Leitfaden zur Implementierung

Virtuelles Wartezimmer auf AWS

Virtuelles Wartezimmer auf AWS: Leitfaden zur Implementierung

Table of Contents

Absorbieren Sie große Besucherströme auf Ihrer Website, wenn der virtuelle Warteraum aktiviert ist AWS

Veröffentlichungsdatum: November 2021 (letzte Aktualisierung: September 2024)

Die AWS Lösung Virtual Waiting Room on hilft Ihnen dabei, eingehende Benutzeranfragen an Ihre Website bei hohem Besucheraufkommen zu kontrollieren. Sie erstellt eine Cloud-Infrastruktur, die darauf ausgelegt ist, eingehenden Traffic vorübergehend auf Ihre Website zu verlagern, und bietet Optionen zur Anpassung und Integration eines virtuellen Warteraums. Diese Lösung kann entweder in neue oder bestehende Websites integriert werden, um sich nahtlos an plötzliche Verkehrsspitzen anzupassen.

Zu den Beispielen für Großveranstaltungen, die zu einem Anstieg des Website-Traffics führen könnten, gehören:

- Beginn des Verkaufs von Tickets für Konzerte oder Sportveranstaltungen
- Feuerverkauf oder andere große Einzelhandelsverkäufe wie Black Friday
- Einführung neuer Produkte mit umfassenden Marketingankündigungen
- Zugang zu Prüfungen und Teilnahme an Kursen für Online-Tests und Online-Unterricht
- Freigabe von Terminen für medizinische Termine
- Einführung eines neuen direct-to-customer Dienstes, für den Kontoerstellung und Zahlungen erforderlich sind

Die Lösung dient als Wartebereich für Besucher Ihrer Website und ermöglicht die Weiterleitung von Traffic, wenn genügend Kapazität vorhanden ist. Die von Besuchern verwendete Client-Software kann so konfiguriert werden, dass der Verkehr transparent durch den Wartebereich geleitet wird, bis die Website die maximale Kapazität erreicht hat. An diesem Punkt hält der Warteraum Besucher zurück. Wenn Ihre Website mehr Traffic aufnehmen kann, generiert die Lösung JSONWeb-Token (JWT), mit denen Benutzer auf die Website zugreifen können. Wenn Sie beispielsweise eine Veranstaltung haben, die zwei Stunden dauert und Ihre Website 50 Benutzer pro Sekunde verarbeiten kann, Sie aber ein Volumen von 250 pro Sekunde erwarten, können Sie diese Lösung verwenden, um den Verkehr zu regulieren und gleichzeitig den Benutzern zu ermöglichen, ihre Position in der Warteschlange zu behalten.

Diese Lösung bietet die folgenden Hauptfunktionen:

- Strukturierte Warteschlange von Benutzern auf Ihrer Website
- Skalierbarkeit zur Steuerung des Datenverkehrs bei sehr großen Veranstaltungen
- JSONGenerierung von Web-Tokens, um den Zugriff auf die Ziel-Site zu ermöglichen
- Die gesamte Funktionalität wird gesteuert durch REST APIs
- Turnkey API Gateway Authorizer für Kundenlösungen
- Eigenständige Integration oder Verwendung mit OpenID

In diesem Implementierungsleitfaden werden architektonische Überlegungen und Konfigurationsschritte für die Bereitstellung von Virtual Waiting Room AWS in der Amazon Web Services (AWS) Cloud beschrieben. Es enthält Links zu AWS CloudFormationVorlagen, mit denen die für die Bereitstellung dieser Lösung erforderlichen AWS Dienste gestartet und konfiguriert werden können. Dabei werden AWS bewährte Methoden für Sicherheit und Verfügbarkeit verwendet.

Der Leitfaden richtet sich an IT-Architekten, Entwickler, DevOps Mitarbeiter, Datenanalysten und Marketingtechnologieexperten, die über praktische Erfahrung mit der Architektur in der AWS Cloud verfügen.

Kosten

Sie sind für die Kosten der AWS Dienste verantwortlich, die beim Betrieb dieser Lösung in Anspruch genommen werden. Zum jetzigen Zeitpunkt belaufen sich die Kosten für den Betrieb dieser Lösung mit den Standardeinstellungen in der Region USA Ost (Nord-Virginia) auf etwa 10,00 USD/Tag pro Stack zuzüglich Gebühren für API-Anfragen und Datenverkehr im Verhältnis zur Größe des Ereignisses.

Die täglichen Kosten für die Wartung der Lösung ohne Zwischenfälle

AWS Service nicht zulässig	Anfragen/Zeit	Kosten [USD]
Amazon API Gateway	0	0,00$
Amazon CloudFront	0	0,00$
Amazon CloudWatch	0	0,00$
Amazon-DynamoDB	0	0,00$
Amazon ElastiCache	Knotenstunden berechnen (Redis)	~6,00 $
AWS Lambda	Kostenloses Kontingent*	0,00$
AWS Secrets Manager	Kostenloses Kontingent*	0,00$
Amazon-Simple-Storage-Service (Amazon-S3)	Kostenloses Kontingent*	0,00$
Amazon Virtual Private Cloud (Amazon VPC)	Stunden der VPC-Endpunkte Öffnungszeiten des NAT-Gateways	~5,00 $
INSGESAMT:		~11,00 $

*Der Kostenvoranschlag basiert auf einer sauberen Umgebung. Wenn Sie diesen AWS-Service außerhalb dieser Lösung verwenden, überschreiten Sie möglicherweise das Kontingent für das kostenlose Kontingent.

Die folgenden Tabellen zeigen die geschätzten Kosten für einen Warteraum mit 50.000 Benutzern und 100.000 Benutzern bei einer Veranstaltungsdauer von 2 bis 4 Stunden mit 500 Benutzern/Sekunde eingehenden und 1.000 Benutzern/Minute ausgehenden Ereignissen. Die Preise sind freibleibend. Vollständige Informationen finden Sie auf der Webseite mit den Preisen für die einzelnen in dieser Lösung verwendeten Dienste. AWS

Geschätzte Kosten für 50.000 Nutzer im Wartezimmer während einer zweistündigen Veranstaltung

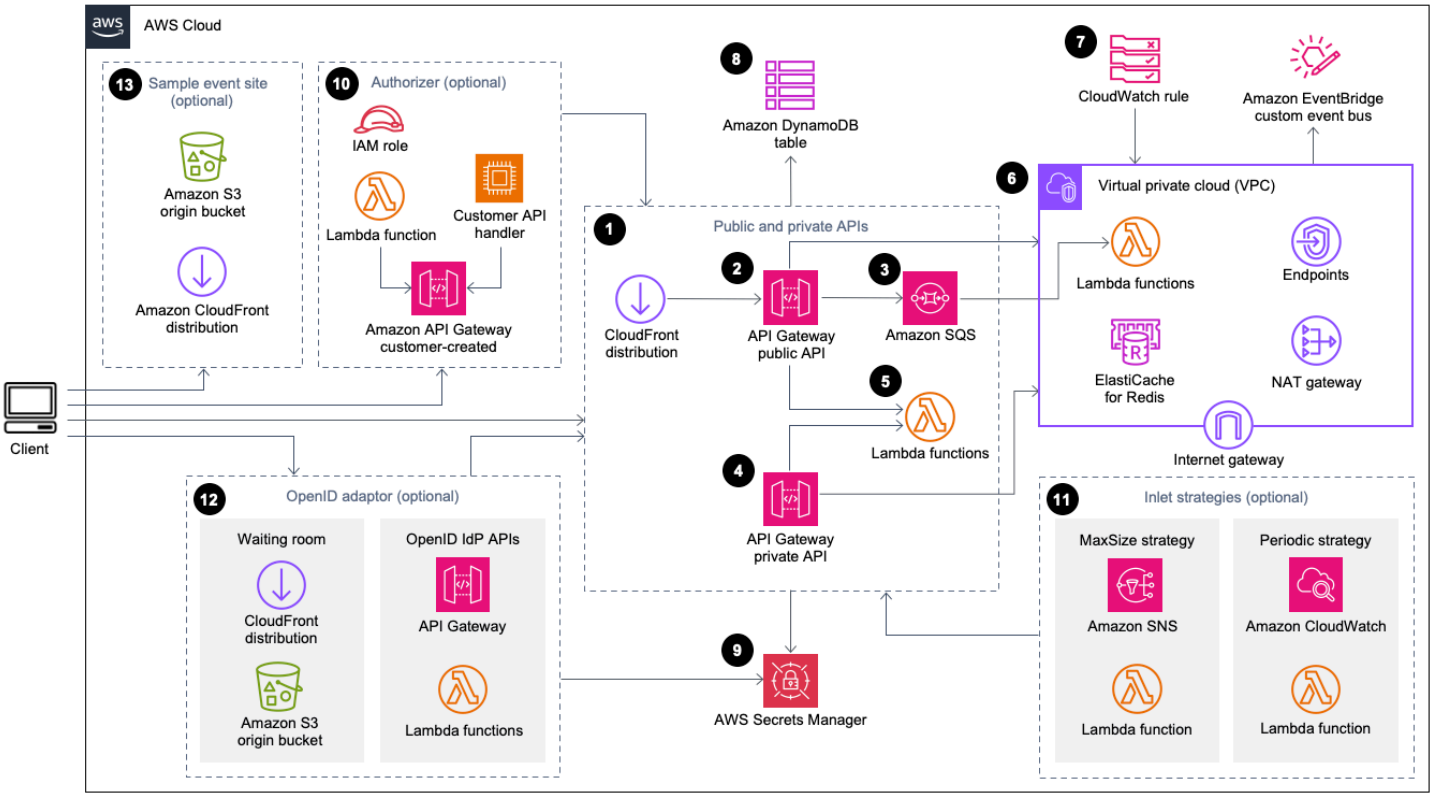
AWS Service nicht zulässig	Dimensions (Abmessungen)	Kosten [USD]
Amazon API Gateway	Anforderungen	2,00$
CloudFront	Anfragen, Bandbreite	75,00$
CloudWatch	Metriken, Alarme, Speicher	1,00$
CloudWatch Amazon-Veranstaltungen	Ereignisse	1,00$
DynamoDB	Lese-/Schreibeinheiten, Speicher	1,00$
ElastiCache	Knotenstunden	8,00$
Lambda	Anfragen, Rechenzeit	1,00$
AWS Secrets Manager	Geheimnisse, Anfragen	1,00$
Amazon S3	Anfragen, Aufbewahrung	1,00$
Amazon VPC	Datenübertragung, Endpunktzeit	2,00$
INSGESAMT		94,00$

Geschätzte Kosten für 100.000 Nutzer im Wartezimmer während einer zweistündigen Veranstaltung

AWS Service nicht zulässig	Dimensions (Abmessungen)	Kosten [USD]
Amazon API Gateway	Anforderungen	4,00$
CloudFront	Anfragen, Bandbreite	296,00$
CloudWatch	Metriken, Alarme, Speicher	1,00$
CloudWatch Ereignisse	Ereignisse	1,00$
DynamoDB	Lese-/Schreibeinheiten, Speicher	4,00$
ElastiCache	Knotenstunden	32,00$
Lambda	Anfragen, Rechenzeit	1,00$
AWS Secrets Manager	Geheimnisse, Anfragen	1,00$
Amazon-Simple-Queue-Service (Amazon SQS)	Anforderungen	1,00$
Amazon S3	Anfragen, Aufbewahrung	1,00$
Amazon VPC	Datenübertragung, Endpunktzeit	6,00$
INSGESAMT		348,00$

Übersicht über die Architektur

Durch die Bereitstellung dieser Lösung mit den erforderlichen und optionalen Vorlagen unter Verwendung von Standardparametern wird die folgende Umgebung in der AWS Cloud erstellt.



