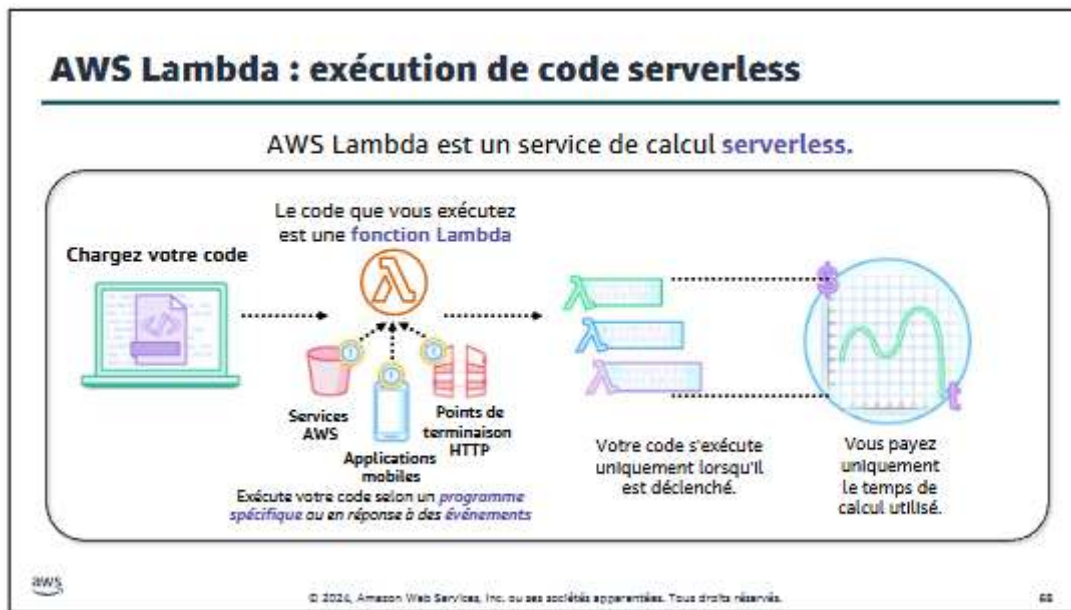


## Introduction à AWS Lambda



AWS propose de nombreuses options de calcul. Par exemple, Amazon EC2 fournit des machines virtuelles. Comme autre exemple, Amazon ECS et Amazon EKS sont des services de calcul basés sur des conteneurs.

Cependant, il existe une autre approche de calcul qui ne vous oblige pas à mettre en service ni à gérer des serveurs. Cette troisième approche est souvent appelée informatique serverless.

AWS Lambda est un service de calcul serverless dicté par des événements. Il vous permet d'exécuter du code sans avoir à mettre en service ou gérer des serveurs.

Vous devez créer une fonction Lambda, qui correspond à la ressource AWS qui contient le code que vous chargerez. Vous devez ensuite configurer la fonction Lambda pour qu'elle soit déclenchée soit de manière planifiée, soit en réponse à un événement.

Le code ne s'exécutera que lorsqu'il sera déclenché. Vous payez uniquement le temps de calcul utilisé. Vous ne payez rien lorsque votre code n'est pas exécuté.

Avec AWS Lambda, inutile d'apprendre un nouveau langage, outil ou framework. Lambda prend en charge plusieurs langages de programmation, notamment Java, Go, PowerShell, Node.js, C#, Python et Ruby. Votre code peut utiliser n'importe quelle bibliothèque, native ou tierce.

## Avantages de Lambda



**AWS  
Lambda**



Prise en charge de plusieurs langages de programmation



Automatisation complète de l'administration



Tolérance aux pannes intégrée



Prise en charge de l'orchestration de plusieurs fonctions



Tarification à l'utilisation



© 2024, Amazon Web Services, Inc. ou ses sociétés apparentées. Tous droits réservés.

