SÉCURITÉ DES APPLICATIONS DANS AWS : CONFIGURATION ET GESTION DES SERVICES SANS SERVEUR

Ana-Maria VECHIU

Université Technique de Moldavie, Faculté Ordinateur, Informatique et Microélectronique,Département Génie Logiciel et Automatique, gr. TI-231M, Chișinău, République de Moldavie

Auteur correspondant: vechiu.ana-maria@isa.utm.md

Coordinateur scientifique: **Daniela ISTRATI**, asist. univ., dr.

Résumé: *Dans un monde numérique en pleine évolution, les services sans serveur proposés par AWS sont de plus en plus populaires auprès des développeurs d'applications. Cependant, une préoccupation majeure associée à l'adoption de ces services est la sécurité des applications. Cet article se concentre sur l'importance de configurer et de gérer correctement les services serverless dans AWS pour garantir un niveau de sécurité élevé. Tout d'abord, nous explorons les avantages offerts par l'architecture serverless, tels que l'évolutivité automatique et l'absence de gestion de l'infrastructure. Cependant, ces avantages s'accompagnent de risques de sécurité spécifiques, tels que l'octroi excessif de privilèges, l'injection de code, l'exposition de données sensibles, et plus encore. Pour faire face à ces risques, nous allons examiner les meilleures pratiques pour configurer et gérer les services sans serveur dans AWS.*

Mots-clés: *Amazon Web Services (AWS), Gestion des identités et des accès (IAM), serverless, infrastructure, chiffrement.*

Introduction

Ces dernières années, le cloud est devenu un élément central de l'infrastructure informatique de nombreuses organisations, offrant d'importantes opportunités d'innovation et d'efficacité. Dans cet environnement numérique en constante évolution, Amazon Web Services (AWS) a acquis une position dominante, en proposant une gamme diversifiée de services cloud pour répondre aux divers besoins des clients. Parmi ces services (Figure 1), l'architecture sans serveur est devenue de plus en plus populaire, permettant aux développeurs de créer et d'exécuter des applications sans gérer l'infrastructure sous-jacente [1].

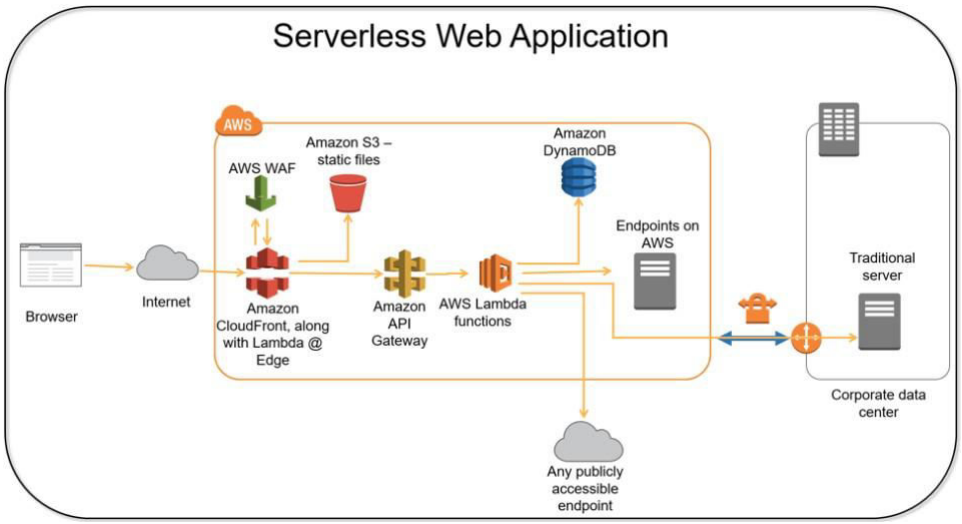

Figure 1. Principe des services sans serveur [1].

