**AWS**

**Projet**

L’idée est de charger une grosse base de données (choisir mongodb, cassandra, neo4j) et derrière de faire des analyses sur ces données via AWS.

Ne pas oublier d’éteindre AWS.

Projet en groupe (même que projet fil rouge).

1. **What is AWS ?**

**Définition**

Ensemble de service cloud proposé par Amazon et chaque catégorie de services est spécialisée dans un domaine particulier :

**Infrastructure as a service (IaaS)** : ensemble qui permet de créer des machines virtuelles, service de stockage de données (on peut choisir la zone géographique dans laquelle on veut stocker nos données, pratique quand on est contraint juridiquement)

**Plateforme as a service (PaaS)**: surcouche sur l’IaaS, on a déjà le matériel alloué mais on a aussi l’OS qui est fournit avec. On s’occupe juste de déployer les applications sur le serveur.

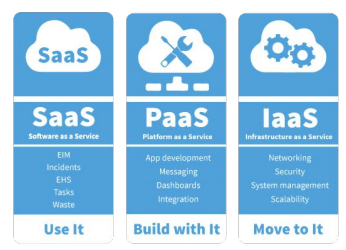
**Software as a service (SaaS)**: il s’agit juste des logiciels dans le cloud qu’on utilise, sans se soucier de l’infra derrière et des OS sur lesquels ils sont exécutés.

*Ex : boite mail*

**Service de cloud storage**: stockage de données brutes (n’importe quel format)

**Global cloud platform**

On peut jouer avec les trois briques principales pour migrer une infrastructure sur le cloud.



*Quelques exemples :*

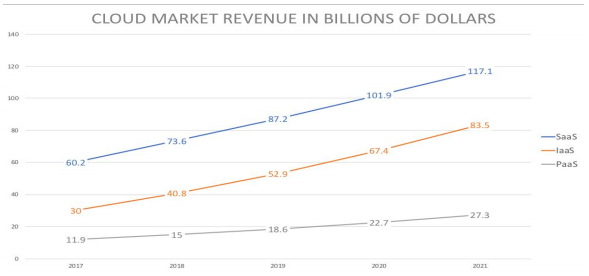
* *Netflix est full AWS* 🡪 *stocke tous les films sur AWS S3, qui permet de stocker des fichiers bruts sur le cloud, les films sont stocker sur les buckets S3, dupliqués dans plusieurs zones géographiques. Derrière il y a d’autres services qui permettent de streamer le contenu vers le lecteur vidéo, qui est lui exécuté sur le serveur. Au moment où il y a des pics d’utilisation, d’autres instances sont créé automatiquement et ça s’autoscale pour supporter la charge.*
* *Airbnb fait la même chose* 🡪 *tous leurs serveurs sont sur AWS.*
* *Spotify*
* *Pinterest* 🡪 *stocke les images et rafraichit quotidiennement leur index pour permettre une recherche très rapide ou quasiment instantanée.*

**Avantages**

On ne perd pas beaucoup de temps avec la maintenance de l’infrastructure, les mises à jour des OS etc., et on peut s’occuper plutôt de développer de nouvelles fonctionnalités.

**Quelques chiffres**

En 2019, 90 milliards de dollars pour le SaaS et en 2021 arrive à 120 milliards de dollars.



AWS ont presque 42% des parts de marchés sur les services infrastructures (IaaS).

Sur tous les services confondus, AWS a 62% tandis que Azure et Google ont 20% et 12% respectivement.

|  |  |
| --- | --- |
| Macintosh HD:Users:camillecochener:Desktop:Capture d’écran 2019-12-12 à 12.04.34.png | Macintosh HD:Users:camillecochener:Desktop:Capture d’écran 2019-12-12 à 12.04.44.png |

1. **Service overview**

**Services de calcul**

* **EC2**

Service qui permet d’allouer une machine virtuelle sur le cloud, la rendre accessible au public et dessus, on peut créer l’OS que l’on veut et faire ce que l’on veut avec cette machine. On ne peut pas faire de gaming dessus car il n’y a pas d’interface graphique (commande shell).

