Livre blanc AWS

Présentation de la sécurité d'AWS Lambda

Présentation de la sécurité d'AWS Lambda: Livre blanc AWS

Table of Contents

Présentation de la sécurité d'AWS Lambda

Date de publication : 12 février 2021 (Révisions du document)

Résumé

Ce livre blanc présente en détail le service AWS Lambda en mettant l'accent sur la sécurité. Il propose une image complète du service, qui peut être utile aux nouveaux utilisateurs, et permet aux utilisateurs actuels de mieux comprendre Lambda.

Ce livre blanc s'adresse notamment aux responsables de la sécurité informatique (CISO), aux ingénieurs en sécurité de l'information, aux architectes d'entreprise, aux équipes de conformité et à toute autre personne qui souhaite comprendre les fondements d'AWS Lambda.

Introduction

À l'heure actuelle, de plus en plus de charges de travail utilisent AWS Lambda par souci de capacité de mise à l'échelle, de performances et de rentabilité, sans avoir à gérer l'infrastructure sous-jacente. Ces charges de travail s'adaptent à des milliers de demandes simultanées par seconde. Lambda est l'un des nombreux services importants proposés par AWS aujourd'hui. Chaque mois, des centaines de milliers de clients Amazon Web Services (AWS) utilisent Lambda pour traiter des billions de demandes.

Lambda convient aux applications stratégiques dans de nombreux secteurs d'activité. Un large éventail de clients, des médias et du divertissement aux services financiers et à d'autres secteurs réglementés, y ont recours. Ces clients réduisent leurs délais de mise sur le marché, optimisent les coûts et améliorent l'agilité en se concentrant sur ce qu'ils font de mieux : gérer leur entreprise.

Grâce au modèle d'environnement d'exécutiongéré, Lambda peut gérer la plupart des détails d'implémentation des charges de travail sans serveur en cours d'exécution. Ce modèle réduit davantage la surface d'attaque tout en simplifiant la sécurité du cloud. Ce livre blanc présente les fondements de ce modèle, ainsi que les bonnes pratiques, aux développeurs, aux analystes de sécurité, aux équipes de sécurité et de conformité et aux autres parties prenantes.

À propos d'AWS Lambda

AWS Lambda est un service de calcul sans serveur guidé par les événements qui étend d'autres services AWS avec une logique personnalisée, ou crée d'autres services backend pouvant être mis à l'échelle, performants et sécuritaires. Lambda peut exécuter automatiquement le code en réponse à plusieurs événements, tels que les requêtes HTTP via Amazon API Gateway, les modifications d'objets dans des compartiments Amazon S3, les mises à jour de tables dans Amazon DynamoDB ou les transitions d'état dans AWS Step Functions. Vous pouvez également exécuter le code directement depuis n'importe quelle application web ou mobile. Lambda exécute le code sur une infrastructure informatique à haute disponibilité et effectue toute l'administration de la plateforme sous-jacente, y compris la maintenance des serveurs et du système d'exploitation, l'allocation et la mise à l'échelle automatique des capacités, l'application de correctifs, ainsi que la surveillance et la journalisation du code.

Grâce à Lambda, il vous suffit de charger le code et de configurer le moment où l'appeler. Lambda réalise toutes les autres tâches nécessaires pour exécuter le code avec une haute disponibilité. Lambda s'intègre à de nombreux autres services AWS. Il permet de créer des applications sans serveur ou des services backend, allant de simples tâches d'automatisation déclenchées périodiquement à des applications de microservices complètes.

Lambda peut également être configuré pour accéder aux ressources de votre Amazon Virtual Private Cloud et, par extension, à vos ressources sur site.

Vous pouvez facilement intégrer Lambda à une posture de sécurité renforcée à l'aide d'AWS Identity and Access Management (IAM) et d'autres techniques abordées dans ce livre blanc, afin de maintenir un niveau élevé de sécurité et d'audit, en fonction de vos besoins de conformité.

Rubriques

Avantages de Lambda

Les clients qui souhaitent libérer la créativité et la rapidité de leurs organisations de développement, sans compromettre la capacité de leur équipe informatique à fournir une infrastructure évolutive, rentable et gérable, estiment qu'AWS Lambda leur permet de se défaire de la complexité

opérationnelle au profit de l'agilité et de meilleurs prix, sans compromettre la mise à l'échelle ou la fiabilité.

Lambda offre de nombreux avantages, notamment :

Aucun serveur à gérer

Lambda exécute votre code sur une infrastructure hautement disponible et tolérante aux pannes, répartie sur plusieurs zones de disponibilité dans une seule région, en déployant du code de manière transparente et en assurant toute l'administration, la maintenance et l'application de correctifs de l'infrastructure. Lambda intègre également des fonctions de journalisation et de surveillance, y compris l'intégration à Amazon CloudWatch, à CloudWatch Logs et à AWS CloudTrail.

Mise à l'échelle continue

Lambda gère précisément la mise à l'échelle de vos fonctions (ou applications) en exécutant le code déclenché par un événement en parallèle et en traitant chaque événement individuellement.

Calcul à la milliseconde

Avec AWS Lambda, les frais s'appliquent à chaque milliseconde (ms) d'exécution du code et selon le nombre de fois où il est déclenché. Ainsi, vous payez pour un débit ou une durée d'exécution constants, plutôt que par unité de serveur.

Innovation accrue

Lambda prend en charge la gestion de l'infrastructure, libérant ainsi vos ressources de programmation qui peuvent alors se concentrer sur l'innovation et le développement de la logique métier.