Cependant, une préoccupation majeure associée à l'adoption d'une architecture sans serveur est la sécurité des applications. Une configuration et une gestion appropriées de ces services deviennent essentielles pour garantir un environnement sécurisé et protégé contre les cybermenaces. Par conséquent, cet article vise à explorer en profondeur l'importance de la sécurité des applications dans AWS, en se concentrant sur la configuration et la gestion appropriées des services sans serveur.

Nous étudierons ensuite les risques de sécurité spécifiques associés à l'architecture sans serveur, explorerons les meilleures pratiques de protection des applications et passerons en revue les outils et technologies disponibles pour mettre en œuvre des solutions de sécurité efficaces. Ce faisant, nous visons à fournir aux lecteurs une solide compréhension des aspects clés de la sécurité des applications dans l'environnement AWS et à faciliter le développement de stratégies robustes pour protéger les données et l'infrastructure cloud [1].

1. Qu'est-ce qu'AWS et à quoi sort-il ?

AWS, ou Amazon Web Services, est une suite de services cloud fournis par Amazon. Ces services sont conçus pour fournir une large gamme de fonctionnalités de calcul, de stockage et autres pour aider les organisations à développer et à exécuter leurs applications sans avoir à gérer l'infrastructure physique.

L'architecture sans serveur est un modèle de développement et de déploiement d'applications dans lequel les développeurs se concentrent uniquement sur le code de l'application sans avoir à gérer l'infrastructure sous-jacente. En d'autres termes, dans un environnement sans serveur, le fournisseur de cloud s'occupe de la gestion de l'infrastructure et de la mise à l'échelle automatique des ressources, et les développeurs peuvent concentrer leurs efforts sur le développement du code et la logique des applications.

AWS propose un certain nombre de services sans serveur qui facilitent le développement et le déploiement d'applications sans avoir besoin de gérer des serveurs ou une infrastructure. Parmi les services sans serveur les plus populaires proposés par AWS figurent:

1. AWS Lambda, c'est un service qui permet au code de s'exécuter sans avoir besoin de gérer des serveurs. Les développeurs peuvent charger des fonctions Lambda et les exécuter en fonction d'événements ou de requêtes HTTP.
2. Amazon API Gateway, c'est un service qui permet la création, la publication, la gestion et la protection des API. Il est couramment utilisé avec AWS Lambda pour créer des API sans serveur.
3. Amazon DynamoDB, il s'agit d'un service de base de données NoSQL entièrement géré qui peut être intégré à des services sans serveur pour stocker et accéder aux données d'application.
4. AWS Step Functions, il s'agit d'un service qui permet aux développeurs de coordonner et de gérer les interactions entre différentes fonctions Lambda ou d'autres composants d'application.
5. AWS AppSync, il s'agit d'un service entièrement géré qui permet aux développeurs de créer rapidement des applications basées sur les données avec des données hors ligne et des fonctionnalités en temps réel.

En utilisant les services sans serveur fournis par AWS, les développeurs peuvent créer et déployer des applications plus rapidement sans avoir à se soucier de la gestion de l'infrastructure. Ce modèle peut également contribuer à réduire les coûts et à faire évoluer automatiquement les ressources en fonction des besoins des applications [2].

2. Que signifie un service sans serveur et comment est-il configuré ?

Un service sans serveur au sein d'AWS signifie créer et déployer des applications sans avoir à se soucier de la gestion de l'infrastructure sous-jacente. Essentiellement, ce modèle de

développement vous permet de vous concentrer uniquement sur le code de votre application, tandisqu'AWS s'occupe de la gestion et de la mise à l'échelle de vos ressources de calcul.

Les principaux composants d'un service sans serveur incluent les fonctions de calcul, les événements et les déclencheurs, le stockage et les bases de données, ainsi que API Gateway pour exposer les fonctions en tant que points de terminaison d'API [2].

