Guia de implementação

Sala de espera virtual na AWS

Sala de espera virtual na AWS: Guia de implementação

Table of Contents

Absorva grandes explosões de tráfego em seu site com a Sala de Espera Virtual ativada AWS

Data de publicação: novembro de 2021

A AWS solução Virtual Waiting Room on ajuda a controlar as solicitações recebidas dos usuários em seu site durante grandes surtos de tráfego. Ele cria uma infraestrutura de nuvem projetada para transferir temporariamente o tráfego de entrada para seu site e fornece opções para personalizar e integrar uma sala de espera virtual. Essa solução pode ser integrada a sites novos ou existentes para escalar perfeitamente para lidar com picos repentinos de tráfego.

Exemplos de eventos de grande escala que podem produzir um aumento no tráfego do site incluem:

- Início da venda de ingressos para shows ou eventos esportivos
- Venda rápida ou outra grande venda a retalho, como a Black Friday
- Lançamento de novo produto com amplos anúncios de marketing
- Acesso ao exame e frequência às aulas para testes e aulas on-line
- Liberação de vagas para consultas médicas
- Lançamento de um novo direct-to-customer serviço que exige criação de contas e pagamentos

A solução atua como uma área de retenção para os visitantes do seu site e permite que o tráfego passe quando há capacidade suficiente. O software cliente usado pelos visitantes pode ser configurado para permitir o tráfego de forma transparente pela sala de espera até que o site atinja a capacidade máxima; nesse ponto, a sala de espera retém os visitantes. Quando seu site tem capacidade para mais tráfego, a solução gera JSON Web Tokens (JWT) que permitem que os usuários acessem o site. Por exemplo, se você tem um evento que dura duas horas e seu site pode processar 50 usuários por segundo, mas você espera um volume de 250 por segundo, você pode usar essa solução para regular o tráfego e permitir que os usuários mantenham sua posição na fila.

Essa solução fornece os seguintes recursos principais:

- Filas estruturadas de usuários em seu site
- Escalabilidade para controlar o tráfego para eventos de grande porte
- Geração de token web JSON para permitir a entrada no site de destino
- Todas as funcionalidades são controladas por meio do REST APIs

- Autorizador Turnkey API Gateway para soluções de clientes
- Integração autônoma ou uso com o OpenID

Este guia de implementação descreve considerações arquitetônicas e etapas de configuração para a implantação da Sala de Espera Virtual AWS na Amazon Web Services (AWS) Cloud. Ele inclui links para AWS CloudFormationmodelos que iniciam e configuram os AWS serviços necessários para implantar essa solução usando as AWS melhores práticas de segurança e disponibilidade.

O guia é destinado a arquitetos de TI, desenvolvedores, DevOps funcionários, analistas de dados e profissionais de tecnologia de marketing com experiência prática em arquitetura na AWS nuvem.

Custo

Você é responsável pelo custo dos AWS serviços usados durante a execução desta solução. A partir dessa revisão, o custo da execução dessa solução com as configurações padrão na região Leste dos EUA (Norte da Virgínia) é de aproximadamente 10,00 USD/dia por pilha, mais cobranças por solicitações de API e tráfego de dados em relação ao tamanho do evento.

Custo diário para manter a solução sem eventos

AWS serviço	Solicitações/Horário	Custo [USD]
Amazon API Gateway	0	$0,00
Amazon CloudFront	0	$0,00
Amazon CloudWatch	0	$0,00
Amazon DynamoDB	0	$0,00
Amazon ElastiCache	Horas do nó de computação (Redis)	~$6,00
AWS Lambda	Nível gratuito*	$0,00
AWS Secrets Manager	Nível gratuito*	$0,00
Amazon Simple Storage Service (Amazon S3)	Nível gratuito*	$0,00
Amazon Virtual Private Cloud (Amazon VPC)	Horários do VPC endpoint Horários do gateway NAT	~$5,00
TOTAL:		~$11,00

*A estimativa de custo é baseada em um ambiente limpo. Se você estiver usando esse serviço da AWS fora dessa solução, poderá exceder a cota de nível gratuito.

As tabelas a seguir mostram os custos estimados para uma sala de espera de 50.000 usuários e 100.000 usuários com uma duração de evento variando de 2 a 4 horas, com 500 de saída. users/second incoming and 1,000 users/min Os preços estão sujeitos a alterações. Para obter detalhes completos, consulte a página de preços de cada AWS serviço usado nesta solução.

Custo estimado para 50.000 usuários da sala de espera durante o evento de 2 horas

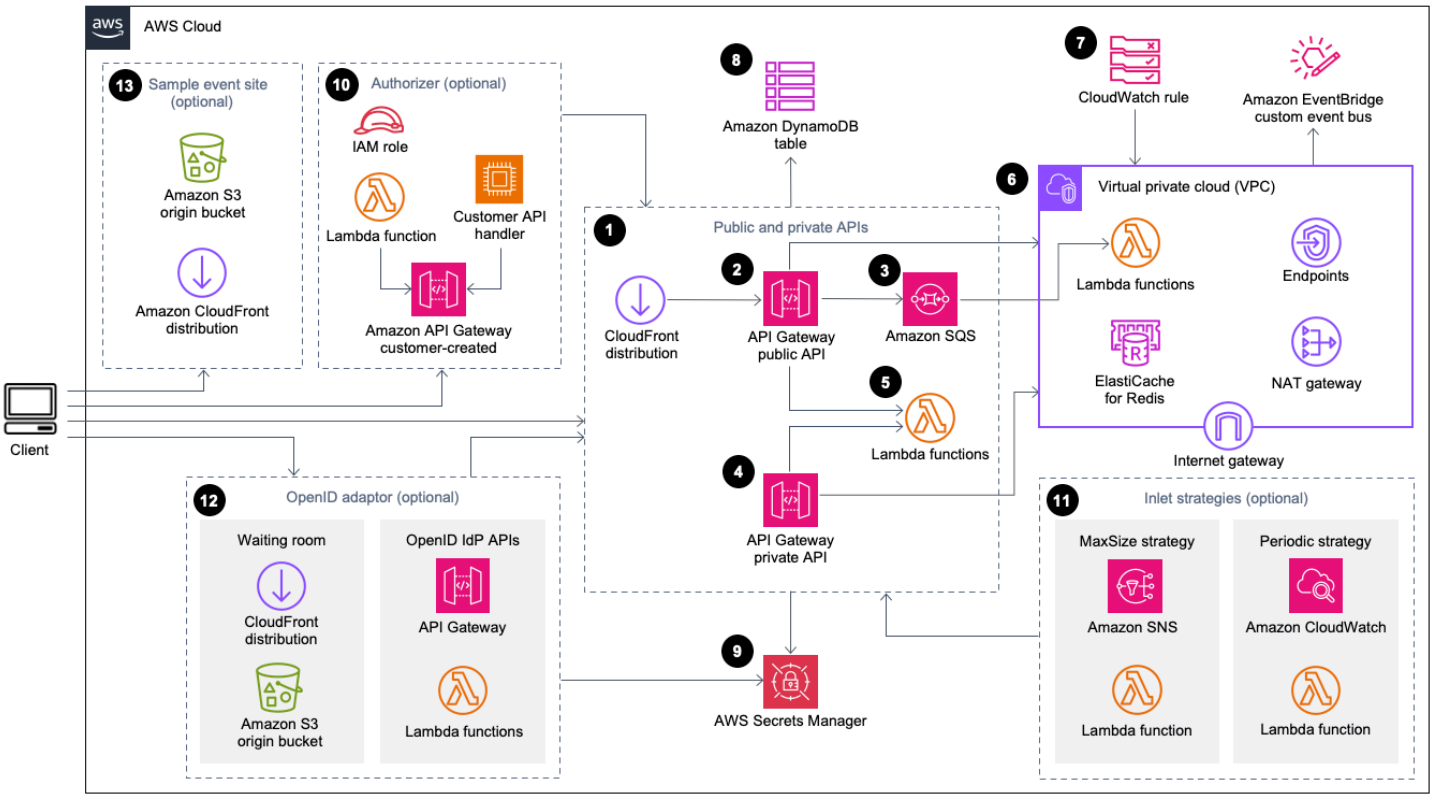
AWS serviço	Dimensões	Custo [USD]
Amazon API Gateway	Solicitações	$2,00
CloudFront	Solicitações, largura de banda	$75,00
CloudWatch	Métricas, alarmes, armazenamento	$1,00
CloudWatch Eventos da Amazon	Eventos	$1,00
DynamoDB	Unidades de leitura/gravação, Armazenamento	$1,00
ElastiCache	Horas de nó	$8,00
Lambda	Solicitações, Tempo de computação	$1,00
AWS Secrets Manager	Segredos, solicitações	$1,00
Amazon S3	Solicitações, Armazenamento	$1,00
Amazon VPC	Transferência de dados, hora do endpoint	$2,00
TOTAL		$94,00

Custo estimado para 100.000 usuários da sala de espera durante o evento de 2 horas

AWS serviço	Dimensões	Custo [USD]
Amazon API Gateway	Solicitações	$4,00
CloudFront	Solicitações, largura de banda	$296,00
CloudWatch	Métricas, alarmes, armazenamento	$1,00
CloudWatch Eventos	Eventos	$1,00
DynamoDB	Unidades de leitura/gravação, Armazenamento	$4,00
ElastiCache	Horas de nó	$32,00
Lambda	Solicitações, Tempo de computação	$1,00
AWS Secrets Manager	Segredos, solicitações	$1,00
Amazon Simple Queue Service (Amazon SQS)	Solicitações	$1,00
Amazon S3	Solicitações, Armazenamento	$1,00
Amazon VPC	Transferência de dados, hora do endpoint	$6,00
TOTAL		$348,00

Visão geral da arquitetura

A implantação dessa solução com os modelos obrigatórios e opcionais, usando parâmetros padrão, cria o seguinte ambiente na AWS nuvem.



Sala de espera virtual sobre AWS arquitetura

Os AWS CloudFormation modelos implantam a seguinte infraestrutura:

1. Uma CloudFront distribuição da Amazon para fornecer chamadas públicas de API para o cliente.
2. Recursos públicos de API do Amazon API Gateway para processar solicitações de fila da sala de espera virtual, rastrear a posição da fila e apoiar a validação de tokens que permitem acesso ao site de destino.
3. Uma fila do Amazon Simple Queue Service (Amazon SQS) para regular o tráfego para a função que processa AWS Lambdaas mensagens da fila. Em vez de invocar a função Lambda para cada solicitação, a fila do SQS agrupa as rajadas de solicitações recebidas.
4. Recursos de API privados do API Gateway para dar suporte às funções administrativas.
5. O Lambda funciona para validar e processar solicitações de API públicas e privadas e retornar as respostas apropriadas.