Virtuelles Wartezimmer zur AWS Architektur

Die AWS CloudFormation Vorlagen stellen die folgende Infrastruktur bereit:

1. Eine CloudFrontAmazon-Distribution zur Durchführung öffentlicher API Anrufe für den Kunden.
2. Öffentliche API Ressourcen von Amazon API Gateway zur Bearbeitung von Warteschlangenanfragen aus dem virtuellen Warteraum, zur Nachverfolgung der Warteschlangenposition und zur Unterstützung der Validierung von Token, die den Zugriff auf die Zielwebsite ermöglichen.
3. Eine Amazon Simple Queue Service (AmazonSQS) -Warteschlange zur Regulierung des Datenverkehrs zu der AWS LambdaFunktion, die die Warteschlangennachrichten verarbeitet. Anstatt die Lambda-Funktion für jede Anfrage aufzurufen, bündelt die SQS Warteschlange die eingehenden Anfragen.
4. APIPrivate API Ressourcen als Gateway zur Unterstützung administrativer Funktionen nutzen

5. Lambda-Funktionen validieren und verarbeiten öffentliche und private API Anfragen und geben die entsprechenden Antworten zurück.
6. Amazon Virtual Private Cloud (VPC) zum Hosten der Lambda-Funktionen, die direkt mit dem Elasticache (Redis OSS) -Cluster interagieren. VPCEndpunkte ermöglichen es Lambda-Funktionen in derVPC, mit Diensten innerhalb der Lösung zu kommunizieren. Darüber hinaus ermöglicht NAT das Gateway Lambda-Funktionen, CloudFront Endpunkte VPC zu verbinden und den Cache nach Bedarf ungültig zu machen.
7. Eine CloudWatchAmazon-Regel zum Aufrufen einer Lambda-Funktion, die mit einem benutzerdefinierten EventBridgeAmazon-Bus zusammenarbeitet, um regelmäßig Status-Updates zu senden.
8. Amazon DynamoDB-Tabellen zum Speichern von Token-, Warteschlangenposition- und Bereitstellungszählerdaten.
9. AWS Secrets Manager zum Speichern von Schlüsseln für Token-Operationen und andere sensible Daten.
10. (Optional) Authorizer-Komponente, bestehend aus einer AWS Identity and Access Management(IAM) -Rolle und einer Lambda-Autorisierungsfunktion zur Verwendung mit Gateway. API
11. (Optional) Amazon Simple Notification Service (AmazonSNS) und Lambda-Funktionen zur Unterstützung von zwei Eingangsstrategien. CloudWatch
12. (Optional) OpenID-Adapterkomponente mit API Gateway- und Lambda-Funktionen, die es einem OpenID-Anbieter ermöglicht, Benutzer auf Ihrer Website zu authentifizieren. CloudFront Verteilung mit einem Amazon Simple Storage Service (Amazon S3) -Bucket für die Wartezimmerseite für diese Komponente.
13. (Optional) CloudFront Verteilung mit Amazon S3 S3-Origin-Bucket für die Muster-Webanwendung für Wartezimmer.

So funktioniert die Lösung

In diesem Abschnitt werden die Schritte eines Workflows für AWS virtuelle Wartezimmer auf allgemeiner Ebene beschrieben. Einzelheiten zum Erstellen, Anpassen und Integrieren eines Warteraums GitHub für Ihre Website finden Sie im Entwicklerhandbuch unter.

Die Öffentlichkeit des Wartezimmers API kann sich hinter der Sicherheitskontrolle Ihres Standorts befinden oder er kann ohne Genehmigung zugänglich sein. Je nachdem, welchen Ansatz Sie für die Integration des Warteraums in die Website verwenden, muss sich der Benutzer möglicherweise zuerst auf der Website authentifizieren, bevor er zum Wartezimmer navigieren und sich eine Position in der Warteschlange sichern darf.

Die Client-Software muss über die Event-ID verfügen, um den Warteraum betreten und andere Anfragen stellen zu können. Eine Event-ID ist eine eindeutige ID, die für die meisten Anfragen an öffentliche und private Nutzer erforderlich istAPIs. Die Event-ID wird während der Installation des API Core-Stacks festgelegt. Während des Betriebs kann die Event-ID als URL Parameter oder Cookie über die Warteraum-Seite bereitgestellt werden. Sie kann als Teil von Authentifizierungstoken-Ansprüchen bereitgestellt werden oder sie kann über einen anderen Datenpfad an die Clients verteilt werden.

Es gibt Fälle, in denen der Client sowohl die Event-ID als auch die Anforderungs-ID benötigt, um bestimmte API Anrufe tätigen zu können. Die Anfrage-ID ist eine eindeutige ID, die vom Wartezimmer ausgestellt wird und einen bestimmten Kunden in der Warteschlange repräsentiert.

Die folgenden Schritte beschreiben den Ablauf von API Anfragen für den Eintritt in die Warteschlange, das Warten auf den Fortgang der Warteschlange und das Verlassen des Warteraums mit einem Zugriffstoken für die Website.

Der Benutzer betritt den Warteraum:

1. Dem Benutzer wird ein Bildschirm oder eine Seite angezeigt, die den Eingangspunkt für den Warteraum darstellt. Sie entscheiden sich dafür, in die Warteschlange einzutreten, und die Client-Software (Browser, Mobilgerät, Gerät) ruft die `assign_queue_num` Öffentlichkeit API auf, um eine Position in der Warteschlange anzufordern.
2. Die API Anfrage wird sofort von API Gateway an die SQS Amazon-Warteschlange übermittelt.
3. Der `assign_queue_num` API Anruf kehrt zurück, wenn die Anfrage in die Warteschlange gestellt wird. Der Client erhält eine eindeutige Anforderungs-ID, die später verwendet werden kann, um die Warteschlangenposition, die Uhrzeit der Anfrage und ein Zugriffstoken abzurufen.

4. Die `AssignQueueNum` Lambda-Funktion empfängt Stapel von bis zu zehn Anfragen aus der SQS Warteschlange. Der Lambda-Service fächert Aufrufe auf, um mehrere Batches von Anfragen zu verarbeiten.
5. Die `AssignQueueNum` Lambda-Funktion validiert jede Nachricht in ihrem Batch, erhöht den Warteschlangenzähler in Elasticache (RedisOSS) und speichert jede Anfrage in Elasticache (RedisOSS) mit der zugehörigen Warteschlangenposition.
6. Jede Nachricht wird gelöscht, wenn sie erfolgreich verarbeitet wurde. Nachrichten, bei denen ein Fehler aufgetreten ist, werden in einem späteren Batch einmal erneut verarbeitet. Nach einem zweiten Ausfall werden sie an eine dead-letter-queue CloudWatchAlarmanlage gesendet.
7. Der Client kann mit der Abfrage beginnen, `queue_num` API nachdem er die Anforderungs-ID aus dem `assign_queue_num` Anruf erhalten hat. Der Client sendet die Ereignis-ID und die Anforderungs-ID an den `queue_num` API und erhält eine numerische Warteschlangenposition oder eine Antwort, die angibt, dass die Anfrage noch nicht bearbeitet wurde. Bei großen Ereignissen muss der Client diesen Anruf möglicherweise mehrmals tätigen. Die `GetQueueNum` Lambda-Funktion wird von API Gateway aufgerufen und gibt die numerische Position des Clients in der Warteschlange von DynamoDB zurück.

Der Benutzer wartet im Wartezimmer:

8. Sobald der Client seine Position in der Warteschlange gefunden hat, kann er in regelmäßigen Abständen mit der `serving_num` API Abfrage beginnen. Der `serving_num` API wird mit der Event-ID aufgerufen und gibt die aktuelle Bereitstellungsposition der Warteschlange zurück. Die Antwort von `serving_num` API teilt dem Kunden mit, wann er vom Wartezimmer zum eigentlichen Zielort wechseln kann, an dem die endgültige Transaktion stattfinden kann. Die `GetServingNum` Lambda-Funktion gibt die aktuelle Servierposition des Wartezimmers zurück.
9. Wenn die Bereitstellungsposition gleich oder größer als die Warteschlangenposition (Anfrage) des Clients ist, kann der Client ein JSON Web-Token (JWT) von der Öffentlichkeit API anfordern. Das Token kann zusammen mit der Ziel-Site verwendet werden, um die Transaktion abzuschließen. Das `generate_token` API wird mit der Event-ID und der Anforderungs-ID aufgerufen. APIGateway ruft die `GenerateToken` Lambda-Funktion mit den Parametern auf.
10. Die `GenerateToken` Lambda-Funktion validiert die Anfrage und prüft, ob dieses Token zuvor generiert wurde. Die Lambda-Funktion fragt die DynamoDB-Tabelle nach einem passenden Token ab. Wenn dieses Token gefunden wird, wird es an den Aufrufer zurückgegeben und nicht neu generiert. Dieser Prozess verhindert, dass eine einzelne Anforderungs-ID verwendet wird, um mehrere unterschiedliche Token mit neuen Ablaufzeiten zu generieren.

11. Wenn das Token nicht in DynamoDB gefunden wird, ruft die Lambda-Funktion Schlüssel zur Erstellung des Tokens ab und speichert das Token in DynamoDB mit der Event-ID und der Anforderungs-ID des Clients. Die Lambda-Funktion schreibt ein Ereignis, um EventBridge zu signalisieren, dass ein neues Token generiert wurde. Die Lambda-Funktion erhöht einen Elasticache-Zähler (RedisOSS), der die Anzahl der für das Ereignis generierten Token verfolgt.
12. Wenn aktiviert `queue_pos_expiry` ist, kann der Client die verbleibende Zeit vor ihrem Ablauf abfragen, indem er den aufruft `queue_pos_expiry`API, der die `GetQueuePositionExpiryTime` Lambda-Funktion aufruft.

Der Benutzer verlässt das Wartezimmer:

13. Wenn der Client sein Token erhält, betritt er die Ziel-Site, um mit der Transaktion zu beginnen. Je nachdem, wie Ihre Infrastruktur eine Integration mit unterstütztJWT, muss der Client das Token möglicherweise in einem Anforderungsheader, einem Cookie oder auf andere Weise präsentieren. Der Autorisierer für API Gateway kann verwendet werden, um das in der Anfrage eines Kunden enthaltene Token zu validieren. Alle kommerziellen oder Open-Source-Bibliotheken für die Validierung und Verwaltung JWTs können mit Virtual Waiting Room auf Tokens verwendet werden. AWS Wenn das Token gültig ist, darf der Kunde seine Transaktion fortsetzen.
14. Nachdem der Client seine Transaktion abgeschlossen hat, API wird ein privater Befehl aufgerufen, um den Status des Kunden-Tokens zu aktualisieren. Dieser Vorgang wird in DynamoDB abgeschlossen.

Ablauf der Warteschlangenposition:

15. Wenn diese Funktion aktiviert ist, kann anhand der Anforderungs-ID, die einer bestimmten Warteschlangenposition entspricht, nur für ein bestimmtes Zeitintervall ein Token generiert werden.

Erhöhen Sie den Bereitstellungszähler bei Ablauf der Warteschlangenposition:

16. Wenn diese Funktion aktiviert ist, wird der Leistungszähler automatisch auf der Grundlage abgelaufener Warteschlangenpositionen erhöht, für die keine Tokens generiert werden konnten.

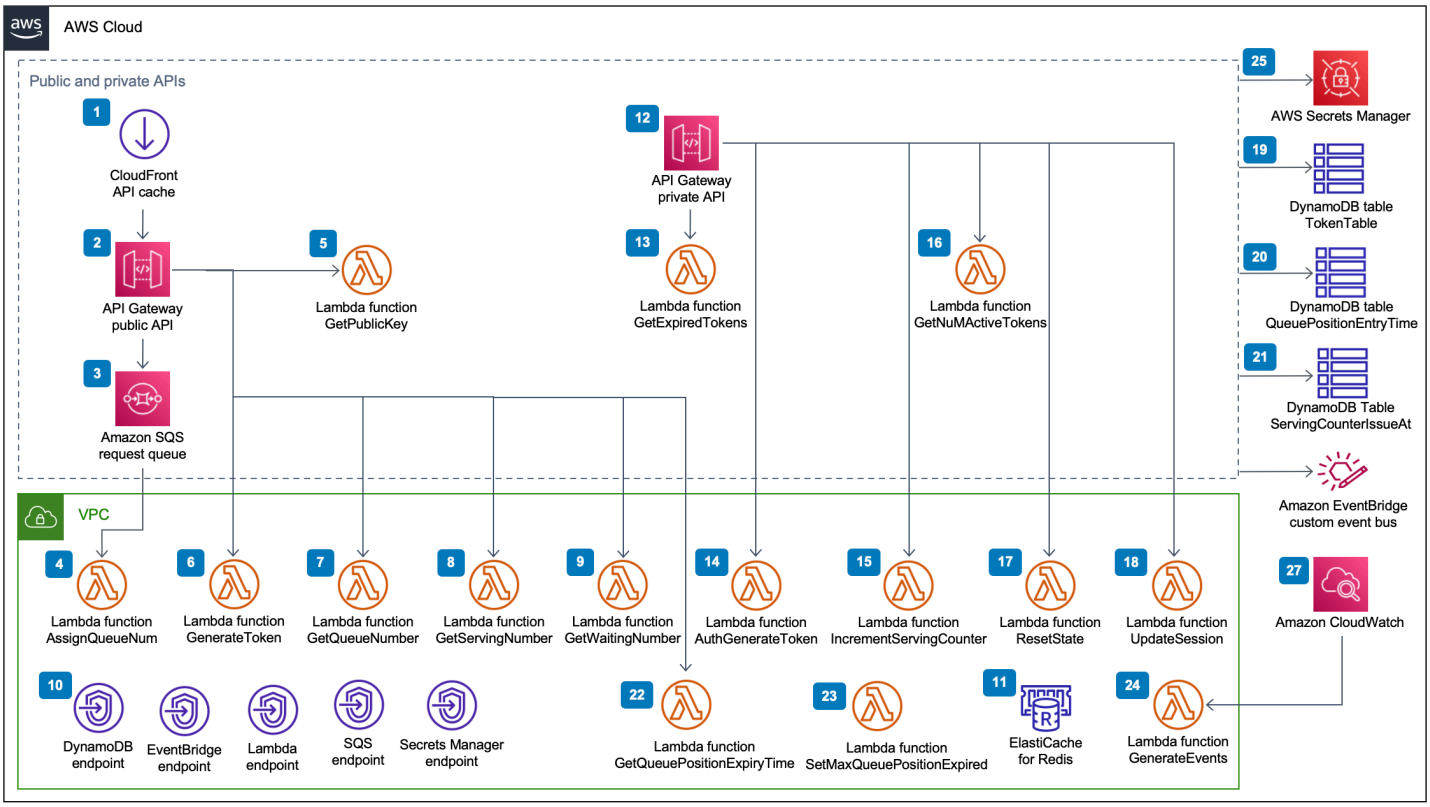
Komponenten der Lösung

Wartezimmer, öffentlich und privat APIs

Der Hauptzweck der Virtual Waiting Room AWS On-Lösung besteht darin, die Generierung von JSON Web-Tokens (JWT) für Clients auf kontrollierte Weise zu kontrollieren, um zu verhindern, dass neue Benutzer die Zielwebsite überlasten. JWTsSie können für den Schutz von Websites verwendet werden, indem der Zugriff auf Webseiten verhindert wird, bis das Wartezimmer-Token abgerufen wurde, und auch für die API Zugriffsautorisierung.