Mettre en place un service sans serveur dans AWS implique avant tout de créer et de définir des fonctions de calcul, qui sont des unités de code qui seront exécutées de manière réactive à des événements spécifiques. Ensuite, vous devez configurer les événements et les déclencheurs qui déclencheront l'exécution des fonctions de calcul. Il est également important d'accorder les autorisations appropriées aux fonctions de calcul pour accéder à d'autres services AWS, tels que le stockage ou la base de données.

De plus, si les fonctions de calcul doivent être exposées en tant que points de terminaison d'API, vous devrez créer et configurer une passerelle API pour gérer les routes et les requêtes HTTP.

Une fois toutes ces configurations terminées, l'application peut être testée et déployée en production pour être utilisée par les utilisateurs, AWS prenant en charge la gestion de l'infrastructure et mettant automatiquement à l'échelle les ressources en fonction des besoins de l'application [3].

Par conséquent, l'utilisation de services sans serveur au sein d'AWS facilite le développement et le déploiement d'applications, supprimant la complexité de la gestion de l'infrastructure et permettant aux développeurs de se concentrer sur la création de valeur via le code d'application.

Pour configurer un service sans serveur à l'aide d'AWS (figure 2), les étapes typiques incluent :

Création de fonctions Lambda, les développeurs doivent écrire et télécharger le code de la fonction Lambda sur la console AWS Lambda ou utiliser les outils de développement CLI et SDK pour créer et télécharger les fonctions.

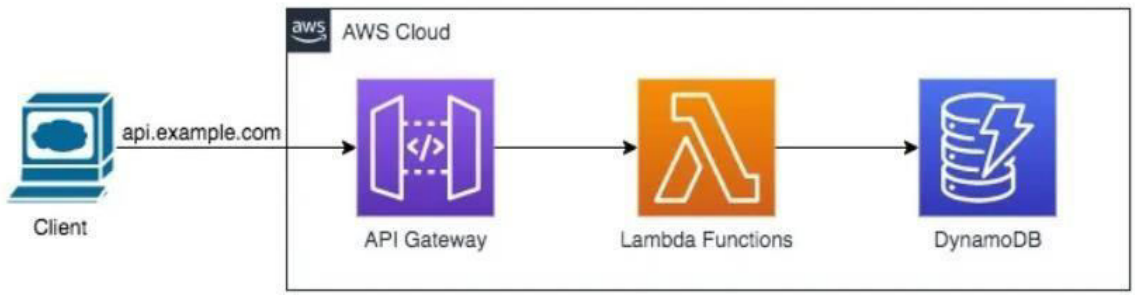

Figure 2. Modèle d'architecture sans serveur – Service API back-end [3].

Définition du déclencheur de fonction, c'est l'étape où les développeurs précisent les événements qui déclencheront l'exécution de la fonction Lambda. Cela peut être configuré à l'aide de la console AWS Lambda ou de l'API Lambda.

Configuration de la stratégie d'autorisations IAM, il est important d'accorder aux fonctions Lambda les autorisations nécessaires pour accéder à d'autres services AWS, tels qu'Amazon S3 ou Amazon DynamoDB. Ceci peut être réalisé en définissant correctement les politiques d'autorisation IAM.

Créer et configurer API Gateway (si nécessaire) si les fonctions Lambda doivent être exposées en tant que points de terminaison d'API, les développeurs doivent créer et configurer une API Gateway pour gérer les routes et les requêtes HTTP.

Test et déploiement, après avoir configuré les fonctions Lambda et les autres ressources associées, il est important de tester l'application pour s'assurer que tout fonctionne comme prévu.

Après les tests, l'application peut être déployée et mise en production pour être utilisée par lesutilisateurs [3].