6. Amazon Virtual Private Cloud (VPC) para hospedar as funções Lambda que interagem diretamente com o cluster Elasticache (Redis OSS). Os VPC endpoints permitem que as funções Lambda na VPC se comuniquem com os serviços dentro da solução. Além disso, o gateway NAT permite que as funções Lambda na VPC CloudFront conectem endpoints e invalidem o cache conforme necessário.
7. Uma CloudWatch regra da Amazon para invocar uma função Lambda que funciona com um barramento personalizado da EventBridgeAmazon para transmitir periodicamente atualizações de status.
8. Tabelas do Amazon DynamoDB para armazenar dados de token, posição da fila e contador de atendimento.
9. AWS Secrets Manager para armazenar chaves para operações de token e outros dados confidenciais.
10. (Opcional) Componente autorizador que consiste em uma função AWS Identity and Access Management(IAM) e uma função autorizadora Lambda para uso com o API Gateway.
11. (Opcional) O Amazon Simple Notification Service (Amazon SNS) e o Lambda funcionam para oferecer suporte a duas estratégias de entrada. CloudWatch
12. (Opcional) Componente adaptador OpenID com funções API Gateway e Lambda para permitir que um provedor OpenID autentique usuários em seu site. CloudFront distribuição com um bucket do Amazon Simple Storage Service (Amazon S3) para a página da sala de espera desse componente.
13. CloudFront Distribuição (opcional) com o bucket de origem do Amazon S3 para o exemplo de aplicativo web de sala de espera.

Como a solução funciona

Esta seção descreve as etapas em um fluxo de trabalho de sala de espera AWS virtual em alto nível. Consulte o Guia do desenvolvedor em GitHub para obter detalhes sobre como criar, personalizar e integrar uma sala de espera para seu site.

A API pública da sala de espera pode estar localizada atrás da segurança do perímetro do seu site ou pode estar disponível sem qualquer autorização. Dependendo da abordagem usada para integrar a sala de espera ao site, talvez seja necessário que o usuário primeiro se autentique no site antes de poder navegar até a sala de espera e obter uma posição na fila.

O software cliente deve ter a ID do evento para entrar na sala de espera e fazer outras solicitações. Uma ID de evento é uma ID exclusiva necessária para a maioria das solicitações públicas e privadas APIs. O ID do evento é definido durante a instalação da pilha principal da API. Durante a operação, o ID do evento pode ser fornecido como um parâmetro de URL ou cookie por meio da página da sala de espera; ele pode ser fornecido como parte das declarações do token de autenticação ou pode ser distribuído aos clientes por meio de um caminho de dados diferente.

Há casos em que o cliente precisa do ID do evento e do ID da solicitação para fazer determinadas chamadas de API. O ID da solicitação é um ID exclusivo emitido pela sala de espera que representa um cliente específico na fila.

As etapas a seguir descrevem o fluxo de solicitações de API para entrar na fila, aguardar o progresso da fila e sair da sala de espera com um token de acesso para o site.

O usuário entra na sala de espera:

1. O usuário recebe uma tela ou página que representa o ponto de entrada da sala de espera. Eles optam por entrar na fila e o software cliente (navegador, celular, dispositivo) chama a API `assign_queue_num` pública para solicitar uma posição na fila.
2. A solicitação de API é entregue imediatamente à fila do Amazon SQS pelo API Gateway.
3. A chamada `assign_queue_num` da API retorna quando a solicitação é colocada na fila. O cliente recebe um ID de solicitação exclusivo que pode ser usado posteriormente para recuperar a posição da fila, a hora da solicitação e um token de acesso.
4. A função `AssignQueueNum` Lambda recebe lotes de até dez solicitações da fila SQS. O serviço Lambda distribui invocações para processar vários lotes de solicitações.

5. A função `AssignQueueNum` Lambda valida cada mensagem em seu lote, incrementa o contador de filas no Elasticache (Redis OSS) e armazena cada solicitação no Elasticache (Redis OSS) com sua posição de fila associada.
6. Cada mensagem é excluída quando processada com sucesso. As mensagens envolvidas em uma condição de erro são reprocessadas uma vez em um lote posterior. Após uma segunda falha, eles são enviados para um dead-letter-queue conectado a um CloudWatchalarme.
7. O cliente pode começar a pesquisar a `queue_num` API depois de receber o ID da solicitação da `assign_queue_num` chamada. O cliente envia a ID do evento e a ID da solicitação para a `queue_num` API e recebe uma posição numérica na fila ou uma resposta indicando que a solicitação ainda não foi processada. Talvez o cliente precise fazer essa ligação mais de uma vez durante grandes eventos. A função `GetQueueNum` Lambda é invocada pelo API Gateway e retorna a posição numérica do cliente na fila do DynamoDB.

O usuário espera na sala de espera:

8. Depois que o cliente tiver sua posição na fila, ele poderá começar a pesquisar a `serving_num` API em intervalos regulares. A `serving_num` API é chamada com o ID do evento e retorna a posição atual de atendimento da fila. A resposta da `serving_num` API informa ao cliente quando ele pode passar da sala de espera para o local de destino real, onde a transação final pode ocorrer. A função `GetServingNum` Lambda retorna a posição de serviço atual da sala de espera.
9. Quando a posição de serviço é igual ou maior que a posição na fila (solicitação) do cliente, o cliente pode solicitar um JSON Web Token (JWT) da API pública. O token pode ser usado com o site de destino para finalizar a transação. A `generate_token` API é chamada com o ID do evento e o ID da solicitação. O API Gateway invoca a função `GenerateToken` Lambda com os parâmetros.
10. A função `GenerateToken` Lambda valida a solicitação e verifica se esse token foi gerado anteriormente. A função Lambda consulta a tabela do DynamoDB em busca de um token correspondente. Se encontrado, esse token é retornado ao chamador e não é regenerado. Esse processo impede que um único ID de solicitação seja usado para gerar vários tokens diferentes com novos prazos de expiração.
11. Se o token não for encontrado no DynamoDB, a função Lambda recuperará as chaves para criar o token e salvará o token no DynamoDB com a ID do evento e a ID da solicitação do cliente. A função Lambda grava um evento para EventBridge sinalizar que um novo token foi gerado. A função Lambda incrementa um contador Elasticache (Redis OSS) que acompanha o número de tokens gerados para o evento.

12. Se `queue_pos_expiry` estiver ativado, o cliente poderá consultar o tempo restante antes de sua expiração chamando a `queue_pos_expiry` API que invoca a função `GetQueuePositionExpiryTime` Lambda.

O usuário sai da sala de espera:

13. Quando o cliente recebe seu token, ele entra no site de destino para iniciar a transação. Dependendo de como sua infraestrutura suporta uma integração com o JWT, o cliente pode precisar apresentar o token em um cabeçalho de solicitação, um cookie ou de outra forma. O autorizador do API Gateway pode ser usado para validar o token incluído na solicitação de um cliente. Qualquer biblioteca comercial ou de código aberto para validação e gerenciamento JWTs pode ser usada com a Sala de Espera Virtual em AWS tokens. Se o token for válido, o cliente poderá continuar a transação.
14. Depois que o cliente conclui a transação, uma API privada é chamada para atualizar o status do token do cliente e é concluída no DynamoDB.

Expiração da posição na fila:

15. Quando esse recurso é ativado, o ID de solicitação correspondente a uma posição específica na fila é elegível para gerar um token somente por um intervalo de tempo especificado.

Incremente o contador de atendimento ao expirar a posição da fila:

16. Quando esse recurso é ativado, o contador de serviço é automaticamente incrementado com base nas posições de fila expiradas que não conseguiram gerar tokens.

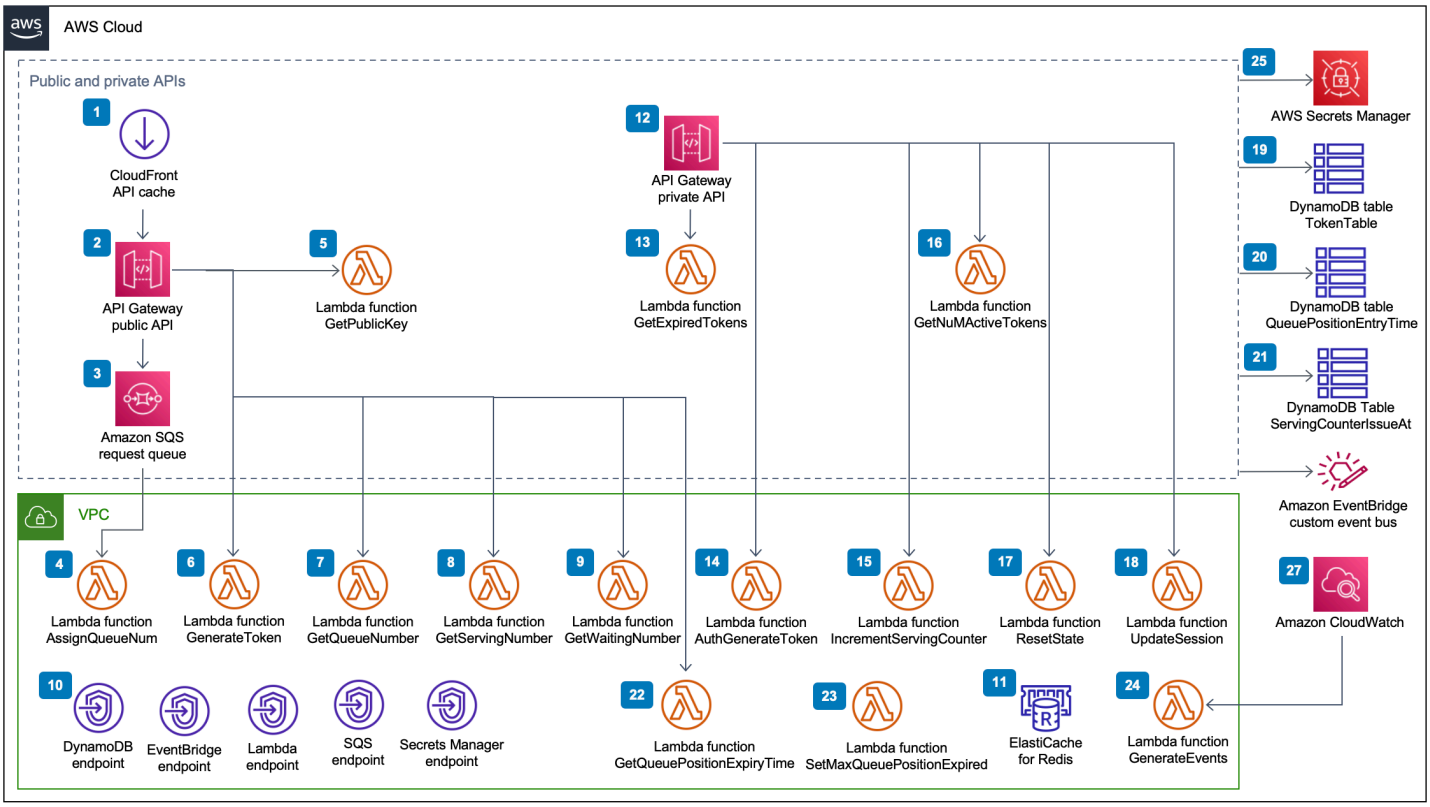
Componentes da solução

Sala de espera pública e privada APIs

O objetivo principal da AWS solução Virtual Waiting Room on é controlar a geração de JSON Web Tokens (JWT) para clientes de forma controlada, a fim de evitar surtos de novos usuários que possam sobrecarregar o site de destino. Eles JWTs podem ser usados para proteção do site, impedindo o acesso a páginas da web até que o token da sala de espera seja obtido e também para autorização de acesso à API.