* **Lightsail**

Comme un EC2 léger, pratique pour les développeurs qui veut créer un site internet. Il y a un WordPress intégré.

* **AWS batch**

Permet de programmer un ensemble de tâche régulièrement. Code qui est exécuté régulièrement. On peut mettre des ETL dedans.

*Ex : chez les furets, il y a des ETL qui tournent après minuit qui permettent de restructurer des données et de les stocker dans une autre base de données qui sert à générer des rapports*.

* **Amazon lambda**

Autre service qui permet de générer du code via des trigger.

*Ex : site facebook, upload d’une nouvelle photo de profil, Amazon Lambda va la compresser et la stocker ailleurs.*

* **Elastic load balancer (ELB)**

Service qui se met devant plusieurs instances EC2. Ces instances contiennent plusieurs instances du même site et ELB va distribuer ces instances.

**Services de stockage**

* Amazon S3 (voir TP/Projet)
* **Amazon EFS**: service qui permet d’étendre temporairement le stockage d’une instance EC2.
* **Snowmobile**: ils ont un camion qui va d’un datacenter à un autre. Snowball edge est une version légère.

**Services de base de données**

* **Amazon RDS**(relational database service) : permet de créer des BDD et de choisir notre moteur de BDD (MySQL, MariaDB, OracleDB, SQLserver)
* **Amazon DynamoDB** : service de BDD NoSQL propriétaire
* **ElastiCache** : BDD stockée en mémoire, très pratique à utiliser, populaire auprès des développeurs de jeux vidéo (accès en temps réel). Moins de latence.
* **Amazon Neptune**: BDD propriétaire. Elle est orientée graphe. Permet de créer des nœuds et des relations.

**Services de migration**

* **Database migration service**: comme un ETL, permet de choisir une BDD input et output et au milieu on structure nos données. (utile pour le projet)

**Networking & content delivery**

* **VPC**: crée un sous réseau dans le cloud et on peut décider derrière si c’est accessible au public ou pas.
* **Route53**: acheminer les requêtes des clients vers les instances EC2.

**Machine learning**

* **SageMaker**: plateforme entièrement gérée qui permet de créer, de former et de déployer des modèles de ML sur le cloud.
* **Amazon Lex**: service de reconnaissance vocal et textuel (service utilisé par les enceintes Alexa).
* **Amazon rekognition**: service de reconnaissance d’objets.

**Analytics**

* **Amazon Quicksight**: on peut le connecter à une BDD quelconque et derrière afficher les données qu’on veut, faire des dashboards etc. permet de prendre des décisions dans les entreprises.

**Security, Identity & compliance**

* **AWS IAM** : service qui permet de créer des groupes d’utilisateurs et leur accorder des droits spécifiques. Service très utile pour créer les utilisateurs et les bons droits et contrôler l’accès au BDD, bucket S3.

**AWS Cost management**

Très utile pour gérer le budget.

Attention, il ne se met pas à jour en temps réel.

Il y a d’autres catégories :

* **Developer Tools**: versioning de code comme Github, Gitlab, IDE sur le cloud
* **Management Tools**: contient des services de management du cloud, historique de ce qui a été fait sur le compte, contient un service cloud formation qui permet de capturer ou de transformer l’infrastructure qu’on vient de créer (instances EC2) et connecter par exemple les instances EC2 et un bucket S3. Il va contenir un script qui pourra être rechargé plus tard. Pour le projet, on va tester différentes infrastructures (2 nœuds, 3 nœuds…) et voir ce qui nous arrange le mieux niveau performance et budget. Si on fait la même manip plusieurs fois, cela va prendre du temps.
* **Media Services**: services qui permettent de traiter du son, des images, des vidéos.
* **Mobile Services**: permettent de tester les applications mobiles en cours de développement.
* **IoT**: objets connectés
* **Software**: marketplace où on peut trouver des services personnalisés, créés par des entreprises ou des particuliers.

