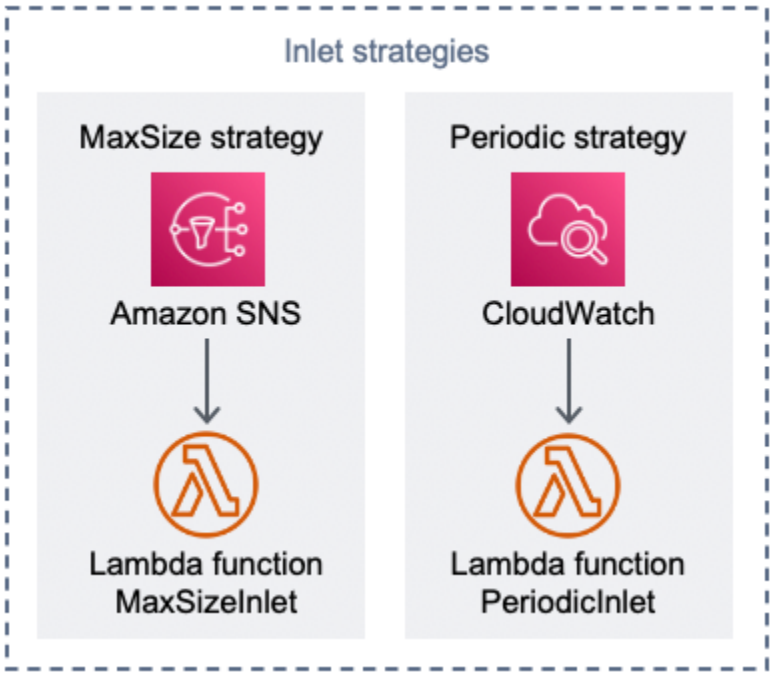
AWS OpenID 어댑터 구성 요소의 가상 대기실

1. CloudFront 배포는 S3 버킷의 콘텐츠를 사용자에게 제공합니다.
2. S3 버킷은 샘플 대기실 페이지를 호스팅합니다.
3. Amazon API Gateway API는 OIDC 자격 증명 공급자의 Lambda 권한 부여 함수를 지원하는 기존 웹 호스팅 소프트웨어와 함께 사용할 수 있는 OIDC 호환 APIs 세트를 제공합니다.
4. `APIHandler` Lambda 함수는 모든 API Gateway 리소스 경로에 대한 요청을 처리합니다. 동일한 모듈 내의 서로 다른 Python 함수가 각 API 경로에 매핑됩니다. 예를 들어 API Gateway의 `/authorize` 리소스 경로는 Lambda 함수 `authorize()` 내에서 호출합니다.
5. OIDC 설정은 Secrets Manager에 저장됩니다.

샘플 인렛 전략

주입구 전략은 대상 사이트에서 더 많은 사용자를 수용하기 위해 솔루션의 서빙 카운터를 진행해야 하는 시기를 결정합니다. 대기실 입구 전략에 대한 자세한 개념 정보는 설계 고려 사항을 참조하세요.

솔루션에서 제공하는 두 가지 샘플 인렛 전략은 MaxSize와 Periodic입니다.



AWS 주입구 전략 구성 요소의 가상 대기실

최대 크기 인렛 전략 옵션:

1. 클라이언트는 메시지 페이로드를 기반으로 서빙 카운터를 늘리기 위해 `MaxSizeInlet` Lambda 함수를 호출하는 Amazon SNS 알림을 발행합니다.
2. `MaxSizeInlet` Lambda 함수는 사용하는 메시지를 수신할 것으로 예상하여 서빙 카운터를 얼마나 늘릴지 결정합니다.

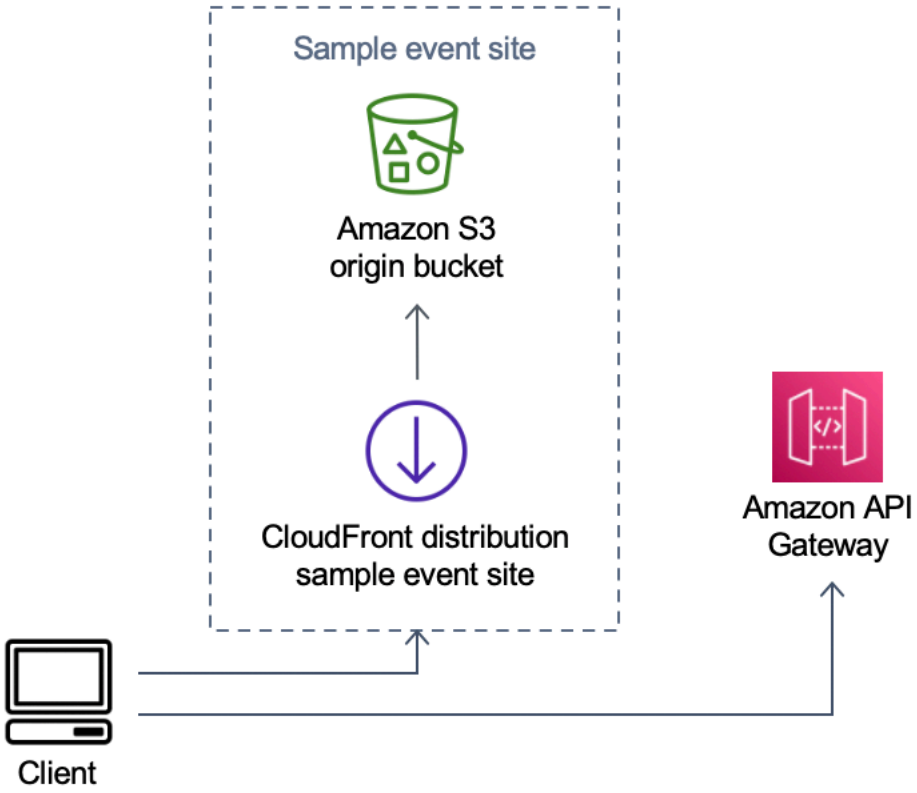
주기적 인렛 전략 옵션:

3. CloudWatch 규칙은 1분마다 Lambda 함수를 호출하여 서빙 카운터를 고정된 수량만큼 늘립니다.
4. `PeriodicInlet` Lambda 함수는 제공된 시작 시간과 종료 시간 사이에 있는 경우 지정된 크기만큼 서빙 카운터를 증가시킵니다. 선택적으로 CloudWatch 경보를 확인하고 경보가 OK 상태인 경우 증분을 수행하고 그렇지 않으면 건너뜁니다.

대기실 샘플

샘플 대기실은 사용자 지정 권한 부여자 외에도 퍼블릭 및 프라이빗 APIs와 통합되어 최소한의 end-to-end 대기실 솔루션을 보여줍니다. 기본 웹 페이지는 S3 버킷에 저장되며 CloudFront의 오리진으로 사용됩니다. 사용자는 다음 단계를 거칩니다.