Modernisez vos applications

Lambda permet d'utiliser des fonctions avec des modèles de machine learning préentraînés pour injecter facilement de l'intelligence artificielle dans des applications. Une seule demande d'interface de programmation d'application (API) permet de classer des images, d'analyser des vidéos, de convertir la parole en texte, d'effectuer un traitement du langage naturel, etc.

Un écosystème riche

Lambda facilite le travail des développeurs grâce à AWS Serverless Application Repository pour la détection, le déploiement et la publication d'applications sans serveur, au modèle d'application

sans serveur AWS pour la génération d'applications sans serveur et les intégrations à divers environnements de développement intégrés (IDE) tels que AWS Cloud9, à AWS Toolkit pour Visual Studio, à AWS Tools pour Visual Studio Team Services et à plusieurs autres. Lambda est intégré à des services AWS supplémentaires afin de vous doter d'un écosystème riche pour la génération d'applications sans serveur.

Coût d'exécution des applications basées sur Lambda

Lambda propose un modèle de tarification avec paiement à l'utilisation granulaire. Grâce à ce modèle, vous payez en fonction du nombre d'appels de fonctions et de leur durée (c'est-à-dire du temps nécessaire à l'exécution du code). Outre ce modèle de tarification flexible, Lambda propose également 1 million de demandes perpétuellement gratuites par mois, ce qui permet à de nombreux clients d'automatiser leur processus sans aucun coût.

Le modèle de responsabilité partagée

La sécurité et la conformité sont une responsabilité partagée entre AWS et le client. Ce modèle de responsabilité partagée peut atténuer votre charge opérationnelle, car AWS exploite, gère et commande les composants depuis le système d'exploitation hôte jusqu'à la sécurité physique des installations dans lesquelles les services sont exploités, en passant par la couche de virtualisation.

Pour AWS Lambda, AWS gère les services d'infrastructure et de base sous-jacents, le système d'exploitation et la plateforme d'applications. Vous êtes responsable de la sécurité de votre code et de la gestion des identités et des accès (IAM) au service Lambda et au sein de votre fonction.

La figure 1 illustre le modèle de responsabilité partagée tel qu'il s'applique aux composantes communes et distinctes d'AWS Lambda. Les responsabilités d'AWS sont illustrées sous la ligne pointillée orange, les responsabilités du client au-dessus de la ligne pointillée bleue.

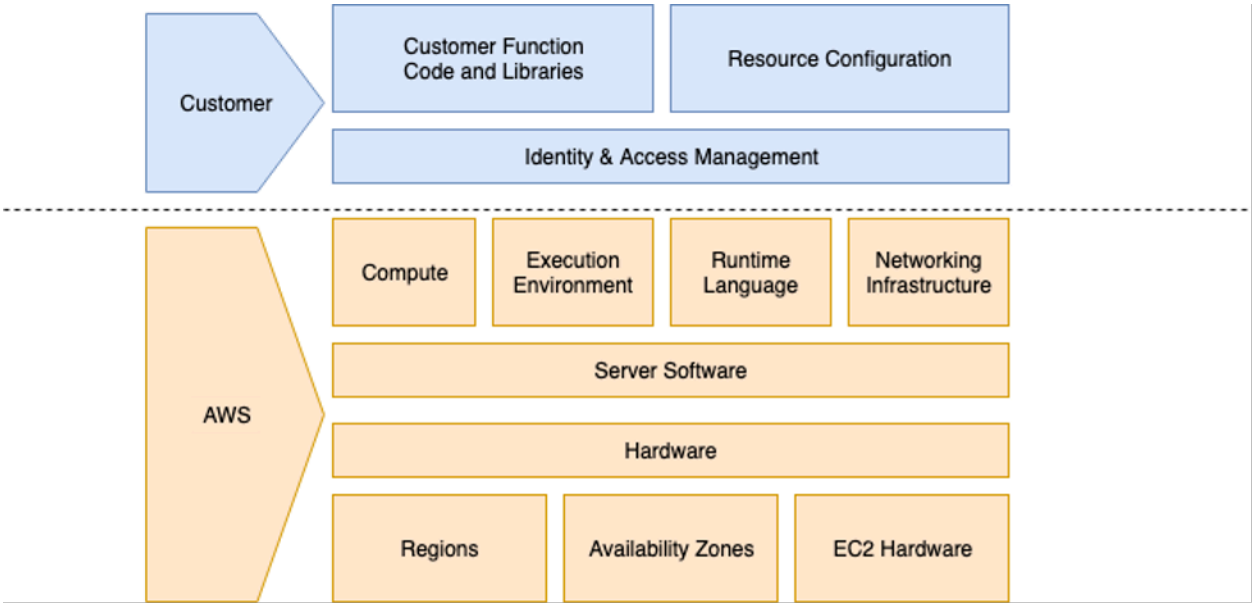

Figure 1 – Modèle de responsabilité partagée pour AWS Lambda

Fonctions et couches AWS Lambda

Grâce à Lambda, vous pouvez exécuter du code virtuellement sans aucune administration de l'infrastructure sous-jacente. Vous êtes uniquement responsable du code que vous fournissez à Lambda et de la configuration de la façon dont Lambda exécute ce code en votre nom. À l'heure actuelle, Lambda prend en charge deux types de ressources de code : les fonctions et les couches.