O modelo principal instala uma API pública e uma API privada (autorizada pelo IAM) usadas na maioria das operações da Sala de Espera Virtual. AWS A API pública é configurada com uma CloudFront distribuição com várias políticas de armazenamento em cache com base no caminho da API. Uma tabela do DynamoDB EventBridge e um barramento de eventos são criados. O modelo adiciona uma nova VPC com duas zonas de disponibilidade (AZs), um cluster Elasticache (Redis OSS) em ambas e várias funções Lambda. AZs As funções Lambda que interagem com o Elasticache (Redis OSS) têm interfaces de rede na VPC e todas as outras funções do Lambda têm conectividade de rede padrão. O núcleo APIs é a camada mais baixa de interação com a solução. Outras funções do Lambda, a instância do Amazon Elastic Compute Cloud (Amazon EC2) e os contêineres podem atuar como extensões e chamar o núcleo APIs para criar salas de espera, controlar o tráfego de entrada e reagir aos eventos gerados pela solução.

Além disso, a pilha principal cria um alarme para todos os erros e condições de aceleração da função Lambda, bem como alarmes para cada implantação do API Gateway para códigos de status 4XX e 5XX.



Sala de espera virtual no APIs componente público e privado da AWS

1. CloudFront A distribuição fornece chamadas públicas de API para o cliente e armazena em cache o resultado quando apropriado.
2. A API pública do Amazon API Gateway processa solicitações de fila da sala de espera virtual, rastreia a posição da fila e oferece suporte à validação de tokens que permitem acesso ao site de destino.
3. A fila SQS regula o tráfego para a AWS Lambda função que processa as mensagens da fila.
4. A função `AssignQueueNum` Lambda valida cada mensagem em seu lote recebido, incrementa o contador de filas no Elasticache (Redis OSS) e armazena cada solicitação no Elasticache (Redis OSS) com sua posição de fila associada.
5. A função `GetPublicKey` Lambda recupera o valor da chave pública do Secrets Manager.
6. A função `GenerateToken` Lambda gera um JWT para uma solicitação válida que foi autorizada a concluir sua transação no site de destino. Ele grava um evento no barramento de eventos personalizado da sala de espera informando que um token foi gerado. Se um token tiver sido gerado anteriormente para essa solicitação, nenhum novo token será gerado.

7. A função `GetQueueNumber` Lambda recupera e retorna a posição numérica do cliente na fila do Elasticache (Redis OSS).
8. A função `GetServingNumber` Lambda recupera e retorna o número atualmente servido pela sala de espera do Elasticache (Redis OSS).
9. A função `GetWaitingNum` Lambda retorna o número atualmente na fila na sala de espera e ainda não recebeu um token.
10. Os VPC endpoints permitem que as funções Lambda na VPC se comuniquem com os serviços dentro da solução.
11. O cluster Elasticache (Redis OSS) armazena todas as solicitações para entrar na sala de espera com uma ID de evento válida. Ele também armazena vários contadores, como número de solicitações enfileiradas, número atualmente atendido, número de tokens gerados, número de sessões concluídas e número de sessões abandonadas.
12. Recursos de API privados do API Gateway para dar suporte às funções administrativas. Os privados APIs são autenticados pelo AWS IAM.
13. A função `GetExpiredTokens` Lambda retorna uma lista de solicitações IDs com tokens expirados.
14. A função `AuthGenerateToken` Lambda gera um token para uma solicitação válida que foi autorizada a concluir sua transação no site de destino. O emissor e o período de validade de um token inicialmente definido durante a implantação da pilha principal podem ser substituídos. Ele grava um evento no barramento de eventos personalizado da sala de espera informando que um token foi gerado. Se o token tiver sido gerado anteriormente para essa solicitação, nenhum novo token será gerado.
15. A função `IncrementServingCounter` Lambda incrementa o contador de atendimento da sala de espera armazenado no Elasticache (Redis OSS), dado um incremento por valor.
16. A função `GetNumActiveTokens` Lambda consulta o DynamoDB para saber o número de tokens que ainda não expiraram, não foram usados para concluir sua transação e não foram marcados como abandonados.
17. A função `ResetState` Lambda redefine todos os contadores armazenados no Elasticache (Redis OSS). Ele também exclui e recria as tabelas `TokenTableQueuePositionEntryTime`, e do DynamoDB`ServingCounterIssuedAt`. Além disso, ele executa a invalidação CloudFront do cache.
18. A função `UpdateSession` Lambda atualiza o status de uma sessão (token) armazenada na tabela do DynamoDB`TokenTable`. O status da sessão é indicado por um número inteiro. Sessões definidas com um status de `1` indicam concluídas e `-1` indicam abandonadas. Ele grava

um evento no barramento de eventos personalizado da sala de espera informando que uma sessão foi atualizada.

19 A tabela do `TokenTable` DynamoDB armazena dados de token.

20 A tabela do `QueuePositionEntryTime` DynamoDB armazena a posição da fila e os dados do horário de entrada.

21 A tabela do `ServingCounterIssuedAt` DynamoDB armazena atualizações no contador de atendimento.

22 A função `GetQueuePositionExpireTime` Lambda é invocada quando o cliente solicita o tempo restante de expiração da posição na fila.

23 A função `SetMaxQueuePositionExpired` Lambda define a posição máxima da fila que expirou correspondente aos valores da tabela. `ServingCounterIssuedAt` Ele é executado a cada minuto se o `IncrSvcOnQueuePositionExpiry` parâmetro for definido como `true` durante a implantação da pilha principal.

24 A função `GenerateEvents` Lambda grava várias métricas da sala de espera no barramento de eventos personalizado da sala de espera. Ele é executado a cada minuto se o parâmetro Enable Events Generation estiver definido como `true` durante a implantação da pilha principal.

25 AWS O Secrets Manager armazena chaves para operações de token e outros dados confidenciais.

26 O Amazon EventBridge Custom Event Bus recebe um evento toda vez que um token é gerado e uma sessão é atualizada na tabela do `TokenTable` DynamoDB. Ele também recebe eventos quando o contador de serviço é movido no `SetMaxQueuePositionExpired` Lambda. Ele é gravado com várias métricas de sala de espera, se ativado durante a implantação da pilha principal.

27 A regra de CloudWatch eventos da Amazon é criada se o parâmetro Enable Events Generation for definido como verdadeiro durante a implantação da pilha principal. Essa regra de evento inicia a função `GenerateEvents` Lambda a cada minuto.

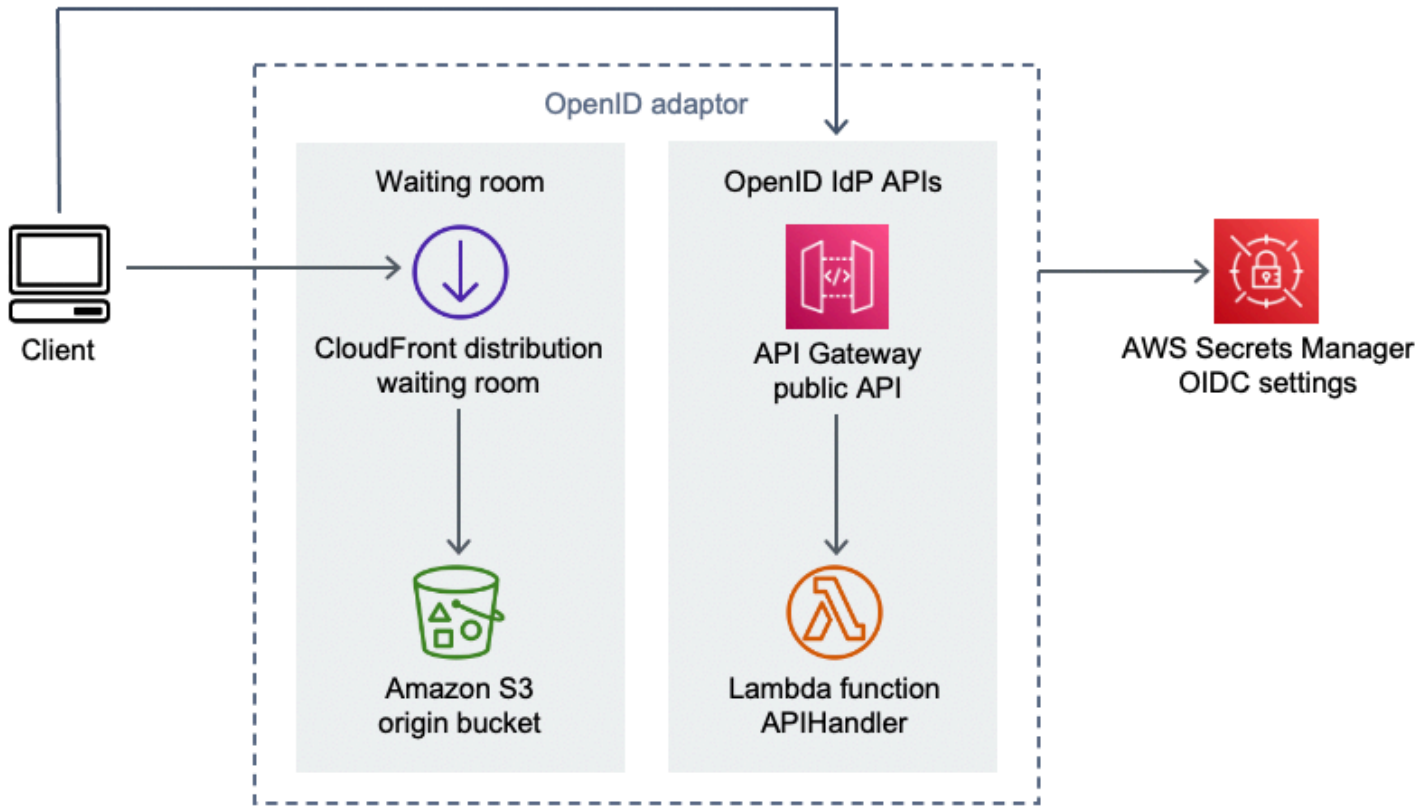
Authorizers

A solução inclui uma pilha de autorizadores Lambda do API Gateway. A pilha consiste em uma função do IAM e uma função do Lambda. A função `APIGatewayAuthorizer` Lambda é uma autorizadora do API Gateway que pode validar a assinatura e as declarações de um token emitido pela Sala de Espera Virtual na API. AWS A função Lambda fornecida com a pilha pode ser usada para proteger a nuvem APIs até que o usuário passe pela sala de espera e receba um token de

acesso. O autorizador recupera e armazena automaticamente em cache a chave pública e a configuração da API principal para verificação do token. Ele pode ser usado sem modificação e pode ser instalado em qualquer AWS região que suporte AWS Lambda.