1. 대기실에서 현장에 들어가기 위해 줄을 서세요.
2. 클라이언트의 위치를 일직선으로 구합니다.
3. 대기실의 서빙 위치를 확인합니다.
4. 서빙 위치가 클라이언트의 위치와 같거나 크면 토큰 세트를 가져옵니다.
5. 토큰을 사용하여 Lambda 권한 부여자로 보호되는 API를 호출합니다.



AWS 샘플 이벤트 사이트 구성 요소의 가상 대기실

1. S3 버킷은 대기실 및 제어판에 대한 샘플 콘텐츠를 호스팅합니다.
2. CloudFront 배포는 사용자에게 S3 버킷 콘텐츠를 제공합니다.

3. `/search` 및와 같은 쇼핑과 유사한 리소스 경로를 사용하여 API Gateway 배포를 샘플링합니다`/checkout`. 이 API는 스택에 의해 설치되고 토큰 권한 부여자로 구성됩니다. 대기실로 API를 보호하는 간단한 방법의 예입니다. 유효한 토큰이 있는 요청은 Lambda로 전달되며, 그렇지 않으면 오류가 반환됩니다. 연결된 Lambda 함수의 응답 외에는 API에 대한 기능이 없습니다.

보안

AWS 인프라를 기반으로 시스템을 구축하면 보안 책임은 사용자와 간에 공유됩니다 AWS. 이 공유 모델은 호스트 운영 체제 AWS , 가상화 계층, 서비스가 운영되는 시설의 물리적 보안을 비롯한 구성 요소를 운영, 관리 및 제어하기 때문에 운영 부담을 줄입니다. AWS 보안에 대한 자세한 내용은 AWS 클라우드 보안을 참조하십시오.

Elasticache(Redis OSS)에는 프라이빗 VPC 내부의 네트워크 인터페이스가 할당됩니다. Elasticache(Redis OSS)와 상호 작용하는 Lambda 함수에도 VPC 내의 네트워크 인터페이스가 할당됩니다. 다른 모든 리소스는 공유 네트워크 공간에 AWS 네트워크 연결이 있습니다. 다른 AWS 서비스와 상호 작용하는 VPC 인터페이스가 있는 Lambda 함수는 VPC 엔드포인트를 사용하여 이러한 서비스에 연결합니다.

JSON 웹 토큰을 생성하고 검증하는 데 사용되는 퍼블릭 및 프라이빗 키는 배포 시 생성되어 Secrets Manager에 저장됩니다. Elasticache(Redis OSS)에 연결하는 데 사용되는 암호도 배포 시 생성되어 Secrets Manager에 저장됩니다. 프라이빗 키 및 Elasticache(Redis OSS) 암호는 솔루션 API를 통해 액세스할 수 없습니다.

퍼블릭 API는 CloudFront를 통해 액세스해야 합니다. 이 솔루션은 CloudFront`x-api-key`에서 사용자 지정 헤더의 값으로 사용되는 API Gateway용 API 키를 생성합니다. CloudFront는 오리진 요청을 할 때 이 헤더를 포함합니다. 자세한 내용은 Amazon CloudFront 개발자 안내서의 오리진 요청에 사용자 지정 헤더 추가를 참조하세요.

프라이빗 APIs는 호출을 위해 AWS IAM 인증을 요구하도록 구성됩니다. 솔루션은 프라이빗 APIs를 호출할 수 있는 적절한 권한을 가진 `ProtectedAPIGroup` IAM 사용자 그룹을 생성합니다. 이 그룹에 추가된 IAM 사용자에게는 프라이빗 APIs.

솔루션에서 생성한 다양한 리소스에 연결된 역할 및 권한에 사용되는 IAM 정책은 필요한 작업을 수행하는 데 필요한 권한만 부여합니다.

솔루션에서 생성된 S3 버킷, SQS 대기열 및 SNS 주제와 같은 리소스의 경우 가능한 경우 저장 중 및 전송 중 암호화가 활성화됩니다.

모니터링

코어 API 스택에는 솔루션이 작동하는 동안 문제를 감지하기 위해 모니터링할 수 있는 여러 CloudWatch 경보가 포함되어 있습니다. 스택은 Lambda 함수 오류 및 스로틀 조건에 대한 경보를 생성하고 오류 또는 스로틀 조건이 1분 동안 발생하는 `OK` `ALARM` 경우 경보의 상태를에서 로 변경합니다.

또한 스택은 4XX 및 5XX 상태 코드에 대한 각 API Gateway 배포에 대한 경보를 생성합니다. 1분 이내에 API에서 0K ALARM 4XX 또는 5XX 상태 코드가 반환되면 경보 상태가에서 로 변경됩니다.

이러한 경보는 1분 동안 오류나 스로틀이 없으면 0K 상태로 돌아갑니다.

IAM 역할

AWS Identity and Access Management (IAM) 역할을 통해 고객은 AWS 클라우드의 서비스 및 사용자에게 세분화된 액세스 정책 및 권한을 할당할 수 있습니다. 이 솔루션은 솔루션의 AWS Lambda 함수에 리전 리소스를 생성할 수 있는 액세스 권한을 부여하는 IAM 역할을 생성합니다.

Amazon CloudFront

대기실의 코어 퍼블릭 및 프라이빗 APIs를 생성하는 `virtual-waiting-room-on-aws.template` CloudFormation 템플릿은 퍼블릭 API에 대한 CloudFront 배포도 배포합니다. CloudFront는 퍼블릭 API의 응답을 캐싱하여 작업을 수행하는 API Gateway 및 Lambda 함수에 대한 부하를 줄입니다.

또한이 솔루션에는 Amazon Simple Storage Service(Amazon S3) 버킷에 호스팅되는 간단한 웹 애플리케이션을 배포하는 선택적 샘플 대기실 템플릿이 있습니다. 지연 시간을 줄이고 보안을 강화하기 위해 Amazon CloudFront 배포는 오리진 액세스 ID, 즉 솔루션의 웹 사이트 버킷 콘텐츠에 대한 퍼블릭 액세스를 제공하는 CloudFront 사용자와 함께 배포됩니다. 자세한 내용을 알아보려면 Amazon CloudFront 개발자 안내서의 오리진 액세스 ID(OAI)를 사용하여 Amazon S3 콘텐츠에 대한 액세스 제한을 참조하세요.

보안 그룹

이 솔루션에서 생성된 VPC 보안 그룹은 Elasticache(Redis OSS)에 대한 네트워크 트래픽을 제어하고 격리하도록 설계되었습니다. Elasticache(Redis OSS)와 통신해야 하는 Lambda는 Elasticache(Redis OSS)와 동일한 보안 그룹에 배치됩니다. 배포가 시작되고 실행되면 보안 그룹을 검토하고 필요에 따라 액세스를 추가로 제한하는 것이 좋습니다.

