

Guide de mise en œuvre

# Salle d'attente virtuelle sur AWS



# Salle d'attente virtuelle sur AWS: Guide de mise en œuvre

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

# Table of Contents

Présentation de la solution .....	1
Coût .....	3
Coût quotidien de maintenance de la solution sans aucun événement .....	3
Coût pour 50 000 utilisateurs de la salle d'attente pendant un événement de 2 heures .....	4
Coût pour 100 000 utilisateurs de la salle d'attente pendant un événement de 2 heures .....	5
Présentation de l'architecture .....	6
Fonctionnement de la solution .....	8
Composants de la solution .....	11
API publiques et privées pour salles d'attente .....	11
Mécanismes d'autorisation .....	14
Adaptateur OpenID .....	15
Stratégies d'entrée d'échantillons .....	16
Exemple de salle d'attente .....	18
Sécurité .....	20
Surveillance .....	21
Rôles IAM .....	21
Amazon CloudFront .....	21
Groupes de sécurité .....	22
Considérations relatives à la conception .....	23
Options de déploiement .....	23
Protocoles pris en charge .....	23
Stratégies d'entrée dans les salles d'attente .....	23
MaxSize .....	24
Périodique .....	24
Personnalisation et extension de la solution .....	24
Quotas .....	25
Déploiements régionaux .....	26
AWS CloudFormation modèles .....	27
Déploiement automatique .....	29
Prérequis .....	29
Vue d'ensemble du déploiement .....	29
Étape 1. Lancez la pile de démarrage .....	30
Étape 2. (Facultatif) Testez la salle d'attente .....	32
Générez des AWS clés pour appeler les API sécurisées IAM .....	32

Ouvrez le panneau de commande de la salle d'attente d'échantillons .....	33
Testez l'exemple de salle d'attente .....	33
Déploiement de piles distinctes .....	34
1. Lancez le core stack .....	34
2. (Facultatif) Lancez la pile d'autorisations .....	36
3. (Facultatif) Lancez la pile OpenID .....	37
4. (Facultatif) Lancez la pile de stratégies d'entrée d'échantillons .....	39
5. (Facultatif) Lancez l'exemple de pile de salles d'attente .....	42
Mise à jour de la pile à partir d'une version précédente .....	44
Données de performance .....	45
Conclusions .....	45
Résolution des problèmes .....	47
Contacter AWS Support .....	48
Créer un dossier .....	48
Comment pouvons-nous vous aider ? .....	48
Informations supplémentaires .....	49
Aidez-nous à résoudre votre cas plus rapidement .....	49
Résolvez maintenant ou contactez-nous .....	49
Ressources supplémentaires .....	50
Désinstallez la solution .....	51
À l'aide du AWS Management Console .....	51
En utilisant AWS Command Line Interface .....	51
Suppression des compartiments Amazon S3 .....	51
Code source .....	53
Collaborateurs .....	54
Révisions .....	55
Avis .....	57
AWS Glossaire .....	58
.....	lix

# Absorbez les fortes rafales de trafic vers votre site Web grâce à la salle d'attente virtuelle activée AWS

Date de publication : novembre 2021 ([dernière mise à jour](#) : juin 2024)

La AWS solution Virtual Waiting Room on permet de contrôler les demandes des utilisateurs entrantes sur votre site Web lors de fortes rafales de trafic. Il crée une infrastructure cloud conçue pour décharger temporairement le trafic entrant sur votre site Web et propose des options pour personnaliser et intégrer une salle d'attente virtuelle. Cette solution peut être intégrée à des sites Web nouveaux ou existants pour s'adapter facilement aux pics soudains de trafic.

Voici des exemples d'événements de grande envergure susceptibles de provoquer une augmentation du trafic sur le site Web :

- Début de la vente des billets de concert ou d'événement sportif
- Vente au feu ou autre grande vente au détail, comme le Black Friday
- Lancement d'un nouveau produit avec de larges annonces marketing
- Accès aux examens et participation aux cours pour les tests et les leçons en ligne
- Libération des créneaux de rendez-vous médicaux
- Lancement d'un nouveau direct-to-customer service qui nécessite la création de comptes et le paiement

La solution agit comme une zone d'attente pour les visiteurs de votre site Web et permet au trafic de passer lorsque la capacité est suffisante. Le logiciel client utilisé par les visiteurs peut être configuré pour autoriser de manière transparente le trafic dans la salle d'attente jusqu'à ce que le site Web atteigne sa capacité maximale ; la salle d'attente retient alors les visiteurs. Lorsque votre site Web a une capacité de trafic accrue, la solution génère des [jetons Web JSON](#) (JWT) qui permettent aux utilisateurs d'accéder au site Web. Par exemple, si vous avez un événement qui dure deux heures et que votre site Web peut traiter 50 utilisateurs par seconde, mais que vous vous attendez à un volume de 250 par seconde, vous pouvez utiliser cette solution pour réguler le trafic tout en permettant aux utilisateurs de conserver leur position dans la file d'attente.

Cette solution fournit les principales fonctionnalités suivantes :

- Mise en file d'attente structurée des utilisateurs sur votre site Web

- Évolutivité permettant de contrôler le trafic lors d'événements de très grande envergure
- Génération de jetons Web JSON pour autoriser l'accès au site cible
- Toutes les fonctionnalités sont contrôlées par des API REST
- Autorisateur API Gateway clé en main pour les solutions client
- Intégration autonome ou utilisation avec OpenID

Ce guide de mise en œuvre décrit les considérations architecturales et les étapes de configuration pour le déploiement de Virtual Waiting Room AWS dans le cloud Amazon Web Services (AWS). Il inclut des liens vers des [AWS CloudFormation](#) modèles qui lancent et configurent les AWS services requis pour déployer cette solution en utilisant les AWS meilleures pratiques en matière de sécurité et de disponibilité.

Le guide est destiné aux architectes informatiques, aux développeurs, au DevOps personnel, aux analystes de données et aux professionnels des technologies marketing ayant une expérience pratique de l'architecture dans le AWS cloud.

# Coût

Vous êtes responsable du coût des AWS services utilisés lors de l'exécution de cette solution. À partir de cette révision, le coût d'exécution de cette solution avec les paramètres par défaut dans la région de l'est des États-Unis (Virginie du Nord) est d'environ 10 dollars par jour par pile, plus les frais liés aux demandes d'API et au trafic de données par rapport à la taille de l'événement.

## Coût quotidien de maintenance de la solution sans aucun événement

AWS web	Demandes/Heure	Coût [USD]
Amazon API Gateway	0	0,00\$
Amazon CloudFront	0	0,00\$
Amazon CloudWatch	0	0,00\$
Amazon DynamoDB	0	0,00\$
Amazon ElastiCache	Heures des nœuds de calcul (Redis)	~6,00 \$
AWS Lambda	Niveau gratuit*	0,00\$
AWS Secrets Manager	Niveau gratuit*	0,00\$
Amazon Simple Storage Service (Amazon S3)	Niveau gratuit*	0,00\$
Amazon Virtual Private Cloud (Amazon VPC)	Heures d'ouverture des terminaux VPC Horaires de la passerelle NAT	~5,00 \$
<b>TOTAL :</b>		<b>~11,00 \$</b>

\*L'estimation des coûts est basée sur un environnement propre. Si vous utilisez ce service AWS en dehors de cette solution, vous risquez de dépasser le quota du niveau gratuit.

Les tableaux suivants indiquent les coûts estimés pour une salle d'attente de 50 000 utilisateurs et une salle d'attente de 100 000 utilisateurs avec une durée d'événement de 2 à 4 heures avec 500 utilisateurs/seconde en entrée et 1 000 utilisateurs/minute en sortie. Les prix sont susceptibles d'être modifiés. Pour plus de détails, consultez la page Web de tarification de chaque AWS service utilisé dans cette solution.

## Coût estimé pour 50 000 utilisateurs de la salle d'attente pendant un événement de 2 heures

AWS web	Dimensions	Coût [USD]
Amazon API Gateway	Requêtes	2,00\$
CloudFront	Demandes, bande passante	75,00\$
CloudWatch	Métriques, alarmes, stockage	1,00\$
CloudWatch Événements Amazon	Événements	1,00\$
DynamoDB	Unités de lecture/écriture, stockage	1,00\$
ElastiCache	Heures d'utilisation des nœuds	8,00\$
Lambda	Demandes, temps de calcul	1,00\$
AWS Secrets Manager	Secrets, demandes	1,00\$
Amazon S3	Demandes, stockage	1,00\$
Amazon VPC	Transfert de données, heure du point de terminaison	2,00\$
<b>TOTAL</b>		<b>94,00\$</b>

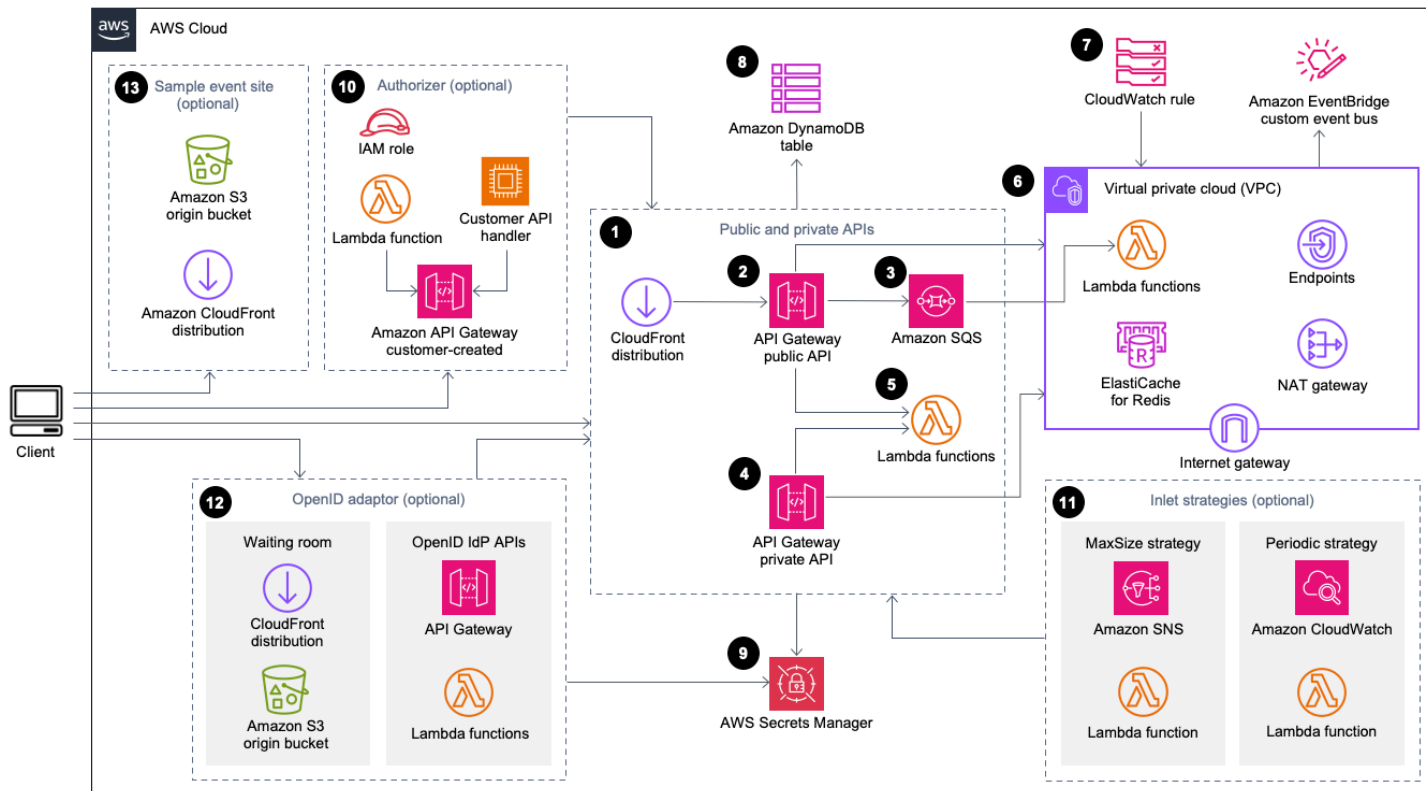


## Coût estimé pour 100 000 utilisateurs de la salle d'attente pendant un événement de 2 heures

AWS web	Dimensions	Coût [USD]
Amazon API Gateway	Requêtes	4,00\$
CloudFront	Demandes, bande passante	296,00\$
CloudWatch	Métriques, alarmes, stockage	1,00\$
CloudWatch Évènements	Évènements	1,00\$
DynamoDB	Unités de lecture/écriture, stockage	4,00\$
ElastiCache	Heures d'utilisation des nœuds	32,00\$
Lambda	Demandes, temps de calcul	1,00\$
AWS Secrets Manager	Secrets, demandes	1,00\$
Amazon Simple Queue Service (Amazon SQS)	Requêtes	1,00\$
Amazon S3	Demandes, stockage	1,00\$
Amazon VPC	Transfert de données, heure du point de terminaison	6,00\$
<b>TOTAL</b>		<b>348,00\$</b>