adaptador OpenID

A pilha de adaptadores OpenID implanta um API Gateway e funções Lambda que atuam como um provedor de identidade OpenID. O adaptador OpenID fornece um conjunto compatível com OIDC APIs que pode ser usado com o software de hospedagem na web existente que oferece suporte a provedores de identidade OIDC, como AWS Elastic Load Balancers WordPress, ou como um provedor de identidade federado para o Amazon Cognito ou serviço similar. O adaptador permite que o cliente use a sala de espera no fluxo Authn/Authz ao usar software de hospedagem off-the-shelf na web com opções de integração limitadas. A pilha também instala uma CloudFront distribuição com um bucket Amazon S3 como origem e outro bucket S3 para registrar solicitações. O adaptador OpenID exibe uma página de amostra da sala de espera, semelhante à fornecida na pilha de amostra da sala de espera, mas projetada para um fluxo de autenticação OpenID. O processo de autenticação envolve obter uma posição na fila da sala de espera e esperar até que a posição de serviço seja igual ou maior que a posição na fila do cliente. A página da sala de espera do OpenID redireciona de volta para o site de destino, que usa a API OpenID para concluir a aquisição do token e a configuração da sessão para o cliente. Os endpoints da API dessa solução são mapeados diretamente para a especificação de fluxo name-for-name oficial do OpenID Connect 1.0,. Consulte Autenticação do OpenID Connect Core 1.0 para obter detalhes.



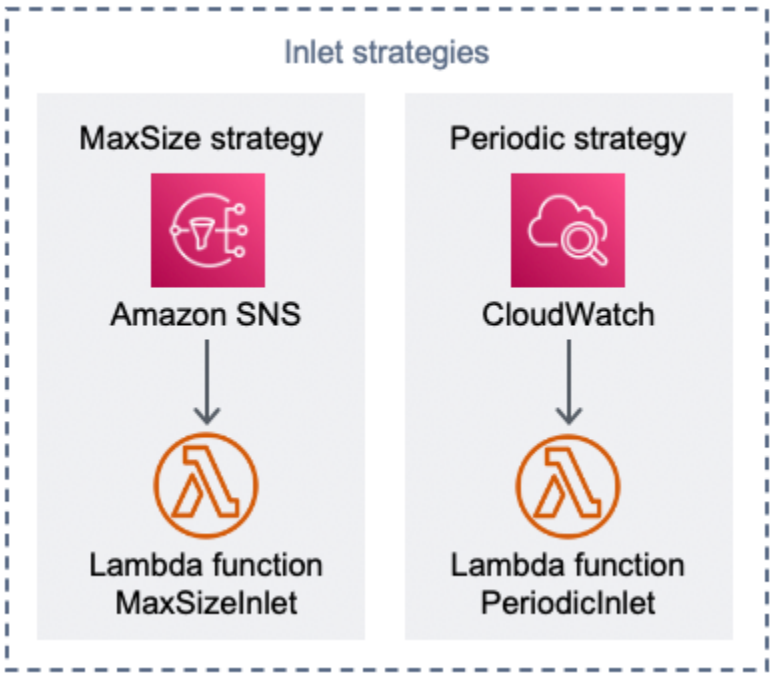
Sala de espera virtual no AWS componente adaptador OpenID

1. CloudFront a distribuição serve o conteúdo do bucket do S3 para o usuário.
2. O bucket do S3 hospeda exemplos de páginas da sala de espera.
3. A API do Amazon API Gateway fornece um conjunto compatível com OIDC APIs que pode ser usado com o software de hospedagem na web existente que suporta a função de autorização Lambda do provedor de identidade OIDC.
4. A função `APIHandler` Lambda processa solicitações para todos os caminhos de recursos do API Gateway. Diferentes funções do Python no mesmo módulo são mapeadas para cada caminho da API. Por exemplo, o caminho do `/authorize` recurso no API Gateway é invocado `authorize()` dentro da função Lambda.
5. As configurações do OIDC são armazenadas no Secrets Manager.

Estratégias de entrada de amostras

As estratégias de entrada determinam quando o balcão de atendimento da solução deve avançar para acomodar mais usuários no site de destino. Para obter mais informações conceituais sobre estratégias de entrada na sala de espera, consulte Considerações de design.

Existem dois exemplos de estratégias de entrada fornecidos pela solução: MaxSizee periódica.



Sala de espera virtual no componente AWS de estratégias de entrada

Opção de estratégia de entrada de tamanho máximo:

1. Um cliente emite uma notificação do Amazon SNS que invoca a função `MaxSizeInlet` Lambda para aumentar o contador de atendimento com base na carga útil da mensagem.
2. A função `MaxSizeInlet` Lambda espera receber uma mensagem que ela usa para determinar quanto incrementar o contador de atendimento.

Opção de estratégia de entrada periódica:

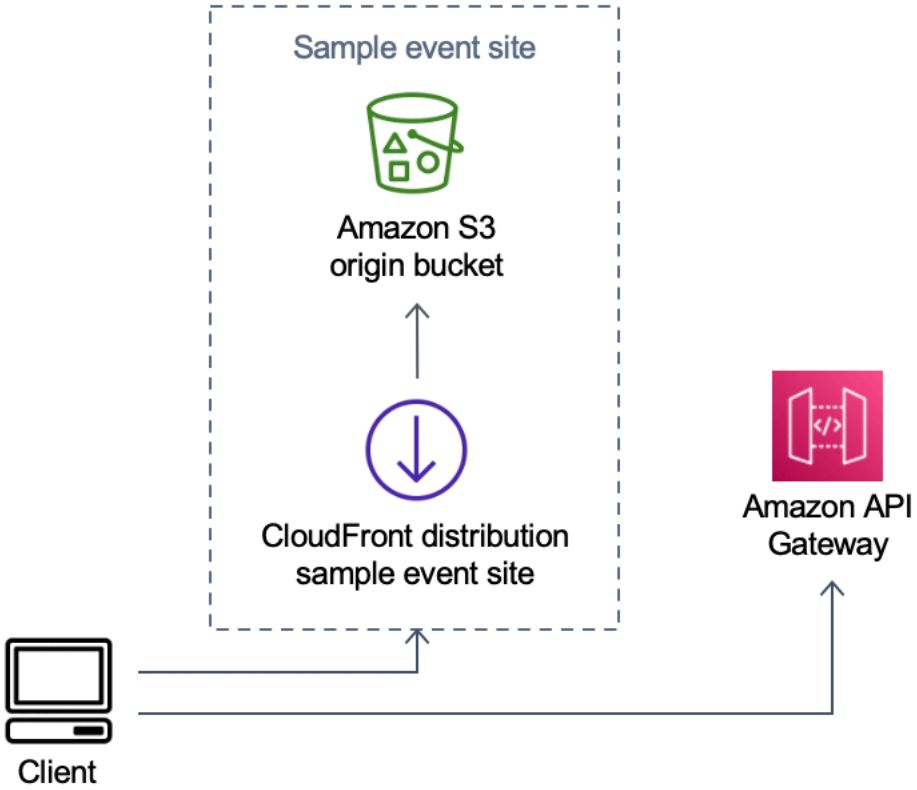
3. Uma CloudWatch regra invoca uma função Lambda a cada minuto para aumentar o contador de porções em uma quantidade fixa.
4. A função `PeriodicInlet` Lambda incrementa o contador de serviço pelo tamanho determinado se o tempo estiver entre o horário de início e término fornecido. Opcionalmente, ele verifica um

CloudWatch alarme e, se o alarme estiver em 0K estado, executa o incremento, caso contrário, o ignora.

Exemplo de sala de espera

A amostra de sala de espera se integra ao público e ao privado, APIs além do autorizador personalizado, para demonstrar uma solução mínima de sala de end-to-end espera. A página principal da web é armazenada em um bucket do S3 e usada como origem para CloudFront. Ele conduz o usuário pelas seguintes etapas:

1. Entre na fila da sala de espera para entrar no local.
2. Obtenha a posição do cliente na fila.
3. Obtenha a posição de serviço da sala de espera.
4. Obtenha um conjunto de tokens quando a posição de serviço for igual ou maior à posição do cliente.
5. Use o token para chamar uma API protegida pelo autorizador Lambda.



Sala de espera virtual no componente AWS Sample Event Site

1. O bucket do S3 hospeda o conteúdo de amostra para a sala de espera e o painel de controle.
2. CloudFront a distribuição serve o conteúdo do bucket do S3 para o usuário.
3. Exemplo de implantação do API Gateway com caminhos de recursos semelhantes a compras, como e. `/search` `/checkout` Essa API é instalada pela pilha e configurada com o autorizador de token. O objetivo é ser um exemplo de uma maneira simples de proteger uma API com a sala de espera. Solicitações que apresentam um token válido são encaminhadas para o Lambda, caso contrário, um erro será retornado. Não há nenhuma funcionalidade na API além da resposta da função Lambda anexada.

Segurança

Quando você cria sistemas na AWS infraestrutura, as responsabilidades de segurança são compartilhadas entre você AWS e. Esse modelo compartilhado reduz sua carga operacional porque AWS opera, gerencia e controla os componentes, incluindo o sistema operacional do host, a camada de virtualização e a segurança física das instalações nas quais os serviços operam. Para obter mais informações sobre AWS segurança, acesse AWS Cloud Security.