Une fonction est une ressource qui peut être invoquée pour exécuter le code dans Lambda. Les fonctions peuvent inclure une ressource commune ou partagée appelée couche. Les couches peuvent être utilisées pour partager du code ou des données communs entre différentes fonctions ou différents comptes AWS. Vous êtes responsable de la gestion de tout le code contenu dans vos fonctions ou couches. Lorsque Lambda reçoit le code fonction ou couche d'un client, il en protège l'accès en le chiffrant au repos à l'aide d'une clé AWS Key Management Service (AWS KMS) et en transit à l'aide du protocole TLS 1.2+.

Vous pouvez gérer l'accès à vos fonctions et couches à l'aide de stratégies AWS Lambda ou d'autorisations basées sur les ressources. Pour obtenir la liste complète des fonctions IAM prises en charge sur IAM, consultez Services AWS qui fonctionnent avec IAM.

Vous pouvez également contrôler l'ensemble du cycle de vie de vos fonctions et couches au moyen des API de plan de contrôle de Lambda. Par exemple, vous pouvez supprimer la fonction en appelant `DeleteFunction` ou révoquer les autorisations d'un autre compte en appelant `RemovePermission`.

Modes d'appel Lambda

L'API Invoke peut être appelée selon deux modes : le mode événement et le mode demande-réponse.

- Le mode événement place en file d'attente la charge utile pour un appel asynchrone.
- Le mode demande-réponse appelle de manière synchrone la fonction avec la charge utile fournie et renvoie une réponse immédiatement.

Dans les deux cas, l'exécution de la fonction est toujours effectuée dans un environnement d'exécution Lambda, mais la charge utile emprunte des chemins différents. Pour en savoir plus, consultez la section « Environnements d'exécution Lambda » dans ce document.

Vous pouvez également utiliser d'autres services AWS qui effectuent des appels en votre nom. Le mode d'appel utilisé dépend du service AWS que vous utilisez et de la façon dont il est configuré. Pour en savoir plus sur la façon dont les autres services AWS s'intègrent à Lambda, consultez Utilisation d'AWS Lambda avec d'autres services.

Lorsque Lambda reçoit un appel demande-réponse, celui-ci est transmis directement au service d'appel. Si le service d'appel n'est pas disponible, les appelants peuvent temporairement placer en file d'attente la charge utile côté client afin de retenter l'appel un certain nombre de fois. Si le service d'appel reçoit la charge utile, il tente alors d'identifier un environnement d'exécution disponible pour la demande, puis transmet la charge utile à cet environnement d'exécution afin de terminer l'appel. S'il n'existe aucun environnement d'exécution ou qu'aucun n'est approprié, il en sera créé un de manière dynamique en réponse à la demande. Pendant le transit, les charges utiles d'appel envoyées au service d'appel sont sécurisées à l'aide du protocole TLS 1.2+. Le trafic au sein du service Lambda (depuis l'équilibreur de charge vers le bas) passe par un cloud privé virtuel (VPC) interne isolé, détenu par le service Lambda, au sein de la région AWS à laquelle la demande a été envoyée.

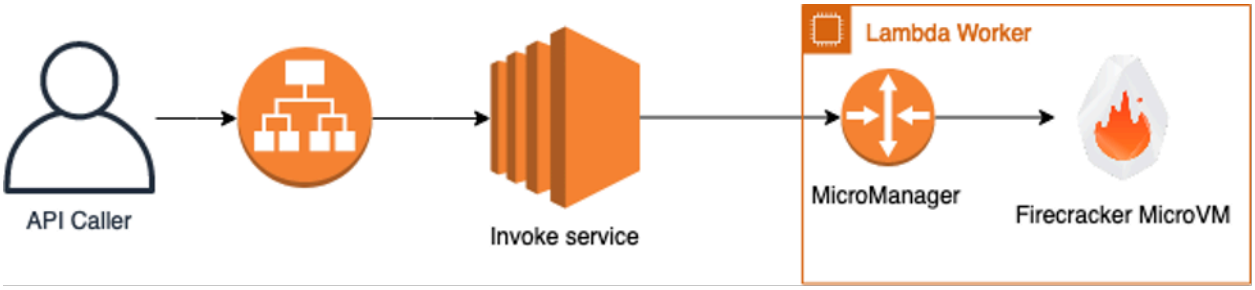

Figure 2 – Modèle d'appel selon le mode AWS Lambdademande-réponse

Les charges utiles du mode d'appel événement sont toujours placées en file d'attente pour être traitées avant l'appel. Toutes les charges utiles sont placées, à des fins de traitement, dans une file d'attente Amazon Simple Queue Service (Amazon SQS). Les événements en file d'attente sont toujours sécurisés en transit à l'aide du protocole TLS 1.2+, mais ils ne sont actuellement pas chiffrés au repos. Les files d'attente Amazon SQS utilisées par Lambda sont gérées par le service Lambda. Vous ne pouvez pas les voir en tant que client. Les événements en file d'attente peuvent être stockés dans une file d'attente partagée, mais ils peuvent être migrés ou affectés à des files d'attente dédiées en fonction d'un certain nombre de facteurs que les clients ne contrôlent pas directement (par exemple, le taux d'appel, la taille des événements, etc.).

Les événements en file d'attente sont récupérés par lots par la flotte d'interrogateurs de Lambda. La flotte d'interrogateurs est un groupe d'instances EC2 dont le but est de traiter les appels en mode événement en file d'attente qui n'ont pas encore été traités. Lorsque la flotte d'interrogateurs récupère un événement en file d'attente qu'elle doit traiter, elle le fait en le transmettant au service d'appel, comme le ferait un client dans le cas d'un appel en mode demande-réponse.

