Amazon Web Services (AWS)

Table of Contents

[Amazon Web Services (AWS) 3](#_Toc185186323)

[AWS Lambda 4](#_Toc185186324)

[Qu’est-ce que AWS Lambda? 4](#_Toc185186325)

[Comment est-ce que AWS Lambda fonctionne? 4](#_Toc185186326)

[AWS Lambda vs Amazon Elastic Compute Cloud (EC2) 5](#_Toc185186327)

[Les cas d’utilisation d’AWS Lambda 5](#_Toc185186328)

[Amazon S3 (Simple Storage Service) 6](#_Toc185186329)

[Qu’est-ce que Amazon S3? 6](#_Toc185186330)

[Ses bénéfices 6](#_Toc185186331)

[Comment est-ce que Amazon S3 fonctionne? 6](#_Toc185186332)

[Les cas d’utilisation d’Amazon S3 7](#_Toc185186333)

[Amazon DynamoDB 8](#_Toc185186334)

[Qu’est-ce qu’Amazon DynamoDB? 8](#_Toc185186335)

[Comment est-ce que DynamoDB fonctionne? 8](#_Toc185186336)

[DynamoDB vs SimpleDB 9](#_Toc185186337)

[DynamoDB vs MongoDB 9](#_Toc185186338)

[Les cas d’utilisation de DynamoDB 9](#_Toc185186339)

[AWS Amplify 10](#_Toc185186340)

[Qu’est-ce que AWS Amplify? 10](#_Toc185186341)

[Comment est-ce que AWS Amplify fonctionne? 10](#_Toc185186342)

[AWS Amplify vs Google Firebase 11](#_Toc185186343)

[Les cas d’utilisation d’Amplify 11](#_Toc185186344)

[Bibliographie 12](#_Toc185186345)

[Veille technologique 14](#_Toc185186346)

[Moteurs de recherches spécialisés 14](#_Toc185186347)

[Google Alerts 16](#_Toc185186348)

[Flux RSS 16](#_Toc185186349)

[Agent de surveillance (Infominder) 17](#_Toc185186350)

[Média social (Twitter) 18](#_Toc185186351)

# Amazon Web Services (AWS)

Amazon Web Services (AWS) est une plateforme cloud offerte par Amazon qui recouvre plus de 200 services cloud venant des centres de données du monde entier. Leurs services sont utilisés par des millions de clients, incluant les petites, moyennes et grandes entreprises.

Comme services clés, AWS offre :

* Des outils comme Elastic Compute Cloud (EC2) permettant d’exécuter les applications dans des serveurs virtuels qui peuvent grandir.
* Le Simple Storage Service (S3) comme espace de stockage pour les objets.
* Des bases de données SQL et NoSQL comme Relational Database Service (RDS) et DynamoDB.
* Des services comme Virtual Private Cloud (VPC) et CloudFront pour créer des réseaux isolés et délivrer le contenu globallement.
* Des outils comme SageMaker pour construire, entrainer et déployer des modèles d’apprentissage automatique.
* Des outils pour les développeurs comme CodeBuild, CodeDeploy et CodePipeline.

De façon générale, ces services sont évolutifs, c’est-à-dire qu’on peut facilement augmenter ou diminuer les ressources d’AWS selon nos besoins. Le coût de ces services dépend de notre utilisation, nous amenant à dépenser moins en nous faisant payer uniquement pour ce qu’on utilise.

Dans ce document, nous allons aborder des services gratuits dont plusieurs seront utilisés pour notre site Web qu’on développera dans le laboratoire.

# AWS Lambda

## Qu’est-ce que AWS Lambda?

AWS Lambda est un service offert par Amazon Web Services qui nous permet d’exécuter nos codes sans devoir maintenir un serveur par nous-même. Ce service gère donc les serveurs, les systèmes d’exploitation, le couche réseau et le restant de l’infrastructure pour nous, nous permettant de mettre tous nos attentions sur le codage de l’application.

AWS Lambda est capable d’exécuter toutes sortes de code comme celui du code backend d’un site Web et les appels aux APIs. Ce service supporte Java, Go, PowerShell, Node.js, C#, Python et Ruby.