O Elasticache (Redis OSS) recebe uma interface de rede dentro da VPC privada. As funções Lambda que interagem com o Elasticache (Redis OSS) também recebem interfaces de rede em uma VPC. Todos os outros recursos têm conectividade de rede no espaço AWS de rede compartilhado. As funções Lambda com interfaces de VPC que interagem com outros serviços AWS usam VPC endpoints para se conectar a esses serviços.

As chaves públicas e privadas usadas para criar e validar tokens web JSON são geradas no momento da implantação e armazenadas no Secrets Manager. A senha usada para se conectar ao Elasticache (Redis OSS) também é gerada no momento da implantação e armazenada no Secrets Manager. A chave privada e a senha do Elasticache (Redis OSS) não podem ser acessadas por meio de nenhuma API de solução.

A API pública deve ser acessada por meio de CloudFront. A solução gera uma chave de API para o API Gateway, que é usada como o valor de um cabeçalho personalizado`x-api-key`,, em CloudFront. CloudFront inclui esse cabeçalho ao fazer solicitações de origem. Para obter detalhes adicionais, consulte Adicionar cabeçalhos personalizados às solicitações de origem no Amazon CloudFront Developer Guide.

Os privados APIs são configurados para exigir autorização AWS do IAM para invocação. A solução cria o grupo de usuários do `ProtectedAPIGroup` IAM com as permissões apropriadas para invocar o privado APIs. Um usuário do IAM adicionado a esse grupo está autorizado a invocar o privado APIs.

As políticas do IAM usadas em funções e permissões anexadas a vários recursos criados pela solução concedem somente as permissões necessárias para realizar as tarefas necessárias.

Para recursos como buckets S3, filas SQS e tópicos de SNS gerados pela solução, a criptografia em repouso e durante o trânsito é ativada sempre que possível.

Monitoramento

A pilha principal de APIs inclui vários CloudWatch alarmes que podem ser monitorados para detectar problemas enquanto a solução está em operação. A pilha cria um alarme para erros da função Lambda e condições de aceleração e altera o estado do alarme de `ALARM` para se ocorrer um erro ou condição `OK` de aceleração em um período de um minuto.

A pilha também cria alarmes para cada implantação do API Gateway para códigos de status 4XX e 5XX. O estado do alarme muda de `OK` para `ALARM` se um código de status 4XX ou 5XX for retornado da API em um período de um minuto.

Esses alarmes retornam a um `OK` estado após um minuto sem erros ou acelerações.

Perfis do IAM

AWS Identity and Access Management As funções (IAM) permitem que os clientes atribuam políticas e permissões de acesso granulares a serviços e usuários na AWS nuvem. Essa solução cria funções do IAM que concedem às AWS Lambda funções da solução acesso para criar recursos regionais.

Amazon CloudFront

O `virtual-waiting-room-on-aws.template` CloudFormation modelo, que cria o núcleo público e privado APIs da sala de espera, também implanta uma CloudFront distribuição para a API pública. CloudFront armazena em cache as respostas da API pública, reduzindo assim a carga no API Gateway e nas funções do Lambda que realizam o trabalho.

Essa solução também tem um exemplo opcional de modelo de sala de espera que implanta um aplicativo web simples hospedado em um bucket do Amazon Simple Storage Service (Amazon S3). Para ajudar a reduzir a latência e melhorar a segurança, uma CloudFront distribuição da Amazon é implantada com uma identidade de acesso de origem, que é um CloudFront usuário que fornece acesso público ao conteúdo do bucket do site da solução. Para obter mais informações, consulte Restringir o acesso ao conteúdo do Amazon S3 usando uma identidade de acesso de origem no CloudFront Amazon Developer Guide.

Grupos de segurança

Os grupos de segurança da VPC criados nessa solução foram projetados para controlar e isolar o tráfego de rede para o Elasticache (Redis OSS). Os lambdas que precisam se comunicar com o

Elasticache (Redis OSS) são colocados no mesmo grupo de segurança do Elasticache (Redis OSS). Recomendamos que você revise os grupos de segurança e restrinja ainda mais o acesso conforme necessário quando a implantação estiver em execução.

Considerações sobre design

Opções de implantação

Se esta for a primeira instalação, ou se você não tiver certeza do que instalar, implante o CloudFormation modelo `virtual-waiting-room-on-aws-getting-started.template` aninhado, que instala o núcleo, os autorizadores e os modelos de amostra da sala de espera. Isso fornece uma sala de espera mínima com um fluxo simples.

Protocolos compatíveis

A AWS solução Virtual Waiting Room on pode ser integrada com o seguinte:

- Bibliotecas e ferramentas de verificação do JSON Web Token
- Implantações existentes do API Gateway
- Clientes da API REST
- Clientes e provedores OpenID

Estratégias de entrada na sala de espera

As estratégias de entrada encapsulam a lógica e os dados necessários para mover os clientes da sala de espera para o site. Uma estratégia de entrada pode ser implementada como uma função Lambda, contêiner, instância da EC2 Amazon ou qualquer outro recurso computacional. Ele não precisa ser um recurso de nuvem, desde que possa chamar a sala de espera de pública e privada APIs. A estratégia de entrada recebe eventos sobre a sala de espera, o site ou outros indicadores externos que a ajudam a decidir quando mais clientes podem emitir tokens e entrar no site. Existem várias abordagens para estratégias de entrada. Qual deles você adota depende dos recursos disponíveis para você e das restrições no design do site que está sendo protegido.

A principal ação tomada pela estratégia de entrada é chamar a API privada do `increment_serving_num` Amazon API Gateway com um valor relativo que indica quantos clientes a mais podem entrar no site. Esta seção descreve dois exemplos de estratégias de entrada. Eles podem ser usados como estão, personalizados ou você pode empregar uma abordagem completamente diferente.

MaxSize

Usando a MaxSize estratégia, a função `MaxSizeInlet` Lambda é configurada com o número máximo de clientes que podem usar o site simultaneamente. Esse é um valor fixo. Um cliente emite uma notificação do Amazon SNS que invoca a função `MaxSizeInlet` Lambda para aumentar o contador de atendimento com base na carga útil da mensagem. A origem da mensagem do SNS pode vir de qualquer lugar, incluindo o código no site ou uma ferramenta de monitoramento que observa o nível de utilização do site.

A função `MaxSizeInlet` Lambda espera receber uma mensagem que pode incluir:

- `exited :`número de transações que foram concluídas
- lista de solicitações IDs a serem marcadas como concluídas
- lista de solicitações IDs a serem marcadas como abandonadas

Esses dados são usados para determinar quanto incrementar o contador de porções. Pode haver casos em que não haja capacidade adicional para incrementar o contador, com base no número atual de clientes.

Periódico

Ao usar a estratégia periódica, uma CloudWatch regra invoca a função `PeriodicInlet` Lambda a cada minuto para aumentar o contador de porções em uma quantidade fixa. A entrada periódica é parametrizada com a hora de início, a hora de término e o valor do incremento do evento. Opcionalmente, essa estratégia também verifica um CloudWatch alarme e, se o alarme estiver em `OK` estado, executa o incremento, caso contrário, o ignora. Os integradores do site podem conectar uma métrica de utilização a um alarme e usar esse alarme para pausar a entrada periódica. Essa estratégia só altera a posição de serviço enquanto a hora atual está entre os horários de início e término e, opcionalmente, o alarme especificado está no `OK` estado.

Personalizando e estendendo a solução

O administrador do site da sua organização deve decidir sobre os métodos de integração a serem usados com a sala de espera. Existem duas opções:

1. Integração básica usando diretamente os APIs autorizadores do API Gateway.
2. Integração com o OpenID por meio de um provedor de identidade.

Além da integração acima, talvez seja necessário configurar o redirecionamento do nome de domínio. Você também é responsável por implantar uma página personalizada do site da sala de espera.

A AWS solução Virtual Waiting Room on foi projetada para extensão por meio de dois mecanismos: EventBridge para notificação unidirecional de eventos e REST APIs para comunicação bidirecional.

Cotas

A principal limitação de escala para a Sala de Espera Virtual em AWS é o limite de aceleração do Lambda para a região instalada. AWS Quando instalada em uma AWS conta com a cota padrão de execução simultânea do Lambda, a AWS solução Virtual Waiting Room on pode lidar com até 500 clientes por segundo solicitando uma posição na fila. A taxa de 500 clientes por segundo é baseada na solução com todos os limites de cota simultâneos da função Lambda disponíveis exclusivamente. Se a região na conta for compartilhada com outras soluções que invocam funções Lambda, a sala de espera virtual AWS na solução deverá ter pelo menos 1.000 invocações simultâneas disponíveis. Você pode usar CloudWatch métricas para traçar as invocações simultâneas do Lambda em sua conta ao longo do tempo para fazer uma determinação. Você pode usar o console Service Quotas para solicitar aumentos. Aumentar o limite de aceleração do Lambda só aumenta as cobranças mensais da conta se invocações adicionais realmente ocorrerem.

Para cada 500 clientes adicionais por segundo, aumente seu limite de aceleração em 1.000.

Espera-se a entrada de usuários por segundo	Cota recomendada de execução simultânea
0-500	1.000 (padrão)
501-1.000	2.000
1.001-1.500	3.000

O Lambda tem um limite fixo de intermitência de 3.000 invocações simultâneas. Para obter mais informações, consulte Dimensionamento da função Lambda. O código do cliente deve esperar e repetir algumas chamadas de API se um código de erro for retornado indicando uma situação temporária de aceleração. O exemplo de cliente de sala de espera inclui esse código como um exemplo de como projetar clientes usados em eventos de alta capacidade e alta intensidade.

Essa solução também é compatível com a simultaneidade reservada e provisionada do Lambda com etapas de configuração personalizadas. Para obter detalhes, consulte Gerenciando a simultaneidade reservada do Lambda.

O limite superior de usuários que podem entrar na sala de espera, receber um token e continuar com uma transação é limitado pelo limite superior dos contadores do Elasticache (Redis OSS). Os contadores são usados para a posição de atendimento da sala de espera e para rastrear o estado resumido da solução. Os contadores usados no Elasticache (Redis OSS) têm um limite superior de 9.223.372.036.854.775.807. Uma tabela do DynamoDB é usada para armazenar uma cópia de cada token emitido para um usuário da sala de espera. O DynamoDB não tem limite prático para o tamanho de uma tabela.

Implantações regionais

Os serviços usados por essa solução são compatíveis em todas as AWS regiões. Para obter a disponibilidade mais atual dos AWS serviços por região, consulte a Lista de serviços AWS regionais.

AWS CloudFormation modelos

Para automatizar a implantação, essa solução usa os seguintes AWS CloudFormation modelos, que você pode baixar antes da implantação.