# Présentation de l'architecture

Le déploiement de cette solution avec les modèles obligatoires et facultatifs, à l'aide des paramètres par défaut, permet de créer l'environnement suivant dans le AWS cloud.



## Salle d'attente virtuelle sur AWS l'architecture

Les AWS CloudFormation modèles déploient l'infrastructure suivante :

1. Une CloudFront distribution [Amazon](#) pour envoyer des appels d'API publics au client.
2. Ressources d'[API publiques Amazon API Gateway](#) pour traiter les demandes de file d'attente depuis la salle d'attente virtuelle, suivre la position de la file d'attente et prendre en charge la validation des jetons permettant d'accéder au site Web cible.
3. Une [file d'attente Amazon Simple Queue Service](#) (Amazon SQS) pour réguler le trafic vers la [AWS Lambda](#) fonction qui traite les messages de file d'attente. Au lieu d'appeler la fonction Lambda pour chaque demande, la file d'attente SQS regroupe les rafales de demandes entrantes.
4. Ressources d'API privées API Gateway pour prendre en charge les fonctions administratives.
5. Fonctions Lambda pour valider et traiter les demandes d'API publiques et privées, et renvoyer les réponses appropriées.

6. [Amazon Virtual Private Cloud](#) (VPC) pour héberger les fonctions Lambda qui interagissent directement avec le cluster [Amazon ElastiCache](#) for Redis. Les points de terminaison VPC permettent aux fonctions Lambda du VPC de communiquer avec les services de la solution. En outre, la passerelle NAT permet aux fonctions Lambda du VPC de connecter des CloudFront points de terminaison et d'invalider le cache selon les besoins.
7. CloudWatch Règle [Amazon](#) invoquant une fonction Lambda qui fonctionne avec un EventBridge bus [Amazon](#) personnalisé pour diffuser régulièrement des mises à jour de statut.
8. Tables [Amazon DynamoDB](#) pour stocker les données relatives aux jetons, à la position de la file d'attente et aux compteurs de service.
9. [AWS Secrets Manager](#) pour stocker les clés pour les opérations liées aux jetons et autres données sensibles.
- 10.(Facultatif) Composant d'autorisation composé d'un rôle [AWS Identity and Access Management](#)(IAM) et d'une fonction d'autorisation Lambda à utiliser avec API Gateway.
- 11.(Facultatif) [Amazon Simple Notification Service](#) (Amazon SNS) et fonctions Lambda pour prendre en charge deux stratégies d'entrée. CloudWatch
- 12.(Facultatif) Composant adaptateur OpenID avec fonctions API Gateway et Lambda pour permettre à un fournisseur OpenID d'authentifier les utilisateurs sur votre site Web. CloudFront distribution avec un compartiment [Amazon Simple Storage Service](#) (Amazon S3) pour la page de la salle d'attente pour ce composant.
- 13.CloudFront Distribution (facultative) avec le compartiment d'origine Amazon S3 pour l'exemple d'application Web de salle d'attente.

# Fonctionnement de la solution

Cette section décrit les étapes d'un flux de travail de salle d'attente AWS virtuelle de haut niveau. Consultez le [guide du développeur GitHub pour plus de détails sur](#) la création, la personnalisation et l'intégration d'une salle d'attente pour votre site Web.

L'API publique de la salle d'attente peut être située derrière le périmètre de sécurité de votre site ou elle peut être disponible sans autorisation. Selon l'approche que vous utilisez pour intégrer la salle d'attente au site Web, l'utilisateur devra peut-être d'abord s'authentifier sur le site Web avant d'être autorisé à accéder à la salle d'attente et à obtenir une position dans la file d'attente.

Le logiciel client doit disposer de l'ID d'événement pour entrer dans la salle d'attente et faire d'autres demandes. Un ID d'événement est un identifiant unique requis pour la plupart des demandes adressées aux API publiques et privées. L'ID d'événement est défini lors de l'installation de la pile d'API principale. Pendant le fonctionnement, l'identifiant de l'événement peut être fourni sous forme de paramètre d'URL ou de cookie via la page de la salle d'attente ; il peut être fourni dans le cadre de demandes de jetons d'authentification ou il peut être distribué aux clients via un chemin de données différent.

Dans certains cas, le client a besoin à la fois de l'ID d'événement et de l'ID de demande pour effectuer certains appels d'API. L'ID de demande est un identifiant unique émis par la salle d'attente et représentant un client spécifique dans la file d'attente.

Les étapes suivantes décrivent le flux des demandes d'API pour entrer dans la file d'attente, attendre que la file progresse et quitter la salle d'attente avec un jeton d'accès au site Web.

L'utilisateur entre dans la salle d'attente :

1. L'utilisateur voit apparaître un écran ou une page représentant le point d'entrée de la salle d'attente. Ils choisissent d'entrer dans la file d'attente et le logiciel client (navigateur, mobile, appareil) appelle l'API `assign_queue_num` publique pour demander une position dans la file d'attente.
2. La demande d'API est immédiatement envoyée à la file d'attente Amazon SQS par API Gateway.
3. L'appel `assign_queue_num` d'API revient lorsque la demande est placée dans la file d'attente. Le client reçoit un identifiant de demande unique qui peut être utilisé ultérieurement pour récupérer la position de la file d'attente, l'heure de la demande et un jeton d'accès.
4. La fonction `AssignQueueNum` Lambda reçoit des lots contenant jusqu'à dix requêtes depuis la file d'attente SQS. Le service Lambda répartit les invocations pour traiter plusieurs lots de demandes.

5. La fonction `AssignQueueNum` Lambda valide chaque message de son lot, incrémente le compteur de files d'attente pour Redis et enregistre chaque demande `ElastiCache` pour Redis avec sa position de file `ElastiCache` d'attente associée.
6. Chaque message est supprimé au fur et à mesure de son traitement. Les messages concernés par une condition d'erreur sont retraités une fois dans un lot ultérieur. Après une deuxième panne, ils sont envoyés à un système `dead-letter-queue` connecté à une [CloudWatchalarme](#).
7. Le client peut commencer à interroger `l'queue_numAPI` après avoir reçu l'ID de demande de `l'assign_queue_numappel`. Le client envoie l'ID d'événement et l'ID de demande à `l'queue_numAPI` et reçoit une position numérique dans la file d'attente ou une réponse indiquant que la demande n'a pas encore été traitée. Le client peut avoir besoin de passer cet appel plusieurs fois lors d'événements importants. La fonction `GetQueueNum` Lambda est invoquée par `API Gateway` et renvoie la position numérique du client dans la file d'attente depuis `DynamoDB`.

L'utilisateur attend dans la salle d'attente :

8. Une fois que le client a trouvé sa position dans la file d'attente, il peut commencer à `serviing_num` interroger l'API à intervalles réguliers. `L'serviing_numAPI` est appelée avec l'ID d'événement et renvoie la position de service actuelle de la file d'attente. La réponse de `l'serviing_numAPI` indique au client quand il peut passer de la salle d'attente au site cible où la transaction finale peut avoir lieu. La fonction `GetServiingNum` Lambda renvoie la position de service actuelle de la salle d'attente.
9. Lorsque la position de service est égale ou supérieure à la position de la file d'attente (demande) du client, celui-ci peut demander un jeton `Web JSON (JWT)` à l'API publique. Le jeton peut être utilisé avec le site cible pour finaliser la transaction. `L'generate_tokenAPI` est appelée avec l'ID d'événement et l'ID de demande. `API Gateway` appelle la fonction `GenerateToken` Lambda avec les paramètres.
- 10 La fonction `GenerateToken` Lambda valide la demande et vérifie si ce jeton a déjà été généré. La fonction Lambda interroge la table `DynamoDB` pour trouver un jeton correspondant. S'il est trouvé, ce jeton est renvoyé à l'appelant et il n'est pas régénéré. Ce processus empêche l'utilisation d'un seul ID de demande pour générer plusieurs jetons différents avec de nouvelles dates d'expiration.
- 11 Si le jeton n'est pas trouvé dans `DynamoDB`, la fonction Lambda récupère les clés pour créer le jeton et enregistre le jeton dans `DynamoDB` avec l'ID d'événement et l'ID de demande du client. La fonction Lambda écrit un événement dans `pour EventBridge` signaler qu'un nouveau jeton a été généré. La fonction Lambda incrémente un compteur `ElastiCache` pour Redis qui assure le suivi du nombre de jetons générés pour l'événement.

12. Si `queue_pos_expiry` est activé, le client peut demander le temps restant avant son expiration en appelant l'API `queue_pos_expiry` qui invoque la fonction `GetQueuePositionExpiryTime` Lambda.

L'utilisateur quitte la salle d'attente :

13. Lorsque le client reçoit son jeton, il entre sur le site cible pour commencer sa transaction. Selon la manière dont votre infrastructure prend en charge une intégration avec JWT, le client devra peut-être présenter le jeton dans un en-tête de demande, un cookie ou d'une autre manière. L'autorisateur d'API Gateway peut être utilisé pour valider le jeton inclus dans la demande d'un client. Toutes les bibliothèques commerciales ou open source permettant de valider et de gérer les JWT peuvent être utilisées avec `Virtual Waiting Room on tokens`. AWS. Si le jeton est valide, le client est autorisé à poursuivre sa transaction.

14. Une fois que le client a terminé sa transaction, une API privée est appelée pour mettre à jour le statut du jeton du client et est terminée dans DynamoDB.

Expiration de la position dans la file

15. Lorsque cette fonctionnalité est activée, l'ID de demande correspondant à une position de file d'attente particulière est éligible pour générer un jeton uniquement pendant un intervalle de temps spécifié.

Augmentez le compteur de service à l'expiration de la position de la file d'attente :

16. Lorsque cette fonctionnalité est activée, le compteur de service est automatiquement incrémenté en fonction des positions de file d'attente expirées qui n'ont pas pu générer de jetons.