Contrairement à EC2, un autre service d’exécution de code dans le cloud également offert par AWS, mais qui est actif pendant tout le temps, AWS Lambda est seulement actif lors du déclenchement qu’on définit pour exécuter seulement le code nécessaire.

## Comment est-ce que AWS Lambda fonctionne?

Chacune des fonctions de Lambda est exécutée dans leur propre conteneur, qui contient tous les éléments nécessaires pour être exécuté dans n’importe quel environnement. Lorsqu’on crée une fonction, cette dernière est mise dans un nouveau conteneur qui est ensuite exécuté par les machines gérées par AWS. Avant même que les fonctions soient exécutées, le conteneur de chacune est alloué les ressources comme la capacité nécessaire de la mémoire et du processeur. Une fois que les fonctions aient fini d’être exécutées, la mémoire allouée au départ est multipliée par le temps d’exécution de ces fonctions, déterminant le frais du service au client.

## AWS Lambda vs Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud est un autre service d’AWS qui nous permet d’exécuter le code dans le cloud. Par contre, comme on a mentionné plus tôt, EC2 est actif pendant tout le temps, alors que Lambda ne l’est que quand l’utilise. Cette différence est importante lorsqu’on parle de coût et Lambda comporte moins de temps d’exécution inutile coutant de l’argent au client. De façon générale, Lambda est mieux placé pour gérer les requêtes qui, par exemple, manipulent les données. Toutefois, après un certain temps d’inactivité, le Lambda pourrait être affecté par un *Cold start*, ce qui signifie qu’il comporterait un délai. De l’autre côté, EC2 est plus efficace lorsqu’on le dédie à certaines tâches en particulier qui sont répétitives. De plus, EC2 est capable de réutiliser les ressources et les configurations du programme, ce qui le permettrait d’émuler, par exemple, un site Web existant comme stratégie pour répondre aux imprévus.

Ces deux services ont chacun leurs forces et faiblesses, les rendant meilleurs dans certaines situations. Il est donc à nous de déterminer quel utilisé.

## Les cas d’utilisation d’AWS Lambda

De façon générale, AWS Lambda est utilisé pour les applications où les tâches individuelles sont autonomes et exécutées pour une courte durée, et où il y a une grande différence entre le plus bas niveau et le plus haut niveau en termes de la charge de travail de l’application.

Comme exemple plus concret pour illustrer son cas d’utilisation, une application Web contient de nombreuses différentes tâches qui sont seulement exécutées pendant peu de temps lorsqu’on fait une requête pour une action telle que l’inscription et la connexion d’un utilisateur. Un autre exemple de cas d’utilisation serait les tâches de maintenance qui ne sont exécutée qu’à un intervalle prédéfini.

# Amazon S3 (Simple Storage Service)

## Qu’est-ce que Amazon S3?

Amazon Simple Storage Service (Amazon S3) est un service de stockage d’objet permettant aux données d’être accédées de partout via Internet avec Amazon Console et l’API de S3. C’est en quelque sorte un hébergement pour les fichiers de grande taille comme les images et les vidéos, ce qui le diffère des stockages de documents de format JSON comme MongoDB.

## Ses bénéfices

#### Flexibilité

Amazon S3 peut stocker un nombre illimité de données avec une bonne performance. Ce service est élastique et change de taille automatiquement selon notre utilisation.

#### Durabilité et disponibilité

Amazon S3 offre une durabilité exceptionnelle de 99.999999999% (11 neufs), empêchant les données de devenir corrompues avec le temps. De plus, ce service est disponible 99,99% du temps, nous enlevant l’inquiétude qu’il ne soit pas utilisable à certains moments.

## Comment est-ce que Amazon S3 fonctionne?

Les données sont stockées en tant qu’objets, ce qui permet un stockage très flexible avec sa taille dans le cloud, une caractéristique possible en répartissant les objets dans plusieurs différents disques physiques d’un centre de données et en utilisant des matériels et des logiciels spécialisés.

Les données sont automatiquement stockées dans plusieurs endroits et distribuées à travers de plusieurs disques physiques, une pratique assurant que les données sont protégées avec des *backups*. Elles sont régulièrement inspectées pour vérifier leur intégrité et sont restaurées dans le cas de corruption.