Se for a primeira vez que instala, ou se você não tiver certeza do que instalar, implante o `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation modelo, que instala os modelos principais, os autorizadores e os exemplos de código da sala de espera. Isso permite testar uma sala de espera em funcionamento com um fluxo simples.

virtual-waiting-room-on- aws-api-gateway-cw -logs-role.template: use esse modelo para adicionar um ARN de função padrão ao API Gateway no nível da conta para obter permissões de registro. CloudWatch Consulte Pré-requisitos para obter detalhes sobre se sua conta exige a implantação desse modelo ou não.

virtual-waiting-room-on- aws-getting-started .template: use esse modelo aninhado para instalar o núcleo, os autorizadores e as pilhas de amostra da sala de espera.

virtual-waiting-room-on-aws.template: use esse modelo principal para instalar os principais serviços REST APIs públicos e privados e de nuvem para criar eventos de sala de espera. Instale esse modelo na conta e na região em que você precisa da tabela REST APIs, Elasticache (Redis OSS) e DynamoDB da sala de espera.

virtual-waiting-room-on-aws-authorizers.template: use esse modelo para instalar o autorizador Lambda projetado para verificar tokens emitidos pela sala de espera e destinado a proteger os usuários finais. APIs Requer a pilha principal. Algumas saídas da pilha principal são necessárias como parâmetros para implantar essa pilha. Esse é um modelo opcional.

View template

virtual-waiting-room-on-aws-openid.template: use esse modelo para instalar um provedor de identidade OpenID para integração da sala de espera com interfaces de autorização. Requer a pilha principal. Algumas saídas da pilha principal são necessárias para implantar essa pilha. Esse é um modelo opcional.

View template

virtual-waiting-room-on- aws-sample-inlet-strategy .template: use esse modelo para instalar exemplos de estratégias de entrada destinadas ao uso entre um local de destino e a sala de espera. As estratégias de entrada ajudam a encapsular a lógica para determinar quando permitir que mais usuários entrem no site de destino. Requer a pilha principal. As saídas da pilha principal são necessárias para implantar essa pilha. Esse é um modelo opcional.

View template

virtual-waiting-room-on-aws-sample.template: use esse modelo para instalar um exemplo de configuração mínima do Web e do API Gateway para uma sala de espera e um site de destino. Requer as pilhas principais e de autorizadores. As saídas das pilhas principais e de autorizadores são necessárias como parâmetros para implantar essa pilha. Esse é um modelo opcional.

Implantação automatizada

Antes de lançar a solução, analise o custo, a arquitetura, a segurança da rede e outras considerações discutidas neste guia. Siga as step-by-step instruções nesta seção para configurar e implantar a solução em sua conta.

Tempo de implantação: aproximadamente 30 minutos (somente para a pilha inicial)

Pré-requisitos

- AWS permissões do console da conta equivalentes ao Acesso do administrador.
- Ative o CloudWatch registro no API Gateway:
  - Faça login no console do API Gateway e selecione a região em que você planeja instalar as pilhas.

    Se você tiver uma existência APIs definida nesta região:

    1. Selecione qualquer API.
    2. No painel de navegação à esquerda, selecione Configurações.
    3. Verifique se há um valor no campo ARN da função de CloudWatch registro.

    - Se não houver ARN, instale o. `virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template`
    - Se houver um ARN, comece iniciando a pilha de introdução.

    Se não houver nenhum APIs definido nesta região, instale `virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template` o.
- Conhecimento da arquitetura e dos detalhes de implementação do local de destino a ser protegido.

Visão geral da implantação

Use as etapas a seguir para implantar essa solução em AWS. Para obter instruções detalhadas, siga os links para cada etapa.

Etapa 1. Inicie a pilha de introdução

- Inicie o AWS CloudFormation modelo em sua AWS conta.
- Revise os parâmetros dos modelos e insira ou ajuste os valores padrão conforme necessário.

Etapa 2. (Opcional) Teste a sala de espera

- Gere AWS chaves para chamar o IAM de protegido APIs.
- Abra o painel de controle da sala de espera da amostra.
- Teste a sala de espera da amostra.

Etapa 1. Inicie a pilha de introdução

Esse AWS CloudFormation modelo automatizado implanta os modelos principais, os autorizadores e os exemplos de sala de espera que permitem visualizar e testar uma sala de espera em funcionamento. Você deve ler e entender os pré-requisitos antes de lançar a pilha.

> **Note**
>
> Você é responsável pelo custo dos AWS serviços usados durante a execução desta solução. Para obter mais detalhes, visite a seção Custo neste guia e consulte a página de preços de cada AWS serviço usado nesta solução.

1. Faça login no AWS Management Consolee selecione o botão para iniciar o `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation modelo.

   Launch solution

   Como alternativa, você pode baixar o modelo como ponto de partida para sua própria implementação.
2. Por padrão, esse modelo é iniciado na região Leste dos EUA (Norte da Virgínia). Para iniciar a solução em uma AWS região diferente, use o seletor de região na barra de navegação do console.
3. Na página Criar pilha, verifique se o URL do modelo correto está na caixa de texto URL do Amazon S3 e escolha Avançar.
4. Na página Especificar detalhes da pilha, insira um nome para a pilha. Para obter informações sobre limitações de nomes de caracteres, consulte Limites de IAM e STS no Guia do AWS Identity and Access Management usuário.
5. Em Parâmetros, revise os parâmetros desse modelo de solução e modifique-os conforme necessário. Essa solução usa os seguintes valores padrão.

Parameter	Padrão	Descrição
ID do evento	`Sample`	ID exclusiva para essa instância da sala de espera, formato GUID sugerido.
Período de validade	`3600`	Período de validade do token em segundos.
Habilitar geração de eventos	`false`	Se definido como`true`, as métricas relacionadas à Sala de Espera são gravadas em seu barramento de eventos a cada minuto
Porta Elasticache (Redis OSS)	`1785`	O número da porta a ser usada para se conectar ao servidor Elasticache (Redis OSS). É recomendável não usar a porta padrão do Elasticache (Redis OSS) do. `6379`
EnableQueuePositionExpiry	`true`	Se definido como`false`, o período de expiração da posição na fila não será aplicado.
QueuePositionExpiryPeriod	`900`	É o intervalo de tempo em segundos além do qual uma posição na fila não é elegível para gerar um token.

Parameter	Padrão	Descrição
IncrSvcOnQueuePosi tionExpiry	`false`	Se definido como`true`, o contador de serviço é automaticamente avançado com base nas posições de fila expiradas que não geraram tokens com sucesso.

6. Escolha Próximo.
7. Na página Configurar opções de pilha, selecione Avançar.
8. Na página Revisar, verifique e confirme as configurações. Marque a caixa confirmando que o modelo cria recursos AWS Identity and Access Management (IAM).
9. Selecione Create stack (Criar pilha) para implantar a pilha.

   Você pode ver o status da pilha no AWS CloudFormation console na coluna Status. Você deve receber o status CREATE_COMPLETE em aproximadamente 30 minutos.

Etapa 2. (Opcional) Teste a sala de espera

Se você implantou a pilha de introdução, as etapas a seguir ajudarão você a testar a funcionalidade da sala de espera. Para concluir o teste, você precisa de AWS chaves com permissões para chamar o IAM protegido APIs na pilha principal.

Gere AWS chaves para chamar o IAM protegido APIs

1. Crie ou use um usuário do IAM na AWS conta em que o `aws-virtual-waiting-room-getting-started.template` CloudFormation modelo foi implantado.
2. Conceda ao usuário do IAM acesso programático. Ao criar um novo conjunto de chaves de acesso para o usuário do IAM, baixe o arquivo da chave quando apresentado. Você precisa do ID da chave de acesso e da chave de acesso secreta do usuário do IAM para testar a sala de espera.
3. Adicione o usuário do IAM ao grupo de usuários do APIGroup IAM protegido criado pelo modelo.

Abra o painel de controle da sala de espera da amostra

1. Faça login no AWS CloudFormation console e selecione a pilha de introdução da solução.
2. Escolha a guia Outputs.
3. Na coluna Chave, localize o ControlPanelURL e selecione o valor correspondente.
4. Abra o painel de controle em uma nova guia ou janela do navegador.
5. No painel de controle, expanda a seção Configuração.
6. Insira o ID da chave de acesso e a chave de acesso secreta que você recuperou em Gerar AWS chaves para chamar o IAM protegido APIs. Os endpoints e o ID do evento são preenchidos a partir dos parâmetros do URL.
7. Escolha Usar. O botão é ativado depois que você fornece as credenciais.

Teste a sala de espera da amostra

1. No AWS CloudFormation console, selecione a pilha de introdução da solução.
2. Escolha a guia Outputs.
3. Na coluna Chave, localize o WaitingRoomURL e selecione o valor correspondente.
4. Abra a sala de espera e escolha Reservar para entrar na sala de espera.
5. Navegue de volta até a guia do navegador que tem o painel de controle.
6. Em Increment Serving Counter, escolha Alterar. Isso permite que 100 usuários passem da sala de espera para o site de destino.
7. Volte para a sala de espera e escolha Finalizar compra agora! Agora você será redirecionado para o site de destino.
8. Escolha Comprar agora para concluir sua transação no site de destino.

Implantação de pilhas separadas

A pilha principal é a única pilha necessária para obter a funcionalidade principal da sala de espera. Todas as outras pilhas são opcionais. Inicie a pilha de autorizadores se você ainda não tiver uma maneira de validar os tokens emitidos pela sala de espera ou proteger os que você já tenha APIs . Inicie a pilha OpenID se você precisar de um provedor de identidade OpenID para integração da sala de espera com interfaces de autorização. O exemplo de pilha de estratégias de entrada fornece alguns exemplos de como e quando permitir que mais usuários acessem o site que você está tentando proteger.

1. Inicie a pilha principal

Tempo para implantação: aproximadamente 20 minutos

Esse AWS CloudFormation modelo automatizado implanta a Sala de Espera Virtual AWS na AWS nuvem. Você deve preencher os pré-requisitos antes de lançar a pilha.

> **Note**
>
> Você é responsável pelo custo dos AWS serviços usados durante a execução desta solução. Para obter mais detalhes, visite a seção Custo neste guia e consulte a página de preços de cada AWS serviço usado nesta solução.

1. Faça login no AWS Management Consolee selecione o botão para iniciar o `aws-virtual-waiting-room-on-aws.template` AWS CloudFormation modelo.