# Composants de la solution

## API publiques et privées pour salles d'attente

L'objectif principal de la AWS solution Virtual Waiting Room est de contrôler la génération de jetons Web JSON (JWT) pour les clients de manière contrôlée afin d'éviter les rafales de nouveaux utilisateurs susceptibles de submerger le site Web de destination. Les JWT peuvent être utilisés pour protéger le site, empêcher l'accès aux pages Web jusqu'à ce que le jeton de salle d'attente soit obtenu, et également pour l'autorisation d'accès aux API.

Le modèle principal installe une API publique et une API privée (autorisée par l'IAM) utilisées pour la plupart des opérations de salle d'attente virtuelle. AWS L'API publique est configurée avec une CloudFront distribution avec plusieurs politiques de mise en cache basées sur le chemin de l'API. Une table DynamoDB EventBridge et un bus d'événements sont créés. Le modèle ajoute un nouveau VPC avec deux zones de disponibilité (AZ), un cluster ElastiCache pour Redis dans les deux zones de disponibilité, et plusieurs fonctions Lambda. Les fonctions Lambda qui interagissent avec Redis ont ElastiCache des interfaces réseau au sein du VPC et toutes les autres fonctions Lambda disposent d'une connectivité réseau par défaut. Les API de base constituent le niveau d'interaction le plus bas avec la solution. Les autres fonctions Lambda, l'instance Amazon Elastic Compute Cloud (Amazon EC2) et les conteneurs peuvent agir comme des extensions et appeler les API principales pour créer des salles d'attente, contrôler le trafic entrant et réagir aux événements générés par la solution.

En outre, la pile principale crée une alarme pour toutes ses erreurs de fonction Lambda et ses conditions d'accélération, ainsi que des alarmes pour chaque déploiement d'API Gateway pour les codes d'état 4XX et 5XX.





7. La fonction `GetQueueNumber` Lambda récupère et renvoie la position numérique du client dans la file d'attente pour Redis. `ElastiCache`
8. La fonction `GetServingNumber` Lambda récupère et renvoie le numéro actuellement desservi par la salle d'attente pour Redis. `ElastiCache`
9. La fonction `GetWaitingNum` Lambda renvoie le numéro actuellement en file d'attente dans la salle d'attente et pour lequel aucun jeton n'a encore été émis.
10. Les points de terminaison VPC permettent aux fonctions Lambda du VPC de communiquer avec les services de la solution.
11. `ElastiCache` for Redis cluster stocke toutes les demandes d'entrée dans la salle d'attente avec un identifiant d'événement valide. Il stocke également plusieurs compteurs tels que le nombre de demandes mises en file d'attente, le nombre de demandes actuellement traitées, le nombre de jetons générés, le nombre de sessions terminées et le nombre de sessions abandonnées.
12. Ressources d'API privées API Gateway pour prendre en charge les fonctions administratives. Les API privées sont authentifiées AWS par IAM.
13. La fonction `GetExpiredTokens` Lambda renvoie une liste d'identifiants de demande contenant des jetons expirés.
14. La fonction `AuthGenerateToken` Lambda génère un jeton pour une demande valide qui a été autorisée à terminer sa transaction sur le site cible. L'émetteur et la période de validité d'un jeton initialement définis lors du déploiement du core stack peuvent être annulés. Il écrit un événement dans le bus d'événements personnalisé de la salle d'attente indiquant qu'un jeton a été généré. Si un jeton a déjà été généré pour cette demande, aucun nouveau jeton n'est généré.
15. La fonction `IncrementServingCounter` Lambda incrémente le compteur de service de la salle d'attente stocké dans Redis en fonction `ElastiCache` d'un incrément par valeur.
16. La fonction `GetNumActiveTokens` Lambda interroge DynamoDB pour connaître le nombre de jetons qui n'ont pas encore expiré, qui n'ont pas été utilisés pour terminer sa transaction et qui n'ont pas été marqués comme abandonnés.
17. La fonction `ResetState` Lambda réinitialise tous les compteurs enregistrés dans Redis. `ElastiCache` Il supprime et recrée également les `tablesTokenTable`, `QueuePositionEntryTime`, et `DynamoDBServingCounterIssuedAt`. En outre, il effectue l'invalidation CloudFront du cache.
18. La fonction `UpdateSession` Lambda met à jour le statut d'une session (jeton) stockée dans la table `DynamoDBTokenTable`. L'état de la session est indiqué par un entier. Les sessions définies sur le statut 1 indiquent qu'elles sont terminées et -1 qu'elles sont abandonnées. Il écrit

- un événement dans le bus d'événements personnalisé de la salle d'attente indiquant qu'une session a été mise à jour.
- 19 La table `TokenTable` DynamoDB stocke les données des jetons.
- 20 La table `QueuePositionEntryTime` DynamoDB stocke les données relatives à la position de la file d'attente et à l'heure d'entrée.
- 21 La table `ServingCounterIssuedAt` DynamoDB stocke les mises à jour du compteur de service.
- 22 La fonction `GetQueuePositionExpireTime` Lambda est invoquée lorsque le client demande l'heure d'expiration de la position de file d'attente restante.
- 23 La fonction `SetMaxQueuePositionExpired` Lambda définit la position maximale de la file d'attente expirée correspondant aux valeurs de la `ServingCounterIssuedAt` table. Il s'exécute toutes les minutes si le `IncrSvcOnQueuePositionExpiry` paramètre est défini sur `true` lors du déploiement du Core Stack.
- 24 La fonction `GenerateEvents` Lambda écrit diverses métriques de salle d'attente dans le bus d'événements personnalisé de la salle d'attente. Il est exécuté toutes les minutes si le paramètre `Enable Events Generation` est défini sur `true` lors du déploiement du Core Stack.
- 25 AWS Secrets Manager stocke les clés pour les opérations liées aux jetons et autres données sensibles.
- 26 Le bus d'événements `EventBridge` personnalisé Amazon reçoit un événement chaque fois qu'un jeton est généré et qu'une session est mise à jour dans la table `TokenTable` DynamoDB. Il reçoit également des événements lorsque le compteur de service est déplacé dans le `SetMaxQueuePositionExpired` Lambda. Il est écrit avec diverses métriques de salle d'attente, s'il est activé lors du déploiement du Core Stack.
- 27 La règle `CloudWatch` d'événement Amazon est créée si le paramètre `Enable Events Generation` est défini sur `true` lors du déploiement de Core Stack. Cette règle d'événement lance la fonction `GenerateEvents` Lambda toutes les minutes.

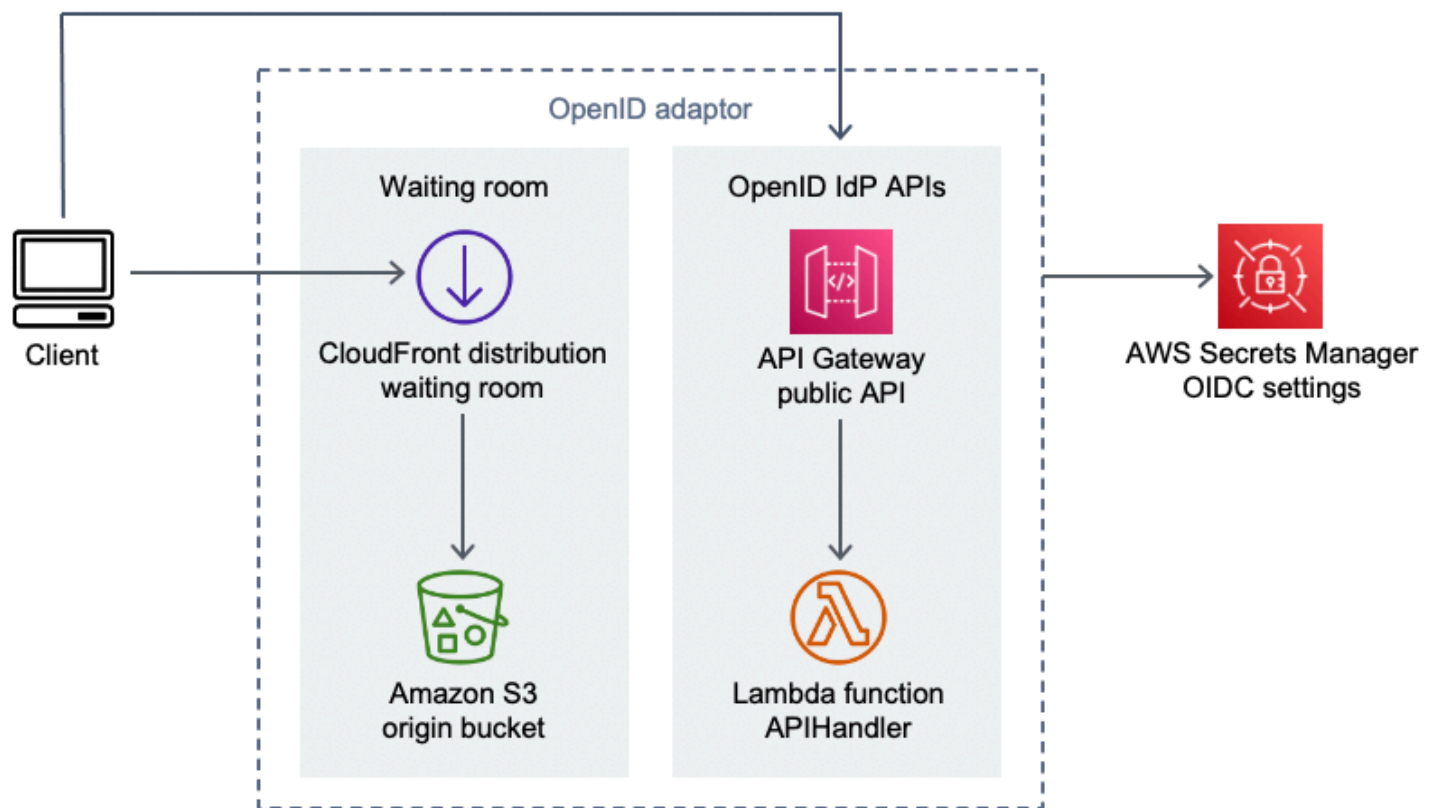
## Mécanismes d'autorisation

La solution inclut une pile d'autorisateurs Lambda API Gateway. La pile se compose d'un rôle IAM et d'une fonction Lambda. La fonction `APIGatewayAuthorizer` Lambda est un autorisateur pour API Gateway qui peut valider la signature et les revendications d'un jeton émis par la Virtual Waiting Room on API. AWS La fonction Lambda fournie avec la pile peut être utilisée pour protéger les API cloud jusqu'à ce qu'un utilisateur ait franchi la salle d'attente et reçoive un jeton d'accès. L'autorisateur récupère et met automatiquement en cache la clé publique et la configuration à partir

de l'API principale pour la vérification des jetons. Il peut être utilisé sans modification et peut être installé dans toutes les AWS régions compatibles AWS Lambda.