Les objets sont stockés dans des *buckets*, c’est-à-dire des conteneurs. Chaque *bucket* peut avoir un nombre illimité d’objets non structurés comme des photos, des documents ou des vidéos et un objet individuel pourrait aller jusqu’à 5TB. Par contre, dans la version gratuite (essai de 12 mois), la limite de stockage totale est de 5 GB.

## Les cas d’utilisation d’Amazon S3

L’utilité de ce service est vaste, mais de façon générale, on peut l’utiliser pour créer des *backups* grâce à la durabilité et la disponibilité de S3 qui nous permettent de restaurer nos données à tout moment lorsqu’on en a besoin. De plus, ce service peut être utilisé pour partager les gros fichiers, ce qui est un bon outil utilisé par les organisations pour distribuer leurs logiciels et leurs mises à jour à leurs clients. Amazon S3 peut également être utilisé pour simplement stocker les images et les documents,

# Amazon DynamoDB

## Qu’est-ce qu’Amazon DynamoDB?

Amazon DynamoDB est une base de données NoSQL (comme MongoDB) nécessitant aucun serveur de notre part et utilisant la méthode de clé-valeur pour stocker les données, dans laquelle la valeur est associée à la clé, qui est l’identifiant unique d’un élément. Cette base de données ne supporte pas nativement le SQL, mais plutôt le JSON.

Ce service est construit d’une certaine façon pour se pencher vers la simplicité, l’évolutivité et la performance, des aspects recherchés par les développeurs et idéals pour toute sorte d’applications utilisant des données, surtout ceux qui nécessitent un accès rapide aux données pour un grand nombre d’utilisateurs. Il est également disponible 99.999% du temps, nous assurant que notre utilisation de leur service ne serait que rompu dans de situations très rares. De plus, pour prévenir les imprévus du côté d’AWS et la corruption résultant d’un temps de stockage de long durée, les données sont stockées dans plusieurs lieus pour permettre la restauration en cas de problème.

## Comment est-ce que DynamoDB fonctionne?

Les détails sur le fonctionnement de DynamoDB ne sont pas connus par le publique. Par contre, il est possible de savoir comment on interagit avec celui-ci. Pour commencer, lorsqu’on débute notre utilisation de DynamoDB sur AWS, on ne définit pas la base de données en choisissant une prédéfinie ou en allouant un certain nombre d’espace sur le disque, mais plutôt en définissant le nombre de transactions et le nombre de kilooctets de trafique supportés par seconde. Les utilisateurs doivent spécifier le niveau de *Read capacity units* (RCUs) et celui de *Write capacity units* (WCUs).

De plus, pour communiquer avec DynamoDB à partir de leurs applications, les utilisateurs ne font pas d’appel direct à un API de DynamoDB. À la place, on ajoute un SDK d’AWS dans leurs applications, qui va gérer toutes les communications au niveau du backend avec le serveur d’AWS.

## DynamoDB vs SimpleDB

Tous les deux sont des services de base de données non-relationnelle offerts par AWS. Par contre, DynamoDB se concentre sur l’évolutivité et la performance. Il utilise alors les SSDs pour un temps de réponse plus bas et il n’y aucune limite du nombre de requêtes et la taille d’une table. De l’autre côté, un domaine de SimpleDB a une limite de stockage de 10 GB. Le nombre de requêtes est également limité à 25 écritures par seconde.

## DynamoDB vs MongoDB

MongoDB est une base de données également NoSQL que plusieurs d’entre nous ont utilisé dans certains de nos cours au cégep. Bien que les DynamoDB et MongoDB offrent des fonctionnalités similaires, DynamoDB est dépend d’AWS alors que MongoDB est indépendant de la plateforme. Cette dépendance n’est pas nécessairement mauvaise puisque dans le cas de DynamoDB, son intégration avec les autres services d’AWS est plus simple. De plus, cela rend son utilisation plus simple puisqu’AWS gère une bonne partie des infrastructures et des configurations de la base de données, tandis que dans MongoDB, qui permet un plus grand contrôle sur celle-ci, tous ces derniers sont gérés par les utilisateurs

## Les cas d’utilisation de DynamoDB

Les cas d’utilisation possibles avec DynamoDB sont nombreux. Il est formidable pour développer des applications nécessitant beaucoup de connections pour des millions d’utilisateurs et des millions de requêtes par seconde. On pourrait également l’utiliser pour les jeux vidéo en les fournissant un lieu de stockage assuré pour les données de chacun des millions de joueurs, leur historique de jeu et leurs scores.