Das Core-Template installiert ein öffentliches API und ein privates (IAM-autorisiertes) Template, das für die meisten virtuellen Wartezimmer während des AWS Betriebs API verwendet wird. Die öffentliche Version API ist mit einer CloudFront Distribution mit mehreren Caching-Richtlinien konfiguriert, die auf dem API Pfad basieren. Eine DynamoDB-Tabelle und ein EventBridge Event-Bus werden erstellt. Die Vorlage fügt eine neue VPC mit zwei Availability Zones (AZs), einem Elasticache-Cluster (RedisOSS) in beiden AZs und mehreren Lambda-Funktionen hinzu. Lambda-Funktionen, die mit Elasticache (RedisOSS) interagieren, verfügen über Netzwerkschnittstellen innerhalb von, VPC und alle anderen Lambda-Funktionen verfügen über Standard-Netzwerkkonnektivität. Der Kern APIs ist die unterste Interaktionsebene mit der Lösung. Andere Lambda-Funktionen, die Amazon Elastic Compute Cloud (AmazonEC2) -Instance und Container können als Erweiterungen fungieren und den Kern aufrufen, APIs um Warteräume einzurichten, den Eingangsverkehr zu kontrollieren und auf Ereignisse zu reagieren, die von der Lösung generiert werden.

Darüber hinaus erzeugt der Core-Stack einen Alarm für alle seine Lambda-Funktionsfehler und Drosselungsbedingungen sowie Alarme für jede API Gateway-Bereitstellung für die 4XX- und 5XX-Statuscodes.



Virtueller Warteraum auf AWS öffentlicher und privater APIs Komponente

1. CloudFront Die Verteilung führt öffentliche API Anrufe für den Kunden durch und speichert die Ergebnisse gegebenenfalls im Cache.
2. Amazon API Gateway Public API verarbeitet Warteschlangenanfragen aus dem virtuellen Warteraum, verfolgt die Warteschlangenposition und unterstützt die Validierung von Token, die den Zugriff auf die Zielwebsite ermöglichen.
3. SQSDie Warteschlange reguliert den Verkehr zu der AWS Lambda Funktion, die die Warteschlangennachrichten verarbeitet.
4. Die `AssignQueueNum` Lambda-Funktion validiert jede Nachricht in ihrem empfangenen Batch, erhöht den Warteschlangenzähler in Elasticache (RedisOSS) und speichert jede Anfrage in Elasticache (RedisOSS) mit der zugehörigen Warteschlangenposition.
5. Die `GetPublicKey` Lambda-Funktion ruft den Wert des öffentlichen Schlüssels aus Secrets Manager ab.
6. Die `GenerateToken` Lambda-Funktion generiert eine JWT für eine gültige Anfrage, die ihre Transaktion auf der Ziel-Site abschließen durfte. Sie schreibt ein Ereignis in den

benutzerdefinierten Event-Bus des Wartezimmers, dass ein Token generiert wurde. Wenn zuvor ein Token für diese Anfrage generiert wurde, wird kein neues Token generiert.

7. Die `GetQueueNumber` Lambda-Funktion ruft die numerische Position des Clients in der Warteschlange von Elasticache (Redis) ab und gibt sie zurück. OSS
8. Die `GetServingNumber` Lambda-Funktion ruft die Nummer, die derzeit vom Wartezimmer bedient wird, von Elasticache (Redis) ab und gibt sie zurück. OSS
9. Die `GetWaitingNum` Lambda-Funktion gibt die Nummer zurück, die sich derzeit im Wartezimmer befindet und für die noch kein Token ausgestellt wurde.
10. VPCEndpunkte ermöglichen es Lambda-Funktionen in derVPC, mit Diensten innerhalb der Lösung zu kommunizieren.
11. Der Elasticache-Cluster (RedisOSS) speichert alle Anfragen zum Betreten des Warteraums mit einer gültigen Event-ID. Es speichert auch mehrere Zähler wie die Anzahl der Anfragen in der Warteschlange, die Anzahl der aktuell bearbeiteten Anfragen, die Anzahl der generierten Token, die Anzahl der abgeschlossenen Sitzungen und die Anzahl der abgebrochenen Sitzungen.
12. APIGateway für private API Ressourcen zur Unterstützung administrativer Funktionen. Die privaten APIs sind AWS IAM authentifiziert.
13. Die `GetExpiredTokens` Lambda-Funktion gibt eine Liste von Anfragen IDs mit abgelaufenen Tokens zurück.
14. Die `AuthGenerateToken` Lambda-Funktion generiert ein Token für eine gültige Anfrage, die ihre Transaktion auf der Ziel-Site abschließen durfte. Der Aussteller und die Gültigkeitsdauer eines Tokens, die ursprünglich während der Core-Stack-Bereitstellung festgelegt wurden, können außer Kraft gesetzt werden. Es schreibt ein Ereignis in den benutzerdefinierten Event-Bus des Warteraums, dass ein Token generiert wurde. Wenn für diese Anfrage bereits ein Token generiert wurde, wird kein neues Token generiert.
15. Die `IncrementServingCounter` Lambda-Funktion erhöht den in Elasticache (RedisOSS) gespeicherten Servierzähler des Wartezimmers, wenn er um einen Wert erhöht wird.
16. Die `GetNumActiveTokens` Lambda-Funktion fragt DynamoDB nach der Anzahl der Token ab, die noch nicht abgelaufen sind, nicht zum Abschluss der Transaktion verwendet wurden und nicht als verlassen markiert wurden.
17. Die `ResetState` Lambda-Funktion setzt alle in Elasticache (Redis) gespeicherten Zähler zurück. OSS Außerdem werden die ,- und `ServingCounterIssuedAt` DynamoDB-Tabellen `TokenTable` gelöscht und neu erstellt. `QueuePositionEntryTime` Darüber hinaus führt es eine Cache-Invalidierung durch. CloudFront

18 Die `UpdateSession` Lambda-Funktion aktualisiert den Status einer Sitzung (Token), die in der `TokenTable` DynamoDB-Tabelle gespeichert ist. Der Sitzungsstatus wird mit einer Ganzzahl angegeben. Sitzungen, die auf den Status gesetzt wurden, bedeuten abgeschlossen und `1` `-1` bedeutet, dass sie aufgegeben wurden. Es schreibt ein Ereignis in den benutzerdefinierten Event-Bus des Wartezimmers, dass eine Sitzung aktualisiert wurde.

19 In der `TokenTable` DynamoDB-Tabelle werden Token-Daten gespeichert.

20 In der `QueuePositionEntryTime` DynamoDB-Tabelle werden Daten zur Warteschlangenposition und zur Eintrittszeit gespeichert.

21 In der `ServingCounterIssuedAt` DynamoDB-Tabelle werden Aktualisierungen des Serving-Counter gespeichert.

22 Die `GetQueuePositionExpireTime` Lambda-Funktion wird aufgerufen, wenn der Client die Ablaufzeit der verbleibenden Warteschlangenposition anfordert.

23 Die `SetMaxQueuePositionExpired` Lambda-Funktion legt die maximale Warteschlangenposition fest, die abgelaufen ist, entsprechend den `ServingCounterIssuedAt` Tabellenwerten. Sie wird jede Minute ausgeführt, wenn der `IncrSvcOnQueuePositionExpiry` Parameter `true` während der Core-Stack-Bereitstellung auf eingestellt ist.

24 Die `GenerateEvents` Lambda-Funktion schreibt verschiedene Metriken für den Warteraum in den benutzerdefinierten Event-Bus des Wartezimmers. Sie wird jede Minute ausgeführt, wenn der Parameter Enable Events Generation `true` während der Core-Stack-Bereitstellung auf gesetzt ist.

25 AWS Secrets Manager speichert Schlüssel für Token-Operationen und andere sensible Daten.

26 Der EventBridge benutzerdefinierte Amazon Event Bus empfängt jedes Mal ein Ereignis, wenn ein Token generiert und eine Sitzung in der `TokenTable` DynamoDB-Tabelle aktualisiert wird. Es empfängt auch Ereignisse, wenn der Serving-Zähler im `SetMaxQueuePositionExpired` Lambda bewegt wird. Es wird mit verschiedenen Metriken für den Warteraum beschrieben, sofern es während der Core-Stack-Bereitstellung aktiviert wird.

27 Die CloudWatch Amazon-Ereignisregel wird erstellt, wenn der Parameter Enable Events Generation während der Core-Stack-Bereitstellung auf true gesetzt ist. Diese Ereignisregel initiiert jede Minute die `GenerateEvents` Lambda-Funktion.

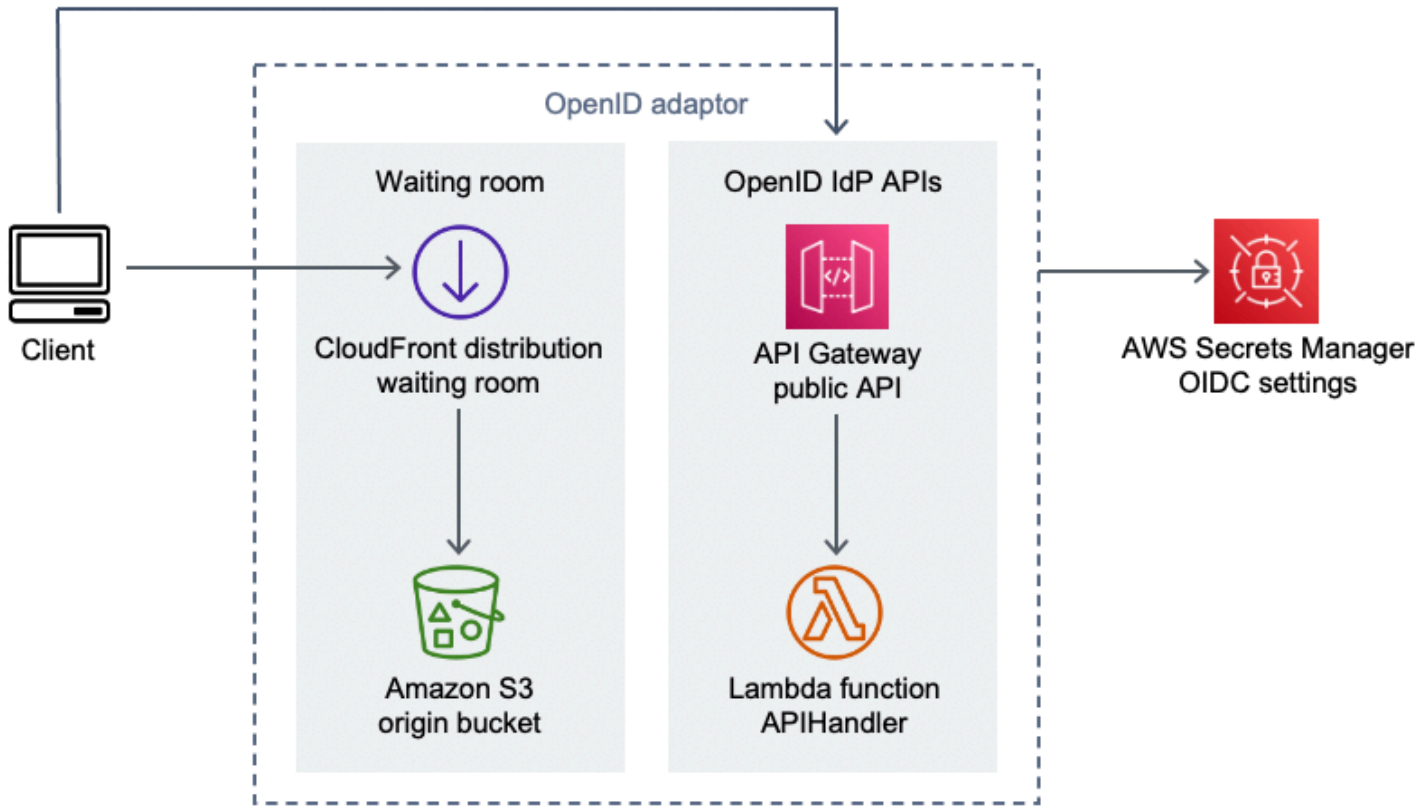
Authorizers

Die Lösung umfasst einen API Gateway Lambda Authorizer-Stack. Der Stack besteht aus einer IAM Rolle und einer Lambda-Funktion. Die `APIGatewayAuthorizer` Lambda-Funktion ist ein Autorisierer für API Gateway, der die Signatur und die Ansprüche eines vom Virtual Waiting Room

ausgegebenen Tokens validieren kann. AWS API Die im Stack enthaltene Lambda-Funktion kann zum Schutz der Cloud verwendet werden, APIs bis ein Benutzer den Wartebereich durchquert hat und ein Zugriffstoken erhält. Der Autorisierer ruft den öffentlichen Schlüssel und die Konfiguration automatisch aus dem Core ab und speichert sie im Cache, um das Token zu verifizieren. API Es kann ohne Änderungen verwendet und in jeder AWS Region installiert werden, die dies unterstützt. AWS Lambda