3. Défis de sécurité des applications dans AWS

Les défis en matière de sécurité (Figure 3) des applications au sein d'AWS peuvent être variés et complexes compte tenu de l'environnement dynamique et distribué des services cloud. Voici quelques-uns de ces défis :

1. Accès non autorisé, une bonne gestion des accès et des autorisations est essentielle dans AWS. Les utilisateurs ne doivent avoir accès qu'aux ressources et fonctionnalités requises pour leur rôle spécifique. Un accès non autorisé peut entraîner une exposition des données ou une compromission des systèmes.
2. Mauvaises configurations, les mauvaises configurations des services AWS peuvent créer des vulnérabilités de sécurité. Par exemple, une stratégie d'accès ou des paramètres de groupe de sécurité incorrects peuvent permettre un accès non autorisé ou une exposition des données.
3. Inspections et audits de sécurité insuffisants, le manque de surveillance et de journalisation détaillées peut faire passer des activités suspectes inaperçues. Il est important de mener des inspections et des audits réguliers pour détecter et enquêter sur les incidents de sécuritépotentiels.
4. Cryptage des données insuffisant, les données stockées dans le cloud peuvent présenter un risque d'accès non autorisé ou d'interception en cours de transit. Le chiffrement des données aurepos et en transit est essentiel à la protection de la confidentialité des informations.
5. Vulnérabilités des applications, les applications développées et déployées dans AWS peuvent être vulnérables à divers types d'attaques, telles que l'injection de code, le cross-site scripting (XSS) ou l'exploitation de vulnérabilités de sécurité connues.
6. Problèmes de conformité, les organisations doivent se conformer à diverses réglementations et normes en matière de sécurité et de confidentialité des données lorsqu'elles utilisent les services AWS. Le non-respect de ces normes peut entraîner des amendes et d'autres conséquences juridiques.
7. Réponse inadéquate aux incidents, une fois qu'un incident de sécurité a été détecté, il est essentiel de réagir rapidement et efficacement pour limiter l'impact et remédier aux vulnérabilités. L'absence d'un plan de réponse aux incidents ou d'une capacité de surveillance adéquate peut entraîner des retards dans la détection et la résolution des incidents [4].

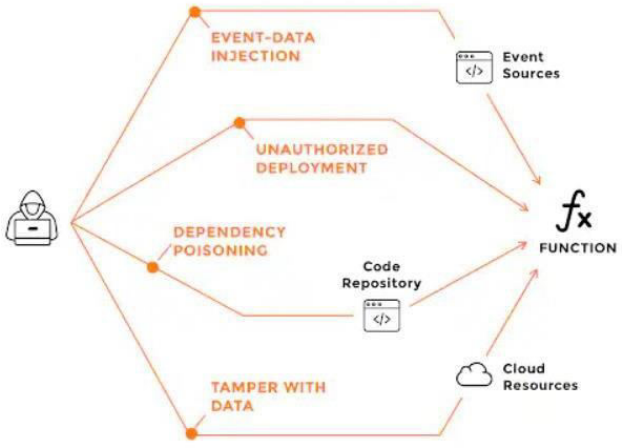

Figure 3. Surface d'attaque par injection de données d'événement [4].

La sécurité des applications dans AWS implique la gestion d'un large éventail de menaces et de défis, et y répondre nécessite une attention constante aux détails, la mise en œuvre de politiques et de pratiques de sécurité solides et l'utilisation d'outils appropriés pour surveiller et gérer les risques de sécurité.

4. Meilleures pratiques pour la sécurité des applications dans AWS

La sécurisation des applications au sein d'AWS est une tâche complexe et essentielle à la protection des données et de l'infrastructure contre les cybermenaces. La mise en œuvre des meilleures pratiques est cruciale pour créer un environnement cloud sécurisé et fiable.

L'une des pratiques les plus importantes consiste à gérer l'accès avec des stratégies IAM, ce qui implique de définir et d'appliquer des stratégies d'accès granulaires pour les utilisateurs et les rôles. La surveillance et la journalisation des activités sont également essentielles pour détecter et enquêter sur les activités suspectes ou indésirables. Le cryptage des données est une autre mesure cruciale pour protéger la confidentialité des informations, tandis que la maintenance des mises à jour et des correctifs est importante pour corriger les vulnérabilités connues. L'utilisation des services de sécurité fournis par AWS, tels que WAF et Shield, complète l'arsenal de sécurité d'une application [5].