Dans le laboratoire, nous allons utiliser DynamoDB pour simplement stocker les données des utilisateurs.

# AWS Amplify

## Qu’est-ce que AWS Amplify?

AWS Amplify est un ensemble d’outils et de services servant à bâtir et lancer les applications web ou mobiles facilement. Ce service inclut des fonctionnalités comme l’authentification, les APIs, le stockage et l’hébergement permettant d’intégrer facilement le backend à nos applications.

AWS Amplify support de nombreux Server-Side Rendering (SSR) frameworks, single-page application (SPA) frameworks et de générateurs de site statique :

* **SSR frameworks**: Next.js, Nuxt, Astro et SvelteKit avec adapteur communautaire, et n’importe quel SSR frameworks avec un adapteur personnalisé.
* **SPA frameworks**: React, Angular, Vue.js, Ionic, Ember.
* **Générateurs de site statique**: Eleventy, Gatsby, Hugo, Jekyll, VuePress.

## Comment est-ce que AWS Amplify fonctionne?

D’une certaine façon, AWS Amplify est comme une librairie de JavaScript qui nous permet de bâtir et déployer des applications sans serveur dans le cloud, combinant le code du frontend et celui du backend. Pour expliquer simplement ce service, on peut le voir comme en trois composants principaux :

* **Librairies** : Permettent de se connecter, d’intégrer et d’interagir avec les services cloud d’AWS. Elles rendent l’ajout de l’authentification, le stockage de fichiers et de données, les APIs et plusieurs autres fonctionnalités plus facile.
* **Interface utilisateur (UI)**: Il comporte des composants de l’interface utilisateur qui sont conçus autour du travail sur le cloud de notre application.
* **Interface de la ligne de commande (CLI)**: Ceci nous aide à agrandir notre application.

## AWS Amplify vs Google Firebase

Google Firebase est un outil de backend que plusieurs d’entre nous ont déjà utilisé dans le cours d’application mobile. Étant un service de Google, Firebase interagit bien avec les autres services de Google, ce qui n’est un avantage exclusif à Firebase puisqu’Amplify fonctionne confortablement avec les autres services d’AWS. Par contre, dans le cas d’Amplify seulement, les développeurs seront limités à déployer uniquement à l’infrastructure d’AWS, ce qui n’est pas un problème avec Firebase.

En termes de fonctionnalités, les deux services sont similaires. Le choix reste donc sur quel écosystème entre Google et AWS on choisit d’utiliser.

## Les cas d’utilisation d’Amplify

Puisqu’Amplify nous permet de bâtir et déployer des applications sans serveur dans le cloud, combinant le code du frontend et celui du backend, on pourrait créer une application web sans serveur utilisant l’authentification, les APIs, et le stockage offerts par Amplify. Cela nous permettrait de se connecter, de télécharger des photos et d’afficher les images.

De plus, on pourrait développer une application mobile utilisant les données en temps réel avec AWS AppSync. Cela nous permettrait de, par exemple, créer une application mobile de messagerie dans laquelle les utilisateurs pourraient envoyer et recevoir des messages en temps réel.