OpenID-Adapter

Der OpenID-Adapter-Stack stellt ein API Gateway und Lambda-Funktionen bereit, die als OpenID-Identitätsanbieter fungieren. Der OpenID-Adapter bietet ein OIDC kompatibles SetAPIs, das mit vorhandener Webhosting-Software, die OIDC Identitätsanbieter unterstützt, wie AWS Elastic Load Balancers WordPress, oder als föderierter Identitätsanbieter für Amazon Cognito oder ähnliche Dienste verwendet werden kann. Der Adapter ermöglicht es einem Kunden, den Warteraum im Authn/AuthZ-Flow zu nutzen, wenn er off-the-shelf Webhosting-Software mit eingeschränkten Integrationsoptionen verwendet. Der Stack installiert auch eine CloudFront Distribution mit einem Amazon S3 S3-Bucket als Ursprung und einem weiteren S3-Bucket für die Protokollierung von Anfragen. Der OpenID-Adapter stellt eine Beispielseite für einen Warteraum bereit, die der im Wartezimmer-Beispielstapel bereitgestellten Seite ähnelt, aber für einen OpenID-Authentifizierungsablauf konzipiert ist. Bei der Authentifizierung müssen Sie sich eine Position in der Warteschlange im Wartezimmer sichern und warten, bis die Servierposition gleich oder größer als die Warteschlangenposition des Kunden ist. Die OpenID-Warteraum-Seite leitet zurück zur Ziel-Site, die die OpenID verwendet, API um die Token-Erfassung und die Sitzungskonfiguration für den Client abzuschließen. Die API Endpunkte dieser Lösung sind direkt der offiziellen OpenID Connect 1.0-Flow-Spezifikation zugeordnet. name-for-name Einzelheiten finden Sie unter OpenID Connect Core 1.0-Authentifizierung.



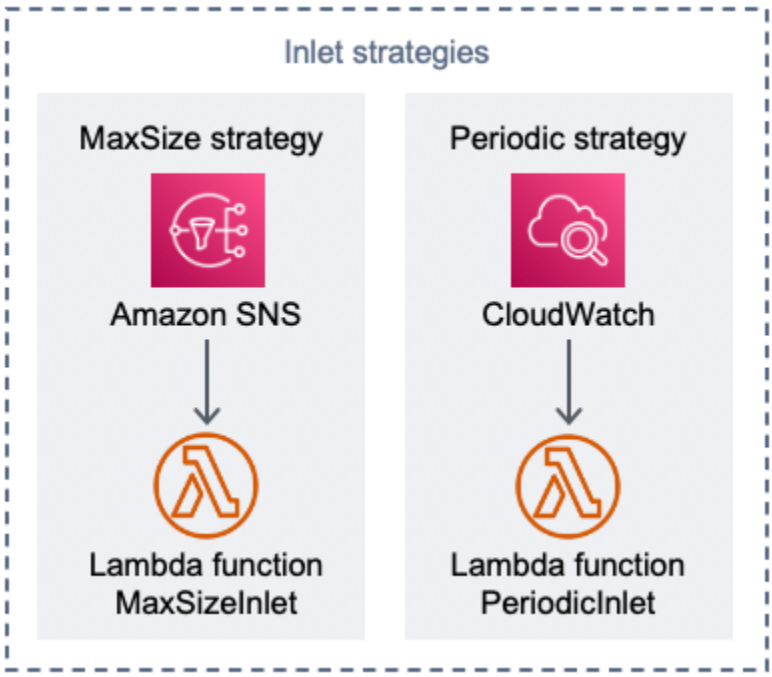
Virtueller Warteraum auf der AWS OpenID-Adapterkomponente

1. CloudFront Die Verteilung stellt dem Benutzer den Inhalt des S3-Buckets zur Verfügung.
2. Der S3-Bucket hostet Beispielseiten für Wartezimmer.
3. Amazon API Gateway API bietet eine Reihe OIDC kompatibler FunktionenAPIs, die mit vorhandener Webhosting-Software verwendet werden können, die die Lambda-Autorisierungsfunktion des OIDC Identitätsanbieters unterstützt.
4. Die `APIHandler` Lambda-Funktion verarbeitet Anfragen für alle API Gateway-Ressourcenpfade. Jedem API Pfad werden verschiedene Python-Funktionen innerhalb desselben Moduls zugeordnet. Beispielsweise wird der `/authorize` Ressourcenpfad in API Gateway `authorize()` innerhalb der Lambda-Funktion aufgerufen.
5. OIDCEinstellungen werden im Secrets Manager gespeichert.

Beispiele für Einlassstrategien

Inlet-Strategien legen fest, wann der Servierzähler der Lösung weiterentwickelt werden sollte, um mehr Benutzer am Zielstandort unterzubringen. Weitere konzeptionelle Informationen zu Strategien für den Zutritt in Wartezimmer finden Sie unter Überlegungen zum Design.

Die Lösung bietet zwei Strategien für den Probeneingang: MaxSizeund Periodisch.



Komponente für Strategien im virtuellen Wartezimmer am AWS Eingang

Strategie-Option „Max. Größe“:

1. Ein Kunde gibt eine SNS Amazon-Benachrichtigung aus, die die `MaxSizeInlet` Lambda-Funktion aufruft, um den Bereitstellungszähler basierend auf der Nachrichtennutzlast zu erhöhen.
2. Die `MaxSizeInlet` Lambda-Funktion erwartet den Empfang einer Nachricht, dass sie bestimmt, um wie viel der Serving-Zähler erhöht werden soll.

Strategieoption „Periodischer Eingang“:

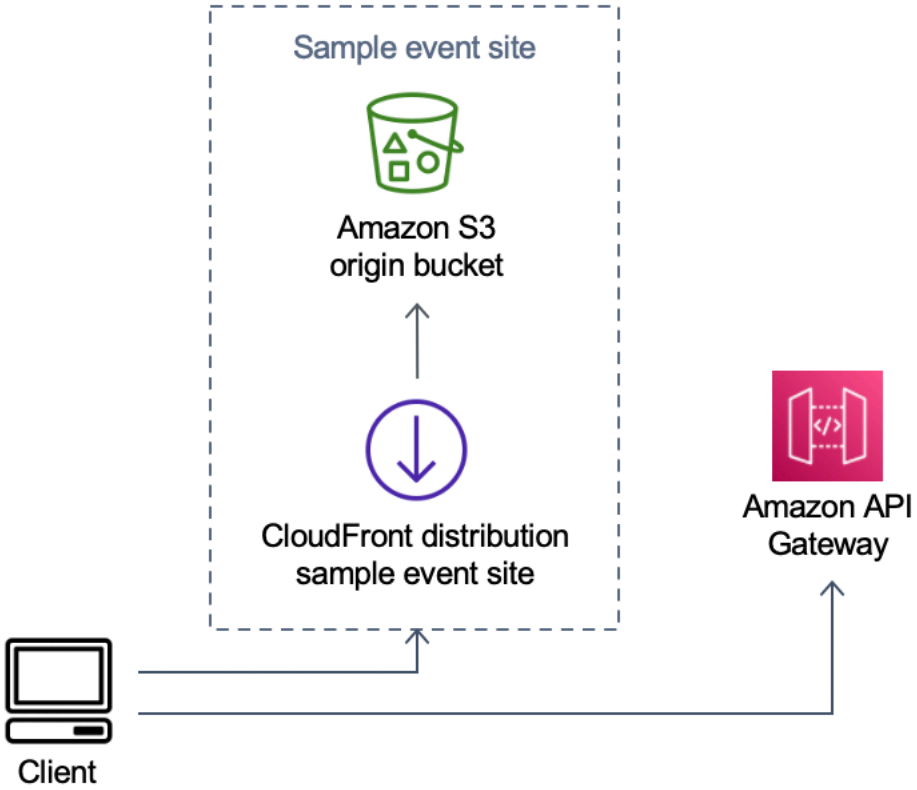
3. Eine CloudWatch Regel ruft jede Minute eine Lambda-Funktion auf, um den Servierzähler um eine feste Menge zu erhöhen.
4. Die `PeriodicInlet` Lambda-Funktion erhöht den Leistungszähler um die angegebene Größe, wenn die Zeit zwischen der angegebenen Start- und Endzeit liegt. Optional überprüft sie einen

CloudWatch Alarm und führt, falls der Alarm aktiv ist, die Erhöhung 0K durch, andernfalls überspringt sie sie.

Beispiel für ein Wartezimmer

Der Musterwarteraum lässt sich zusätzlich zum benutzerdefinierten Authorizer APIs in den öffentlichen und privaten Bereich integrieren, um zu demonstrieren, dass die end-to-end Wartezimmerlösung auf ein Minimum beschränkt ist. Die Hauptwebseite wird in einem S3-Bucket gespeichert und als Quelle für CloudFront verwendet. Sie führt den Benutzer durch die folgenden Schritte:

1. Stellen Sie sich im Wartezimmer in die Warteschlange, um die Website betreten zu können.
2. Ermitteln Sie die Position des Kunden in der Schlange.
3. Besorgen Sie sich die Servierposition des Wartezimmers.
4. Besorgen Sie sich ein Token-Set, sobald die Servierposition der Position des Kunden entspricht oder größer ist.
5. Verwenden Sie das Token, um einen vom Lambda API geschützten Autorisierer aufzurufen.



Virtueller Warteraum auf einer AWS Beispielkomponente für eine Event-Website

1. Der S3-Bucket hostet den Beispielinhalt für den Warteraum und das Control Panel.
2. CloudFront Die Verteilung stellt dem Benutzer den Inhalt des S3-Buckets zur Verfügung.
3. Beispiel für eine API Gateway-Bereitstellung mit einkaufsähnlichen Ressourcenpfaden wie und. `/search` `/checkout` Dies API wird vom Stack installiert und mit dem Token-Authorizer konfiguriert. Es ist als Beispiel für eine einfache Möglichkeit gedacht, einen API mit dem Wartezimmer zu schützen. Anfragen, die ein gültiges Token vorlegen, werden an das Lambda weitergeleitet, andernfalls wird ein Fehler zurückgegeben. Es gibt keine API andere Funktionalität als die Antwort der angehängten Lambda-Funktion.

Sicherheit

Wenn Sie Systeme auf der AWS Infrastruktur aufbauen, teilen Sie sich die Sicherheitsverantwortung zwischen Ihnen und AWS. Dieses gemeinsame Modell reduziert Ihren betrieblichen Aufwand, da AWS die Komponenten wie das Host-Betriebssystem, die Virtualisierungsebene und die physische Sicherheit der Einrichtungen, in denen die Services ausgeführt werden, betrieben, verwaltet und kontrolliert werden. Weitere Informationen zur AWS Sicherheit finden Sie unter AWS Cloud Security.

Elasticache (RedisOSS) wird eine Netzwerkschnittstelle innerhalb des privaten Bereichs zugewiesen. VPC Den Lambda-Funktionen, die mit Elasticache (RedisOSS) interagieren, werden ebenfalls Netzwerkschnittstellen innerhalb eines zugewiesen. VPC Alle anderen Ressourcen verfügen über Netzwerkkonnektivität im gemeinsam genutzten AWS Netzwerkbereich. Lambda-Funktionen mit VPC Schnittstellen, die mit anderen AWS Diensten interagieren, verwenden VPC Endpunkte, um eine Verbindung zu diesen Diensten herzustellen.

Die öffentlichen und privaten Schlüssel, die für die Erstellung und Validierung von JSON Web-Token verwendet werden, werden bei der Bereitstellung generiert und in Secrets Manager gespeichert. Das für die Verbindung mit Elasticache (RedisOSS) verwendete Passwort wird ebenfalls bei der Bereitstellung generiert und in Secrets Manager gespeichert. Auf den privaten Schlüssel und das Elasticache-Passwort (RedisOSS) kann über keine Lösung zugegriffen werden. API

Der Zugriff auf die Öffentlichkeit API muss über erfolgen. CloudFront Die Lösung generiert einen API Schlüssel für API Gateway, der als Wert eines benutzerdefinierten Headers (`x-api-key`, in) verwendet wird CloudFront. CloudFront schließt diesen Header ein, wenn ursprüngliche Anfragen gestellt werden. Weitere Informationen finden Sie unter Hinzufügen von benutzerdefinierten Headern zu ursprünglichen Anfragen im Amazon CloudFront Developer Guide.