Si l'appel ne peut pas être effectué, la flotte d'interrogateurs stocke temporairement l'événement, en mémoire, sur l'hôte jusqu'à ce qu'elle soit en mesure de terminer l'exécution ou jusqu'à ce que le nombre de tentatives d'exécution soit dépassé. Aucune donnée de charge utile n'est jamais écrite sur le disque de la flotte d'interrogateurs à proprement parler. La flotte d'interrogateurs peut être affectée à tous les clients AWS, ce qui permet de réduire le délai d'appel au maximum. Pour en savoir plus sur les services susceptibles d'accepter le mode d'appel événement, consultez la section Utilisation d'AWS Lambda avec d'autres services.

Exécutions Lambda

Lorsque Lambda exécute une fonction en votre nom, il gère l'allocation et la configuration des systèmes sous-jacents nécessaires à l'exécution du code. Les développeurs peuvent ainsi se concentrer sur la logique métier et l'écriture de code, et non sur l'administration et la gestion des systèmes sous-jacents.

Le service Lambda est réparti entre le plan de contrôle et le plan de données. Chaque plan a une fonction distincte dans le service. Le plan de contrôle fournit les API de gestion (par exemple, `CreateFunction`, `UpdateFunctionCode`, `PublishLayerVersion`, etc.) et gère les intégrations à tous les services AWS. Les communications vers le plan de contrôle de Lambda sont protégées en transit par TLS. Toutes les données client stockées dans le plan de contrôle de Lambda sont chiffrées au repos en utilisant AWS KMS, qui est conçu pour les protéger contre toute divulgation non autorisée ou falsification.

Le plan de données est l'API Invoke de Lambda déclenche l'appel des fonctions Lambda. Lorsqu'une fonction Lambda est invoquée, le plan de données alloue un environnement d'exécution sur un worker AWS Lambda (ou simplement worker, qui est un type d'instance Amazon EC2) à cette version de fonction, ou choisit un environnement d'exécution existant déjà configuré pour cette version de fonction, qu'il utilise ensuite pour terminer l'appel. Pour en savoir plus, consultez la section « Micro-machines virtuelles et workers AWS Lambda » de ce document.

Environnements d'exécution Lambda

Chaque appel est acheminé par le service d'appel de Lambda vers un environnement d'exécution sur un worker capable de traiter la demande. En dehors du plan de données, les clients et les autres utilisateurs ne peuvent pas directement initier des communications réseau entrantes/d'entrée avec un environnement d'exécution. Il est ainsi possible de garantir que les communications vers votre environnement d'exécution sont authentifiées et autorisées.

Les environnements d'exécution sont réservés à une version de fonction spécifique et ne peuvent pas être réutilisés entre des versions de fonction, des fonctions ou des comptes AWS. Cela signifie qu'une fonction unique qui peut avoir deux versions différentes aboutirait à au moins deux environnements d'exécution uniques.

Chaque environnement d'exécution peut être utilisé pour un seul appel simultané à la fois, et il peut être réutilisé pour plusieurs appels de la même version de fonction pour des raisons de

performances. Selon un certain nombre de facteurs (par exemple, le taux d'appel, la configuration de la fonction, etc.), un ou plusieurs environnements d'exécution peuvent exister pour une version de fonction donnée. Grâce à cette approche, Lambda est en mesure de fournir à ses clients une isolation au niveau de la version des fonctions.

Lambda n'isole actuellement pas les appels dans l'environnement d'exécution d'une version de fonction. Cela signifie qu'un appel peut laisser un état susceptible d'affecter le prochain appel (par exemple, des fichiers écrits dans /tmp ou des données en mémoire). Si vous voulez vous assurer qu'un appel ne peut pas affecter un autre appel, Lambda recommande de créer des fonctions distinctes supplémentaires. Par exemple, vous pouvez créer des fonctions distinctes pour des opérations d'analyse complexes qui sont plus sujettes aux erreurs, et réutiliser des fonctions qui n'effectuent pas d'opérations affectant la sécurité. Actuellement, Lambda ne limite pas le nombre de fonctions que les clients peuvent créer. Pour en savoir plus sur les restrictions, consultez la page Quotas Lambda.

Lambda surveille et gère en permanence les environnements d'exécution, qui peuvent être créés ou détruits pour un certain nombre de raisons, notamment :

- Un nouvel appel arrive et il n'existe aucun environnement d'exécution approprié.
- Un déploiement d'environnement d'exécution interne ou de logiciel de worker se produit.
- Une nouvelle configuration de la simultanéité allouée est publiée.
- La durée de bail sur l'environnement d'exécution, ou le worker, approche de son terme ou l'a dépassé.
- Autres processus de rééquilibrage des charges de travail internes.

Les clients peuvent gérer le nombre d'environnements d'exécution préalloués qui existent pour une version de fonction en configurant la simultanéité allouée sur leur configuration de fonction. Lorsqu'il est configuré pour cela, Lambda crée et gère le nombre configuré d'environnements d'exécution, et veille à ce qu'ils existent toujours. Les clients profitent ainsi d'un plus grand contrôle sur les performances au démarrage de leurs applications sans serveur, quelle que soit l'échelle.

En dehors d'une configuration de la simultanéité allouée, les clients ne peuvent pas contrôler de manière déterministe le nombre d'environnements d'exécution créés ou gérés par Lambda en réponse aux appels.