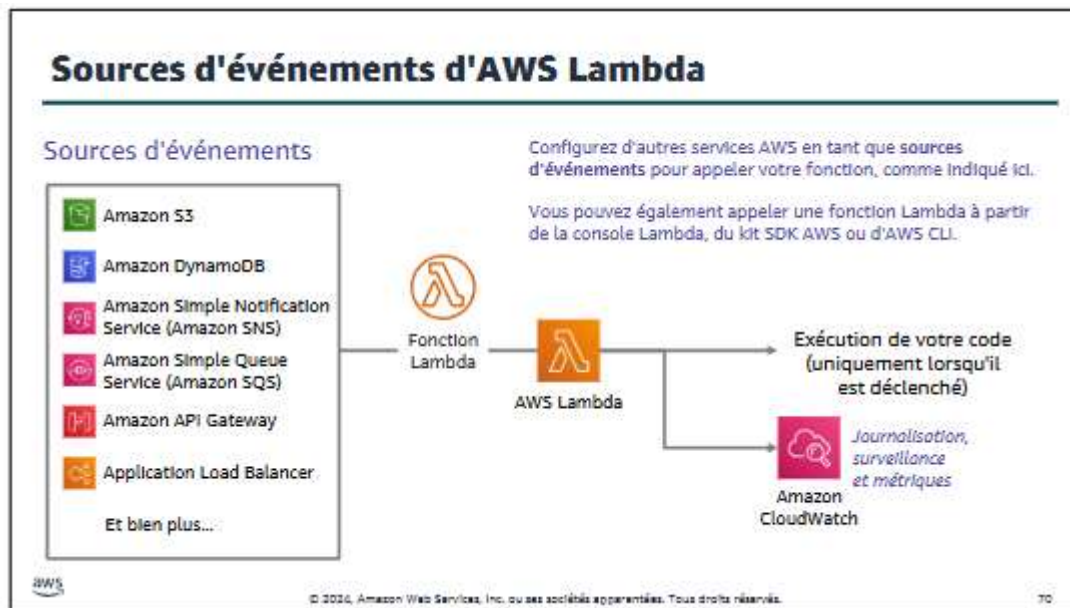
69

Lambda automatise complètement l'administration. Il gère toute l'infrastructure pour exécuter votre code sur une infrastructure à haute disponibilité et tolérante aux pannes, ce qui vous permet de vous concentrer sur la création de services backend différenciés. Lambda déploie aisément votre code et s'occupe des tâches administratives, de la maintenance et des correctifs de sécurité. De plus, il fournit une journalisation intégrée et une surveillance par le biais d'Amazon CloudWatch.

Lambda intègre un système de tolérance aux pannes. Il gère la capacité de calcul sur plusieurs zones de disponibilité au sein de chaque région. Vous bénéficiez ainsi d'une protection renforcée de votre code en cas de panne d'ordinateurs individuels ou de défaillance de l'ensemble de votre centre de données. Il n'y a ni fenêtres de maintenance ni arrêts programmés.

Vous pouvez orchestrer plusieurs fonctions Lambda pour des tâches complexes ou de longue durée en créant des flux de travail avec AWS Step Functions. Utilisez AWS Step Functions pour définir les flux de travail. Ceux-ci déclenchent un ensemble de fonctions Lambda à l'aide d'étapes séquentielles, parallèles, de ramification et de gestion des erreurs. Avec AWS Step Functions et Lambda, vous pouvez créer des processus avec état et de longue durée pour les applications et les backends.

Avec Lambda, vous ne payez que les requêtes distribuées et le temps de calcul requis pour exécuter votre code. La facturation est calculée par incréments de 100 millisecondes, permettant de passer automatiquement, de manière économique et facile, de quelques requêtes par jour à plusieurs milliers de requêtes par seconde.