Private APIs sind so konfiguriert, dass für den Aufruf eine AWS IAM Autorisierung erforderlich ist. Die Lösung erstellt die `ProtectedAPIGroup` IAM Benutzergruppe mit den entsprechenden Berechtigungen zum Aufrufen von Private. APIs Ein IAM Benutzer, der dieser Gruppe hinzugefügt wurde, ist berechtigt, die private Gruppe aufzurufen. APIs

IAMRichtlinien, die in Rollen und Berechtigungen verwendet werden, die mit verschiedenen Ressourcen verknüpft sind, die von der Lösung erstellt wurden, gewähren nur die Berechtigungen, die für die Ausführung der erforderlichen Aufgaben erforderlich sind.

Für Ressourcen wie S3-Buckets, SQS Warteschlangen und SNS Themen, die von der Lösung generiert wurden, ist die Verschlüsselung im Ruhezustand und während der Übertragung, wo immer möglich, aktiviert.

Überwachen

Der API Core-Stack umfasst mehrere CloudWatch Alarme, die überwacht werden können, um Probleme zu erkennen, während die Lösung in Betrieb ist. Der Stack erzeugt einen Alarm für Lambda-Funktionsfehler und Drosselungsbedingungen und ändert den Status des Alarms von `OK` zu, `ALARM` wenn innerhalb einer Minute ein Fehler oder eine Drosselung auftritt.

Der Stack erstellt außerdem Alarme für jede API Gateway-Bereitstellung für die Statuscodes 4XX und 5XX. Der Alarm ändert seinen Status von `OK` zu, `ALARM` wenn API innerhalb einer Minute ein 4XX- oder 5XX-Statuscode zurückgegeben wird.

Diese Alarme kehren nach einer Minute in einen `OK` Zustand zurück, in dem es keine Fehler oder Drosselungen gibt.

IAMRollen

AWS Identity and Access Management (IAM) Rollen ermöglichen es Kunden, Diensten und Benutzern in der AWS Cloud detaillierte Zugriffsrichtlinien und -berechtigungen zuzuweisen. Diese Lösung erstellt IAM Rollen, die den AWS Lambda Funktionen der Lösung Zugriff gewähren, um regionale Ressourcen zu erstellen.

Amazon CloudFront

Die `virtual-waiting-room-on-aws.template` CloudFormation Vorlage, die den öffentlichen und privaten Kern APIs des Wartezimmers erstellt, stellt auch eine CloudFront Verteilung für die Öffentlichkeit API bereit. CloudFront speichert die Antworten der Öffentlichkeit API im Cache und reduziert so die Belastung von API Gateway und den Lambda-Funktionen, die ihre Arbeit ausführen.

Diese Lösung bietet auch eine optionale Mustervorlage für Wartezimmer, die eine einfache Webanwendung bereitstellt, die in einem Amazon Simple Storage Service (Amazon S3) -Bucket gehostet wird. Um die Latenz zu reduzieren und die Sicherheit zu verbessern, wird eine CloudFront Amazon-Distribution mit einer Ursprungszugriffsidentität bereitgestellt. Dabei handelt es sich um einen CloudFront Benutzer, der öffentlichen Zugriff auf die Inhalte des Website-Buckets der Lösung gewährt. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Amazon S3 S3-Inhalte mithilfe einer Origin-Zugriffsidentität im Amazon CloudFront Developer Guide.

Sicherheitsgruppen

Die in dieser Lösung erstellten VPCSicherheitsgruppen dienen dazu, den Netzwerkverkehr zum Elasticache (OSSRedis) zu kontrollieren und zu isolieren. Lambdas, die mit dem Elasticache (RedisOSS) kommunizieren müssen, werden derselben Sicherheitsgruppe zugeordnet wie die des Elasticache (Redis). OSS Wir empfehlen Ihnen, die Sicherheitsgruppen zu überprüfen und den Zugriff nach Bedarf weiter einzuschränken, sobald die Bereitstellung betriebsbereit ist.

Überlegungen zum Design

Optionen für die Bereitstellung

Wenn dies die erste Installation ist oder Sie sich nicht sicher sind, was Sie installieren sollen, stellen Sie die `virtual-waiting-room-on-aws-getting-started.template` verschachtelte CloudFormation Vorlage bereit, die den Kern, die Autorisierer und die Beispielvorlagen für Wartezimmer installiert. Dies bietet Ihnen einen minimalen Wartezeitraum mit einem einfachen Ablauf.

Unterstützte Protokolle

Die AWS Lösung Virtual Waiting Room on kann in Folgendes integriert werden:

- JSONBibliotheken und Tools zur Überprüfung von Web-Tokens
- Bestehende API Gateway-Bereitstellungen
- RESTAPIKunden
- OpenID-Kunden und Anbieter

Strategien für den Zutritt zum Wartezimmer

Inlet-Strategien beinhalten die Logik und die Daten, die erforderlich sind, um Kunden vom Wartezimmer zur Website zu bewegen. Eine Inlet-Strategie kann als Lambda-Funktion, Container, EC2 Amazon-Instance oder jede andere Rechenressource implementiert werden. Es muss sich nicht um eine Cloud-Ressource handeln, solange sie den Warteraum als öffentlich und privat APIs bezeichnen kann. Bei der Inlet-Strategie werden Ereignisse über den Wartebereich, die Website oder andere externe Indikatoren erfasst, anhand derer entschieden werden kann, wann mehr Kunden Tokens ausgeben lassen und die Website betreten können. Es gibt verschiedene Ansätze für Inlet-Strategien. Welchen Sie wählen, hängt von den Ihnen zur Verfügung stehenden Ressourcen und den Einschränkungen bei der Gestaltung der zu schützenden Website ab.

Die wichtigste Maßnahme der Inlet-Strategie besteht darin, das `increment_serving_num` Amazon API Gateway als privat zu bezeichnen, API wobei ein relativer Wert angegeben wird, der angibt, wie viele weitere Kunden die Site betreten können. In diesem Abschnitt werden zwei Beispiele für Inlet-Strategien beschrieben. Diese können unverändert oder kundenspezifisch verwendet werden, oder Sie können einen völlig anderen Ansatz verwenden.

MaxSize

Mithilfe der MaxSize Strategie wird die `MaxSizeInlet` Lambda-Funktion mit der maximalen Anzahl von Clients konfiguriert, die die Website gleichzeitig nutzen können. Dies ist ein fester Wert. Ein Kunde gibt eine SNS Amazon-Benachrichtigung aus, die die `MaxSizeInlet` Lambda-Funktion aufruft, um den Bereitstellungszähler basierend auf der Nachrichtennutzlast zu erhöhen. Die Quelle der SNS Nachricht kann aus einer beliebigen Quelle stammen, z. B. aus Code auf der Website oder aus einem Überwachungstool, das den Nutzungsgrad der Website beobachtet.

Die `MaxSizeInlet` Lambda-Funktion erwartet den Empfang einer Nachricht, die Folgendes beinhalten kann:

- `exited :`Anzahl der abgeschlossenen Transaktionen
- Liste der AnfragenIDs, die als abgeschlossen markiert werden sollen
- Liste der AnträgeIDs, die als aufgegeben markiert werden sollen

Anhand dieser Daten wird bestimmt, um wie viel der Servierzähler erhöht werden muss. Es kann vorkommen, dass aufgrund der aktuellen Anzahl von Clients keine zusätzliche Kapazität zur Erhöhung des Zählers zur Verfügung steht.

Regelmäßig

Bei Verwendung der periodischen Strategie ruft eine CloudWatch Regel jede Minute die `PeriodicInlet` Lambda-Funktion auf, um den Servierzähler um eine feste Menge zu erhöhen. Der periodische Eingang wird mit der Startzeit, der Endzeit und der Inkrementmenge des Ereignisses parametrisiert. Wahlweise überprüft diese Strategie auch einen CloudWatch Alarm. Wenn sich der Alarm im `OK` Status befindet, wird das Inkrement ausgeführt, andernfalls wird es übersprungen. Die Standortintegratoren können eine Nutzungsmetrik mit einem Alarm verbinden und diesen Alarm verwenden, um die regelmäßige Eingabe zu unterbrechen. Bei dieser Strategie wird die Bereitstellungsposition nur geändert, solange die aktuelle Uhrzeit zwischen der Start- und Endzeit liegt, und optional befindet sich der angegebene Alarm im `OK` Status.

Anpassung und Erweiterung der Lösung

Der Site-Administrator Ihres Unternehmens muss entscheiden, welche Integrationsmethoden für den Warteraum verwendet werden sollen. Es gibt zwei Optionen:

1. Grundlegende Integration direkt mithilfe von APIs API Gateway-Autorisierern.
2. OpenID-Integration über einen Identitätsanbieter.

Zusätzlich zu der oben genannten Integration müssen Sie möglicherweise die Domainnamenumleitung konfigurieren. Sie sind auch für die Bereitstellung einer maßgeschneiderten Webseite für Wartezimmer verantwortlich.

Die AWS Lösung Virtual Waiting Room on ist so konzipiert, dass sie durch zwei Mechanismen erweitert werden kann: EventBridge für die unidirektionale Benachrichtigung über Ereignisse und REST APIs für die bidirektionale Kommunikation.

Kontingente

Die primäre Skalierungsbeschränkung für Virtual Waiting Room on AWS ist die Lambda-Drosselgrenze für die installierte AWS Region. Bei der Installation in einem AWS Konto mit dem standardmäßigen Lambda-Kontingent für gleichzeitige Ausführung kann die Virtual Waiting Room AWS On-Lösung bis zu 500 Clients pro Sekunde verarbeiten, die eine Position in der Warteschlange anfordern. Die Rate von 500 Clients pro Sekunde basiert auf der Lösung, bei der alle gleichzeitigen Quotenlimits für alle Lambda-Funktionen exklusiv verfügbar sind. Wenn die Region im Konto mit anderen Lösungen geteilt wird, die Lambda-Funktionen aufrufen, sollte der virtuelle Warteraum auf der AWS Lösung mindestens 1.000 gleichzeitige Aufrufe verfügbar sein. Mithilfe von CloudWatch Kennzahlen können Sie die gleichzeitigen Lambda-Aufrufe in Ihrem Konto im Zeitverlauf grafisch darstellen und so eine Entscheidung treffen. Sie können die Service Quotas Quotas-Konsole verwenden, um Erhöhungen zu beantragen. Durch die Erhöhung des Lambda-Drosselllimits werden die monatlichen Kontogebühren nur dann erhöht, wenn tatsächlich zusätzliche Aufrufe erfolgen.

Erhöhen Sie Ihr Drosselllimit für jede weitere 500 Clients pro Sekunde um 1.000.

Eingehende Benutzer pro Sekunde werden erwartet	Empfohlenes Kontingent für gleichzeitige Ausführung
0-500	1.000 (Standard)
501-1.000	2.000
1.001-1.500	3,000

Lambda hat ein festes Burst-Limit von 3.000 gleichzeitigen Aufrufen. Weitere Informationen finden Sie unter Lambda-Funktionsskalierung. Der Client-Code sollte einige API Aufrufe erwarten und erneut versuchen, wenn ein Fehlercode zurückgegeben wird, der auf eine vorübergehende Drosselung hinweist. Der Beispielclient für Wartezimmer enthält diesen Code als Beispiel für das Entwerfen von Clients, die bei Ereignissen mit hoher Kapazität und hohen Auslastungsraten eingesetzt werden.

Diese Lösung ist auch mit reservierter und bereitgestellter Lambda-Parallelität mit benutzerdefinierten Konfigurationsschritten kompatibel. Einzelheiten finden Sie unter Managing Lambda Reserved Concurrency.

Die Obergrenze für Benutzer, die den Warteraum betreten, ein Token erhalten und mit einer Transaktion fortfahren können, ist durch die Obergrenze der Elasticache-Zähler (OSSRedis) begrenzt. Die Zähler werden für die Bereitstellungsposition im Wartezimmer und für die Erfassung des Gesamtstatus der Lösung verwendet. Die in Elasticache (RedisOSS) verwendeten Zähler haben eine Obergrenze von 9.223.372.036.854.775.807. Eine DynamoDB-Tabelle wird verwendet, um eine Kopie jedes Tokens zu speichern, das an einen Benutzer im Wartezimmer ausgegeben wurde. DynamoDB hat keine praktische Beschränkung für die Größe einer Tabelle.

Regionale Bereitstellungen

Die von dieser Lösung verwendeten Dienste werden in allen AWS Regionen unterstützt. Die aktuelle Verfügbarkeit von AWS Diensten nach Regionen finden Sie in der Liste der AWS regionalen Dienste.