설계 고려 사항

배포 옵션

설치가 처음이거나 무엇을 설치해야 할지 잘 모르는 경우 코어, 권한 부여자 및 샘플 대기실 템플릿을 설치하는 `virtual-waiting-room-on-aws-getting-started.template` 중첩된 CloudFormation 템플릿을 배포합니다. 이렇게 하면 간단한 흐름으로 대기실을 최소화할 수 있습니다.

지원되는 프로토콜

의 가상 대기실 AWS 솔루션은 다음과 통합할 수 있습니다.

- JSON 웹 토큰 확인 라이브러리 및 도구
- 기존 API Gateway 배포
- REST API 클라이언트
- OpenID 클라이언트 및 공급자

대기실 입구 전략

주입구 전략은 클라이언트를 대기실에서 웹 사이트로 이동하는 데 필요한 로직과 데이터를 캡슐화합니다. 입력 전략은 Lambda 함수, 컨테이너, Amazon EC2 인스턴스 또는 기타 컴퓨팅 리소스로 구현할 수 있습니다. 대기실 퍼블릭 및 프라이빗 APIs를 호출할 수 있는 한 클라우드 리소스일 필요는 없습니다. 인렛 전략은 대기실, 웹 사이트 또는 기타 외부 지표에 대한 이벤트를 수신하여 더 많은 클라이언트가 토큰을 발급하고 사이트에 들어갈 수 있는 시기를 결정하는 데 도움이 됩니다. 인렛 전략에는 몇 가지 접근 방식이 있습니다. 어떤 리소스를 채택할지는 사용 가능한 리소스와 보호 대상 웹 사이트 설계의 제약 조건에 따라 달라집니다.

인렛 전략에서 수행하는 기본 작업은 사이트에 들어갈 수 있는 클라이언트 수를 나타내는 상대 값으로 `increment_serving_num` Amazon API Gateway 프라이빗 API를 호출하는 것입니다. 이 섹션에서는 두 가지 샘플 인렛 전략에 대해 설명합니다. 이는 있는 그대로 사용하거나, 사용자 지정하거나, 완전히 다른 접근 방식을 사용할 수 있습니다.

MaxSize

MaxSize 전략을 사용하면 `MaxSizeInlet` Lambda 함수가 웹 사이트를 동시에 사용할 수 있는 최대 클라이언트 수로 구성됩니다. 고정된 값입니다. 클라이언트는 메시지 페이로드를 기반으로 서빙 카운

터를 늘리기 위해 MaxSizeInlet Lambda 함수를 호출하는 Amazon SNS 알림을 발행합니다. SNS 메시지의 소스는 웹 사이트의 코드 또는 사이트의 사용률 수준을 관찰하는 모니터링 도구를 포함하여 어디에서나 올 수 있습니다.

MaxSizeInlet Lambda 함수는 다음을 포함할 수 있는 메시지를 수신할 것으로 예상합니다.

- exited : 완료된 트랜잭션 수
- 완료로 표시할 요청 IDs 목록
- 중단됨으로 표시할 요청 IDs 목록

이 데이터는 서빙 카운터를 얼마나 늘릴지 결정하는 데 사용됩니다. 현재 클라이언트 수에 따라 카운터를 늘릴 추가 용량이 없는 경우가 있을 수 있습니다.

주기적

주기적 전략을 사용할 때 CloudWatch 규칙은 1분마다 PeriodicInlet Lambda 함수를 호출하여 서빙 카운터를 고정된 수량만큼 늘립니다. 주기적 입력은 이벤트 시작 시간, 종료 시간 및 증분 양으로 파라미터화됩니다. 선택적으로이 전략은 CloudWatch 경보도 확인하고, 경보가 OK 상태인 경우 증분을 수행하며, 그렇지 않으면 건너뜁니다. 사이트 통합자는 사용률 지표를 경보에 연결하고 해당 경보를 사용하여 주기적 입력자를 일시 중지할 수 있습니다. 이 전략은 현재 시간이 시작 시간과 종료 시간 사이에 있는 동안에만 제공 위치를 변경하고, 선택적으로 지정된 경보가 OK 상태에 있습니다.

솔루션 사용자 지정 및 확장

조직의 사이트 관리자는 대기실과 함께 사용할 통합 방법을 결정해야 합니다. 두 가지 옵션이 있습니다.

1. APIs 및 API Gateway 권한 부여자를 사용한 기본 통합.
2. 자격 증명 공급자를 통한 OpenID 통합.

위의 통합 외에도 도메인 이름 리디렉션을 구성해야 할 수 있습니다. 또한 사용자 지정 대기실 사이트 페이지를 배포해야 합니다.

의 가상 대기실 AWS 솔루션은 단방향 이벤트 알림을 위한 EventBridge와 양방향 통신을 위한 REST APIs라는 두 가지 메커니즘을 통해 확장되도록 설계되었습니다.

할당량

의 가상 대기실에 대한 기본 규모 제한 AWS 은 설치된 AWS 리전에 대한 Lambda 스로틀 제한입니다. 기본 Lambda 동시 실행 할당량을 사용하여 AWS 계정에 설치하면의 가상 대기실 AWS 솔루션은 대기열의 위치를 요청하는 초당 최대 500개의 클라이언트를 처리할 수 있습니다. 초당 500 클라이언트 속도는 모든 Lambda 함수 동시 할당량 제한을 독점적으로 사용할 수 있는 솔루션을 기반으로 합니다. 계정의 리전이 Lambda 함수를 호출하는 다른 솔루션과 공유되는 경우 솔루션의 가상 대기실에는 최소 1,000개의 동시 호출이 사용 가능 AWS 해야 합니다. CloudWatch 지표를 사용하여 시간 경과에 따라 계정의 Lambda 동시 호출을 차트로 작성하여 결정할 수 있습니다. Service Quotas 콘솔을 사용하여 증가를 요청할 수 있습니다. Lambda 스로틀 제한을 늘리면 추가 호출이 실제로 발생하는 경우에만 월별 계정 요금이 발생합니다.

초당 클라이언트 500개마다 스로틀 제한을 1,000씩 늘립니다.

초당 수신 사용자 예상	권장 동시 실행 할당량
0-500	1,000(기본값)
501~1,000	2,000
1,001~1,500	3,000

Lambda의 고정 버스트 제한은 동시 호출 3,000개입니다. 자세한 내용은 Lambda 함수 조정을 참조하세요. 클라이언트 코드는 일시적인 스로틀 상황을 나타내는 오류 코드가 반환되는 경우 일부 API 호출을 예상하고 다시 시도해야 합니다. 샘플 대기실 클라이언트에는 대용량 및 높은 버스트 이벤트에 사용되는 클라이언트를 설계하는 방법의 예로이 코드가 포함되어 있습니다.