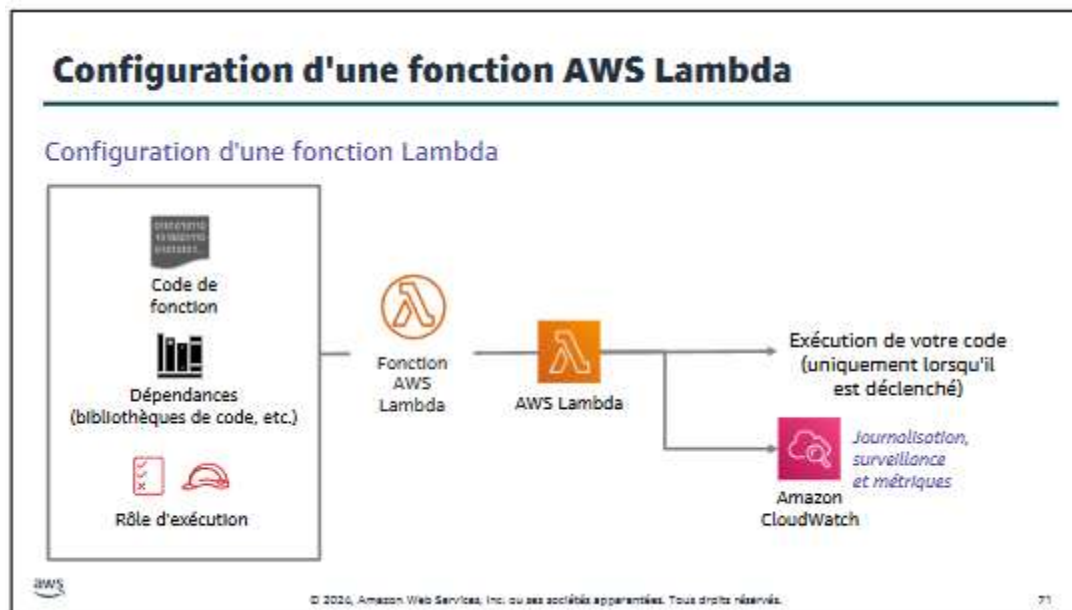
Une source d'événement est un service AWS ou une application créée par un développeur qui déclenche l'exécution d'une fonction AWS Lambda. Certains services publient des événements sur Lambda en appelant directement la fonction Lambda. Les services qui appellent les fonctions Lambda de manière asynchrone incluent, mais sans s'y limiter, Amazon S3, Amazon Simple Notification Service (Amazon SNS) et Amazon CloudWatch Events.

Lambda peut également interroger des ressources dans d'autres services qui ne publient pas d'événements sur Lambda. Par exemple, Lambda peut extraire des enregistrements d'une file d'attente Amazon Simple Queue Service (Amazon SQS) et exécuter une fonction Lambda pour chaque message récupéré. Lambda peut également lire les événements à partir d'Amazon DynamoDB.

Certains services, tels qu'Elastic Load Balancing (Application Load Balancer) et Amazon API Gateway peuvent appeler directement votre fonction Lambda.

Vous pouvez appeler les fonctions Lambda directement avec la console Lambda, l'API Lambda, le kit SDK AWS, AWS CLI et les boîtes à outils AWS. L'approche d'appel direct peut être utile, par exemple lorsque vous développez une application mobile et que vous souhaitez qu'elle appelle des fonctions Lambda.

AWS Lambda surveille automatiquement les fonctions Lambda à l'aide d'Amazon CloudWatch. Pour vous aider à dépanner les défaillances d'une fonction, Lambda journalise toutes les demandes traitées par votre fonction. Il stocke également automatiquement les journaux générés par votre code via Amazon CloudWatch Logs.



N'oubliez pas qu'une fonction Lambda est le code personnalisé que vous écrivez pour traiter des événements et que Lambda exécute cette fonction en votre nom.

Lorsque vous utilisez la Console de gestion AWS pour créer une fonction Lambda, vous donnez d'abord un nom à la fonction. Puis, vous spécifiez les détails suivants :

- L'environnement de runtime que la fonction utilisera (par exemple, une version de Python ou Node.js)
- Un rôle d'exécution (pour accorder l'autorisation IAM à la fonction afin qu'elle puisse interagir avec d'autres services AWS si nécessaire).