Rôle d'exécution

Chaque fonction Lambda doit également être configurée avec un rôle d'exécution, qui est un rôle IAM assumé par le service Lambda lors de l'exécution d'opérations de plan de contrôle et de plan de données liées à la fonction. Le service Lambda assume ce rôle afin de récupérer les informations d'identification de sécurité temporaires qui sont ensuite disponibles en tant que variables d'environnement lors de l'appel d'une fonction. Pour des raisons de performances, le service Lambda met en mémoire cache ces informations d'identification et peut les réutiliser dans différents environnements d'exécution qui reposent sur le même rôle d'exécution.

Conformément au principe du moindre privilège, Lambda recommande que chaque fonction possède son propre rôle unique, configuré avec le jeu minimum d'autorisations nécessaires.

Le service Lambda peut également assumer le rôle d'exécution afin d'effectuer certaines opérations du plan de contrôle, telles que celles liées à la création et à la configuration d'interfaces réseau Elastic (ENI) pour les fonctions VPC, à l'envoi de journaux vers Amazon CloudWatch Application Insights ou à l'envoi de traces vers AWS X-Ray, ou encore d'autres opérations non liées à l'appel. Les clients peuvent toujours passer en revue et auditer ces cas d'utilisation en consultant les journaux d'audit dans AWS CloudTrail.

Pour en savoir plus à ce sujet, consultez la page de documentation du rôle d'exécution AWS Lambda.

Micro-machines virtuelles et workers AWS Lambda

Lambda crée ses environnements d'exécution sur une flotte d'instances Amazon EC2 appelées workers AWS Lambda. Les workers sont des instances EC2 Nitro de matériel nu qui sont lancées et gérées par Lambda dans un compte AWS isolé distinct non visible pour les clients. Ils s'accompagnent d'une ou de plusieurs micro-machines virtuelles (MVM) virtualisées par le matériel et créées par Firecracker. Firecracker est un moniteur de machine virtuelle (VMM) open source qui utilise la machine virtuelle basée sur le noyau (KVM) de Linux pour créer et gérer des micro-machines virtuelles. Il est spécialement conçu pour créer et gérer des conteneurs multi-locataire sécurisés et des services basés sur des fonctions qui fournissent des modèles opérationnels sans serveur. Pour en savoir plus sur le modèle de sécurité de Firecracker, consultez le site web du projet Firecracker.

Dans le cadre du modèle de responsabilité partagée, Lambda est responsable du maintien de la configuration de sécurité, des contrôles et du niveau de correctifs des workers. L'équipe Lambda

utilise Amazon Inspector pour détecter les problèmes de sécurité potentiels connus, ainsi que d'autres mécanismes de notification des problèmes de sécurité personnalisés et des listes de prédivulgation, afin que les clients n'aient pas à gérer la posture de sécurité sous-jacente de leur environnement d'exécution.

Figure 3 – Modèle d'isolation pour les workers AWS Lambda

Les workers ont une durée de bail maximale de 14 heures. Lorsqu'un worker approche du terme de la durée de bail, aucun autre appel n'est acheminé vers lui, les micro-machines virtuelles sont arrêtées de manière appropriée et l'instance sous-jacente du worker est arrêtée. Lambda surveille en permanence les activités du cycle de vie de sa flotte et émet des alarmes.

Toutes les communications du plan de données vers les workers sont chiffrées selon la norme de chiffrement AES-GCM (Advanced Encryption Standard with Galois/Counter Mode). En dehors des opérations de plan de données, les clients ne peuvent pas interagir directement avec un worker, car celui-ci est hébergé dans un Amazon VPC isolé du réseau géré par Lambda dans les comptes de service de Lambda.

Lorsqu'un worker doit créer un nouvel environnement d'exécution, il reçoit une autorisation limitée dans le temps pour accéder aux artefacts de fonction du client. Ces artefacts sont spécifiquement optimisés pour l'environnement d'exécution et les workers de Lambda. Le code de fonction chargé au format ZIP est optimisé une fois, puis stocké dans un format chiffré à l'aide d'une clé gérée par AWS et de la norme AES-GCM.

Les fonctions chargées vers Lambda au format d'image de conteneur sont également optimisées. L'image de conteneur est d'abord téléchargée à partir de sa source d'origine, optimisée en fragments distincts, puis stockée sous la forme de fragments chiffrés à l'aide d'un procédé de chiffrement convergent authentifié qui utilise une combinaison des normes AES-CTR et AES-GCM et d'un code SHA-256 MAC. Grâce à la méthode de chiffrement convergent, Lambda peut dédupliquer en toute sécurité les fragments chiffrés. Toutes les clés nécessaires au déchiffrement des données du client sont protégées à l'aide de AWS KMSclés principales client (CMK) gérées par le client. Les clients peuvent consulter l'utilisation de clés CMK par le service Lambda dans les journaux AWS CloudTrail à des fins de suivi et d'audit.

Technologies d'isolation Lambda

Pour protéger les workers et les environnements d'exécution, Lambda utilise diverses technologies d'isolation propriétaires et open source. Chaque environnement d'exécution contient une copie dédiée des éléments suivants :

- Le code de la version de fonction particulière
- Toutes les couches AWS Lambda sélectionnées pour la version de votre fonction
- Le composant d'exécution de la fonction choisie (par exemple, Java 11, NodeJS 12, Python 3.8, etc.) ou le composant d'exécution personnalisé de la fonction
- Un répertoire /tmp accessible en écriture
- Un espace utilisateur Linux minimal basé sur Amazon Linux 2

Les environnements d'exécution sont isolés les uns des autres à l'aide de plusieurs technologies de type conteneur intégrées au noyau Linux, ainsi que des technologies d'isolation propriétaires d'AWS. Notamment :