이 솔루션은 사용자 지정 구성 단계를 통해 Lambda 예약 및 프로비저닝된 동시성과도 호환됩니다. 자세한 내용은 Lambda 예약 동시성 관리를 참조하세요.

대기실에 들어가 토큰을 받고 트랜잭션을 계속할 수 있는 사용자의 상한은 Elasticache(Redis OSS) 카운터의 상한에 의해 제한됩니다. 카운터는 대기실 서빙 위치와 솔루션의 추적 요약 상태에 사용됩니다. Elasticache(Redis OSS)에 사용되는 카운터의 상한은 9,223,372,036,854,775,807입니다. DynamoDB 테이블은 대기실 사용자에게 발급된 각 토큰의 사본을 저장하는 데 사용됩니다. DynamoDB는 테이블 크기에 대한 실질적인 제한이 없습니다.

리전 배포

이 솔루션에서 사용하는 서비스는 모든 AWS 리전에서 지원됩니다. 리전별 AWS 서비스의 최신 가용성은 AWS 리전 서비스 목록을 참조하세요.

AWS CloudFormation 템플릿

배포를 자동화하기 위해이 솔루션은 배포 전에 다운로드할 수 있는 다음 AWS CloudFormation 템플릿을 사용합니다.

설치가 처음이거나 무엇을 설치해야 할지 잘 모르는 경우 코어, 권한 부여자 및 샘플 대기실 코드 `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation 템플릿을 설치하는 템플릿을 배포합니다. 이렇게 하면 간단한 흐름으로 작업 대기실을 테스트할 수 있습니다.

virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template:이 템플릿을 사용하여 CloudWatch 로깅 권한에 대한 계정 수준에서 API Gateway에 기본 역할 ARN을 추가합니다. 계정에이 템플릿의 배포가 필요한지 여부에 대한 자세한 내용은 사전 조건을 참조하세요.

virtual-waiting-room-on-aws-getting-started.template:이 중첩 템플릿을 사용하여 코어, 권한 부여자 및 샘플 대기실 스택을 설치합니다.

virtual-waiting-room-on-aws.template:이 코어 템플릿을 사용하여 대기실 이벤트를 생성하기 위한 코어 퍼블릭 및 프라이빗 REST APIs와 클라우드 서비스를 설치합니다. 대기실 REST APIs, Elasticache(Redis OSS) 및 DynamoDB 테이블이 필요한 계정 및 리전에이 템플릿을 설치합니다.

virtual-waiting-room-on-aws-authorizers.template:이 템플릿을 사용하여 대기실에서 발급한 토큰을 확인하도록 설계되고 최종 사용자의 APIs를 보호하기 위한 Lambda 권한 부여자를 설치합니다. 코어 스택이 필요합니다. 코어 스택의 일부 출력은이 스택을 배포하기 위한 파라미터로 필요합니다. 이는 선택적 템플릿입니다.

virtual-waiting-room-on-aws-openid.template:이 템플릿을 사용하여 권한 부여 인터페이스와의 대기실 통합을

위한 OpenID 자격 증명 공급자를 설치합니다. 코어 스택이 필요합니다. 이 스택을 배포하려면 코어 스택의 일부 출력이 필요합니다. 이는 선택적 템플릿입니다.

View template

virtual-waiting-room-on-aws-sample-inlet-strategy.template:이 템플릿을 사용하여 대상 사이트와 대기실 간에 사용할 샘플 인렛 전략을 설치합니다. 주입구 전략은 로직을 캡슐화하여 대상 사이트에 더 많은 사용자를 허용할 시기를 결정하는 데 도움이 됩니다. 코어 스택이 필요합니다. 이 스택을 배포하려면 코어 스택의 출력이 필요합니다. 이는 선택적 템플릿입니다.

View template

virtual-waiting-room-on-aws-sample.template:이 템플릿을 사용하여 대기실 및 대상 사이트에 대한 샘플 최소 웹 및 API Gateway 구성을 설치합니다. 코어 및 권한 부여자 스택이 필요합니다. 코어 및 권한 부여자 스택의 출력은이 스택을 배포하기 위한 파라미터로 필요합니다. 이는 선택적 템플릿입니다.

배포 자동화

솔루션을 시작하기 전에이 가이드에서 설명하는 비용, 아키텍처, 네트워크 보안 및 기타 고려 사항을 검토하세요. 이 섹션의 step-by-step 지침에 따라 솔루션을 구성하고 계정에 배포합니다.

배포 시간: 약 30분(시작된 스택만 해당)

사전 조건

- AWS 관리자 액세스와 동등한 계정 콘솔 권한.
- API Gateway에서 CloudWatch 로깅 활성화:
  - API Gateway 콘솔에 로그인하고 스택을 설치할 리전을 선택합니다.

    이 리전에 정의된 기존 APIs 있는 경우:

    1. API를 선택합니다.
    2. 왼쪽 탐색 창에서 설정을 선택합니다.
    3. CloudWatch 로그 역할 ARN 필드에서 값을 확인합니다.
    - ARN이 없는 경우를 설치합니다`virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template`.
    - ARN이 있는 경우 시작하기 스택 시작부터 시작합니다.

    이 리전에 정의된 기존 APIs가 없는 경우를 설치합니다`virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template`.
- 보호할 대상 사이트의 아키텍처 및 구현 세부 정보에 대한 지식입니다.

배포 개요

다음 단계에 따라이 솔루션을 배포합니다 AWS. 자세한 지침은 각 단계에 대한 링크를 따르세요.

1단계. 시작하기 스택 시작

- AWS 계정으로 AWS CloudFormation 템플릿을 시작합니다.
- 템플릿 파라미터를 검토하고 필요에 따라 기본값을 입력하거나 조정합니다.

2단계. (선택 사항) 대기실 테스트

- AWS 키를 생성하여 IAM 보안 APIs.
- 샘플 대기실의 제어판을 엽니다.
- 샘플 대기실을 테스트합니다.

1단계. 시작하기 스택 시작

이 자동 AWS CloudFormation 템플릿은 작업 대기실을 보고 테스트할 수 있는 코어, 권한 부여자 및 샘플 대기실 템플릿을 배포합니다. 스택을 시작하기 전에 사전 조건을 읽고 이해해야 합니다.

> **Note**
>
> 이 솔루션을 실행하는 동안 사용되는 AWS 서비스의 비용은 사용자가 부담합니다. 자세한 내용은이 가이드의 비용 섹션을 참조하고이 솔루션에 사용되는 각 AWS 서비스의 요금 웹 페이지를 참조하세요.