Il est également essentiel de mettre en œuvre un modèle de sécurité dès la conception, qui impliqued'intégrer les principes de sécurité dès le début du développement des applications. Un examen régulier de la configuration et des politiques de sécurité est également important pour garantir la conformité et résoudre les problèmes de sécurité.

En appliquant ces bonnes pratiques et en adoptant une approche proactive en matière de sécurité, les organisations peuvent créer un environnement sécurisé et protégé pour leurs applications sur AWS [5].

Conclusion

L'article souligne l'importance cruciale de sécuriser les applications dans l'environnement en constante évolution d'Amazon Web Services (AWS), en mettant particulièrement l'accent sur la configuration et la gestion appropriées des services sans serveur. Alors que le cloud computing devient la norme pour de nombreuses organisations, l'adoption croissante des services sans serveur offre une agilité et une évolutivité inégalées, tout en déplaçant la responsabilité de la gestion de l'infrastructure vers le fournisseur de services cloud. Cependant, cette évolution rapide pose des défis significatifs en matière de sécurité des applications.

L'article explore en profondeur les risques spécifiques associés à l'architecture sans serveur, allant de l'octroi excessif de privilèges à l'exposition de données sensibles. Face à ces défis, il est impératif de mettre en œuvre des pratiques de sécurité rigoureuses. Cela comprend une gestion précise des accès avec des politiques IAM bien définies, une surveillance active des activités pour détecter les comportements suspects, le cryptage des données pour protéger leur confidentialité, et une maintenance rigoureuse des mises à jour et des correctifs pour remédier aux vulnérabilités connues.

En suivant ces meilleures pratiques, les organisations peuvent renforcer la sécurité de leurs applications dans AWS, réduisant ainsi le risque de violations de données et de perturbations des services. De plus, une collaboration étroite avec les équipes de sécurité et l'adoption de technologies de sécurité avancées telles que les services de protection contre les attaques distribuées par déni de service (DDoS) et les pare-feu d'applications Web (WAF) renforcent la posture de sécurité globale de l'infrastructure cloud.

En fin de compte, la sécurité des applications dans AWS n'est pas seulement une responsabilité technique, mais aussi une priorité commerciale critique. Les organisations doivent adopter une approche holistique de la sécurité, intégrant des mesures préventives, des mécanismes de détection et des plans d'intervention en cas d'incident pour protéger leurs actifs numériques et préserver la confiance de leurs clients. En s'engageant dans un effort continu pour rester à jour

avec les meilleures pratiques de sécurité et en favorisant une culture de la sécurité au sein de l'organisation, les entreprises peuvent prospérer dans le paysage numérique actuel tout en minimisant les risques de sécurité.

Bibliographie

[1] Catherine Recanati.Cours « *What is serveless-security* » [Accédé le 10.03.24], sur le site web : https://www.paloaltonetworks.com/cyberpedia/what-is-serverless-security

[2] Jean Caelen. *Systèmes AWS*, [Accédé le 10.03.24], sur le site web : https://waswani.medium.com/serverless-architecture-patterns-in-aws-edeab0e46a323

[3] Olivier Chapui, *AWS-Serverless services and thei use case,* [Accédé le 15.03.24], sur le siteweb : https://blog.kloudmate.com/aws-serverless-services-and-their-use-cases-part-1-fd3f238ec6cd4

[4] Michel Beaudouin cours « *AWS basics »* [Accédé le 15.03.24], sur le site web : https://www.spiceworks.com/tech/cloud/articles/aws-basics/

[5] Michel Beaudouin-Lafon , *All AWS service list,* [Accédé le 15.03.24], sur le site web : https://allcode.com/tag/all-aws-services-list/