## Adaptateur OpenID

La pile d'[adaptateurs OpenID](#) déploie une API Gateway et des fonctions Lambda qui agissent en tant que fournisseur d'identité OpenID. L'adaptateur OpenID fournit un ensemble d'API compatibles OIDC qui peuvent être utilisées avec les logiciels d'hébergement Web existants qui prennent en charge les fournisseurs d'identité OIDC, tels que les AWS Elastic Load Balancers WordPress, ou en tant que fournisseur d'identité fédéré pour Amazon Cognito ou un service similaire. L'adaptateur permet à un client d'utiliser la salle d'attente dans le flux Authn/Authz lorsqu'il utilise un logiciel d'hébergement off-the-shelf Web avec des options d'intégration limitées. La pile installe également une CloudFront distribution avec un compartiment Amazon S3 comme origine et un autre compartiment S3 pour les requêtes de journalisation. L'adaptateur OpenID fournit un exemple de page de salle d'attente, similaire à celle fournie dans la pile d'exemples de salles d'attente, mais conçue pour un flux d'authentification OpenID. Le processus d'authentification consiste à obtenir une position dans la file d'attente de la salle d'attente et à attendre que la position de service soit égale ou supérieure à celle du client dans la file d'attente. La page de la salle d'attente OpenID redirige vers le site cible, qui utilise l'API OpenID pour terminer l'acquisition du jeton et la configuration de session pour le client. Les points de terminaison de l'API de cette solution correspondent directement à la spécification de flux name-for-name officielle OpenID Connect 1.0,. Reportez-vous à la section [Authentification OpenID Connect Core 1.0](#) pour plus de détails.



## Salle d'attente virtuelle sur le AWS composant adaptateur OpenID

1. CloudFront la distribution fournit le contenu du compartiment S3 à l'utilisateur.
2. Le compartiment S3 héberge des exemples de pages de salle d'attente.
3. L'API Amazon API Gateway fournit un ensemble d'API compatibles OIDC qui peuvent être utilisées avec les logiciels d'hébergement Web existants qui prennent en charge la fonction d'autorisation Lambda du fournisseur d'identité OIDC.
4. La fonction APIHandler Lambda gère les demandes pour tous les chemins de ressources d'API Gateway. Différentes fonctions Python au sein d'un même module sont mappées à chaque chemin d'API. Par exemple, le chemin de /authorize ressource dans API Gateway est invoqué authorize() dans la fonction Lambda.
5. Les paramètres OIDC sont stockés dans Secrets Manager.

## Stratégies d'entrée d'échantillons

Les stratégies d'entrée déterminent à quel moment le comptoir de service de la solution doit être déplacé pour accueillir un plus grand nombre d'utilisateurs sur le site cible. Pour plus d'informations

conceptuelles sur les stratégies d'entrée dans les salles d'attente, reportez-vous à la section [Considérations relatives à la conception](#).

La solution propose deux stratégies d'entrée d'échantillons : MaxSizeet périodique.



Composante des stratégies de salle d'attente virtuelle sur AWS Inlet

Option de stratégie d'entrée Max Size :

1. Un client émet une notification Amazon SNS qui invoque la fonction MaxSizeInlet Lambda pour augmenter le compteur de service en fonction de la charge utile des messages.
2. La fonction MaxSizeInlet Lambda s'attend à recevoir un message indiquant qu'elle l'utilise pour déterminer dans quelle mesure le compteur de service doit être incrémenté.

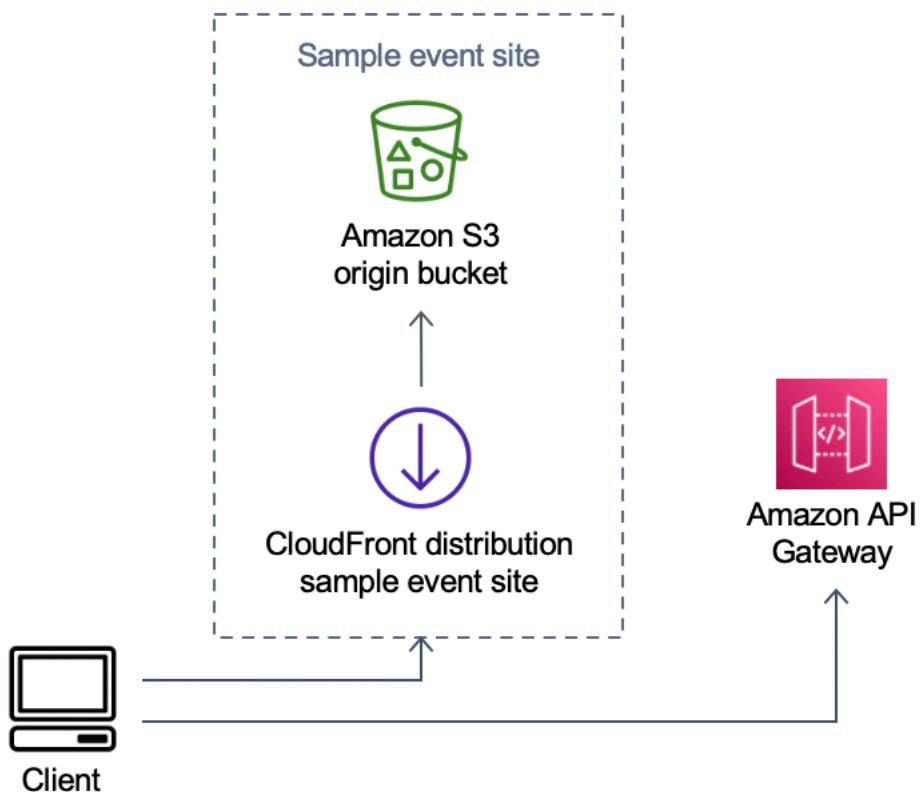
Option de stratégie d'entrée périodique :

3. Une CloudWatch règle invoque une fonction Lambda toutes les minutes pour augmenter le compteur de service d'une quantité fixe.
4. La fonction PeriodicInlet Lambda incrémente le compteur de service de la taille donnée si le temps se situe entre l'heure de début et l'heure de fin indiquée. Facultativement, il vérifie une CloudWatch alarme et, si l'alarme est activeOK, effectue l'incrément, sinon elle l'ignore.

## Exemple de salle d'attente

L'exemple de salle d'attente s'intègre aux API publiques et privées en plus de l'autorisateur personnalisé pour démontrer une solution de salle end-to-end d'attente minimale. La page Web principale est stockée dans un compartiment S3 et utilisée comme origine pour CloudFront. Il guide l'utilisateur à travers les étapes suivantes :

1. Faites la queue dans la salle d'attente pour accéder au site.
2. Obtenez la position du client dans la ligne.
3. Obtenez le poste de service de la salle d'attente.
4. Obtenez un jeu de jetons une fois que le poste de service est égal ou supérieur à celui du client.
5. Utilisez le jeton pour appeler une API protégée par l'autorisateur Lambda.



Salle d'attente virtuelle sur AWS un exemple de composant du site de l'événement

1. Le compartiment S3 héberge les exemples de contenu pour la salle d'attente et le panneau de commande.
2. CloudFront la distribution fournit le contenu du compartiment S3 à l'utilisateur.
3. Exemple de déploiement d'API Gateway avec des chemins de ressources de type shopping tels que `/search` `/checkout` Cette API est installée par la pile et configurée avec l'autorisateur de jetons. Il s'agit d'un exemple de moyen simple de protéger une API avec la salle d'attente. Les demandes présentant un jeton valide sont transmises au Lambda, sinon une erreur est renvoyée. L'API ne comporte aucune fonctionnalité autre que la réponse de la fonction Lambda attachée.

# Sécurité

Lorsque vous créez des systèmes sur une AWS infrastructure, les responsabilités en matière de sécurité sont partagées entre vous et AWS. Ce [modèle partagé](#) réduit votre charge opérationnelle car il AWS exploite, gère et contrôle les composants, notamment le système d'exploitation hôte, la couche de virtualisation et la sécurité physique des installations dans lesquelles les services fonctionnent. Pour plus d'informations sur AWS la sécurité, consultez [AWS Cloud Security](#).

ElastiCache pour Redis se voit attribuer une interface réseau au sein du VPC privé. Les fonctions Lambda qui interagissent avec Redis se voient également attribuer ElastiCache des interfaces réseau au sein d'un VPC. Toutes les autres ressources disposent d'une connectivité réseau dans l'espace AWS réseau partagé. Les fonctions Lambda dotées d'interfaces VPC qui interagissent avec d'autres services AWS utilisent des points de terminaison VPC pour se connecter à ces services.

Les clés publiques et privées utilisées pour créer et valider les jetons Web JSON sont générées au moment du déploiement et stockées dans Secrets Manager. Le mot de passe utilisé ElastiCache pour se connecter à Redis est également généré au moment du déploiement et stocké dans Secrets Manager. La clé privée et le mot ElastiCache de passe Redis ne sont accessibles via aucune API de solution.

L'API publique doit être accessible via CloudFront. La solution génère une clé d'API pour API Gateway, qui est utilisée comme valeur d'un en-tête personnalisé `-api-key`, dans CloudFront. CloudFront inclut cet en-tête lors des demandes d'origine. Pour plus de détails, consultez la section [Ajout d'en-têtes personnalisés aux demandes d'origine](#) dans le manuel Amazon CloudFront Developer Guide.

Les API privées sont configurées pour nécessiter une autorisation AWS IAM pour l'invocation. La solution crée le groupe d'utilisateurs `ProtectedAPIGroup` IAM avec les autorisations appropriées pour appeler les API privées. Un utilisateur IAM ajouté à ce groupe est autorisé à appeler les API privées.

Les politiques IAM utilisées dans les rôles et les autorisations attachés aux différentes ressources créées par la solution n'accordent que les autorisations requises pour effectuer les tâches nécessaires.

Pour les ressources telles que les compartiments S3, les files d'attente SQS et les rubriques SNS générées par la solution, le chiffrement au repos et pendant le transit est activé dans la mesure du possible.



## Surveillance

La pile d'API principale comprend plusieurs CloudWatch alarmes qui peuvent être surveillées pour détecter les problèmes pendant le fonctionnement de la solution. La pile crée une alarme en cas d'erreur de fonction Lambda et de conditions d'accélérateur, et fait passer l'état de l'alarme OK à ALARM si une erreur ou une condition d'accélérateur survient sur une période d'une minute.

La pile crée également des alarmes pour chaque déploiement d'API Gateway pour les codes d'état 4XX et 5XX. L'alarme passe de l'état OK à ALARM si un code d'état 4XX ou 5XX est renvoyé par l'API dans un délai d'une minute.

Ces alarmes reprennent OK leur état après une minute sans erreur ni accélération.

## Rôles IAM

AWS Identity and Access Management Les rôles (IAM) permettent aux clients d'attribuer des politiques d'accès et des autorisations détaillées aux services et aux utilisateurs sur le AWS cloud. Cette solution crée des rôles IAM qui accordent aux AWS Lambda fonctions de la solution l'accès pour créer des ressources régionales.

## Amazon CloudFront

Le `virtual-waiting-room-on-aws.template` CloudFormation modèle, qui crée les principales API publiques et privées de la salle d'attente, déploie également une CloudFront distribution pour l'API publique. CloudFront met en cache les réponses de l'API publique, réduisant ainsi la charge sur API Gateway et les fonctions Lambda qui exécutent leur travail.

Cette solution propose également un exemple de modèle de salle d'attente optionnel qui déploie une application Web simple [hébergée](#) dans un bucket Amazon Simple Storage Service (Amazon S3). Pour réduire la latence et améliorer la sécurité, une CloudFront distribution Amazon est déployée avec une identité d'accès d'origine, c'est-à-dire un CloudFront utilisateur fournissant un accès public au contenu du bucket du site Web de la solution. Pour plus d'informations, reportez-vous à la section [Restreindre l'accès au contenu Amazon S3 à l'aide d'une identité d'accès d'origine](#) dans le manuel Amazon CloudFront Developer Guide.

## Groupes de sécurité

Les [groupes de sécurité VPC](#) créés dans cette solution sont conçus pour contrôler et isoler le trafic réseau vers Redis ElastiCache . Les Lambdas qui doivent communiquer avec le ElastiCache for Redis sont placés dans le même groupe de sécurité que celui du ElastiCache for Redis. Nous vous recommandons de passer en revue les groupes de sécurité et de restreindre davantage l'accès, le cas échéant, une fois le déploiement terminé.