# Bibliographie

1. AMAZON WEB SERVICES, « Qu'est-ce qu'AWS? », *Amazon Web Services*, [en ligne], <https://aws.amazon.com/fr/what-is-aws/> , consulté le 10 décembre 2024.
2. SERVERLESS, « AWS Lambda The Ultimate Guide», Serverless, [en ligne], <https://www.serverless.com/aws-lambda> , consulté le 10 décembre 2024.
3. AMAZON WEB SERVICES, «FAQ sur AWS Lambda », *Amazon Web Services*, [en ligne], <https://aws.amazon.com/fr/lambda/faqs/#:~:text=AWS%20Lambda%20natively%20supports%20Java,C%23%2C%20Go%2C%20and%20PowerShell> , consulté le 10 décembre 2024.
4. MIKAC, Erik, «AWS Lambda vs EC2 : Which to Use and When », *CBTnuggets*, [en ligne], <https://www.cbtnuggets.com/blog/certifications/cloud/aws-lambda-vs-ec2-which-to-use-and-when> , consulté le 10 décembre 2024.
5. MIKAC, Erik, «AWS Lambda vs EC2 : Which to Use and When », *CBTnuggets*, [en ligne], <https://www.cbtnuggets.com/blog/certifications/cloud/aws-lambda-vs-ec2-which-to-use-and-when> , consulté le 10 décembre 2024.
6. AMAZON WEB SERVICES, « Amazon S3 », *Amazon Web Services*, [en ligne], <https://aws.amazon.com/s3/?nc1=h_ls> , consulté le 10 décembre 2024.
7. GOLAS, Henry, « What is S3 storage? », *Cloudian*, [en ligne], <https://cloudian.com/blog/s3-storage-behind-the-scenes/> , consulté le 11 décembre 2024.
8. AMAZON WEB SERVICES, « Amazon DynamoDB », *Amazon Web Services*, [en ligne], <https://aws.amazon.com/dynamodb/> , consulté le 11 décembre 2024.
9. SCYLLADB, « Introduction to DynamoDB », *ScyllaDB*, [en ligne], <https://www.scylladb.com/learn/dynamodb/introduction-to-dynamodb/>, consulté le 11 décembre 2024.
10. AMAZON WEB SERVICES, « Amazon SimpleDB FAQs », *Amazon Web Services*, [en ligne], <https://aws.amazon.com/simpledb/faqs/>, consulté le 12 décembre 2024.
11. WICKRAMASINGHE, Shanika, « DynamoDB vs MongoDB : Comparing NoSQL Databases », *bmc blogs*, [en ligne], <https://www.bmc.com/blogs/mongodb-vs-dynamodb/>, consulté le 12 décembre 2024.
12. JAVATPOINT, « What is AWS Amplify », *Javatpoint*, [en ligne], <https://www.javatpoint.com/what-is-aws-amplify>, consulté le 12 décembre 2024.
13. SERVIFYSPHERESOLUTIONS, « What is AWS Amplify », *Medium*, [en ligne], <https://medium.com/@servifyspheresolutions/what-is-aws-amplify-3d15a10fbc4a>, consulté le 12 décembre 2024.
14. AMAZON WEB SERVICES, « Welcome to AWS Amplify Hosting », *Amazon Web Services*, [en ligne], <https://docs.aws.amazon.com/amplify/latest/userguide/welcome.html>, consulté le 13 décembre 2024.
15. EROLIN, Justice, « Amplify vs. Firebase : Which Is Best Suited for Your Project? », *BairesDevBlog*, [en ligne], <https://www.bairesdev.com/blog/amplify-vs-firebase-which-one-is-best/>, consulté le 13 décembre 2024.
16. DUMINIL, Nicolas, « AWS - Gérez votre infrastructure sur la plateforme cloud d’Amazon », s. l., Éditions ENI, 2019, 280 p.
17. Tobias Schmidt, X, [en ligne], <https://x.com/tpschmidt_>
18. AWS Developers, X, [en ligne], <https://x.com/awsdevelopers>
19. Amazon Web Services, X, [en ligne], <https://x.com/awscloud>

# Veille technologique

## Moteurs de recherches spécialisés

1. Google News sur le sujet d’AWS (Actualité)

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

1. Google Scholar sur AWS (Articles scientifiques)

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

1. Google Books (Livres)

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

## Google Alerts

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

## Flux RSS

1. RSS Feed Reader (Annonces dans aws.amazon.com/new/, Wired et TechRadar)

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

## Agent de surveillance (Infominder)

* Techcrunch se portant sur Amazon : <https://techcrunch.com/tag/amazon/>
* CNBC se portant sur la technology : <https://www.cnbc.com/tech-guide/>
* AboutAmazon se portant sur AWS : <https://www.aboutamazon.com/amazon-aws-news>
* SiliconeAngle se portant sur le cloud : <https://siliconangle.com/category/cloud/>
* DataCenterDynamics se portant sur le cloud : <https://www.datacenterdynamics.com/en/the-cloud-and-hybrid-channel/>

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

## Média social (Twitter)

* Tobias Schmidt
* AWS Developers
* Amazon Web Services

Une image contenant texte, capture d’écran, logiciel, Site web

Description générée automatiquement