- cgroups : technologie utilisée pour restreindre l'accès de la fonction au processeur et à la mémoire.
- espaces de noms : chaque environnement d'exécution s'exécute dans un espace de noms dédié. Pour ce faire, des ID de processus de groupe uniques, des ID utilisateur, des interfaces réseau et d'autres ressources gérées par le noyau Linux sont en place.
- seccomp-bpf : permet de limiter les appels système (syscalls) qui peuvent être utilisés depuis l'environnement d'exécution.
- iptables et tables de routage : permet d'empêcher les communications réseau entrantes et d'isoler les connexions réseau entre les micro-machines virtuelles.
- chroot : fournit un accès limité au système de fichiers sous-jacent.
- Configuration de Firecracker : permet de limiter le débit du périphérique de stockage en mode bloc et du périphérique réseau.
- Fonctions de sécurité de Firecracker : pour en savoir plus sur la conception de sécurité actuelle de Firecracker, consultez le dernier document de conception de Firecracker.

Outre les technologies d'isolation propriétaires d'AWS, ces mécanismes fournissent une isolation solide entre les environnements d'exécution.

Stockage et état

Les environnements d'exécution ne sont jamais réutilisés entre différentes versions de fonction ou différents clients. En revanche, un seul environnement peut être réutilisé entre les appels de la même version de fonction. Cela signifie que les données et l'état peuvent persister entre les appels. Les données et/ou l'état peuvent persister pendant des heures avant d'être détruits dans le cadre de la gestion normale du cycle de vie de l'environnement d'exécution. Pour des raisons de performances, les fonctions peuvent tirer parti de ce comportement afin d'améliorer l'efficacité en conservant et en réutilisant des caches locaux ou des connexions de longue durée entre les appels. Dans un environnement d'exécution, ces appels multiples sont gérés par un seul processus, de sorte que tout état à l'échelle du processus (tel qu'un état statique en Java) peut être disponible pour de futurs appels à réutiliser, si l'appel se produit dans un environnement d'exécution réutilisé.

Chaque environnement d'exécution Lambda inclut également un système de fichiers accessible en écriture, disponible dans `/tmp`. Les environnements d'exécution n'ont pas accès à ce stockage et ne peuvent pas le partager. De même que pour l'état du processus, les fichiers écrits dans `/tmp` sont conservés pendant toute la durée de vie de l'environnement d'exécution. Cela permet d'amortir des opérations de transfert coûteuses, telles que le téléchargement de modèles ML (machine learning), sur plusieurs appels. Les fonctions qui ne souhaitent pas conserver les données entre les appels ne doivent pas les écrire dans /tmp ou doivent supprimer leurs fichiers de /tmp entre chaque appel. Le répertoire `/tmp` est soutenu par un stockage d'instances Amazon EC2 et est chiffré au repos.

Les clients qui souhaitent conserver les données dans le système de fichiers en dehors de l'environnement d'exécution doivent envisager d'utiliser l'intégration de Lambda à Amazon Elastic File System (Amazon EFS). Pour en savoir plus, consultez Utilisation d'Amazon EFS avec AWS Lambda.

Si les clients ne souhaitent pas conserver les données ou l'état entre les appels, Lambda leur recommande de ne pas utiliser l'environnement ou le contexte d'exécution pour stocker des données ou un état. S'ils souhaitent empêcher activement les fuites de données ou d'état entre les appels, Lambda leur recommande de créer des fonctions distinctes pour chaque état. Lambda ne recommande pas aux clients d'utiliser ou de stocker les états affectant la sécurité dans l'environnement d'exécution, car ils peuvent subir une mutation entre les appels. Nous vous recommandons plutôt de recalculer l'état à chaque appel.

Maintenance de l'environnement d'exécution dans Lambda

Lambda prend en charge ces composants d'exécution en recherchant et en déployant en permanence des mises à jour compatibles et des correctifs de sécurité, et en effectuant d'autres activités de maintenance de l'environnement d'exécution. Cela permet aux clients de se concentrer uniquement sur la maintenance et la sécurité de tout code inclus dans leur fonction et leur couche. L'équipe Lambda utilise Amazon Inspector afin de détecter les problèmes de sécurité connus, ainsi que d'autres mécanismes de notification des problèmes de sécurité personnalisés et des listes de prédivulgation, afin que les correctifs continuent à être appliqués à nos langages et à notre environnement d'exécution. Si de nouveaux correctifs ou mises à jour sont identifiés, Lambda teste et déploie les mises à jour des composants d'exécution sans aucune intervention des clients. Pour en savoir sur le programme de conformité de Lambda, consultez la section « Lambda et conformité » de ce document.

En règle générale, aucune action n'est requise pour récupérer les derniers correctifs destinés aux composants d'exécution Lambda pris en charge. Toutefois, il arrive qu'une action soit requise pour tester les correctifs avant leur déploiement (par exemple, des correctifs de composants d'exécution incompatibles connus). Si une action est requise par les clients, Lambda les contacte par le biais d'AWS Personal Health Dashboard, par e-mail du compte AWS ou par tout autre moyen, en indiquant les mesures spécifiques à prendre.

Les clients peuvent utiliser d'autres langages de programmation dans Lambda en implémentant un environnement d'exécution personnalisé. Pour les environnements d'exécution personnalisés, la maintenance relève de la responsabilité du client, qui doit notamment s'assurer que cet environnement personnalisé inclut les derniers correctifs de sécurité. Pour en savoir plus, consultez la section Environnements d'exécution AWS Lambda personnalisés dans le Guide du développeur AWS Lambda.