# Considérations relatives à la conception

## Options de déploiement

S'il s'agit de la première installation ou si vous ne savez pas quoi installer, déployez le CloudFormation modèle `virtual-waiting-room-on-aws-getting-started.template` imbriqué, qui installe le noyau, les autorisateurs et des exemples de modèles de salle d'attente. Vous disposez ainsi d'une salle d'attente minimale avec un flux simple.

## Protocoles pris en charge

La AWS solution Virtual Waiting Room on peut être intégrée aux éléments suivants :

- Bibliothèques et outils de vérification des jetons Web JSON
- Déploiements d'API Gateway existants
- Clients de l'API REST
- Clients et fournisseurs OpenID

## Stratégies d'entrée dans les salles d'attente

Les stratégies d'entrée encapsulent la logique et les données nécessaires pour déplacer les clients de la salle d'attente vers le site Web. Une stratégie d'entrée peut être implémentée sous la forme d'une fonction Lambda, d'un conteneur, d'une instance Amazon EC2 ou de toute autre ressource de calcul. Il n'est pas nécessaire qu'il s'agisse d'une ressource cloud tant qu'elle peut appeler les API publiques et privées de la salle d'attente. La stratégie d'entrée reçoit des événements concernant la salle d'attente, le site Web ou d'autres indicateurs extérieurs qui l'aident à décider quand un plus grand nombre de clients peuvent se faire émettre des jetons et accéder au site. Il existe plusieurs approches en matière de stratégies d'admission. Le choix que vous adopterez dépend des ressources mises à votre disposition et des contraintes liées à la conception du site Web à protéger.

La principale action entreprise par la stratégie d'entrée consiste à appeler l'API privée `increment_serving_num` Amazon API Gateway avec une valeur relative indiquant le nombre de clients supplémentaires autorisés à accéder au site. Cette section décrit deux stratégies d'entrée

d'échantillons. Ils peuvent être utilisés tels quels, personnalisés ou vous pouvez utiliser une approche complètement différente.

## MaxSize

À l'aide de MaxSize cette stratégie, la fonction `MaxSizeInlet` Lambda est configurée avec le nombre maximum de clients pouvant utiliser le site Web simultanément. Il s'agit d'une valeur fixe. Un client émet une notification Amazon SNS qui invoque la fonction `MaxSizeInlet` Lambda pour augmenter le compteur de service en fonction de la charge utile des messages. La source du message SNS peut provenir de n'importe où, y compris du code du site Web ou d'un outil de surveillance qui observe le niveau d'utilisation du site.

La fonction `MaxSizeInlet` Lambda s'attend à recevoir un message qui peut inclure :

- `exited` : nombre de transactions terminées
- liste des identifiants de demandes à marquer comme terminés
- liste des identifiants de demandes à marquer comme abandonnés

Ces données sont utilisées pour déterminer dans quelle mesure le compteur de service doit être incrémenté. Il peut arriver qu'il n'y ait pas de capacité supplémentaire pour augmenter le compteur, en fonction du nombre actuel de clients.

## Périodique

Lorsque vous utilisez la stratégie périodique, une CloudWatch règle invoque la fonction `PeriodicInlet` Lambda toutes les minutes pour augmenter le compteur de service d'une quantité fixe. L'entrée périodique est paramétrée avec l'heure de début, l'heure de fin et le montant de l'incrément. Facultativement, cette stratégie vérifie également une CloudWatch alarme et, si l'alarme est OK active, elle effectue l'incrément, sinon elle l'ignore. Les intégrateurs du site peuvent connecter une métrique d'utilisation à une alarme et utiliser cette alarme pour suspendre l'entrée périodique. Cette stratégie ne modifie la position de service que lorsque l'heure actuelle se situe entre les heures de début et de fin, et éventuellement, l'alarme spécifiée est dans l'OK état.

## Personnalisation et extension de la solution

L'administrateur du site de votre organisation doit décider des méthodes d'intégration à utiliser avec la salle d'attente. Deux options s'offrent à vous :

1. Intégration de base directement à l'aide des API et des autorisateurs d'API Gateway.
2. Intégration d'OpenID via un fournisseur d'identité.

Outre l'intégration ci-dessus, vous devrez peut-être configurer la redirection du nom de domaine. Vous êtes également responsable du déploiement d'une page de site de salle d'attente personnalisée.

La AWS solution Virtual Waiting Room on est conçue pour être étendue grâce à deux mécanismes : EventBridge pour la notification unidirectionnelle des événements et les API REST pour la communication bidirectionnelle.

## Quotas

La principale limite d'échelle pour Virtual Waiting Room on AWS est la limite d'accélération Lambda pour la région installée. AWS Lorsque elle est installée sur un AWS compte avec le quota d'exécution simultanée Lambda par défaut, la AWS solution Virtual Waiting Room on peut gérer jusqu'à 500 clients par seconde demandant une position dans la file d'attente. Le taux de 500 clients par seconde est basé sur le fait que toutes les limites de quotas simultanés de la fonction Lambda sont disponibles exclusivement dans la solution. Si la région du compte est partagée avec d'autres solutions qui invoquent des fonctions Lambda, la salle d'attente virtuelle de la AWS solution doit disposer d'au moins 1 000 appels simultanés disponibles. Vous pouvez utiliser CloudWatch des métriques pour cartographier les appels Lambda simultanés sur votre compte au fil du temps afin de prendre une décision. Vous pouvez utiliser la [console Service Quotas](#) pour demander des augmentations. L'augmentation de la limite Lambda n'augmente les frais mensuels du compte que si des appels supplémentaires se produisent réellement.

Pour chaque 500 clients supplémentaires par seconde, augmentez votre limite d'accélération de 1 000.

Nombre d'utilisateurs entrants par seconde attendus	Quota d'exécution simultanée recommandé
0 à 500	1 000 (par défaut)
501 à 1 000	2 000
1 001 à 1 500	3 000

Lambda a une limite de rafale fixe de 3 000 appels simultanés. Pour plus d'informations, reportez-vous à la section Mise à l'[échelle des fonctions Lambda](#). Le code client doit attendre et réessayer certains appels d'API si un code d'erreur est renvoyé indiquant une situation d'accélération temporaire. L'exemple de client de salle d'attente inclut ce code à titre d'exemple de conception de clients utilisés lors d'événements de grande capacité et de forte rafale.

Cette solution est également compatible avec la simultanéité réservée et provisionnée Lambda avec des étapes de configuration personnalisées. Pour plus de détails, reportez-vous à la section [Gestion de la simultanéité réservée Lambda](#).

La limite supérieure d'utilisateurs pouvant entrer dans la salle d'attente, recevoir un jeton et poursuivre une transaction est limitée par la limite supérieure ElastiCache de quatre compteurs Redis. Les compteurs sont utilisés pour indiquer le poste de service de la salle d'attente et pour suivre l'état récapitulatif de la solution. Les compteurs utilisés dans Redis ont une limite supérieure ElastiCache de 9 223 372 036 854 775 807. Une table DynamoDB est utilisée pour stocker une copie de chaque jeton délivré à un utilisateur de la salle d'attente. DynamoDB n'impose aucune limite pratique quant à la taille d'une table.

## Déploiements régionaux

Les services utilisés par cette solution sont pris en charge dans toutes les AWS régions. Pour connaître la disponibilité la plus récente des AWS services par région, consultez la [liste des services AWS régionaux](#).

# AWS CloudFormation modèles

Pour automatiser le déploiement, cette solution utilise les AWS CloudFormation modèles suivants, que vous pouvez télécharger avant le déploiement.

S'il s'agit de la première installation ou si vous ne savez pas quoi installer, déployez le `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation modèle, qui installe le noyau, les autorisateurs et des exemples de modèles de code de salle d'attente. Cela vous permet de tester une salle d'attente fonctionnelle avec un flux simple.

[View template](#)

[virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template](#) : utilisez ce modèle pour ajouter un ARN de rôle par défaut à API Gateway au niveau du compte pour les autorisations de journalisation. CloudWatch Reportez-vous à la section [Conditions préalables](#) pour savoir si votre compte nécessite le déploiement de ce modèle ou non.

[View template](#)

[virtual-waiting-room-on-aws-getting-started.template](#) : utilisez ce modèle imbriqué pour installer le noyau, les autorisateurs et des exemples de piles de salles d'attente.

[View template](#)

[virtual-waiting-room-on-aws.template](#) : utilisez ce modèle de base pour installer les principales API REST publiques et privées et les services cloud pour créer des événements de salle d'attente. Installez ce modèle dans le compte et dans la région où vous avez besoin des API REST de la salle d'attente, ElastiCache pour Redis, et de la table DynamoDB.

[View template](#)

[virtual-waiting-room-on-aws-authorizers.template](#) : utilisez ce modèle pour installer l'autorisateur Lambda conçu pour vérifier les jetons émis par les salles d'attente et destiné à protéger les API des utilisateurs finaux. Nécessite la pile principale. Certaines sorties de la pile principale sont nécessaires en tant que paramètres pour déployer cette pile. Il s'agit d'un modèle facultatif.

**View template**

virtual-

[waiting-room-on-aws-openid.template](#) : utilisez ce modèle pour installer un fournisseur d'identité OpenID afin d'intégrer les interfaces d'autorisation dans les salles d'attente. Nécessite la pile principale. Certaines sorties de la pile principale sont nécessaires pour déployer cette pile. Il s'agit d'un modèle facultatif.

**View template**

virtual-

[waiting-room-on-aws-sample-inlet-strategy.template](#) : utilisez ce modèle pour installer des stratégies de saisie d'échantillons destinées à être utilisées entre un site cible et la salle d'attente. Les stratégies d'entrée aident à encapsuler la logique afin de déterminer quand autoriser un plus grand nombre d'utilisateurs à accéder au site cible. Nécessite la pile principale. Les sorties de la pile principale sont nécessaires pour déployer cette pile. Il s'agit d'un modèle facultatif.

**View template**

virtual-

[waiting-room-on-aws-sample.template](#) : utilisez ce modèle pour installer un exemple de configuration minimale de Web et d'API Gateway pour une salle d'attente et un site cible. Nécessite le noyau et les piles d'autorisations. Les sorties des piles de base et d'autorisation sont requises en tant que paramètres pour déployer cette pile. Il s'agit d'un modèle facultatif.



# Déploiement automatique

Avant de lancer la solution, examinez le coût, l'architecture, la sécurité du réseau et les autres considérations abordées dans ce guide. Suivez les step-by-step instructions de cette section pour configurer et déployer la solution dans votre compte.

Temps de déploiement : environ 30 minutes (pile de démarrage uniquement)

## Prérequis

- AWS autorisations de console de compte équivalentes à [l'accès administrateur](#).
- Activez la CloudWatch journalisation depuis API Gateway :
  - Connectez-vous à la [console API Gateway](#) et sélectionnez la région dans laquelle vous souhaitez installer les stacks.

Si des API sont déjà définies dans cette région :

1. Sélectionnez n'importe quelle API.
2. Dans le menu de navigation de gauche, sélectionnez Réglages.
3. Vérifiez la présence d'une valeur dans le champ ARN du rôle du CloudWatch journal.

- S'il n'y a pas d'ARN, installez le [virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template](#).
- S'il existe un ARN, commencez par [lancer la pile de démarrage](#).