1. 에 로그인AWS Management Console하고 버튼을 선택하여 `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation 템플릿을 시작합니다.

   Launch solution

   또는 템플릿을 자체 구현의 시작점으로 다운로드할 수 있습니다.
2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 시작하려면 콘솔 탐색 모음에서 리전 선택기를 사용합니다.
3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	Default	설명
이벤트 ID	`Sample`	대기실의이 인스턴스에 대한 고유 ID, GUID 형식이 제안되었습니다.
검증 기간	`3600`	초 단위의 토큰 유효 기간입니다.
이벤트 생성 활성화	`false`	로 설정하면 대기실과 관련된 `true`지표가 1분마다 이벤트 버스에 기록됩니다.
Elasticache(Redis OSS) 포트	`1785`	Elasticache(Redis OSS) 서버에 연결하는 데 사용할 포트 번호입니다. 의 기본 Elasticache(Redis OSS) 포트를 사용하지 않는 것이 좋습니다6379.
EnableQueuePositionExpiry	`true`	로 설정하면 `false`대기열 위치 만료 기간이 적용되지 않습니다.
QueuePositionExpiryPeriod	`900`	이 간격은 대기열 위치가 토큰을 생성할 수 없는 초 단위의 시간 간격입니다.
IncrSvcOnQueuePositionExpiry	`false`	로 설정하면 토큰`true`을 성공적으로 생성하지 못한 만료된 대기열 위치를 기반으로 서빙 카운터가 자동으로 진행됩니다.

6. Next(다음)를 선택합니다.
7. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management (IAM) 리소스를 생성했음을 확인하는 확인란을 선택합니다.

9. [스택 생성(Create stack)]을 선택하여 스택을 배포합니다.

   AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 30분 후에 CREATE_COMPLETE 상태가 표시됩니다.

2단계. (선택 사항) 대기실 테스트

시작하기 스택을 배포한 경우 다음 단계는 대기실의 기능을 테스트하는 데 도움이 됩니다. 테스트를 완료하려면 코어 스택에서 IAM 보안 APIs를 호출할 수 있는 권한이 있는 AWS 키가 필요합니다.

IAM 보안 APIs를 호출하기 위한 AWS 키 생성

1. `aws-virtual-waiting-room-getting-started.template` CloudFormation 템플릿이 배포된 AWS 계정에서 IAM 사용자를 생성하거나 사용합니다.
2. IAM 사용자에게 프로그래밍 방식 액세스 권한을 부여합니다. IAM 사용자에 대한 새 액세스 키 세트를 생성할 때 키 파일이 표시되면 다운로드합니다. 대기실을 테스트하려면 IAM 사용자의 액세스 키 ID와 보안 액세스 키가 필요합니다.
3. 템플릿으로 생성된 ProtectedAPIGroup IAM 사용자 그룹에 IAM 사용자를 추가합니다.

샘플 대기실의 제어판 열기

1. AWS CloudFormation 콘솔에 로그인하고 솔루션의 시작 스택을 선택합니다.
2. 출력 탭을 선택합니다.
3. 키 열에서 ControlPanelURL을 찾아 해당 값을 선택합니다.
4. 새 탭 또는 브라우저 창에서 제어판을 엽니다.
5. 제어판에서 구성 섹션을 확장합니다.
6. 키 생성에서 검색한 액세스 키 ID와 보안 액세스 AWS 키를 입력하여 IAM 보안 APIs. 엔드포인트와 이벤트 ID는 URL 파라미터에서 채워집니다.
7. 사용을 선택합니다. 자격 증명을 제공하면 버튼이 활성화됩니다.

샘플 대기실 테스트

1. AWS CloudFormation 콘솔에서 솔루션의 시작 스택을 선택합니다.
2. 출력 탭을 선택합니다.

3. 키 열에서 WaitingRoomURL을 찾아 해당 값을 선택합니다.
4. 대기실을 연 다음 예약을 선택하여 대기실에 들어갑니다.
5. 제어판이 있는 브라우저 탭으로 돌아갑니다.
6. 증분 서비스 카운터에서 변경을 선택합니다. 이렇게 하면 100명의 사용자가 대기실에서 대상 사이트로 이동할 수 있습니다.
7. 대기실로 돌아가 지금 체크아웃을 선택합니다! 이제 대상 사이트로 리디렉션됩니다.
8. 지금 구매를 선택하여 대상 사이트에서 트랜잭션을 완료합니다.

별도의 스택 배포

코어 스택은 대기실의 주요 기능을 가져오는 데 필요한 유일한 스택입니다. 다른 모든 스택은 선택 사항입니다. 대기실에서 발급한 토큰을 검증하거나 이미 가지고 있을 수 있는 APIs를 보호할 방법이 아직 없는 경우 권한 부여자 스택을 시작합니다. 권한 부여 인터페이스와 룸 통합을 대기하기 위해 OpenID 자격 증명 공급자가 필요한 경우 OpenID 스택을 시작합니다. 샘플 인렛 전략 스택은 보호하려는 사이트에 더 많은 사용자를 허용하는 방법과 시기에 대한 몇 가지 예를 제공합니다.

1. 코어 스택 시작

배포에 소요되는 시간: 약 20분

이 자동 AWS CloudFormation 템플릿은 클라우드 AWS 의에 가상 대기실을 AWS 배포합니다. 스택을 시작하기 전에 사전 조건을 완료해야 합니다.

> **Note**
>
> 이 솔루션을 실행하는 동안 사용되는 AWS 서비스의 비용은 사용자가 부담해야 합니다. 자세한 내용은이 가이드의 비용 섹션을 참조하고이 솔루션에 사용되는 각 AWS 서비스의 요금 웹 페이지를 참조하세요.

1. 에 로그인AWS Management Console하고 버튼을 선택하여 `aws-virtual-waiting-room-on-aws.template` AWS CloudFormation 템플릿을 시작합니다.

   Launch solution

   또는 템플릿을 자체 구현의 시작점으로 다운로드할 수 있습니다.
2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 시작하려면 콘솔 탐색 모음에서 리전 선택기를 사용합니다.
3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	Default	설명
이벤트 ID	`Sample`	Waiting Room의이 인스턴스에 대한 고유 ID, GUID 형식이 제안되었습니다.
검증 기간	`3600`	초 단위의 토큰 유효 기간입니다.
이벤트 생성 활성화	`false`	로 설정하면 대기실과 관련된 `true`지표가 1분마다 이벤트 버스에 기록됩니다.
Elasticache(Redis OSS) 포트	`1785`	Elasticache(Redis OSS) 서버에 연결하는 데 사용할 포트 번호입니다. 의 기본 Elasticache(Redis OSS) 포트를 사용하지 않는 것이 좋습니다6379.
EnableQueuePositionExpiry	`true`	로 설정하면 `false`대기열 위치 만료 기간이 적용되지 않습니다.
QueuePositionExpiryPeriod	`900`	이 간격은 대기열 위치가 토큰을 생성할 수 없는 초 단위의 시간 간격입니다.
IncrSvcOnQueuePositionExpiry	`false`	로 설정하면 토큰`true`을 성공적으로 생성하지 못한 만료된 대기열 위치에 따라 서빙 카운터가 자동으로 진행됩니다.