AWS CloudFormation Vorlagen

Um die Bereitstellung zu automatisieren, verwendet diese Lösung die folgenden AWS CloudFormation Vorlagen, die Sie vor der Bereitstellung herunterladen können.

Wenn dies die erste Installation ist oder Sie sich nicht sicher sind, was Sie installieren sollen, stellen Sie die `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation Vorlage bereit, die den Kern, die Autorisierer und die Beispielvorlagen für den Wartezimmercode installiert. Auf diese Weise können Sie einen funktionierenden Warteraum mit einem einfachen Ablauf testen.

View template

virtual-waiting-room-on- aws-api-gateway-cw -logs-role.template: Verwenden Sie diese Vorlage, um API Gateway auf Kontoebene eine Standardrolle für ARN Logging-Berechtigungen hinzuzufügen. CloudWatch Einzelheiten darüber, ob Ihr Konto die Bereitstellung dieser Vorlage erfordert oder nicht, finden Sie unter Voraussetzungen.

View template

virtual-waiting-room-on- aws-getting-started .template: Verwenden Sie diese verschachtelte Vorlage, um den Core, die Autorisierer und die Beispielstapel für Wartezimmer zu installieren.

View template

virtual-waiting-room-on-aws.template: Verwenden Sie diese Kernvorlage, um die wichtigsten öffentlichen und privaten REST APIs Dienste sowie Cloud-Dienste für die Erstellung von Wartezimmerereignissen zu installieren. Installieren Sie diese Vorlage in dem Konto und der Region, in der Sie den Warteraum RESTAPIs, Elasticache (RedisOSS) und die DynamoDB-Tabelle benötigen.

View template

virtual-waiting-room-on-aws-authorizers.template: Verwenden Sie diese Vorlage, um den Lambda-Autorisierer zu installieren, der für die Überprüfung von im Wartezimmer ausgegebenen Token konzipiert ist und Endbenutzer schützen soll. APIs Erfordert den Core-Stack. Einige Ausgaben des Core-Stacks werden als Parameter benötigt, um diesen Stack bereitzustellen. Dies ist eine optionale Vorlage.

View template

virtual-waiting-room-on-aws-openid.template: Verwenden Sie diese Vorlage, um einen OpenID-Identitätsanbieter für die Wartezimmerintegration mit Autorisierungsschnittstellen zu installieren. Erfordert den Core-Stack. Einige Ausgaben des Core-Stacks sind erforderlich, um diesen Stack bereitzustellen. Dies ist eine optionale Vorlage.

View template

virtual-waiting-room-on- aws-sample-inlet-strategy .template: Verwenden Sie diese Vorlage, um beispielhafte Einlassstrategien zu installieren, die für den Einsatz zwischen einem Zielstandort und dem Wartezimmer vorgesehen sind. Inlet-Strategien helfen dabei, die Logik zu kapseln, um zu bestimmen, wann mehr Benutzer die Ziel-Site betreten sollen. Erfordert den Core-Stack. Für die Bereitstellung dieses Stacks sind Ausgaben aus dem Core-Stack erforderlich. Dies ist eine optionale Vorlage.

View template

virtual-waiting-room-on-aws-sample.template: Verwenden Sie diese Vorlage, um eine minimale Web- und API Gateway-Beispielkonfiguration für einen Warteraum und eine Zielsite zu installieren. Erfordert die Core- und Authorizer-Stacks. Für die Bereitstellung dieses Stacks sind Ausgaben aus den Core- und Authorizer-Stacks als Parameter erforderlich. Dies ist eine optionale Vorlage.

Automatisierte Bereitstellung

Bevor Sie die Lösung auf den Markt bringen, sollten Sie sich mit den Kosten, der Architektur, der Netzwerksicherheit und anderen in diesem Handbuch erörterten Überlegungen vertraut machen. Folgen Sie den step-by-step Anweisungen in diesem Abschnitt, um die Lösung zu konfigurieren und in Ihrem Konto bereitzustellen.

Bereitstellungszeit: Ungefähr 30 Minuten (nur Stack für Einsteiger)

Voraussetzungen

- AWS Kontokonsolenberechtigungen entsprechen Administratorzugriff.
- Aktivieren Sie die CloudWatch Protokollierung vom API Gateway aus:
  - Melden Sie sich bei der APIGateway-Konsole an und wählen Sie die Region aus, in der Sie die Stacks installieren möchten.

    Wenn Sie in dieser Region bereits APIs definiert haben:

    1. Wählen Sie eine beliebige ausAPI.
    2. Wählen Sie in der linken Navigationsleiste Einstellungen aus.
    3. Suchen Sie nach einem Wert im ARN Feld für die CloudWatch Protokollrolle.

    - Wenn es keinen gibtARN, installieren Sie den`virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template`.
    - Wenn es einen gibtARN, starten Sie zunächst den Getting-Started-Stack.

    Wenn in dieser Region keine Definitionen APIs vorhanden sind, installieren Sie die. `virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template`
- Kenntnis der Architektur und der Implementierungsdetails der zu schützenden Ziel-Site.

Überblick über die Bereitstellung

Gehen Sie wie folgt vor, um diese Lösung auf bereitzustellen AWS. Ausführliche Anweisungen können über die Links zu den einzelnen Schritten abgerufen werden.

Schritt 1. Starten Sie den Stack für die ersten Schritte

- Starten Sie die AWS CloudFormation Vorlage in Ihrem AWS Konto.

- Überprüfen Sie die Vorlagenparameter und geben Sie die Standardwerte nach Bedarf ein oder passen Sie sie an.

Schritt 2. (Optional) Testen Sie das Wartezimmer

- Generieren Sie AWS Schlüssel, um das IAM Secured aufzurufenAPIs.
- Öffnen Sie das Bedienfeld des Probenwarteraums.
- Testen Sie den Warteraum für die Probe.

Schritt 1. Starten Sie den Stack für die ersten Schritte

Diese automatisierte AWS CloudFormation Vorlage stellt den Kern, die Autorisierer und die Mustervorlagen für Wartezimmer bereit, sodass Sie einen funktionierenden Warteraum anzeigen und testen können. Sie müssen die Voraussetzungen lesen und verstehen, bevor Sie den Stack starten können.

> **Note**
>
> Sie sind für die Kosten der AWS Dienste verantwortlich, die Sie beim Betrieb dieser Lösung in Anspruch nehmen. Weitere Informationen finden Sie im Abschnitt Kosten in diesem Handbuch und auf der Preisseite der einzelnen AWS Dienste, die in dieser Lösung verwendet werden.

1. Melden Sie sich bei an AWS Management Consoleund klicken Sie auf die Schaltfläche, um die `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation Vorlage zu starten.

   Launch solution Alterna

   können Sie die Vorlage als Ausgangspunkt für Ihre eigene Implementierung herunterladen.
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlage im Amazon S3 URL S3-Textfeld URL befindet, und wählen Sie Weiter.

4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter IAMund STS Grenzwerte im AWS Identity and Access Management Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Beschreibung
Ereignis-ID	`Sample`	Eindeutige ID für diese Instanz des Wartezimmers, GUID empfohlenes Format.
Gültigkeitszeitraum	`3600`	Gültigkeitsdauer des Tokens in Sekunden.
Generierung von Ereignissen aktivieren	`false`	Wenn diese Option auf gesetzt ist`true`, werden jede Minute Messwerte für den Warteraum in den zugehörig en Event-Bus geschrieben
Elasticache-Anschluss (RedisOSS)	`1785`	Die Portnummer, die für die Verbindung zum Elasticac he-Server (OSSRedis) verwendet werden soll. Es wird empfohlen, nicht den Standard-Elasticache-Port (RedisOSS) von zu verwenden. `6379`
EnableQueuePositionExpiry	`true`	Wenn auf gesetzt`false`, wird die Ablaufzeit für die Warteschlangenposition nicht angewendet.
QueuePositionExpiryPeriod	`900`	Dies ist das Zeitintervall in Sekunden, nach dessen Ablauf eine Warteschl

Parameter	Standard	Beschreibung
		angenposition nicht mehr zur Generierung eines Tokens berechtigt ist.
IncrSvcOnQueuePositionExpiry	`false`	Wenn diese Option auf gesetzt ist`true`, wird der Bereitstellungszähler auf der Grundlage abgelaufener Warteschlangenpositionen, für die keine erfolgreichen Tokens generiert wurden, automatisch erweitert.

6. Wählen Sie Weiter.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage Ressourcen AWS Identity and Access Management (IAM) erstellt.
9. Wählen Sie Stack erstellen aus, um den Stack bereitzustellen.

   Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Sie sollten in etwa 30 Minuten COMPLETE den Status CREATE _ erhalten.

Schritt 2. (Optional) Testen Sie das Wartezimmer

Wenn Sie den Stack für die ersten Schritte bereitgestellt haben, helfen Ihnen die folgenden Schritte dabei, die Funktionalität des Warteraums zu testen. Um den Test abzuschließen, benötigen Sie AWS Schlüssel mit Berechtigungen zum Aufrufen des IAM gesicherten APIs Stacks im Kern.

Generieren Sie AWS Schlüssel, um das IAM Gesicherte aufzurufen APIs

1. Erstellen oder verwenden Sie einen IAM Benutzer in dem AWS Konto, in dem die `aws-virtual-waiting-room-getting-started.template` CloudFormation Vorlage bereitgestellt wurde.

2. Gewähren Sie dem IAMBenutzer programmatischen Zugriff. Wenn Sie einen neuen Satz von Zugriffsschlüsseln für den IAM Benutzer erstellen, laden Sie die Schlüsseldatei herunter, sobald sie angezeigt wird. Sie benötigen die Zugangsschlüssel-ID und den geheimen Zugriffsschlüssel des IAM Benutzers, um den Warteraum zu testen.
3. Fügen Sie den IAM Benutzer der rotectedAPIGroup IAM P-Benutzergruppe hinzu, die mit der Vorlage erstellt wurde.

Öffnen Sie das Bedienfeld des Warteraums für die Probe

1. Melden Sie sich an der AWS CloudFormation Konsole an und wählen Sie den Stack „Erste Schritte“ der Lösung aus.
2. Wählen Sie die Registerkarte Outputs.
3. Suchen Sie in der Spalte Schlüssel den entsprechenden ControlPanelURLWert und wählen Sie ihn aus.
4. Öffnen Sie das Bedienfeld in einem neuen Tab oder Browserfenster.
5. Erweitern Sie in der Systemsteuerung den Abschnitt Konfiguration.
6. Geben Sie die Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel ein, die Sie unter Generate AWS keys to call the IAM secured abgerufen habenAPIs. Die Endpunkte und die Ereignis-ID werden anhand der URL Parameter eingegeben.
7. Wählen Sie Verwenden. Die Schaltfläche wird aktiviert, nachdem Sie die Anmeldeinformationen eingegeben haben.

Testen Sie das Musterwartezimmer

1. Wählen Sie in der AWS CloudFormation Konsole den Getting-Started-Stack der Lösung aus.
2. Wählen Sie die Registerkarte Outputs.
3. Suchen Sie in der Spalte Schlüssel den entsprechenden WaitingRoomURLWert und wählen Sie ihn aus.
4. Öffnen Sie den Warteraum und wählen Sie dann Reservieren, um den Warteraum zu betreten.
5. Gehen Sie zurück zur Browser-Registerkarte mit dem Bedienfeld.
6. Wählen Sie unter Increment Serving Counter die Option Change aus. Auf diese Weise können 100 Benutzer vom Wartezimmer zum Zielstandort wechseln.

7. Gehen Sie zurück zum Wartezimmer und wählen Sie Jetzt auschecken! Sie werden nun zur Zielseite weitergeleitet.
8. Wählen Sie Jetzt kaufen, um Ihre Transaktion auf der Zielseite abzuschließen.

Separate Stacks bereitstellen

Der Core-Stack ist der einzige Stack, der benötigt wird, um die Hauptfunktionalität des Warteraums zu nutzen. Alle anderen Stapel sind optional. Starten Sie den Autorisierer-Stack, falls Sie noch keine Möglichkeit haben, im Wartezimmer ausgestellte Token zu validieren oder bereits vorhandene APIs zu schützen. Starten Sie den OpenID-Stack, wenn Sie einen OpenID-Identitätsanbieter für die Wartezimmerintegration mit Autorisierungsschnittstellen benötigen. Der Beispiel-Inlet-Strategiestack enthält einige Beispiele dafür, wie und wann mehr Benutzer auf die Site zugreifen können, die Sie schützen möchten.

1. Starten Sie den Core-Stack

Zeit bis zur Bereitstellung: Ungefähr 20 Minuten

Diese automatisierte AWS CloudFormation Vorlage stellt Virtual Waiting Room AWS in der AWS Cloud bereit. Sie müssen die Voraussetzungen erfüllen, bevor Sie den Stack starten können.

> **Note**
>
> Sie sind für die Kosten der AWS Dienste verantwortlich, die Sie beim Betrieb dieser Lösung in Anspruch nehmen. Weitere Informationen finden Sie im Abschnitt Kosten in diesem Handbuch und auf der Preisseite der einzelnen AWS Dienste, die in dieser Lösung verwendet werden.