Si aucune API n'est définie dans cette région, installez le [virtual-waiting-room-on-aws-api-gateway-cw-logs-role.template](#).

- Connaissance de l'architecture et des détails de mise en œuvre du site cible à protéger.

## Vue d'ensemble du déploiement

Suivez les étapes ci-dessous pour déployer cette solution sur AWS. Pour obtenir des instructions détaillées, suivez les liens pour chaque étape.

### [Étape 1. Lancez la pile de démarrage](#)

- Lancez le AWS CloudFormation modèle dans votre AWS compte.

- Passez en revue les paramètres des modèles et entrez ou ajustez les valeurs par défaut selon vos besoins.

## Étape 2. (Facultatif) Testez la salle d'attente

- Générez des AWS clés pour appeler les API sécurisées IAM.
- Ouvrez le panneau de commande de la salle d'attente des échantillons.
- Testez l'exemple de salle d'attente.

## Étape 1. Lancez la pile de démarrage

Ce AWS CloudFormation modèle automatisé déploie le noyau, les autorisateurs et des exemples de modèles de salle d'attente, ce qui vous permet de visualiser et de tester une salle d'attente fonctionnelle. Vous devez lire et comprendre les prérequis avant de lancer la pile.

### Note

Vous êtes responsable du coût des AWS services utilisés lors de l'exécution de cette solution. Pour plus de détails, consultez la section [Coût](#) de ce guide et consultez la page Web de tarification de chaque AWS service utilisé dans cette solution.

1. Connectez-vous au modèle [AWS Management Console](#) et sélectionnez le bouton pour lancer le `virtual-waiting-room-on-aws-getting-started.template` AWS CloudFormation modèle.



Vous

pouvez également [télécharger le modèle](#) comme point de départ pour votre propre implémentation.

2. Le modèle est lancé par défaut dans la région USA Est (Virginie du Nord). Pour lancer la solution dans une autre AWS région, utilisez le sélecteur de région dans la barre de navigation de la console.
3. Sur la page Create stack, vérifiez que l'URL du modèle est correcte dans la zone de texte URL Amazon S3 et choisissez Next.

4. Sur la page Spécifier les détails de la pile, attribuez un nom à votre pile de solutions. Pour plus d'informations sur les limites de dénomination des caractères, reportez-vous aux [limites IAM et STS](#) du guide de l'AWS Identity and Access Management utilisateur.
5. Sous Paramètres, passez en revue les paramètres de ce modèle de solution et modifiez-les si nécessaire. Cette solution utilise les valeurs par défaut suivantes.

Paramètre	Par défaut	Description
ID de l'événement	Sample	ID unique pour cette instance de salle d'attente, format GUID suggéré.
Période de validité	3600	Durée de validité du jeton en secondes.
Activer la génération d'événements	false	Si ce paramètre est défini sur <code>true</code> , les métriques relatives à la salle d'attente sont enregistrées dans son bus d'événements toutes les minutes
Port Redis	1785	Le numéro de port à utiliser pour la connexion au ElastiCache serveur Redis. Il est recommandé de ne pas utiliser le port Redis par défaut ElastiCache de 6379.
EnableQueuePositionExpiry	true	Si ce paramètre est défini sur <code>false</code> , la période d'expiration de la position de la file d'attente n'est pas appliquée.
QueuePositionExpiryPeriod	900	Il s'agit de l'intervalle de temps en secondes au-delà duquel une position de file

Paramètre	Par défaut	Description
		d'attente n'est pas éligible pour générer un jeton.
IncrSvcOnQueuePositionExpiry	false	S'il est défini sur true, le compteur de service est automatiquement avancé en fonction des positions de file d'attente expirées qui n'ont pas généré de jetons avec succès.

6. Choisissez Next (Suivant).
7. Sur la page Configurer les options de pile, choisissez Suivant.
8. Sur la page Vérification, vérifiez et confirmez les paramètres. Cochez la case indiquant que le modèle crée des ressources AWS Identity and Access Management (IAM).
9. Sélectionnez Create stack (Créer une pile) pour déployer la pile.

Vous pouvez consulter l'état de la pile dans la AWS CloudFormation console dans la colonne État. Vous devriez recevoir le statut CREATE\_COMPLETE dans environ 30 minutes.

## Étape 2. (Facultatif) Testez la salle d'attente

Si vous avez déployé la pile de démarrage, les étapes suivantes vous aideront à tester les fonctionnalités de la salle d'attente. Pour terminer les tests, vous avez besoin de AWS clés avec des autorisations pour appeler les API sécurisées IAM de la pile principale.

### Générez des AWS clés pour appeler les API sécurisées IAM

1. [Créez](#) ou utilisez un utilisateur IAM dans le AWS compte sur lequel le `aws-virtual-waiting-room-getting-started.template` CloudFormation modèle a été déployé.
2. Accordez à l'[utilisateur IAM un accès programmatique](#). Lorsque vous créez un nouvel ensemble de clés d'accès pour l'utilisateur IAM, téléchargez le fichier clé lorsqu'il est présenté. Vous avez besoin de l'ID de clé d'accès et de la clé d'accès secrète de l'utilisateur IAM pour tester la salle d'attente.
3. [Ajoutez l'utilisateur IAM au groupe d'utilisateurs IAM ProtectedApiGroup créé par le modèle.](#)

## Ouvrez le panneau de commande de la salle d'attente d'échantillons

1. Connectez-vous à la [AWS CloudFormation console](#) et sélectionnez la pile de démarrage de la solution.
2. Choisissez l'onglet Outputs.
3. Dans la colonne Clé, recherchez ControlPanell'URL et sélectionnez la valeur correspondante.
4. Ouvrez le panneau de commande dans un nouvel onglet ou une nouvelle fenêtre de navigateur.
5. Dans le panneau de commande, développez la section Configuration.
6. Entrez l'ID de clé d'accès et la clé d'accès secrète que vous avez récupérés dans [AWS Generate keys pour appeler les API sécurisées IAM](#). Les points de terminaison et l'ID d'événement sont renseignés à partir des paramètres de l'URL.
7. Choisissez Utiliser. Le bouton s'active une fois que vous avez fourni les informations d'identification.

## Testez l'exemple de salle d'attente

1. Dans la [AWS CloudFormation console](#), sélectionnez la pile de démarrage de la solution.
2. Choisissez l'onglet Outputs.
3. Dans la colonne Clé, recherchez WaitingRooml'URL et sélectionnez la valeur correspondante.
4. Ouvrez la salle d'attente, puis choisissez Réserver pour accéder à la salle d'attente.
5. Retournez à l'onglet du navigateur qui contient le panneau de configuration.
6. Sous Incrément Serving Counter, sélectionnez Modifier. Cela permet à 100 utilisateurs de passer de la salle d'attente au site cible.
7. Retournez dans la salle d'attente et choisissez Check out now ! Vous allez maintenant être redirigé vers le site cible.
8. Choisissez Acheter maintenant pour terminer votre transaction sur le site cible.

# Déploiement de piles distinctes

La pile principale est la seule pile requise pour obtenir les fonctionnalités principales de la salle d'attente. Toutes les autres piles sont facultatives. Lancez la pile des autorisateurs si vous ne disposez pas déjà d'un moyen de valider les jetons émis par les salles d'attente ou de protéger les API que vous possédez déjà. Lancez la pile OpenID si vous avez besoin d'un fournisseur d'identité OpenID pour intégrer les interfaces d'autorisation dans les salles d'attente. La pile de stratégies de saisie d'échantillons fournit quelques exemples expliquant comment et quand autoriser un plus grand nombre d'utilisateurs à accéder au site que vous essayez de protéger.

## 1. Lancez le core stack

Temps de déploiement : environ 20 minutes

Ce AWS CloudFormation modèle automatisé déploie Virtual Waiting Room AWS dans le AWS cloud. Vous devez remplir les [prérequis](#) avant de lancer la pile.

### Note

Vous êtes responsable du coût des AWS services utilisés lors de l'exécution de cette solution. Pour plus de détails, consultez la section [Coût](#) de ce guide et consultez la page Web de tarification de chaque AWS service utilisé dans cette solution.

1. Connectez-vous au modèle [AWS Management Console](#) et sélectionnez le bouton pour lancer le `aws-virtual-waiting-room-on-aws.template` AWS CloudFormation modèle.



Vous

pouvez également [télécharger le modèle](#) comme point de départ pour votre propre implémentation.

2. Le modèle est lancé par défaut dans la région USA Est (Virginie du Nord). Pour lancer la solution dans une autre AWS région, utilisez le sélecteur de région dans la barre de navigation de la console.
3. Sur la page Create stack, vérifiez que l'URL du modèle est correcte dans la zone de texte URL Amazon S3 et choisissez Next.

4. Sur la page Spécifier les détails de la pile, attribuez un nom à votre pile de solutions. Pour plus d'informations sur les limites de dénomination des caractères, reportez-vous aux [limites IAM et STS](#) du guide de l'AWS Identity and Access Management utilisateur.
5. Sous Paramètres, passez en revue les paramètres de ce modèle de solution et modifiez-les si nécessaire. Cette solution utilise les valeurs par défaut suivantes.

Paramètre	Par défaut	Description
ID de l'événement	Sample	ID unique pour cette instance de la salle d'attente, format GUID suggéré.
Période de validité	3600	Durée de validité du jeton en secondes.
Activer la génération d'événements	false	Si ce paramètre est défini sur <code>true</code> , les métriques relatives à la salle d'attente sont enregistrées dans son bus d'événements toutes les minutes.
Port Redis	1785	Le numéro de port à utiliser pour la connexion au ElastiCache serveur Redis. Il est recommandé de ne pas utiliser le port Redis par défaut ElastiCache de 6379.
EnableQueuePositionExpiry	true	Si ce paramètre est défini sur <code>false</code> , la période d'expiration de la position de la file d'attente n'est pas appliquée.
QueuePositionExpiryPeriod	900	Il s'agit de l'intervalle de temps en secondes au-delà duquel une position de file

Paramètre	Par défaut	Description
		d'attente n'est pas éligible pour générer un jeton.
IncrSvcOnQueuePositionExpiry	false	S'il est défini sur true, le compteur de service est automatiquement avancé en fonction des positions de file d'attente expirées qui n'ont pas généré de jetons avec succès.

6. Choisissez Next (Suivant).
7. Sur la page Configurer les options de pile, choisissez Suivant.
8. Sur la page Vérification, vérifiez et confirmez les paramètres. Cochez la case indiquant que le modèle crée des ressources AWS Identity and Access Management (IAM).
9. Sélectionnez Create stack (Créer une pile) pour déployer la pile.

Vous pouvez consulter l'état de la pile dans la AWS CloudFormation console dans la colonne État. Vous devriez recevoir le statut CREATE\_COMPLETE dans environ 20 minutes.

## 2. (Facultatif) Lancez la pile d'autorisations

Temps de déploiement : environ 5 minutes

1. Connectez-vous au modèle [AWS Management Console](#) et sélectionnez le bouton pour lancer le `aws-virtual-waiting-room-on-aws-authorizers` template AWS CloudFormation modèle.

[Launch solution](#)

Vous

pouvez également [télécharger le modèle](#) comme point de départ pour votre propre implémentation.

2. Le modèle est lancé par défaut dans la région USA Est (Virginie du Nord). Pour lancer la solution dans une autre AWS région, utilisez le sélecteur de région dans la barre de navigation de la console.