6. Next(다음)를 선택합니다.
7. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management (IAM) 리소스를 생성했음을 확인하는 확인란을 선택합니다.
9. [스택 생성(Create stack)]을 선택하여 스택을 배포합니다.

AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 20분 후에 CREATE_COMPLETE 상태를 받게 됩니다.

2. (선택 사항) 권한 부여자 스택 시작

배포에 소요되는 시간: 약 5분

1. 에 로그인AWS Management Console하고 버튼을 선택하여 `aws-virtual-waiting-room-on-aws-authorizers.template` AWS CloudFormation 템플릿을 시작합니다.

   Launch solution

   또는 템플릿을 자체 구현의 시작점으로 다운로드할 수 있습니다.
2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 시작하려면 콘솔 탐색 모음에서 리전 선택기를 사용합니다.
3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	Default	설명
퍼블릭 API 엔드포인트	`<## ##>`	가상 대기실 APIs.
대기실 이벤트 ID	`Sample`	대기실의 이벤트 ID입니다.
발급자 URI	`<## ##>`	퍼블릭 키 및 토큰의 발급자 URI입니다.

6. Next(다음)를 선택합니다.
7. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management (IAM) 리소스를 생성했음을 확인하는 확인란을 선택합니다.

9. [스택 생성(Create stack)]을 선택하여 스택을 배포합니다.

   AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 5분 후에 CREATE_COMPLETE 상태가 표시됩니다.

3. (선택 사항) OpenID 스택 시작

배포에 소요되는 시간: 약 5분

1. 에 로그인AWS Management Console하고 버튼을 선택하여 `aws-virtual-waiting-room-on-aws-openid.template` AWS CloudFormation 템플릿을 시작합니다.

   Launch solution

   또는 템플릿을 자체 구현의 시작점으로 다운로드할 수 있습니다.
2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 시작하려면 콘솔 탐색 모음에서 리전 선택기를 사용합니다.
3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	Default	설명
퍼블릭 API 엔드포인트	*<## ##>*	가상 대기실 APIs.
프라이빗 API 엔드포인트	*<## ##>*	가상 대기실 APIs.
API 리전	*<## ##>*	AWS 퍼블릭 및 프라이빗 대기실 APIs.
이벤트 ID	`Sample`	대기실의 이벤트 ID입니다.

6. Next(다음)를 선택합니다.
7. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.

8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management (IAM) 리소스를 생성했음을 확인하는 확인란을 선택합니다.
9. [스택 생성(Create stack)]을 선택하여 스택을 배포합니다.

   AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 5분 후에 CREATE_COMPLETE 상태가 표시됩니다.

4. (선택 사항) 샘플 인렛 전략 스택 시작

배포 시간: 약 2분

1. 에 로그인AWS Management Console하고 버튼을 선택하여 `aws-virtual-waiting-room-sample-inlet-strategy.template` AWS CloudFormation 템플릿을 시작합니다.

   Launch solution 또는 템플릿을 자체 구현의 시작점으로 다운로드할 수 있습니다.
2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 시작하려면 콘솔 탐색 모음에서 리전 선택기를 사용합니다.
3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	Default	설명
이벤트 ID	`Sample`	대기실의 이벤트 ID입니다.
프라이빗 코어 API 엔드포인트	*<## ##>*	가상 대기실 APIs.
코어 API 리전	*<## ##>*	AWS 코어 API가 설치된 리전입니다.

파라미터	Default	설명
주입구 전략	`Periodic`	배포할 주입구 전략입니다.는 1분마다 서비스 수를 `Periodic` 증가시킵니다.는 다운스트림 대상 사이트가 지정된 시간에 처리할 수 있는 최대 트랜잭션 수를 기준으로 서비스 수를 `MaxSize` 증가시킵니다.
증분 기준	*<## ##>*	1분마다 서빙 카운터를 늘려야 하는 양입니다. 주기적 인렛 전략을 선택하는 경우 필요합니다.
시작 시간	*<## ##>*	서빙 번호(초 단위의 에포크 시간) 증가를 시작할 때의 타임스탬프입니다. 주기적 인렛 전략을 선택하는 경우 필요합니다.
종료 시간	*<## ##>*	서비스 번호(초 단위의 에포크 시간) 증가를 중지할 때의 타임스탬프입니다. 0으로 두면 서빙 번호가 무기한 증가합니다. 주기적 인렛 전략을 선택하는 경우 필요합니다.
CloudWatch 경보 이름	*<## ##>*	주기적 인렛 전략과 연결할 선택적 CloudWatch 경보 이름입니다. 제공되고 경보 상태인 경우 제공 번호가 증가하지 않습니다. 주기적 인렛 전략에만 적용됩니다.

파라미터	Default	설명
최대 크기	<## ##>	다운스트림 대상 사이트가 한 번에 처리할 수 있는 최대 트랜잭션 수(MaxSize Strategy).

6. Next(다음)를 선택합니다.
7. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management (IAM) 리소스를 생성했음을 확인하는 확인란을 선택합니다.
9. [스택 생성(Create stack)]을 선택하여 스택을 배포합니다.

   AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 2분 후에 CREATE_COMPLETE 상태가 표시됩니다.

5. (선택 사항) 샘플 대기실 스택 시작

배포에 소요되는 시간: 약 5분

1. 에 로그인AWS Management Console하고 버튼을 선택하여 `aws-virtual-waiting-room-sample.template` AWS CloudFormation 템플릿을 시작합니다.

   Launch solution 또는 템플릿을 자체 구현의 시작점으로 다운로드할 수 있습니다.
2. 이 템플릿은 기본적으로 미국 동부(버지니아 북부) 리전에서 시작됩니다. 다른 AWS 리전에서 솔루션을 시작하려면 콘솔 탐색 모음에서 리전 선택기를 사용합니다.
3. 스택 생성 페이지에서 Amazon S3 URL 텍스트 상자에 올바른 템플릿 URL이 있는지 확인하고 다음을 선택합니다.
4. 스택 세부 정보 지정 페이지에서 솔루션 스택 이름을 할당합니다. 문자 제한 이름 지정에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 및 STS 제한을 참조하세요.
5. 파라미터에서이 솔루션 템플릿의 파라미터를 검토하고 필요에 따라 수정합니다. 이 솔루션은 다음과 같은 기본값을 사용합니다.