   Launch solution Como alternativa, você pode baixar o modelo como ponto de partida para sua própria implementação.
2. Por padrão, esse modelo é iniciado na região Leste dos EUA (Norte da Virgínia). Para iniciar a solução em uma AWS região diferente, use o seletor de região na barra de navegação do console.
3. Na página Criar pilha, verifique se o URL do modelo correto está na caixa de texto URL do Amazon S3 e escolha Avançar.
4. Na página Especificar detalhes da pilha, insira um nome para a pilha. Para obter informações sobre limitações de nomes de caracteres, consulte Limites do IAM e do STS no Guia do AWS Identity and Access Management usuário.

5. Em Parâmetros, revise os parâmetros desse modelo de solução e modifique-os conforme necessário. Essa solução usa os seguintes valores padrão.

Parameter	Padrão	Descrição
ID do evento	`Sample`	ID exclusiva para essa instância da Sala de Espera, formato GUID sugerido.
Período de validade	`3600`	Período de validade do token em segundos.
Habilitar geração de eventos	`false`	Se definido como`true`, as métricas relacionadas à sala de espera são gravadas em seu barramento de eventos a cada minuto.
Porta Elasticache (Redis OSS)	`1785`	O número da porta a ser usada para se conectar ao servidor Elasticache (Redis OSS). É recomendável não usar a porta padrão do Elasticache (Redis OSS) do. `6379`
EnableQueuePositionExpiry	`true`	Se definido como`false`, o período de expiração da posição na fila não será aplicado.
QueuePositionExpiryPeriod	`900`	É o intervalo de tempo em segundos além do qual uma posição na fila não é elegível para gerar um token.

Parameter	Padrão	Descrição
IncrSvcOnQueuePosi tionExpiry	`false`	Se definido como`true`, o contador de serviço é automaticamente avançado com base nas posições de fila expiradas que não geraram tokens com sucesso.

6. Escolha Próximo.
7. Na página Configurar opções de pilha, selecione Avançar.
8. Na página Revisar, verifique e confirme as configurações. Marque a caixa confirmando que o modelo cria recursos AWS Identity and Access Management (IAM).
9. Selecione Create stack (Criar pilha) para implantar a pilha.

   Você pode ver o status da pilha no AWS CloudFormation console na coluna Status. Você deve receber o status CREATE_COMPLETE em cerca de 20 minutos.

2. (Opcional) Inicie a pilha de autorizadores

Tempo para implantação: aproximadamente 5 minutos

1. Faça login no AWS Management Consolee selecione o botão para iniciar o `aws-virtual-waiting-room-on-aws-authorizers.template` AWS CloudFormation modelo.

   Como alternativa, você pode baixar o modelo como ponto de partida para sua própria implementação.
2. Por padrão, esse modelo é iniciado na região Leste dos EUA (Norte da Virgínia). Para iniciar a solução em uma AWS região diferente, use o seletor de região na barra de navegação do console.
3. Na página Criar pilha, verifique se o URL do modelo correto está na caixa de texto URL do Amazon S3 e escolha Avançar.
4. Na página Especificar detalhes da pilha, insira um nome para a pilha. Para obter informações sobre limitações de nomes de caracteres, consulte Limites do IAM e do STS no Guia do AWS Identity and Access Management usuário.

5. Em Parâmetros, revise os parâmetros desse modelo de solução e modifique-os conforme necessário. Essa solução usa os seguintes valores padrão.

Parameter	Padrão	Descrição
Endpoint de API público	*<Requires input>*	Endpoint público para a sala APIs de espera virtual.
ID do evento da sala de espera	`Sample`	ID do evento da sala de espera.
URI do emissor	*<Requires input>*	URI do emissor das chaves e tokens públicos.

6. Escolha Próximo.
7. Na página Configurar opções de pilha, selecione Avançar.
8. Na página Revisar, verifique e confirme as configurações. Marque a caixa confirmando que o modelo cria recursos AWS Identity and Access Management (IAM).
9. Selecione Create stack (Criar pilha) para implantar a pilha.

   Você pode ver o status da pilha no AWS CloudFormation console na coluna Status. Você deve receber o status CREATE_COMPLETE em aproximadamente cinco minutos.

3. (Opcional) Inicie a pilha OpenID

Tempo para implantação: aproximadamente 5 minutos

1. Faça login no AWS Management Consolee selecione o botão para iniciar o `aws-virtual-waiting-room-on-aws-openid.template` AWS CloudFormation modelo.

   Como alternativa, você pode baixar o modelo como ponto de partida para sua própria implementação.
2. Por padrão, esse modelo é iniciado na região Leste dos EUA (Norte da Virgínia). Para iniciar a solução em uma AWS região diferente, use o seletor de região na barra de navegação do console.
3. Na página Criar pilha, verifique se o URL do modelo correto está na caixa de texto URL do Amazon S3 e escolha Avançar.

4. Na página Especificar detalhes da pilha, insira um nome para a pilha. Para obter informações sobre limitações de nomes de caracteres, consulte Limites do IAM e do STS no Guia do AWS Identity and Access Management usuário.
5. Em Parâmetros, revise os parâmetros desse modelo de solução e modifique-os conforme necessário. Essa solução usa os seguintes valores padrão.

Parameter	Padrão	Descrição
Endpoint de API público	*<Requires input>*	URL de endpoint público para a sala APIs de espera virtual.
Endpoint de API privado	*<Requires input>*	URL de endpoint privado para a sala APIs de espera virtual.
Região da API	*<Requires input>*	AWS nome da região para a sala de espera pública e privada APIs.
ID do evento	`Sample`	ID do evento da sala de espera.

6. Escolha Próximo.
7. Na página Configurar opções de pilha, selecione Avançar.
8. Na página Revisar, verifique e confirme as configurações. Marque a caixa confirmando que o modelo cria recursos AWS Identity and Access Management (IAM).
9. Selecione Create stack (Criar pilha) para implantar a pilha.

   Você pode ver o status da pilha no AWS CloudFormation console na coluna Status. Você deve receber o status CREATE_COMPLETE em aproximadamente cinco minutos.

4. (Opcional) Inicie a pilha estratégica de entrada de amostras

Tempo de implantação: aproximadamente dois minutos

1. Faça login no AWS Management Consolee selecione o botão para iniciar o `aws-virtual-waiting-room-sample-inlet-strategy.template` AWS CloudFormation modelo.

Launch solution

Como alternativa, você pode baixar o modelo como ponto de partida para sua própria implementação.

2. Por padrão, esse modelo é iniciado na região Leste dos EUA (Norte da Virgínia). Para iniciar a solução em uma AWS região diferente, use o seletor de região na barra de navegação do console.
3. Na página Criar pilha, verifique se o URL do modelo correto está na caixa de texto URL do Amazon S3 e escolha Avançar.
4. Na página Especificar detalhes da pilha, insira um nome para a pilha. Para obter informações sobre limitações de nomes de caracteres, consulte Limites do IAM e do STS no Guia do AWS Identity and Access Management usuário.
5. Em Parâmetros, revise os parâmetros desse modelo de solução e modifique-os conforme necessário. Essa solução usa os seguintes valores padrão.

Parameter	Padrão	Descrição
ID do evento	`Sample`	ID do evento da sala de espera.
Endpoint de API de núcleo privado	*`<Requires input>`*	URL de endpoint privado para a sala APIs de espera virtual.
Região principal da API	*`<Requires input>`*	AWS Região em que a API principal está instalada.
Estratégia de entrada	`Periodic`	Estratégia de entrada a ser implantada. `Periodic`incrementa o número da porção a cada minuto. `MaxSize`increment a o número de serviço com base no número máximo de transações que o site de destino downstream pode

Parameter	Padrão	Descrição
		processar em um determina do momento.
Incrementar por	*<Requires input>*	Quanto o contador de porções deve ser increment ado a cada minuto. Necessári o ao selecionar a estratégia de entrada periódica.
Hora de início	*<Requires input>*	Registro de data e hora de começar a incrementar o número da porção (tempo de época em segundos) . Necessário ao seleciona r a estratégia de entrada periódica.
End Time	*<Requires input>*	Registro de data e hora de parar de incrementar o número da porção (tempo de época em segundos). Se for deixado 0, o número da porção é incrementado indefinidamente. Necessário ao selecionar a estratégia de entrada periódica.
CloudWatch Nome do alarme	*<Requires input>*	Nome opcional do CloudWatc h alarme a ser associado à estratégia de entrada periódica. Se fornecido e em estado alarmante, o número de porções não é incrementado. Aplicável somente à estratégia de entrada periódica.

Parameter	Padrão	Descrição
Tamanho máximo	*<Requires input>*	O número máximo de transações que o site de destino downstream pode processar por vez (MaxSize Estratégia).

6. Escolha Próximo.
7. Na página Configurar opções de pilha, selecione Avançar.
8. Na página Revisar, verifique e confirme as configurações. Marque a caixa confirmando que o modelo cria recursos AWS Identity and Access Management (IAM).
9. Selecione Create stack (Criar pilha) para implantar a pilha.

   Você pode ver o status da pilha no AWS CloudFormation console na coluna Status. Você deve receber o status CREATE_COMPLETE em aproximadamente dois minutos.

5. (Opcional) Inicie a pilha de amostra da sala de espera

Tempo para implantação: aproximadamente 5 minutos

1. Faça login no AWS Management Consolee selecione o botão para iniciar o `aws-virtual-waiting-room-sample.template` AWS CloudFormation modelo.

   Launch solution Como alternativa, você pode baixar o modelo como ponto de partida para sua própria implementação.
2. Por padrão, esse modelo é iniciado na região Leste dos EUA (Norte da Virgínia). Para iniciar a solução em uma AWS região diferente, use o seletor de região na barra de navegação do console.
3. Na página Criar pilha, verifique se o URL do modelo correto está na caixa de texto URL do Amazon S3 e escolha Avançar.
4. Na página Especificar detalhes da pilha, insira um nome para a pilha. Para obter informações sobre limitações de nomes de caracteres, consulte Limites do IAM e do STS no Guia do AWS Identity and Access Management usuário.
5. Em Parâmetros, revise os parâmetros desse modelo de solução e modifique-os conforme necessário. Essa solução usa os seguintes valores padrão.

Parameter	Padrão	Descrição
Região do API Gateway	*<Requires input>*	AWS Nome da região do API Gateway.
ARN do autorizador	*<Requires input>*	ARN do autorizador Lambda do API Gateway.
ID do evento	`Sample`	ID do evento da sala de espera.
Endpoint de API privado	*<Requires input>*	URL de endpoint privado para a sala APIs de espera virtual.
Endpoint de API público	*<Requires input>*	URL de endpoint público para a sala APIs de espera virtual.