Lorsque les responsables de la maintenance du langage d'exécution en amont marquent la fin de vie de leur langue, Lambda en tient compte en ne prenant plus en charge la version du langage d'exécution. Lorsque les versions de l'environnement d'exécution sont marquées comme obsolètes dans Lambda, Lambda cesse de prendre en charge la création de nouvelles fonctions et la mise à jour des fonctions existantes qui ont été créées dans l'environnement d'exécution obsolète. Pour avertir le client des prochaines obsolescences de l'environnement d'exécution, Lambda envoie des notifications aux clients concernant la prochaine date d'obsolescence et ce à quoi ils peuvent s'attendre. Lambda ne fournit pas non plus de mises à jour de sécurité, de support technique ou de correctifs logiciels pour les composants d'exécution obsolètes et se réserve le droit de désactiver

les appels des fonctions configurées pour s'exécuter sur un composant d'exécution obsolète à tout moment. Si les clients souhaitent continuer à exécuter des versions de composants d'exécution obsolètes ou non prises en charge, ils peuvent créer leur propre environnement d'exécution AWS Lambda personnalisé. Pour en savoir plus sur les cas où les composants d'exécution sont déconseillés, consultez la section Stratégie de prise en charge des environnements d'exécution AWS Lambda.

Surveillance et audit des fonctions Lambda

Vous pouvez surveiller et auditer les fonctions Lambda à l'aide de nombreux services et méthodes AWS, y compris les services suivants.

Amazon CloudWatch

AWS Lambda surveille automatiquement les fonctions Lambda en votre nom. Par le biais d'Amazon CloudWatch, il communique des métriques telles que le nombre de demandes, la durée d'exécution par demande et le nombre de demandes ayant entraîné une erreur. Ces métriques sont exposées au niveau de la fonction, que vous pouvez ensuite exploiter pour définir des alarmes CloudWatch. Pour obtenir la liste des métriques exposées par Lambda, consultez Métriques AWS Lambda.

Amazon CloudTrail

En utilisant AWS CloudTrail, vous pouvez mettre en œuvre la gouvernance, la conformité, l'audit opérationnel et l'audit des risques de l'ensemble de votre compte AWS, y compris Lambda. CloudTrail permet de journaliser, de surveiller en permanence et de conserver l'activité du compte liée aux actions menées sur l'ensemble de votre infrastructure AWS, en fournissant un historique complet des événements des actions effectuées au moyen de AWS Management Console, des kits SDK AWS, des outils de ligne de commande et d'autres services AWS. À l'aide de CloudTrail, vous pouvez éventuellement chiffrer les fichiers journaux à l'aide de AWS KMS ou encore exploiter la validation de l'intégrité des fichiers journaux CloudTrail pour confirmer leur validité.

AWS X-Ray

À l'aide de AWS X-Ray, vous pouvez analyser et déboguer les applications Lambda de production et distribuées pour comprendre les performances de votre application et de ses services sous-jacents, afin que vous puissiez éventuellement identifier et dépanner la cause première des problèmes de performances et des erreurs. Dans X-Ray, la vue complète des demandes au fur et à mesure qu'elles se déplacent dans votre application affiche une carte des composants sous-jacents de l'application, afin que vous puissiez analyser les applications pendant le développement et en production.

AWS Config

Grâce à AWS Config, vous pouvez suivre les modifications de configuration apportées aux fonctions Lambda (y compris les fonctions supprimées), aux environnements d'exécution, aux balises, au nom du gestionnaire, à la taille du code, à l'allocation de mémoire, aux paramètres d'expiration et aux paramètres de simultanéité, ainsi que le rôle d'exécution, le sous-réseau et les associations de groupes de sécurité Lambda IAM. Vous profitez ainsi d'une vue globale du cycle de vie de la fonction Lambda et vous pouvez mettre en évidence ces données pour des exigences d'audit et de conformité potentielles.

Architecture et exploitation des fonctions Lambda

Cette section traite de l'architecture et des opérations Lambda. Pour en savoir plus sur les bonnes pratiques standard relatives aux applications sans serveur, consultez le livre blanc Lentille d'applications serverless, qui définit et explore les piliers du cadre AWS Well-Architected Framework dans un contexte sans serveur.

- Pilier Excellence opérationnelle : capacité de faire fonctionner et de surveiller les systèmes afin de fournir une valeur opérationnelle et d'améliorer en continu les processus et les procédures sous-jacents.
- Pilier Sécurité : capacité de protéger les informations, les systèmes et les ressources tout en offrant une valeur opérationnelle avec des évaluations et l'atténuation des risques.
- Pilier Fiabilité : capacité d'un système à récupérer après des perturbations de l'infrastructure ou du service, à acquérir de manière dynamique les ressources de calcul pour satisfaire la demande et à atténuer les perturbations telles que les erreurs de configuration ou les problèmes réseau temporaires.
- Pilier Efficacité des performances : utilisation efficace des ressources de calcul pour répondre aux exigences et façon de maintenir cette efficacité à mesure que la demande change et que les technologies évoluent.
- Pilier Optimisation des coûts : processus continu de raffinement et d'amélioration visant à garantir que les résultats commerciaux sont atteints tout en minimisant les coûts à mesure que la demande et les technologies évoluent.

Le livre blanc Lentille d'applications serverless inclut des rubriques telles que la journalisation des métriques et des alarmes, la limitation de bande passante et les restrictions, l'attribution d'autorisations aux fonctions Lambda et la mise à disposition de données sensibles pour les fonctions Lambda.