파라미터	Default	설명
API 게이트웨이 리전	<## ##>	AWS API Gateway의 리전 이름입니다.
권한 부여자 ARN	<## ##>	API Gateway Lambda 권한 부여자의 ARN입니다.
이벤트 ID	Sample	대기실의 이벤트 ID입니다.
프라이빗 API 엔드포인트	<## ##>	가상 대기실 APIs.
퍼블릭 API 엔드포인트	<## ##>	가상 대기실 APIs.

6. Next(다음)를 선택합니다.
7. Configure stack options(스택 옵션 구성) 페이지에서 Next(다음)를 선택합니다.
8. 검토 페이지에서 설정을 검토하고 확인합니다. 템플릿이 AWS Identity and Access Management (IAM) 리소스를 생성했음을 확인하는 확인란을 선택합니다.
9. [스택 생성(Create stack)]을 선택하여 스택을 배포합니다.

   AWS CloudFormation 콘솔의 상태 열에서 스택의 상태를 볼 수 있습니다. 약 5분 후에 CREATE_COMPLETE 상태가 표시됩니다.

이전 버전에서 스택 업데이트

스택을 삭제하고 새 버전에 대한 새 스택을 생성하는 것이 좋습니다. 현재 CloudFormation 스택 업데이트를 사용하여 최신 버전으로 마이그레이션하는 것은 지원되지 않습니다. 그런 솔루션 제거 다음 시작하기 스택 시작을 참조하세요.

> **Note**
>
> 진행 중인 이벤트를 지원하기 위해 솔루션을 적극적으로 사용하지 않는 경우 최신 버전으로 마이그레이션하는 것이 좋습니다.

성능 데이터

의 가상 대기실 AWS 은 Locust라는 도구를 사용하여 로드 테스트되었습니다. 시뮬레이션된 이벤트 크기는 클라이언트 10,000~100,000개였습니다. 로드 테스트 환경은 다음 구성으로 구성되었습니다.

- AWS 클라우드 배포에 대한 사용자 지정이 포함된 Locust 2.x
- 4개 AWS 리전(`us-west-1`, `us-west-2`, `us-east-1`, `us-east-2`)
- 리전당 `c5.4xlarge` Amazon EC2 호스트 10개(총 40개)
- 호스트당 32개의 Locust 프로세스
- 시뮬레이션된 사용자가 1,280개의 프로세스 간에 고르게 분산되었습니다.

각 사용자 프로세스에 대한 end-to-end API 테스트 단계:

1. 를 호출`assign_queue_num`하고 요청 ID를 받습니다.
2. 사용자의 대기열 위치(짧은 시간)를 반환할 때까지 요청 ID`queue_num`로 루프합니다.
3. 반환된 값이 >= 사용자의 대기열 위치(장시간)가 될 `serving_num` 때까지 반복합니다.
4. 대기 중인 사용자 수를 검색`waiting_room_size`하려면를 자주 호출하지 않습니다.
5. 대상 사이트에서 사용할 JWT를 호출`generate_token`하고 수신합니다.

조사 결과

대기실을 통해 처리할 수 있는 클라이언트 수에는 실질적인 상한이 없습니다.

사용자가 대기실에 들어가는 속도는 배포된 리전의 Lambda 함수 동시 실행 할당량에 영향을 미칩니다.

로드 테스트가 CloudFront에 사용되는 캐싱 정책으로 초당 10,000개의 기본 API Gateway 요청 제한을 초과할 수 없습니다.

`get_queue_num` Lambda 함수는 대기실로 들어오는 사용자의 호출 속도에 거의 1:1의 호출 속도를 갖습니다. 동시성 제한 또는 버스트 제한으로 인해 들어오는 사용자의 비율이 높을 때이 Lambda 함수가 제한될 수 있습니다. 많은 `get_queue_num` Lambda 함수 호출로 인한 조절은 다른 Lambda 함수에 부작용으로 영향을 미칠 수 있습니다. 클라이언트 소프트웨어가 재시도/백오프 로직으로 이러한 유형의 임시 조정 오류에 적절하게 응답할 수 있는 경우 전체 시스템이 계속 작동합니다.

기본 할당량 구성에서 코어 스택에 의해 구성된 CloudFront 배포는 250,000명의 사용자가 있는 대기실을 처리할 수 있으며 각 사용자는 최소 1초마다 `serving_num` API를 폴링합니다.

문제 해결

이 섹션에서는이 솔루션에 대한 문제 해결 정보를 제공합니다.

이 섹션에서 문제를 해결하지 못하는 경우 AWS Support에 문의하세요.이 솔루션에 대한 AWS 지원 사례를 개설하기 위한 지침을 제공합니다.

APIs의 4xx 응답 상태

- 이는 잘못된 이벤트 ID 또는 요청 ID 또는 둘 다로 인해 발생할 수 있습니다. 이는 관련 Lambda 함수의 CloudWatch Logs에서 발생합니다.
- 프라이빗 APIs는 IAM 인증을 받았으며 클라이언트에는 프라이빗 APIs를 호출할 권한이 있는 AWS 키가 필요합니다. 이는 API Gateway용 CloudWatch Logs에서 발생합니다.

APIs의 5xx 응답 상태

- 제한된 Lambda 또는 API Gateway의 응답은 *<LambdaFunctionName>*ThrottlesAlarm CloudWatch 경보를 확인합니다.
- 백엔드의 구성 오류, 자세한 내용은 *<LambdaFunctionName>*ErrorsAlarm CloudWatch 경보 및 CloudWatch Logs를 확인하세요.

5XXErrorPublic/PrivateApiAlarm

- 이 경보 상태는 API가 60초 이내에 호출자에게 5XX 상태를 반환하는 ALARM 경우입니다.
- 이 경보는 60초 동안 5xx 상태가 반환OK되지 않으면 로 돌아갑니다.
- 이 경보는 API Gateway에 오류를 반환하는 Lambda 함수 또는 Lambda 런타임에 의해 시작될 수 있습니다.

4XXErrorPublic/PrivateApiAlarm

- 이 경보 상태는 API가 60초 내에 호출자에게 4XX 상태를 반환하는 ALARM 경우입니다.
- 이 경보는 4XX 상태가 60초 동안 반환될 OK 때 로 돌아갑니다.
- 이 경보는 잘못된 API URL로 시작할 수 있습니다.

*<LambdaFunctionName>*ThrottlesAlarm

- 이 경보 상태는 명명된 Lambda가 60초 이내에 동시 실행 제한을 발견할 때 ALARM입니다.
- 이 경보는 60초 동안 스로틀이 발생하지 OK 않으면 로 돌아갑니다.
- 계정 리전의 동시성 제한을 늘려야 할 수 있습니다.
- 클라이언트에서 일부 재시도 로직이 필요한 Lambda에 대한 버스트 제한이 발생할 수 있습니다.