3. Sur la page **Create stack**, vérifiez que l'URL du modèle est correcte dans la zone de texte URL Amazon S3 et choisissez **Next**.
4. Sur la page **Spécifier les détails de la pile**, attribuez un nom à votre pile de solutions. Pour plus d'informations sur les limites de dénomination des caractères, reportez-vous aux [limites IAM et STS](#) du guide de l'AWS Identity and Access Management utilisateur.
5. Sous **Paramètres**, passez en revue les paramètres de ce modèle de solution et modifiez-les si nécessaire. Cette solution utilise les valeurs par défaut suivantes.

Paramètre	Par défaut	Description
Point de terminaison d'API public	<Entrée obligatoire>	Point de terminaison public pour les API de la salle d'attente virtuelle.
Numéro d'événement de la salle d'attente	Sample	ID de l'événement de la salle d'attente.
URI de l'émetteur	<Entrée obligatoire>	URI de l'émetteur des clés publiques et des jetons.

6. Choisissez **Next (Suivant)**.
7. Sur la page **Configurer les options de pile**, choisissez **Suivant**.
8. Sur la page **Vérification**, vérifiez et confirmez les paramètres. Cochez la case indiquant que le modèle crée des ressources AWS Identity and Access Management (IAM).
9. Sélectionnez **Create stack (Créer une pile)** pour déployer la pile.

Vous pouvez consulter l'état de la pile dans la AWS CloudFormation console dans la colonne **État**. Vous devriez recevoir le statut **CREATE\_COMPLETE** dans environ cinq minutes.

### 3. (Facultatif) Lancez la pile OpenID

Temps de déploiement : environ 5 minutes

1. Connectez-vous au modèle [AWS Management Console](#) et sélectionnez le bouton pour lancer le `aws-virtual-waiting-room-on-aws-openid.template` AWS CloudFormation modèle.



Vous

pouvez également [télécharger le modèle](#) comme point de départ pour votre propre implémentation.

- Le modèle est lancé par défaut dans la région USA Est (Virginie du Nord). Pour lancer la solution dans une autre AWS région, utilisez le sélecteur de région dans la barre de navigation de la console.
- Sur la page Create stack, vérifiez que l'URL du modèle est correcte dans la zone de texte URL Amazon S3 et choisissez Next.
- Sur la page Spécifier les détails de la pile, attribuez un nom à votre pile de solutions. Pour plus d'informations sur les limites de dénomination des caractères, reportez-vous aux [limites IAM et STS](#) du guide de l'AWS Identity and Access Management utilisateur.
- Sous Paramètres, passez en revue les paramètres de ce modèle de solution et modifiez-les si nécessaire. Cette solution utilise les valeurs par défaut suivantes.

Paramètre	Par défaut	Description
Point de terminaison d'API public	<Entrée obligatoire>	URL du point de terminaison public pour les API de la salle d'attente virtuelle.
Point de terminaison d'API privé	<Entrée obligatoire>	URL du point de terminaison privé pour les API de la salle d'attente virtuelle.
Région de l'API	<Entrée obligatoire>	AWS nom de région pour les API des salles d'attente publiques et privées.
ID de l'événement	Sample	ID de l'événement de la salle d'attente.

- Choisissez Next (Suivant).
- Sur la page Configurer les options de pile, choisissez Suivant.
- Sur la page Vérification, vérifiez et confirmez les paramètres. Cochez la case indiquant que le modèle crée des ressources AWS Identity and Access Management (IAM).

## 9. Sélectionnez Create stack (Créer une pile) pour déployer la pile.

Vous pouvez consulter l'état de la pile dans la AWS CloudFormation console dans la colonne État. Vous devriez recevoir le statut CREATE\_COMPLETE dans environ cinq minutes.

## 4. (Facultatif) Lancez la pile de stratégies d'entrée d'échantillons

Temps de déploiement : environ deux minutes

1. Connectez-vous au modèle [AWS Management Console](#) et sélectionnez le bouton pour lancer le `aws-virtual-waiting-room-sample-inlet-strategy`.template AWS CloudFormation modèle.



Vous

pouvez également [télécharger le modèle](#) comme point de départ pour votre propre implémentation.

2. Le modèle est lancé par défaut dans la région USA Est (Virginie du Nord). Pour lancer la solution dans une autre AWS région, utilisez le sélecteur de région dans la barre de navigation de la console.
3. Sur la page Create stack, vérifiez que l'URL du modèle est correcte dans la zone de texte URL Amazon S3 et choisissez Next.
4. Sur la page Spécifier les détails de la pile, attribuez un nom à votre pile de solutions. Pour plus d'informations sur les limites de dénomination des caractères, reportez-vous aux [limites IAM et STS](#) du guide de l'AWS Identity and Access Management utilisateur.
5. Sous Paramètres, passez en revue les paramètres de ce modèle de solution et modifiez-les si nécessaire. Cette solution utilise les valeurs par défaut suivantes.

Paramètre	Par défaut	Description
ID de l'événement	Sample	ID de l'événement de la salle d'attente.
Point de terminaison d'API de base privée	<i>&lt;Entrée obligatoire&gt;</i>	URL du point de terminaison privé pour les API de la salle d'attente virtuelle.

Paramètre	Par défaut	Description
Région de l'API principale	<i>&lt;Entrée obligatoire&gt;</i>	AWS Région dans laquelle l'API principale est installée.
Stratégie Inlet	Periodic	Stratégie d'entrée à déployer. Periodicaugmente le nombre de portions toutes les minutes. MaxSizeaugmente le nombre de serveurs en fonction du nombre maximum de transactions que le site cible en aval peut traiter à un moment donné.
Incrémenter par	<i>&lt;Entrée obligatoire&gt;</i>	Dans quelle mesure le compteur de service doit être incrémenté chaque minute. Obligatoire si vous sélectionnez une stratégie d'entrée périodique.
Heure de début	<i>&lt;Entrée obligatoire&gt;</i>	Horodatage indiquant quand commencer à incrémenter le nombre de serveurs (durée de l'époque en secondes). Obligatoire si vous sélectionnez une stratégie d'entrée périodique.

Paramètre	Par défaut	Description
End Time (Heure de fin)	<Entrée obligatoire>	Horodatage indiquant quand arrêter d'incrémenter le nombre de serveurs (durée de l'époque en secondes). S'il reste 0, le nombre de portions est incrémenté indéfiniment. Obligatoire si vous sélectionnez une stratégie d'entrée périodique.
CloudWatch Nom de l'alarme	<Entrée obligatoire>	Nom CloudWatch d'alarme facultatif à associer à la stratégie d'entrée périodique. S'il est fourni et qu'il est dans un état alarmant, le numéro de service n'est pas incrémenté. Applicable uniquement à la stratégie d'entrée périodique.
Taille maximale	<Entrée obligatoire>	Nombre maximal de transactions que le site cible en aval peut traiter à la fois (MaxSize stratégie).


6. Choisissez Next (Suivant).
7. Sur la page Configurer les options de pile, choisissez Suivant.
8. Sur la page Vérification, vérifiez et confirmez les paramètres. Cochez la case indiquant que le modèle crée des ressources AWS Identity and Access Management (IAM).
9. Sélectionnez Create stack (Créer une pile) pour déployer la pile.

Vous pouvez consulter l'état de la pile dans la AWS CloudFormation console dans la colonne État. Vous devriez recevoir le statut CREATE\_COMPLETE dans environ deux minutes.

## 5. (Facultatif) Lancez l'exemple de pile de salles d'attente

Temps de déploiement : environ 5 minutes

1. Connectez-vous au modèle [AWS Management Console](#) et sélectionnez le bouton pour lancer le `aws-virtual-waiting-room-sample.template` AWS CloudFormation modèle.



Vous

pouvez également [télécharger le modèle](#) comme point de départ pour votre propre implémentation.

2. Le modèle est lancé par défaut dans la région USA Est (Virginie du Nord). Pour lancer la solution dans une autre AWS région, utilisez le sélecteur de région dans la barre de navigation de la console.
3. Sur la page Create stack, vérifiez que l'URL du modèle est correcte dans la zone de texte URL Amazon S3 et choisissez Next.
4. Sur la page Spécifier les détails de la pile, attribuez un nom à votre pile de solutions. Pour plus d'informations sur les limites de dénomination des caractères, reportez-vous aux [limites IAM et STS](#) du guide de l'AWS Identity and Access Management utilisateur.
5. Sous Paramètres, passez en revue les paramètres de ce modèle de solution et modifiez-les si nécessaire. Cette solution utilise les valeurs par défaut suivantes.

Paramètre	Par défaut	Description
Région API Gateway	<i>&lt;Entrée obligatoire&gt;</i>	AWS Nom de région de l'API Gateway.
ARN de l'autorisateur	<i>&lt;Entrée obligatoire&gt;</i>	ARN de l'autorisateur Lambda API Gateway.
ID de l'événement	Sample	ID d'événement de la salle d'attente.
Point de terminaison d'API privé	<i>&lt;Entrée obligatoire&gt;</i>	URL du point de terminaison privé pour les API de la salle d'attente virtuelle.

Paramètre	Par défaut	Description
Point de terminaison d'API public	<i>&lt;Entrée obligatoire&gt;</i>	URL du point de terminaison public pour les API de la salle d'attente virtuelle.

6. Choisissez Next (Suivant).
7. Sur la page Configurer les options de pile, choisissez Suivant.
8. Sur la page Vérification, vérifiez et confirmez les paramètres. Cochez la case indiquant que le modèle crée des ressources AWS Identity and Access Management (IAM).
9. Sélectionnez Create stack (Créer une pile) pour déployer la pile.

Vous pouvez consulter l'état de la pile dans la AWS CloudFormation console dans la colonne État. Vous devriez recevoir le statut CREATE\_COMPLETE dans environ cinq minutes.

## Mise à jour de la pile à partir d'une version précédente

Nous vous recommandons de supprimer la pile et d'en créer une nouvelle pour la nouvelle version. Actuellement, la migration vers la version la plus récente à l'aide de CloudFormation Stack Update n'est pas prise en charge. Voir [Désinstallez la solution](#) ensuite [Lancer la pile de démarrage](#).

### Note

Nous vous recommandons de migrer vers une version plus récente lorsque vous n'utilisez pas activement la solution pour prendre en charge un événement en cours.



# Données de performance

La charge de Virtual Waiting Room on AWS a été testée à l'aide d'un outil appelé [Locust](#). La taille des événements simulés variait de 10 000 à 100 000 clients. L'environnement de test de charge comprenait la configuration suivante :

- Locust 2.x avec personnalisations pour les déploiements dans le cloud AWS
- Quatre AWS régions (us-west-1,us-west-2,us-east-1,us-east-2)
- 10 hôtes c5.4xlarge Amazon EC2 par région (40 au total)
- 32 processus Locust par hôte
- Les utilisateurs simulés étaient répartis équitablement entre les 1 280 processus

Les étapes de test de l' end-to-end API pour chaque processus utilisateur :

1. Appelez `assign_queue_num` et recevez un numéro de demande.
2. Faites `queue_num` une boucle avec l'ID de demande jusqu'à ce qu'elle renvoie la position de l'utilisateur dans la file d'attente (court laps de temps).
3. Boucle `erving_num` jusqu'à ce que la valeur renvoyée soit  $\geq$  la position de l'utilisateur dans la file d'attente (longue durée).
4. Appelez rarement `waiting_room_size` pour connaître le nombre d'utilisateurs en attente.
5. Appelez `generate_token` et recevez un JWT à utiliser sur le site cible.

## Conclusions

Il n'y a pas de limite maximale pratique au nombre de clients pouvant être traités dans la salle d'attente.

La fréquence à laquelle les utilisateurs entrent dans la salle d'attente a un impact sur les quotas d'exécution simultanée de la fonction Lambda pour la région dans laquelle elle est déployée.