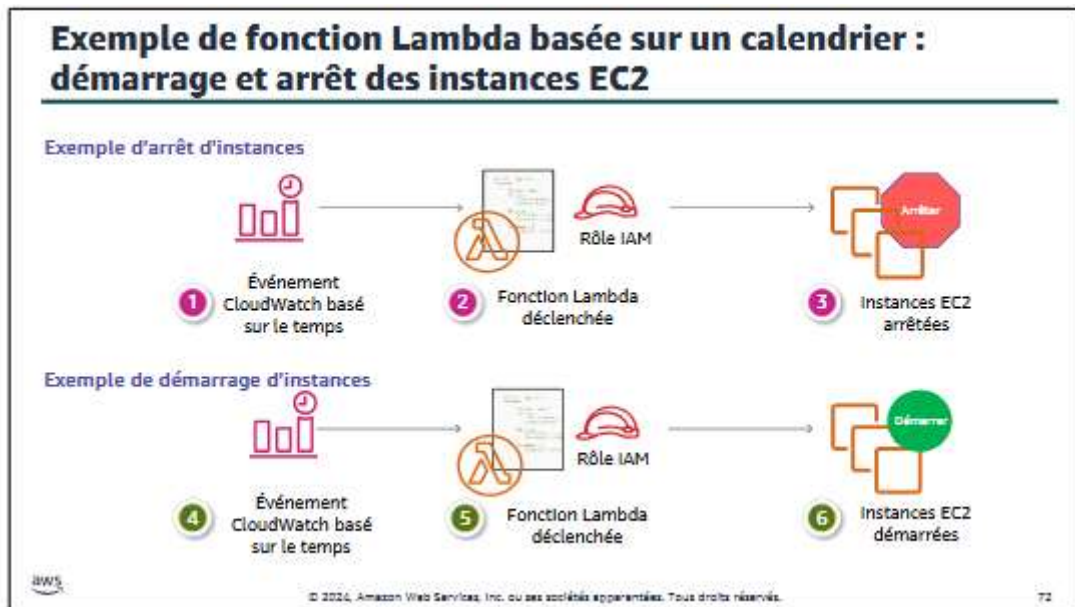
Ensuite, après avoir cliqué sur Créer la fonction, vous devez la configurer. Voici les configurations possibles:

- Ajoutez un déclencheur (spécifiez l'une des sources d'événements disponibles à partir de la diapositive précédente).
- Ajoutez votre code de fonction (utilisez l'éditeur de code fourni ou chargez un fichier contenant votre code).
- Spécifiez la mémoire en Mo à allouer à votre fonction (128Mo à 10240Mo).
- Vous pouvez aussi définir les variables d'environnement, la description, le délai d'expiration, le cloud privé virtuel(VPC) spécifique dans lequel exécuter la fonction, les balises que vous souhaitez utiliser et d'autres paramètres.

Tous les paramètres ci-dessus se retrouvent dans un package de déploiement Lambda qui est un ZIP contenant le code de fonction et vos dépendances.

Lorsque vous utilisez la console Lambda pour créer votre fonction, celle-ci gère

le package pour vous. Cependant, vous devez créer un package de déploiement si vous utilisez l'API Lambda pour gérer les fonctions.



Prenons un exemple de cas d'utilisation pour une fonction Lambda basée sur un calendrier.

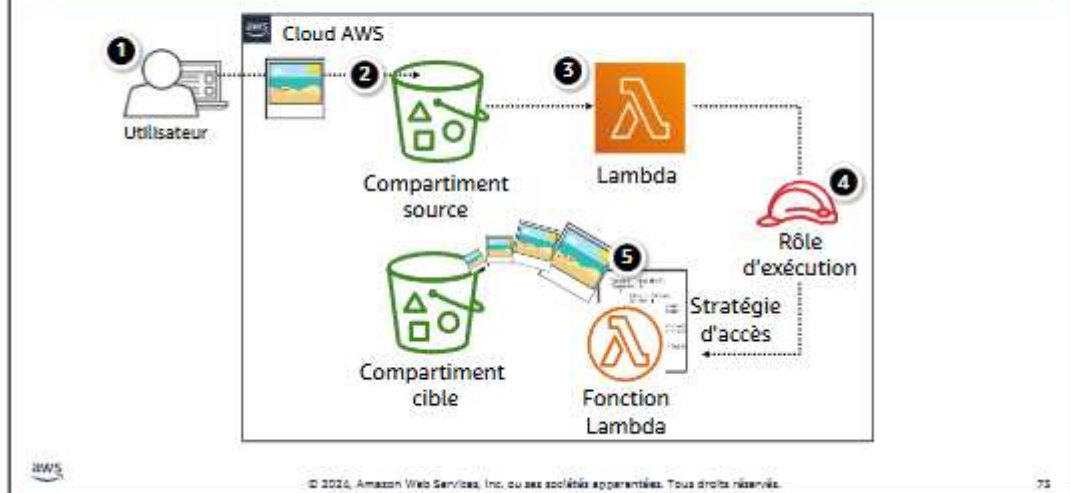
Supposons que vous souhaitiez réduire votre utilisation d'AmazonEC2. Vous décidez d'arrêter les instances à une heure prédéfinie (par exemple, la nuit lorsque personne n'y accède), puis de les redémarrer le matin (avant le début de la journée de travail).