6. Escolha Próximo.
7. Na página Configurar opções de pilha, selecione Avançar.
8. Na página Revisar, verifique e confirme as configurações. Marque a caixa confirmando que o modelo cria recursos AWS Identity and Access Management (IAM).
9. Selecione Create stack (Criar pilha) para implantar a pilha.

   Você pode ver o status da pilha no AWS CloudFormation console na coluna Status. Você deve receber o status CREATE_COMPLETE em aproximadamente cinco minutos.

Atualizando a pilha de uma versão anterior

Recomendamos excluir a pilha e criar uma nova pilha para a nova versão. Atualmente, a migração para a versão mais recente usando a atualização de CloudFormation pilha não é suportada. Veja Desinstalar a solução então Iniciar a pilha de introdução.

> **Note**
>
> Recomendamos migrar para uma versão mais recente quando você não estiver usando ativamente a solução para dar suporte a um evento contínuo.

Dados de desempenho

A sala de espera virtual ativada AWS foi testada em carga com uma ferramenta chamada Locust. Os tamanhos dos eventos simulados variaram de 10.000 a 100.000 clientes. O ambiente de teste de carga consistia na seguinte configuração:

- Locust 2.x com personalizações para implantações na nuvem AWS
- Quatro AWS regiões (`us-west-1`,`us-west-2`,`us-east-1`,`us-east-2`)
- 10 EC2 anfitriões `c5.4xlarge` da Amazon por região (40 no total)
- 32 processos do Locust por host
- Os usuários simulados foram distribuídos uniformemente entre os 1.280 processos

As etapas de teste end-to-end da API para cada processo do usuário:

1. Ligue `assign_queue_num` e receba um ID de solicitação.
2. Faça um loop `queue_num` com o ID da solicitação até que ele retorne a posição da fila do usuário (pouco tempo).
3. Faça um loop `serving_num` até que o valor retornado seja **>=** posição na fila do usuário (longo tempo).
4. Ligue com pouca frequência `waiting_room_size` para recuperar o número de usuários em espera.
5. Ligue `generate_token` e receba um JWT para uso no site de destino.

Descobertas

Não há limite máximo prático para o número de clientes que podem ser processados na sala de espera.

A taxa na qual os usuários entram na sala de espera afeta as cotas de execução simultânea da função Lambda para a região em que ela está implantada.

O teste de carga não conseguiu exceder os limites padrão de solicitação do API Gateway de 10.000 solicitações por segundo com as políticas de armazenamento em cache usadas com CloudFront.

A função `get_queue_num` Lambda tem uma taxa de invocação próxima de 1:1 em relação à taxa de entrada de usuários na sala de espera. Essa função Lambda pode ser limitada durante altas taxas

de entrada de usuários devido a limites de simultaneidade ou limites de intermitência. A limitação causada por um grande número de invocações de funções do `get_queue_num` Lambda pode afetar outras funções do Lambda como efeito colateral. O sistema geral continuará operando se o software cliente puder responder adequadamente a esse tipo de erro temporário de escalabilidade com a lógica de repetição/recuo.

A CloudFront distribuição configurada pela pilha principal em uma configuração de cota padrão pode lidar com uma sala de espera com 250.000 usuários, com cada usuário pesquisando a `serving_num` API pelo menos a cada segundo.

Solução de problemas

Esta seção fornece informações sobre solução de problemas dessa solução.

Se esta seção não resolver seu problema, o Contact AWS Support fornece instruções para abrir um caso do AWS Support para essa solução.

Status de resposta 4xx de APIs

- Isso pode ser causado por uma ID de evento ou ID de solicitação incorreta, ou ambas. Isso ocorre nos CloudWatch registros da função Lambda relacionada.
- APIs Os privados são autenticados pelo IAM e o cliente precisa de AWS chaves que tenham direitos para invocar o privado. APIs Isso ocorre nos CloudWatch registros do API Gateway.

Status de resposta 5xx de APIs

- Resposta do Lambda ou do API Gateway acelerado, verifique o alarme. *<LambdaFunctionName>*`ThrottlesAlarm` CloudWatch
- Configuração incorreta no back-end, verifique o *<LambdaFunctionName>*`ErrorsAlarm` CloudWatch alarme e os CloudWatch registros para obter detalhes.

5 XXError Público/ PrivateApiAlarm

- Esse estado de alarme ocorre `ALARM` quando a API retorna um status 5XX ao chamador em um período de 60 segundos.
- Esse alarme retorna `OK` quando nenhum status 5xx é retornado por 60 segundos.
- Esse alarme pode ser iniciado por uma função do Lambda ou pelo tempo de execução do Lambda que retorna um erro ao API Gateway.

4 XXError Público/ PrivateApiAlarm

- Esse estado de alarme ocorre `ALARM` quando a API retorna um status 4XX ao chamador em um período de 60 segundos.
- Esse alarme retorna `OK` quando o status 4XX é retornado por 60 segundos.
- Esse alarme pode ser iniciado por uma URL de API incorreta.

*<LambdaFunctionName>*ThrottlesAlarm

- Esse estado de alarme é ALARM quando o Lambda nomeado encontra um limite de execução simultânea em um período de 60 segundos.
- Esse alarme retorna OK se nenhum acelerador for encontrado por 60 segundos.
- Talvez seja necessário aumentar o limite de simultaneidade para a região da sua conta.
- Você pode estar encontrando o limite de intermitência do Lambda, o que requer alguma lógica de repetição em seu cliente.

*<LambdaFunctionName>*ErrorsAlarm

- Esse estado de alarme ocorre `ALARM` quando o Lambda nomeado encontra um erro de execução em tempo de execução em um período de 60 segundos.
- Esse alarme retornará OK se nenhum erro for encontrado por 60 segundos.
- Isso pode ser causado por uma configuração incorreta no back-end.
- Isso pode ser causado por um bug no código do Lambda.

Contato Suporte

Se você tem o AWS Developer Support, o AWS Business Support ou o AWS Enterprise Support, você pode usar o Support Center para obter assistência especializada com essa solução. As seções a seguir dão instruções.

Criar caso

1. Faça login no Support Center.
2. Escolha Criar caso.

Como podemos ajudar?

1. Escolha Técnico.
2. Em Serviço, selecione Soluções.
3. Em Categoria, selecione Outras soluções.
4. Em Severidade, selecione a opção que melhor corresponda ao seu caso de uso.

5. Quando você insere o Serviço, a Categoria e a Gravidade, a interface preenche links para perguntas comuns de solução de problemas. Se você não conseguir resolver sua pergunta com esses links, escolha Próxima etapa: Informações adicionais.

Mais informações

1. Em Assunto, insira um texto resumindo sua pergunta ou problema.
2. Em Descrição, descreva o problema em detalhes.
3. Escolha Anexar arquivos.
4. Anexe as informações Suporte necessárias para processar a solicitação.

Ajude-nos a resolver seu caso com mais rapidez

1. Insira as informações solicitadas.
2. Escolha Próxima etapa: solucione ou entre em contato conosco.

Resolva agora ou entre em contato conosco

1. Analise as soluções Solve now.
2. Se você não conseguir resolver seu problema com essas soluções, escolha Fale conosco, insira as informações solicitadas e escolha Enviar.

Recursos adicionais

AWS serviços	
• AWS CloudFormation	• Amazon DynamoDB
• Amazon Simple Storage Service	• Amazon API Gateway
• AWS Lambda	• AWS Secrets Manager
• Amazon CloudFront	• Amazon Simple Queue Service
• Amazon EventBridge	• Amazon CloudWatch
• Cache elástico (Redis OSS)	• Amazon Comprehend
• Amazon Virtual Private Cloud	• AWS Identity and Access Management

Desinstalar a solução

Você pode desinstalar a sala de espera virtual na AWS solução do AWS Management Console ou usando AWS Command Line Interface o. Você deve excluir manualmente os buckets do S3 usados para armazenar registros de vários recursos criados por essa solução. AWS As implementações de soluções não excluem automaticamente esses buckets do S3, então você ainda pode revisar os eventos de registros após a exclusão da solução.

Se você adicionou manualmente um usuário do IAM ao grupo de usuários do `ProtectedAPIGroup` IAM criado pela solução, remova o usuário do IAM do grupo de usuários do IAM antes de desinstalar a solução. Caso contrário, o grupo de usuários do IAM e a política associada do IAM não serão excluídos.

Para cada uma das pilhas implantadas, siga as instruções abaixo.

Usando o AWS Management Console

1. Faça login no console do AWS CloudFormation.
2. Na página Pilhas, selecione a pilha de instalação dessa solução.
3. Escolha Excluir.

Usando AWS Command Line Interface

Determine se o AWS Command Line Interface (AWS CLI) está disponível em seu ambiente. Para obter instruções de instalação, consulte What Is the AWS Command Line Interface? no Guia do AWS CLI usuário. Depois de confirmar que o AWS CLI está disponível, execute o comando a seguir.

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

Excluindo os buckets do Amazon S3

Essa solução é configurada para reter o bucket Amazon S3 criado pela solução (para implantação em uma região opcional) se você decidir excluir a pilha para evitar perda acidental de dados. AWS CloudFormation Depois de desinstalar a solução, você pode excluir manualmente esse bucket do S3 se não precisar reter os dados. Siga estas etapas para excluir o bucket do Amazon S3.

1. Faça login no console do Amazon S3.
2. No painel de navegação à esquerda, escolha Buckets.
3. Localize os *<stack-name>*buckets do S3.
4. Selecione o bucket do S3 e escolha Excluir.

Para excluir o bucket do S3 usando AWS CLI, execute o seguinte comando:

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

Código-fonte

Visite nosso GitHubrepositório para baixar os arquivos de origem dessa solução e compartilhar suas personalizações com outras pessoas.

Colaboradores

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

Revisões

Verifique o arquivo CHANGELOG.md no GitHub repositório para ver todas as alterações e atualizações notáveis do software. O changelog fornece um registro claro das melhorias e correções para cada versão.

Avisos

Os clientes são responsáveis por fazer uma avaliação independente das informações contidas neste documento. Este documento: (a) é apenas para fins informativos, (b) representa ofertas e práticas AWS atuais de produtos, que estão sujeitas a alterações sem aviso prévio, e (c) não cria nenhum compromisso ou garantia de AWS suas afiliadas, fornecedores ou licenciadores. AWS os produtos ou serviços são fornecidos “no estado em que se encontram”, sem garantias, representações ou condições de qualquer tipo, expressas ou implícitas. AWS as responsabilidades e obrigações para com seus clientes são controladas por AWS contratos, e este documento não faz parte nem modifica nenhum acordo entre AWS e seus clientes.

O Virtual Waiting Room on AWS é licenciado sob os termos da Licença Apache Versão 2.0.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.