Le test de charge n'a pas réussi à dépasser les limites de 10 000 requêtes par seconde par défaut d'API Gateway avec les politiques de mise en cache utilisées avec CloudFront.

La fonction `get_queue_num` Lambda a un taux d'invocation proche de 1:1 par rapport au taux d'utilisateurs entrant dans la salle d'attente. Cette fonction Lambda peut être limitée lorsque le

nombre d'utilisateurs entrants est élevé en raison des limites de simultanéité ou des limites de rafale. La limitation provoquée par un grand nombre d'invocations de fonctions `get_queue_num` Lambda peut avoir un impact secondaire sur les autres fonctions Lambda. L'ensemble du système continue de fonctionner si le logiciel client peut répondre de manière appropriée à ce type d'erreur de dimensionnement temporaire avec une logique de réessai/de réduction.

La CloudFront distribution configurée par le noyau dans une configuration de quotas par défaut peut gérer une salle d'attente contenant 250 000 utilisateurs, chaque utilisateur interrogeant `l_serving_numAPI` au moins toutes les secondes.

# Résolution des problèmes

Cette section fournit des informations de dépannage pour cette solution.

Si cette section ne répond pas à votre problème, [contactez le support AWS](#) pour obtenir des instructions pour ouvrir un dossier de support AWS pour cette solution.

## État de réponse 4xx des API

- Cela peut être dû à un identifiant d'événement ou à un identifiant de demande incorrect, ou aux deux. Cela se produit dans les CloudWatch journaux de la fonction Lambda associée.
- Les API privées sont authentifiées par IAM et le client a besoin de AWS clés habilitées à invoquer les API privées. Cela se produit dans les CloudWatch journaux d'API Gateway.

## État de réponse 5xx des API

- Réponse d'un Lambda limité ou d'une API Gateway, alarme de vérification.  
`<LambdaFunctionName>ThrottlesAlarm` CloudWatch
- Mauvaise configuration sur le back-end, vérifiez l'`<LambdaFunctionName>ErrorsAlarm` CloudWatch alarme et les CloudWatch journaux pour plus de détails.

## 5 XX/ErrorPublicPrivateApiAlarm

- Cet état d'alarme se ALARM produit lorsque l'API renvoie un statut 5XX à l'appelant dans un délai de 60 secondes.
- Cette alarme revient OK lorsqu'aucun statut 5xx n'est renvoyé pendant 60 secondes.
- Cette alarme peut être lancée par une fonction Lambda ou un environnement d'exécution Lambda renvoyant une erreur à API Gateway.

## 4 X/ErrorPublicPrivateApiAlarm

- Cet état d'alarme se ALARM produit lorsque l'API renvoie un statut 4XX à l'appelant dans un délai de 60 secondes.
- Cette alarme revient au OK moment où le statut 4XX est renvoyé pendant 60 secondes.
- Cette alarme peut être déclenchée par une URL d'API incorrecte.

### <LambdaFunctionName>ThrottlesAlarm

- Cet état d'alarme est ALARM lorsque le Lambda nommé atteint une limite d'exécution simultanée dans un délai de 60 secondes.
- Cette alarme revient OK si aucun accélérateur n'est détecté pendant 60 secondes.
- Vous devrez peut-être augmenter la limite de simultanéité pour la région de votre compte.
- Vous êtes peut-être confronté à la limite de rafales pour Lambda, ce qui nécessite une certaine logique de nouvelle tentative sur votre client.

### <LambdaFunctionName>ErrorsAlarm

- Cet état d'alarme se produit ALARM lorsque le Lambda nommé rencontre une erreur d'exécution dans un délai de 60 secondes.
- Cette alarme revient OK si aucune erreur n'est détectée pendant 60 secondes.
- Cela peut être dû à une mauvaise configuration du backend.
- Cela peut être dû à un bogue dans le code Lambda.

## Contacter AWS Support

Si vous bénéficiez d'[AWS Developer Support](#), d'[AWS Business Support](#) ou d'[AWS Enterprise Support](#), vous pouvez utiliser le Centre de support pour obtenir l'assistance d'experts concernant cette solution. Les sections suivantes fournissent des instructions.

### Créer un dossier

1. Connectez-vous au [Centre de Support](#).
2. Choisissez Create case (Créer une demande).

### Comment pouvons-nous vous aider ?

1. Choisissez Technique.
2. Dans le champ Service, sélectionnez Solutions.
3. Dans Catégorie, sélectionnez Autres solutions.
4. Pour Severity, sélectionnez l'option qui correspond le mieux à votre cas d'utilisation.

5. Lorsque vous entrez le service, la catégorie et la gravité, l'interface contient des liens vers des questions de dépannage courantes. Si vous ne parvenez pas à résoudre votre question à l'aide de ces liens, sélectionnez **Étape suivante : Informations supplémentaires**.

## Informations supplémentaires

1. Dans le champ **Objet**, saisissez un texte résumant votre question ou problème.
2. Pour la description, décrivez le problème en détail.
3. Choisissez **Joindre des fichiers**.
4. Joignez les informations AWS Support nécessaires au traitement de la demande.

## Aidez-nous à résoudre votre cas plus rapidement

1. Entrez les informations demandées.
2. Choisissez **Next step: Solve now or contact us** (**Étape suivante : résolvez maintenant ou contactez-nous**).

## Résolvez maintenant ou contactez-nous

1. Passez en revue les solutions **Solve now**.
2. Si vous ne parvenez pas à résoudre votre problème avec ces solutions, choisissez **Contactez-nous**, entrez les informations demandées, puis choisissez **Soumettre**.

## Ressources supplémentaires

AWS services	
• <a href="#">AWS CloudFormation</a>	• <a href="#">Amazon DynamoDB</a>
• <a href="#">Amazon Simple Storage Service</a>	• <a href="#">Amazon API Gateway</a>
• <a href="#">AWS Lambda</a>	• <a href="#">AWS Secrets Manager</a>
• <a href="#">Amazon CloudFront</a>	• <a href="#">Amazon Simple Queue Service</a>
• <a href="#">Amazon EventBridge</a>	• <a href="#">Amazon CloudWatch</a>
• <a href="#">Amazon ElastiCache pour Redis</a>	• <a href="#">Amazon Comprehend</a>
• <a href="#">Amazon Virtual Private Cloud</a>	• <a href="#">AWS Identity and Access Management</a>

## Désinstallez la solution

Vous pouvez désinstaller la AWS solution Virtual Waiting Room on à partir du AWS Management Console ou en utilisant le AWS Command Line Interface. Vous devez supprimer manuellement les compartiments S3 utilisés pour stocker les journaux par les différentes ressources créées par cette solution. AWS Les implémentations de solutions ne suppriment pas automatiquement ces compartiments S3. Vous avez donc toujours la possibilité de consulter les journaux des événements après la suppression de la solution.

Si vous avez ajouté manuellement un utilisateur IAM au groupe d'utilisateurs ProtectedAPIGroup IAM créé par la solution, [supprimez-le du groupe d'utilisateurs IAM avant de désinstaller la](#) solution. Dans le cas contraire, le groupe d'utilisateurs IAM et la politique IAM associée ne seront pas supprimés.

Pour chacune des piles déployées, suivez les instructions ci-dessous.

### À l'aide du AWS Management Console

1. Connectez-vous à la [console AWS CloudFormation](#).
2. Sur la page Stacks, sélectionnez la pile d'installation de cette solution.
3. Sélectionnez Delete (Supprimer).

### En utilisant AWS Command Line Interface

Déterminez si le AWS Command Line Interface (AWS CLI) est disponible dans votre environnement. Pour les instructions d'installation, reportez-vous à [Qu'est-ce que le AWS Command Line Interface ?](#) dans le guide de AWS CLI l'utilisateur. Après avoir confirmé que le AWS CLI est disponible, exécutez la commande suivante.

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

### Suppression des compartiments Amazon S3

Cette solution est configurée pour conserver le compartiment Amazon S3 créé par la solution (à déployer dans une région optionnelle) si vous décidez de supprimer la AWS CloudFormation pile

afin d'éviter toute perte de données accidentelle. Après avoir désinstallé la solution, vous pouvez supprimer manuellement ce compartiment S3 si vous n'avez pas besoin de conserver les données. Suivez ces étapes pour supprimer le compartiment Amazon S3.

1. Connectez-vous à la [console Amazon S3](#).
2. Choisissez Buckets dans le volet de navigation de gauche.
3. Localisez les <stack-name>compartiments S3.
4. Sélectionnez le compartiment S3, puis choisissez Supprimer.

Pour supprimer le compartiment S3 à l'aide de AWS CLI, exécutez la commande suivante :

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



## Code source

Consultez notre [GitHub référentiel](#) pour télécharger les fichiers source de cette solution et partager vos personnalisations avec d'autres utilisateurs.

# Collaborateurs

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

# Révisions

Date	Modification
novembre 2021	Première version
Septembre 2022	Version 1.1 : Incrémentation automatique du compteur de service en fonction des positions de file d'attente expirées. Déplacez une partie de l'utilisation de Redis vers DynamoDB. Point de terminaison de l'API publique pour obtenir l'heure d'expiration de la position de file d'attente restante. Pour plus d'informations, reportez-vous au fichier <a href="#">ChangeLog.md</a> dans le référentiel. GitHub
Avril 2023	Version 1.1.1 : impact atténué causé par les nouveaux paramètres par défaut pour la propriété des objets S3 (ACL désactivées) pour tous les nouveaux compartiments S3. Pour plus d'informations, reportez-vous au fichier <a href="#">ChangeLog.md</a> dans le référentiel. GitHub
Novembre 2023	Version 1.1.2 : versions des packages mises à jour pour résoudre les failles de sécurité. Pour plus d'informations, reportez-vous au fichier <a href="#">ChangeLog.md</a> dans le référentiel. GitHub
Mars 2024	Version 1.1.3 : résolution de trois problèmes : les positions de file d'attente expirées persistent dans la taille de la salle d'attente, l'queue_num API renvoyant d'anciens résultats même après une réinitialisation et les défaillances intermittentes de l'API de l'adaptateur OpenID. /userInfo Pour plus d'informa

Date	Modification
	tions, reportez-vous au fichier <a href="#">ChangeLog.md</a> dans le référentiel. GitHub
Avril 2024	Version 1.1.4 : versions des packages mises à jour pour résoudre les failles de sécurité. Pour plus d'informations, reportez-vous au fichier <a href="#">ChangeLog.md</a> dans le référentiel. GitHub
Juin 2024	Version 1.1.5 : versions des packages mises à jour pour résoudre les failles de sécurité. Pour plus d'informations, reportez-vous au fichier <a href="#">ChangeLog.md</a> dans le référentiel. GitHub

## Avis

Il incombe aux clients de procéder à une évaluation indépendante des informations contenues dans le présent document. Ce document : (a) est fourni à titre informatif uniquement, (b) représente les offres de produits et les pratiques AWS actuelles, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ni aucune assurance de la part de AWS ses filiales, fournisseurs ou concédants de licence. AWS les produits ou services sont fournis « tels quels » sans garanties, déclarations ou conditions d'aucune sorte, qu'elles soient explicites ou implicites. AWS les responsabilités et les obligations envers ses clients sont régies par des AWS accords, et ce document ne fait partie ni ne modifie aucun accord entre AWS et ses clients.

Virtual Waiting Room on AWS est licencié selon les termes de la [licence Apache version 2.0](#).

# AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le [AWS glossaire](#) dans la Glossaire AWS référence.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.