*<LambdaFunctionName>*ErrorsAlarm

- 이 경보 상태는 명명된 Lambda에서 60초 이내에 런타임 실행 오류가 발생하는 ALARM 경우입니다.
- 이 경보는 60초 동안 오류가 발생하지 OK 않으면 로 돌아갑니다.
- 이는 백엔드의 잘못된 구성으로 인해 발생할 수 있습니다.
- 이는 Lambda 코드의 버그로 인해 발생할 수 있습니다.

연락처 지원

AWS 개발자 지원, AWS 비즈니스 지원 또는 AWS 엔터프라이즈 지원이 있는 경우 지원 센터를 사용하여이 솔루션에 대한 전문가 지원을 받을 수 있습니다. 이후 단원에서는 그 방법에 대해서 설명합니다.

사례 생성

1. 지원 센터에 로그인합니다.
2. 사례 생성을 선택합니다.

어떻게 도와드릴까요?

1. 기술을 선택합니다.
2. 서비스에서 솔루션을 선택합니다.
3. 범주에서 기타 솔루션을 선택합니다.
4. 심각도에서 사용 사례에 가장 적합한 옵션을 선택합니다.
5. 서비스, 범주 및 심각도를 입력하면 인터페이스가 일반적인 문제 해결 질문에 대한 링크를 채웁니다. 이러한 링크로 질문을 해결할 수 없는 경우 다음 단계: 추가 정보를 선택합니다.

추가 정보

1. 제목에 질문 또는 문제를 요약하는 텍스트를 입력합니다.

2. 설명에서 문제를 자세히 설명합니다.
3. 파일 연결을 선택합니다.
4. 요청을 처리하는 데 지원 필요한 정보를 첨부합니다.

사례를 더 빠르게 해결할 수 있도록 지원

1. 요청된 정보를 입력합니다.
2. 다음 단계: 지금 해결하거나 문의하기를 선택합니다.

지금 해결하거나 문의하기

1. 지금 해결 솔루션을 검토합니다.
2. 이러한 솔루션의 문제를 해결할 수 없는 경우 문의하기를 선택하고 요청된 정보를 입력한 다음 제출을 선택합니다.

추가적인 리소스

AWS 서비스	
• AWS CloudFormation	• Amazon DynamoDB
• Amazon Simple Storage Service(S3)	• Amazon API Gateway
• AWS Lambda	• AWS 보안 관리자
• Amazon CloudFront	• Amazon Simple Queue Service
• Amazon EventBridge	• Amazon CloudWatch
• Elasticache(Redis OSS)	• Amazon Comprehend
• Amazon Virtual Private Cloud	• AWS Identity and Access Management

솔루션 제거

에서 AWS Management Console 또는를 사용하여의 가상 대기실 AWS 솔루션을 제거할 수 있습니다 AWS Command Line Interface. 이 솔루션에서 생성한 다양한 리소스에 의해 로그를 저장하는 데 사용되는 S3 버킷을 수동으로 삭제해야 합니다. AWS 솔루션 구현에서는 이러한 S3 버킷을 자동으로 삭제하지 않으므로 솔루션이 삭제된 후에도 로그 이벤트를 검토할 수 있습니다.

솔루션에서 생성한 IAM 사용자 그룹에 `ProtectedAPIGroup` IAM 사용자를 수동으로 추가한 경우 솔루션을 제거하기 전에 IAM 사용자 그룹에서 IAM 사용자를 제거합니다. 그렇지 않으면 IAM 사용자 그룹과 연결된 IAM 정책이 삭제되지 않습니다.

배포된 각 스택에 대해 아래 지침을 따릅니다.

사용 AWS Management Console

1. AWS CloudFormation 콘솔에 로그인합니다.
2. 스택 페이지에서 이 솔루션의 설치 스택을 선택합니다.
3. Delete(삭제)를 선택합니다.

사용 AWS Command Line Interface

환경에서 AWS Command Line Interface (AWS CLI)를 사용할 수 있는지 확인합니다. 설치 지침은 AWS CLI 사용 설명서의 What Is the AWS Command Line Interface?를 참조하세요. AWS CLI 를 사용할 수 있는지 확인한 후 다음 명령을 실행합니다.

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

Amazon S3 버킷 삭제

이 솔루션은 실수로 데이터가 손실되지 않도록 AWS CloudFormation 스택을 삭제하기로 결정한 경우 솔루션에서 생성한 Amazon S3 버킷(옵트인 리전에 배포용)을 유지하도록 구성됩니다. 솔루션을 제거한 후 데이터를 보존할 필요가 없는 경우이 S3 버킷을 수동으로 삭제할 수 있습니다. 다음 단계에 따라 Amazon S3 버킷을 삭제합니다.

1. Amazon S3 콘솔에 로그인합니다.

2. 왼쪽 탐색 창에서 버킷을 선택합니다.
3. *<stack-name>* S3 버킷을 찾습니다.
4. S3 버킷을 선택하고 삭제를 선택합니다.

를 사용하여 S3 버킷을 삭제하려면 다음 명령을 AWS CLI실행합니다.

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

소스 코드

GitHub 리포지토리를 방문하여이 솔루션의 소스 파일을 다운로드하고 사용자 지정을 다른 사용자와 공유합니다.

기여자

- 짐 타리오
- Thyag Ramachandran
- Joan Morgan
- Justin Pirtle
- 알렌 모하이마니
- 가빗 싱
- Bassem Wanis

개정

GitHub 리포지토리의 CHANGELOG.md 파일을 확인하여 소프트웨어에 대한 모든 주요 변경 사항 및 업데이트를 확인합니다. 변경 로그는 각 버전의 개선 사항 및 수정 사항에 대한 명확한 기록을 제공합니다.

고지 사항

고객은 본 문서의 정보를 독립적으로 평가할 책임이 있습니다. 이 문서는 다음과 같습니다. (a)는 정보 제공 목적으로만 사용됩니다. (b) AWS 현재 제품 제공 및 관행을 나타냅니다. 예고 없이 변경될 수 있습니다. 및 (c)는 AWS 및 그 계열사로부터 어떠한 약정이나 보장도 생성하지 않습니다. 공급업체 또는 licensors. AWS products 또는 서비스는 보증 없이 “있는 그대로” 제공됩니다. 표현, 또는 모든 종류의 조건 명시적이든 묵시적이든, 고객에 대한 책임과 책임은 AWS 계약에 의해 제어됩니다. AWS 이 문서는 수정도 하지 않습니다. AWS 와 고객 간의 모든 계약.

의 가상 대기실 AWS 은 Apache 라이선스 버전 2.0의 약관에 따라 라이선스가 부여됩니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.