1. Melden Sie sich bei an AWS Management Consoleund klicken Sie auf die Schaltfläche, um die `aws-virtual-waiting-room-on-aws.template` AWS CloudFormation Vorlage zu starten.

   Launch solution Alterna

   können Sie die Vorlage als Ausgangspunkt für Ihre eigene Implementierung herunterladen.
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlage im Amazon S3 URL S3-Textfeld URL befindet, und wählen Sie Weiter.

4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter IAMund STS Grenzwerte im AWS Identity and Access Management Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Beschreibung
ID des Ereignisses	`Sample`	Eindeutige ID für diese Instanz des Warteraums, GUID empfohlenes Format.
Gültigkeitszeitraum	`3600`	Gültigkeitsdauer des Tokens in Sekunden.
Generierung von Ereignissen aktivieren	`false`	Wenn diese Option aktiviert ist`true`, werden jede Minute Messwerte für den Warteraum in den zugehörig en Event-Bus geschrieben.
Elasticache-Anschluss (RedisOSS)	`1785`	Die Portnummer, die für die Verbindung zum Elasticac he-Server (OSSRedis) verwendet werden soll. Es wird empfohlen, nicht den Standard-Elasticache-Port (RedisOSS) von zu verwenden. `6379`
EnableQueuePositionExpiry	`true`	Wenn dieser Wert auf gesetzt ist`false`, wird die Ablaufzei t für die Warteschlangenposi tion nicht angewendet.
QueuePositionExpiryPeriod	`900`	Dies ist das Zeitintervall in Sekunden, nach dessen Ablauf eine Warteschl

Parameter	Standard	Beschreibung
		angenposition nicht mehr zur Generierung eines Tokens berechtigt ist.
IncrSvcOnQueuePositionExpiry	`false`	Wenn diese Option auf gesetzt ist`true`, wird der Bereitstellungszähler auf der Grundlage abgelaufener Warteschlangenpositionen, für die keine erfolgreichen Tokens generiert wurden, automatisch erweitert.

6. Wählen Sie Weiter.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage Ressourcen AWS Identity and Access Management (IAM) erstellt.
9. Wählen Sie Stack erstellen aus, um den Stack bereitzustellen.

   Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Sie sollten in etwa 20 Minuten COMPLETE den Status CREATE _ erhalten.

2. (Optional) Starten Sie den Authorizers-Stack

Bereitstellungszeit: ungefähr fünf Minuten

1. Melden Sie sich bei an AWS Management Consoleund klicken Sie auf die Schaltfläche, um die `aws-virtual-waiting-room-on-aws-authorizers.template` AWS CloudFormation Vorlage zu starten.

   Launch solutionAlterna

   können Sie die Vorlage als Ausgangspunkt für Ihre eigene Implementierung herunterladen.

2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlage im Amazon S3 URL S3-Textfeld URL befindet, und wählen Sie Weiter.
4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter IAMund STS Grenzwerte im AWS Identity and Access Management Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Beschreibung
Öffentlicher API Endpunkt	*<Requires input>*	Öffentlicher Endpunkt für das virtuelle WartezimmerAPIs.
Event-ID für das Wartezimmer	`Sample`	Event-ID des Wartezimmers.
Emittent URI	*<Requires input>*	Emittent URI der öffentlichen Schlüssel und Token.

6. Wählen Sie Weiter.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage Ressourcen AWS Identity and Access Management (IAM) erstellt.
9. Wählen Sie Stack erstellen aus, um den Stack bereitzustellen.

   Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Sie sollten in etwa fünf Minuten COMPLETE den Status CREATE _ erhalten.

3. (Optional) Starten Sie den OpenID-Stack

Bereitstellungszeit: ungefähr fünf Minuten

1. Melden Sie sich bei an AWS Management Consoleund klicken Sie auf die Schaltfläche, um die `aws-virtual-waiting-room-on-aws-openid.template` AWS CloudFormation Vorlage zu starten.

   Launch solution Alterna

   können Sie die Vorlage als Ausgangspunkt für Ihre eigene Implementierung herunterladen.
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlage im Amazon S3 URL S3-Textfeld URL befindet, und wählen Sie Weiter.
4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter IAMund STS Grenzwerte im AWS Identity and Access Management Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Beschreibung
Öffentlicher API Endpunkt	*<Requires input>*	Öffentlicher Endpunkt URL für das virtuelle Wartezimm erAPIs.
Privater API Endpunkt	*<Requires input>*	Privater Endpunkt URL für das virtuelle Wartezimm erAPIs.
APIRegion	*<Requires input>*	AWS Name der Region für den öffentlichen und privaten WarteraumAPIs.
ID des Ereignisses	`Sample`	Event-ID des Wartezimmers.

6. Wählen Sie Weiter.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.

8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage Ressourcen AWS Identity and Access Management (IAM) erstellt.
9. Wählen Sie Stack erstellen aus, um den Stack bereitzustellen.

   Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Sie sollten in etwa fünf Minuten COMPLETE den Status CREATE _ erhalten.

4. (Optional) Starten Sie das Beispiel für einen Inlet-Strategie-Stack

Zeit bis zur Bereitstellung: Ungefähr zwei Minuten

1. Melden Sie sich bei der an AWS Management Consoleund klicken Sie auf die Schaltfläche, um die `aws-virtual-waiting-room-sample-inlet-strategy.template` AWS CloudFormation Vorlage zu starten.

   Launch solution Alterna

   können Sie die Vorlage als Ausgangspunkt für Ihre eigene Implementierung herunterladen.
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlage im Amazon S3 URL S3-Textfeld URL befindet, und wählen Sie Weiter.
4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter IAMund STS Grenzwerte im AWS Identity and Access Management Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Beschreibung
ID des Ereignisses	`Sample`	Event-ID des Wartezimmers.

Parameter	Standard	Beschreibung
Privater API Core-Endpunkt	*<Requires input>*	Privater Endpunkt URL für das virtuelle Wartezimmerapis.
APIKernregion	*<Requires input>*	AWS Region, in der der Core installiert API ist.
Inlet-Strategie	`Periodic`	Einlassstrategie, die umgesetzt werden soll. `Periodic`erhöht die Anzahl der Portionen jede Minute. `MaxSize`erhöht die Anzahl der Zustellungen auf der Grundlage der maximalen Anzahl von Transaktionen, die der Downstream-Zielstandort zu einem bestimmten Zeitpunkt verarbeiten kann.
Inkrementieren um	*<Requires input>*	Um wie viel der Servierzähler jede Minute erhöht werden soll. Erforderlich, wenn Sie eine regelmäßige Einlassstrategie wählen.
Startzeit	*<Requires input>*	Zeitstempel, wann mit der Erhöhung der Serviernummer begonnen werden soll (Epochenzeit in Sekunden). Erforderlich, wenn Sie eine Strategie für die regelmäßige Einnahme wählen.

Parameter	Standard	Beschreibung
Endzeit	*<Requires input>*	Zeitstempel, wann die Erhöhung der Serviernummer beendet werden soll (Epochenzeit in Sekunden). Bleibt 0 übrig, wird die Serviernummer auf unbestimmte Zeit erhöht. Erforderlich, wenn Sie eine Strategie für die regelmäßige Zufuhr wählen.
CloudWatch Name des Alarms	*<Requires input>*	Optionaler CloudWatch Alarmname, der der periodischen Eingangsstrategie zugeordnet werden soll. Falls angegeben und im Alarmzustand, wird die Serviernummer nicht erhöht. Gilt nur für die periodische Eingangsstrategie.
Max. Größe	*<Requires input>*	Die maximale Anzahl von Transaktionen, die der Downstream-Zielstandort gleichzeitig verarbeiten kann (MaxSize Strategie).

6. Wählen Sie Weiter.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, um zu bestätigen, dass die Vorlage Ressourcen AWS Identity and Access Management (IAM) erstellt.
9. Wählen Sie Stack erstellen aus, um den Stack bereitzustellen.

Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Sie sollten in etwa zwei Minuten COMPLETE den Status CREATE _ erhalten.

5. (Optional) Starten Sie den Beispielstapel für den Warteraum

Bereitstellungszeit: ungefähr fünf Minuten

1. Melden Sie sich bei an AWS Management Consoleund klicken Sie auf die Schaltfläche, um die `aws-virtual-waiting-room-sample.template` AWS CloudFormation Vorlage zu starten.

   Launch solution

   Alterna

   können Sie die Vorlage als Ausgangspunkt für Ihre eigene Implementierung herunterladen.
2. Die Vorlage wird standardmäßig in der Region USA Ost (Nord-Virginia) gestartet. Um die Lösung in einer anderen AWS Region zu starten, verwenden Sie die Regionsauswahl in der Navigationsleiste der Konsole.
3. Vergewissern Sie sich auf der Seite Stack erstellen, dass sich die richtige Vorlage im Amazon S3 URL S3-Textfeld URL befindet, und wählen Sie Weiter.
4. Weisen Sie Ihrem Lösungsstapel auf der Seite „Stack-Details angeben“ einen Namen zu. Informationen zu Einschränkungen bei der Benennung von Zeichen finden Sie unter IAMund STS Grenzwerte im AWS Identity and Access Management Benutzerhandbuch.
5. Überprüfen Sie unter Parameter die Parameter für diese Lösungsvorlage und ändern Sie sie nach Bedarf. Diese Lösung verwendet die folgenden Standardwerte.

Parameter	Standard	Beschreibung
APIGateway-Region	*<Requires input>*	AWS Name der Region des API Gateways.
Autorisierer ARN	*<Requires input>*	ARNdes API Gateway Lambda Authorizers.
Ereignis-ID	`Sample`	Event-ID des Wartezimmers.

Parameter	Standard	Beschreibung
Privater API Endpunkt	*<Requires input>*	Privater Endpunkt URL für das virtuelle Wartezimmerapis.
Öffentlicher API Endpunkt	*<Requires input>*	Öffentlicher Endpunkt URL für das virtuelle Wartezimmerapis.

6. Wählen Sie Weiter.
7. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
8. Überprüfen und bestätigen Sie die Einstellungen auf der Seite Review. Markieren Sie das Kästchen, das bestätigt, dass die Vorlage Ressourcen AWS Identity and Access Management (IAM) erstellt.
9. Wählen Sie Stack erstellen aus, um den Stack bereitzustellen.

   Sie können den Status des Stacks in der AWS CloudFormation Konsole in der Spalte Status einsehen. Sie sollten in etwa fünf Minuten COMPLETE den Status CREATE _ erhalten.

Aktualisierung des Stacks von einer früheren Version

Wir empfehlen, den Stack zu löschen und einen neuen Stack für die neue Version zu erstellen. Derzeit wird die Migration zur neueren Version mithilfe des CloudFormation Stack-Updates nicht unterstützt. Weitere Informationen finden Deinstallieren Sie die Lösung Sie unter Starten des Stacks für die ersten Schritte.

> **Note**
>
> Wir empfehlen, auf eine neuere Version zu migrieren, wenn Sie die Lösung nicht aktiv zur Unterstützung einer laufenden Veranstaltung verwenden.

Leistungsdaten

Virtual Waiting Room on AWS wurde mit einem Tool namens Locust ausgelastet. Die Größe der simulierten Ereignisse lag zwischen 10.000 und 100.000 Clients. Die Lasttestumgebung bestand aus der folgenden Konfiguration:

- Locust 2.x mit Anpassungen für Cloud-Bereitstellungen AWS
- Vier AWS Regionen (,,,) `us-west-1` `us-west-2` `us-east-1` `us-east-2`
- 10 `c5.4xlarge` Amazon EC2 EC2-Hosts pro Region (40 insgesamt)
- 32 Locust-Prozesse pro Host
- Die simulierten Benutzer verteilten sich gleichmäßig auf die 1.280 Prozesse

Die end-to-end API-Testschritte für jeden Benutzerprozess:

1. Rufen Sie an `assign_queue_num` und erhalten Sie eine Anfrage-ID.
2. Schleife `queue_num` mit der Anforderungs-ID, bis die Warteschlangenposition des Benutzers zurückgegeben wird (kurze Zeit).
3. Schleife `serving_num` solange, bis der zurückgegebene Wert >= die Warteschlangenposition des Benutzers ist (lange Zeit).
4. Rufen Sie selten an`waiting_room_size`, um die Anzahl der wartenden Benutzer abzurufen.
5. Rufen Sie an `generate_token` und erhalten Sie ein JWT zur Verwendung auf der Zielsite.

Funde

Es gibt keine praktische Obergrenze für die Anzahl der Kunden, die im Wartezimmer bearbeitet werden können.

Die Geschwindigkeit, mit der Benutzer den Warteraum betreten, wirkt sich auf die Quoten für die gleichzeitige Ausführung der Lambda-Funktion für die Region aus, in der sie bereitgestellt wird.