Lambda et conformité

Comme indiqué dans la section « Modèle de responsabilité partagée », vous devez déterminer quel régime de conformité s'applique à vos données. Après avoir déterminé les critères de ce régime de conformité, vous pouvez utiliser les différentes fonctions de Lambda pour les respecter. Vous pouvez contacter des experts AWS (tels que des architectes de solutions, des experts dans un domaine, des gestionnaires de compte technique et d'autres ressources humaines) pour obtenir de l'aide. Toutefois, AWS ne peut pas conseiller ses clients sur la question de savoir si (ou quels) régimes de conformité sont applicables à un cas d'utilisation particulier.

Depuis novembre 2020, Lambda entre dans le champ d'application des rapports SOC 1, SOC 2 et SOC 3, qui sont des comptes rendus rédigés suite à un audit indépendant réalisé par un tiers qui démontrent comment AWS réalise les principaux contrôles et objectifs de conformité. Pour obtenir la liste à jour des informations de conformité, consultez la page Services AWS concernés par le programme de conformité.

En raison de la nature sensible de certains rapports de conformité, ceux-ci ne peuvent pas être partagés publiquement. Pour accéder à ces rapports, vous pouvez vous connecter à AWS Management Console et utiliser AWS Artifact, un portail en libre-service gratuit permettant d'accéder à la demande aux rapports de conformité AWS.

Sources d'événements Lambda

Lambda s'intègre à plus de 140 services AWS au moyen de l'intégration directe et du bus d'événements Amazon EventBridge. Les sources d'événements Lambda couramment utilisées sont les suivantes :

- Amazon API Gateway
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon DynamoDB Streams
- Amazon EventBridge
- Amazon Kinesis Data Streams
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Step Functions

Grâce à ces sources d'événements, vous pouvez :

- Utiliser AWS Identity and Access Management pour gérer l'accès au service et aux ressources en toute sécurité.
- Chiffrer vos données au repos*. Tous les services chiffrent les données en transit.
- Accéder aux données depuis votre Amazon Virtual Private Cloud à l'aide de points de terminaison d'un VPC (optimisation par AWS PrivateLink)
- Utiliser Amazon CloudWatch Application Insights pour collecter les métriques, générer des rapports et émettre des alarmes.
- Utiliser AWS CloudTrail pour journaliser, surveiller en permanence et conserver l'activité du compte liée aux actions menées sur l'ensemble de votre infrastructure AWS, en fournissant un historique complet des événements des actions effectuées au moyen de AWS Management Console > kits SDK AWS, des outils de ligne de commande et d'autres services AWS.

* Au moment de la publication, le chiffrement des données au repos n'était pas disponible pour Amazon EventBridge. Consultez régulièrement les pages d'accueil des services pour obtenir des mises à jour sur ces fonctionnalités.

Conclusion

AWS Lambda propose une boîte à outils puissante pour générer des applications sécurisées et évolutives. La plupart des bonnes pratiques en matière de sécurité et de conformité dans Lambda sont les mêmes que dans tous les services AWS, mais certaines sont spécifiques à Lambda. Ce livre blanc décrit les avantages de Lambda, son adéquation aux applications et l'environnement d'exécution géré par Lambda. Il comprend également des informations sur la surveillance et l'audit, ainsi que les bonnes pratiques en matière de sécurité et de conformité. Lorsque vous réfléchissez à votre prochaine implémentation, tenez compte de ce que vous avez appris au sujet d'AWS Lambda et de la façon dont il pourrait améliorer votre prochaine solution de charge de travail.

Participants

Ont contribué à la préparation du présent document :

- Mayank Thakkar, architecte de solutions, secteur mondial des sciences de la vie
- Marc Brooker, ingénieur principal senior (sans serveur)
- Osman Surkatty, ingénieur principal en sécurité (sans serveur)

Autres lectures

Pour en savoir plus, voir :

- Modèle de responsabilité partagée, qui explique comment AWS envisage la sécurité en général.
- Bonnes pratiques de sécurité dans IAM, qui couvre les recommandations relatives au service AWS Identity and Access Management (IAM).
- Lentille d'applications serverless, qui couvre le cadre AWS Well-Architected Framework et identifie les éléments clés pour garantir que vos charges de travail sont architecturées conformément aux bonnes pratiques.
- Introduction à la sécurité dans AWS, qui fournit une introduction générale à la réflexion sur la sécurité dans AWS.
- Risque et conformité AWS, qui fournit un aperçu de la conformité dans AWS.

Révisions du document

Pour être informé des mises à jour de ce livre blanc, abonnez-vous au flux RSS.

update-history-change	update-history-description	update-history-date
Mis à jour	Mises à jour importantes	15 février 2021
Publication initiale	Première publication du livre blanc	3 janvier 2019

Mentions légales

Les clients sont responsables de leur propre évaluation indépendante des informations contenues dans ce document. Le présent document : (a) est fourni à titre informatif uniquement, (b) représente les offres et pratiques actuelles de produits AWS, qui sont susceptibles d'être modifiées sans préavis, et (c) ne crée aucun engagement ou assurance de la part d'AWS et de ses affiliés, fournisseurs ou concédants de licences. Les produits ou services AWS sont fournis « en l'état » sans garantie, représentation ou condition, de quelque nature que ce soit, explicite ou implicite. Les responsabilités et obligations d'AWS envers ses clients sont déterminées par les contrats AWS, et le présent document ne fait pas partie d'un contrat entre AWS et ses clients, ni le modifie.