Dans ce cas, vous pouvez configurer AWS Lambda et Amazon CloudWatch Events pour vous aider à effectuer ces actions automatiquement.

Voici ce qui se passe à chaque étape de cet exemple:

1. Un événement CloudWatch est planifié pour exécuter une fonction Lambda afin d'arrêter vos instances EC2 à (par exemple) 22h GMT.
2. La fonction Lambda est déclenchée et s'exécute avec le rôle IAM qui lui accorde l'autorisation d'arrêter les instances EC2.
3. Les instances EC2 passent à l'état Arrêté.
4. Plus tard, à (par exemple) 05h GMT, un événement CloudWatch est planifié pour exécuter une fonction Lambda afin de démarrer les instances EC2.
5. La fonction Lambda est déclenchée et s'exécute avec le rôle IAM qui lui accorde l'autorisation de démarrer les instances EC2.
6. Les instances EC2 passent à l'état En cours d'exécution.

## Exemple de fonction Lambda basée sur les événements : créer des miniatures



Prenons maintenant un exemple de cas d'utilisation pour une fonction Lambda basée sur un événement. Supposons que vous souhaitez créer une miniature pour chaque image (objets .jpg et .png) qui est importée dans un compartiment S3.

Pour créer une solution, vous pouvez créer une fonction Lambda qu'Amazon S3 appellera lorsque des objets seront importés. Ensuite, la fonction Lambda lira l'objet image à partir du compartiment source et créera une image miniature dans le compartiment cible.

1. Un utilisateur charge un objet dans le compartiment source d'Amazon S3 (événement de création d'objet).
2. AmazonS3 détecte l'événement de création d'objet.
3. AmazonS3 publie l'événement de création d'objet dans Lambda en appelant la fonction Lambda et en transmettant les données d'événement.
4. Lambda exécute la fonction Lambda en assumant le rôle d'exécution que vous avez spécifié au moment de la création de cette fonction.
5. À partir des données d'événement qu'elle reçoit, la fonction Lambda identifie le nom du compartiment source et le nom de clé de l'objet. Elle lit l'objet et crée une miniature à l'aide des bibliothèques graphiques, puis enregistre la miniature dans le compartiment cible.



## Quotas AWS Lambda

Limites souples par région :

- Exécutions simultanées = 1 000
- Stockage des fonctions et des couches = 75 Go

Limites strictes pour les fonctions individuelles :

- Allocation maximale de mémoire de la fonction = 10 240 Mo
- Délai d'expiration de la fonction = 15 minutes
- Taille du package de déploiement = 250 Mo décompressé, y compris les couches
- Taille du package de code d'image de conteneur = 10 Go

Des limites supplémentaires existent également. Pour plus d'informations, consultez la documentation Quotas AWS Lambda :

<https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html>.



Voici quelques points clés à retenir de cette section du module :

- Le calcul serverless permet de concevoir et d'exécuter des applications et des services sans avoir à se soucier de la mise en service ni de la gestion des serveurs.
- AWS Lambda est un service de calcul serverless qui offre une tolérance aux pannes intégrée et une mise à l'échelle automatique.
- Une source d'événement est un service AWS ou une application créée par un développeur qui déclenche l'exécution d'une fonction Lambda.
- L'allocation de mémoire maximale pour une seule fonction Lambda est de 10240Mo.
- La durée d'exécution maximale d'une fonction Lambda est de 15minutes.