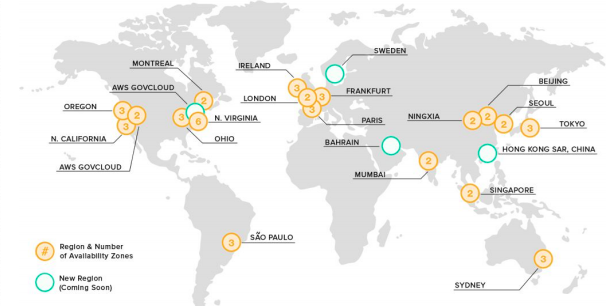
1. **Combien ça coute ?**

**Services de calcul**: on paye à l’utilisation 🡪 éteindre dès qu’on a fini

**Services de stockage**: on paye par giga

**Data transfer**: payant uniquement à la lecture (par giga). Quand on met dans les buckets S3, gratuit, mais les lectures sont payantes.

1. **Quel volume ?**

****

1. **Services principaux (utilisés pour le projet)**

* **AWS management console**
* **S3**

On commence par créer un bucket, on a l’url qui nous permet d’y accéder

* **EC2**

Avant de créer des instances EC2, il faut créer des paires de clé (prive, publique) qui permettent de s’authentifier quand on veut se connecter à une instance EC2 depuis notre terminal. Il faut conserver la paire de clé.

* **IAM**

Ici, on va créer les droits d’admin. On crée un user qu’on va affecter à un groupe d’utilisateur.

Credentials : qu’on va spécifier dans la config du client AWS (fichier credentials.csv).

* **EMR**

Service qui va nous permettre de créer un cluster (plusieurs machines EC2). On choisit l’image qu’on va installer dessus. Les machines sont accessibles depuis le terminal du client AWS.

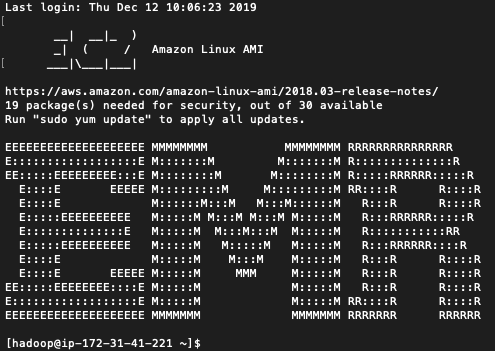
On va utiliser une image Spark et un notebook Zeppelin pendant le TP.

Il faut ouvrir le port 22 en connexion SSH.

1. **Notes TP/Projet :**

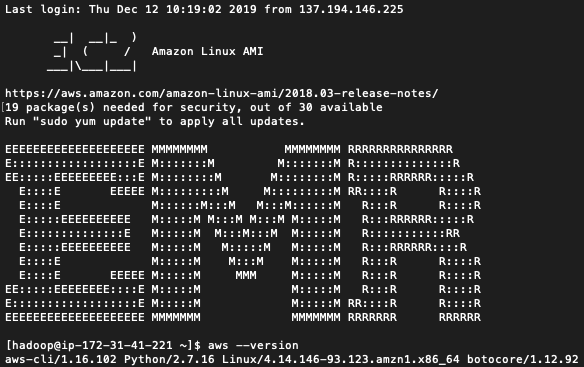
***Etape 6 : Enable Web Connection***

Macintosh HD:Users:camillecochener:Desktop:Capture d’écran 2019-12-12 à 11.46.43.png

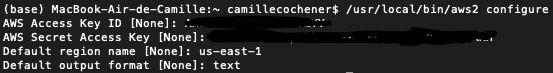


avec la redirection des ports :

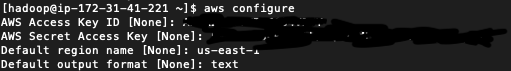




Configuration du client AWS en local :



Configuration du client AWS sur le serveur :



Macintosh HD:Users:camillecochener:Desktop:Capture d’écran 2019-12-12 à 11.49.17.png

Ouverture de Zeppelin : <http://localhost:8890/>