Der Auslastungstest konnte die standardmäßigen API-Gateway-Anforderungslimits von 10.000 Anfragen pro Sekunde mit den verwendeten Caching-Richtlinien nicht überschreiten. CloudFront

Die `get_queue_num` Lambda-Funktion hat eine Aufrufrate von etwa 1:1 im Vergleich zur Rate der eingehenden Benutzer im Wartezimmer. Diese Lambda-Funktion kann bei hoher Anzahl eingehender

Benutzer aufgrund von Parallelitätslimits oder Burst-Limits gedrosselt werden. Eine Drosselung, die durch eine große Anzahl von `get_queue_num` Lambda-Funktionsaufrufen verursacht wird, kann sich als Nebeneffekt auf andere Lambda-Funktionen auswirken. Das gesamte System läuft weiter, wenn die Client-Software mit der Wiederholungs-/Back-Off-Logik angemessen auf diese Art von temporärem Skalierungsfehler reagieren kann.

Die vom Core-Stack in einer Standardkontingentkonfiguration konfigurierte CloudFront Verteilung kann einen Warteraum mit 250.000 Benutzern bewältigen, wobei jeder Benutzer die `serving_num` API mindestens jede Sekunde abfragt.

Fehlerbehebung

Dieser Abschnitt enthält Informationen zur Fehlerbehebung für diese Lösung.

Wenn Ihr Problem in diesem Abschnitt nicht behandelt wird, finden Sie unter Wenden Sie sich an den AWS-Support Anweisungen zum Öffnen einer AWS-Supportanfrage für diese Lösung.

4xx-Antwortstatus von APIs

- Dies kann durch eine falsche Event- oder Anforderungs-ID oder beides verursacht werden. Dies tritt in den CloudWatch Protokollen für die zugehörige Lambda-Funktion auf.
- Private APIs sind IAM-authentifiziert und der Client benötigt AWS Schlüssel, die über Rechte zum Aufrufen der privaten APIs verfügen. Dies tritt in den CloudWatch Protokollen für API Gateway auf.

5xx-Antwortstatus von APIs

- Antwort von gedrosseltem Lambda oder API Gateway, Alarm überprüfen. *<LambdaFunctionName>*`ThrottlesAlarm` CloudWatch
- Fehlkonfiguration im Backend. Einzelheiten finden Sie im *<LambdaFunctionName>*`ErrorsAlarm` CloudWatch Alarm und in den Protokollen. CloudWatch

5XX/ErrorPublicPrivateApiAlarm

- Dieser Alarmstatus liegt vor`ALARM`, wenn die API dem Anrufer innerhalb von 60 Sekunden einen 5XX-Status zurückgibt.
- Dieser Alarm kehrt zurück, `OK` wenn 60 Sekunden lang kein 5xx-Status zurückgegeben wird.
- Dieser Alarm kann durch eine Lambda-Funktion oder eine Lambda-Laufzeit ausgelöst werden, die einen Fehler an API Gateway zurückgibt.

4XX/ErrorPublicPrivateApiAlarm

- Dieser Alarmstatus liegt vor`ALARM`, wenn die API dem Anrufer innerhalb von 60 Sekunden einen 4XX-Status zurückgibt.
- Dieser Alarm kehrt zurück, `OK` wenn der 4XX-Status für 60 Sekunden wiederhergestellt wird.
- Dieser Alarm kann durch eine falsche API-URL ausgelöst werden.

*<LambdaFunctionName>*ThrottlesAlarm

- Dieser Alarmstatus ist ALARM, wenn das benannte Lambda innerhalb eines Zeitraums von 60 Sekunden auf ein Limit für gleichzeitige Ausführung stößt.
- Dieser Alarm kehrt in den Status zurück, OK wenn 60 Sekunden lang keine Drosselungen festgestellt wurden.
- Möglicherweise müssen Sie das Parallelitätslimit für die Region Ihres Kontos erhöhen.
- Möglicherweise stoßen Sie auf das Burst-Limit für Lambda, was eine gewisse Wiederholungslogik auf Ihrem Client erfordert.

*<LambdaFunctionName>*ErrorsAlarm

- Dieser Alarmstatus liegt vorALARM, wenn das benannte Lambda innerhalb eines Zeitraums von 60 Sekunden auf einen Laufzeitfehler stößt.
- Dieser Alarm kehrt zu, OK wenn 60 Sekunden lang keine Fehler aufgetreten sind.
- Dies kann durch eine Fehlkonfiguration im Backend verursacht werden.
- Dies kann durch einen Fehler im Lambda-Code verursacht werden.

Kontakt AWS Support

Wenn Sie über AWS Developer Support, AWS Business Support oder AWS Enterprise Support verfügen, können Sie das Support Center nutzen, um kompetente Unterstützung zu dieser Lösung zu erhalten. In den folgenden Abschnitten finden Sie entsprechende Anweisungen.

Fall erstellen

1. Melden Sie sich im Support Center an.
2. Wählen Sie Create case (Fall erstellen) aus.

Wie können wir helfen?

1. Wählen Sie Technisch.
2. Wählen Sie für Service die Option Lösungen aus.
3. Wählen Sie als Kategorie die Option Andere Lösungen aus.

4. Wählen Sie unter Schweregrad die Option aus, die Ihrem Anwendungsfall am besten entspricht.
5. Wenn Sie den Service, die Kategorie und den Schweregrad eingeben, werden in der Benutzeroberfläche Links zu häufig gestellten Fragen zur Fehlerbehebung angezeigt. Wenn Sie Ihre Frage mit diesen Links nicht lösen können, wählen Sie Nächster Schritt: Zusätzliche Informationen.

Zusätzliche Informationen

1. Geben Sie als Betreff einen Text ein, der Ihre Frage oder Ihr Problem zusammenfasst.
2. Beschreiben Sie das Problem im Feld Beschreibung detailliert.
3. Wählen Sie Dateien anhängen.
4. Hängen Sie die Informationen an, die AWS Support für die Bearbeitung der Anfrage erforderlich sind.

Helfen Sie uns, Ihren Fall schneller zu lösen

1. Geben Sie die angeforderten Informationen ein.
2. Klicken Sie auf Next step: Solve now or contact us (()Nächster Schritt): Jetzt lösen oder Support kontaktieren).

Löse jetzt oder kontaktiere uns

1. Sehen Sie sich die Solve Now-Lösungen an.
2. Wenn Sie Ihr Problem mit diesen Lösungen nicht lösen können, wählen Sie Kontakt, geben Sie die angeforderten Informationen ein und klicken Sie auf Absenden.

Weitere Ressourcen

AWS Dienstleistungen	
• AWS CloudFormation	• Amazon-DynamoDB
• Amazon Simple Storage Service	• APIAmazon-Gateway
• AWS Lambda	• AWS Secrets Manager
• Amazon CloudFront	• Amazon Simple Queue Service
• Amazon EventBridge	• Amazon CloudWatch
• Elasticache (OSSRedis)	• Amazon Comprehend
• Amazon Virtual Private Cloud	• AWS Identity and Access Management

Deinstalliere die Lösung

Sie können den virtuellen Warteraum auf der AWS Lösung von AWS Management Console oder mit dem deinstallieren AWS Command Line Interface. Sie müssen die S3-Buckets, die zum Speichern von Protokollen von verschiedenen mit dieser Lösung erstellten Ressourcen verwendet werden, manuell löschen. AWS Lösungsimplementierungen löschen diese S3-Buckets nicht automatisch, sodass Sie auch nach dem Löschen der Lösung die Protokollereignisse überprüfen können.

Wenn Sie der von der Lösung erstellten IAM-Benutzergruppe manuell einen `ProtectedAPIGroup` IAM-Benutzer hinzugefügt haben, entfernen Sie den IAM-Benutzer aus der IAM-Benutzergruppe, bevor Sie die Lösung deinstallieren. Andernfalls können die IAM-Benutzergruppe und die zugehörige IAM-Richtlinie nicht gelöscht werden.

Folgen Sie für jeden der bereitgestellten Stacks den nachstehenden Anweisungen.

Verwenden Sie den AWS Management Console

1. Melden Sie sich an der AWS CloudFormation -Konsole an.
2. Wählen Sie auf der Seite Stacks den Installationsstapel dieser Lösung aus.
3. Wählen Sie Löschen aus.

Verwenden AWS Command Line Interface

Ermitteln Sie, ob AWS Command Line Interface (AWS CLI) in Ihrer Umgebung verfügbar ist. Installationsanweisungen finden Sie unter Was ist der AWS Command Line Interface? im AWS CLI Benutzerhandbuch. Nachdem Sie sich vergewissert haben, dass das verfügbar AWS CLI ist, führen Sie den folgenden Befehl aus.

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

Löschen der Amazon S3 S3-Buckets

Diese Lösung ist so konfiguriert, dass der von der Lösung erstellte Amazon S3 S3-Bucket (für die Bereitstellung in einer Opt-in-Region) beibehalten wird, falls Sie den AWS CloudFormation Stack löschen möchten, um einen versehentlichen Datenverlust zu verhindern. Nach der Deinstallation der

Lösung können Sie diesen S3-Bucket manuell löschen, wenn Sie die Daten nicht behalten müssen. Gehen Sie wie folgt vor, um den Amazon S3 S3-Bucket zu löschen.

1. Melden Sie sich bei der Amazon S3-Konsole an.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Suchen Sie die <stack-name>S3-Buckets.
4. Wählen Sie den S3-Bucket aus und wählen Sie Löschen.

Um den S3-Bucket mit zu löschen AWS CLI, führen Sie den folgenden Befehl aus:

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

Quellcode

Besuchen Sie unser GitHubRepository, um die Quelldateien für diese Lösung herunterzuladen und Ihre Anpassungen mit anderen zu teilen.

Mitwirkende

- Jim Thario
- Thyag Ramachandran
- Joan Morgan
- Justin Pirtle
- Allen Moheimani
- Garvit Singh
- Bassem Wanis

Überarbeitungen

Datum	Änderung
November 2021	Erstversion
September 2022	Version 1.1: Automatisches Inkrementieren des Servierzählers auf der Grundlage abgelaufe ner Warteschlangenpositionen. Verlagern Sie einen Teil der Elasticache-Nutzung (RedisOSS) nach DynamoDB. Öffentlicher API Endpunkt, um die verbleibende Ablaufzeit der Warteschl angenposition abzurufen. Weitere Informati onen finden Sie in der CHANGELOG.md-Datei im GitHub Repository.
April 2023	Version 1.1.1: Die Auswirkungen, die durch die neuen Standardeinstellungen für S3 Object Ownership (ACLsdeaktiviert) für alle neuen S3-Buckets verursacht wurden, wurden gemildert . Weitere Informationen finden Sie in der CHANGELOG.md-Datei im Repository. GitHub
November 2023	Version 1.1.2: Aktualisierte Paketversionen zur Behebung von Sicherheitslücken. Weitere Informationen finden Sie in der CHANGELOG .md-Datei im GitHub Repository.
März 2024	Version 1.1.3: Drei Probleme wurden behoben: abgelaufene Warteschlangenpositionen, die in der Größe des Warteraums bestehen blieben, alte Ergebnisse auch nach einem Reset `queue_num` API zurückgegeben wurden, und zeitweilige Ausfälle in OpenID-Adaptern. `/ userInfo` API Weitere Informationen finden

Datum	Änderung
	Sie in der CHANGELOG.md-Datei im Repository. GitHub
April 2024	Version 1.1.4: Aktualisierte Paketversionen zur Behebung von Sicherheitslücken. Weitere Informationen finden Sie in der CHANGELOG.md-Datei im GitHub Repository.
Juni 2024	Version 1.1.5: Aktualisierte Paketversionen zur Behebung von Sicherheitslücken. Weitere Informationen finden Sie in der CHANGELOG.md-Datei im GitHub Repository.
August 2024	Version 1.1.6: Aktualisierte Paketversionen zur Behebung von Sicherheitslücken. Weitere Informationen finden Sie in der CHANGELOG.md-Datei im GitHub Repository.
August 2024	Version 1.1.7: Aktualisierte Paketversionen zur Behebung von Sicherheitslücken. Weitere Informationen finden Sie in der CHANGELOG.md-Datei im GitHub Repository.
September 2024	Version 1.1.8: Aktualisierte Paketversionen zur Behebung von Sicherheitslücken. Weitere Informationen finden Sie in der CHANGELOG.md-Datei im GitHub Repository.

Hinweise

Kunden sind dafür verantwortlich, Ihre eigene unabhängige Bewertung der Informationen in diesem Dokument vorzunehmen. Dieses Dokument: (a) dient nur zu Informationszwecken, (b) stellt AWS aktuelle Produktangebote und Praktiken dar, die ohne vorherige Ankündigung geändert werden können, und (c) stellt keine Verpflichtungen oder Zusicherungen von AWS und seinen verbundenen Unternehmen, Lieferanten oder Lizenzgebern dar. AWS Produkte oder Dienstleistungen werden „wie sie sind“ ohne ausdrückliche oder stillschweigende Garantien, Zusicherungen oder Bedingungen jeglicher Art bereitgestellt. AWS Die Verantwortlichkeiten und Verbindlichkeiten gegenüber seinen Kunden werden durch AWS Vereinbarungen geregelt, und dieses Dokument ist weder Teil einer Vereinbarung zwischen AWS und seinen Kunden noch ändert es diese.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.