29 JUIN 2019

SHEET

AWS SOLUTION ARCHITECTE ASSOCIATE

https://www.linkedin.com/in/hatim-ajdaini/

AJDAINI, HATIM

Sommaire

Mots-clés à connaitre

Dictionnaire

- **Availability** : Disponibilité
- **Durability** : Durabilité
- **Resiliency** : Élasticité
- **Reliability** : Fiabilité
- **Consistency** : Cohérence des données
- **Tenancy** : la manière dont sont placées les instances
- **HA (Hight Availability)** : Petit panne tolérée
- **Fault tolerent :** Aucune panne n'est tolérée
- **Service managé** : AWS s'occupe de tous (scalabilité, haute dispo, etc …)
- **Scaling vertical :** Possibilité d'augmenter les ressources d'une machine (ex : CPU, espace disque) sans la redémarrer.
- **Firewall Statefull** : si on crée une règle entrante (inbound) il n'y a pas besoin de créer une règle sortante (outbound), car elle est automatiquement appliquée dans le outbound
- **Firewall stateless** : Une règle inbound n'est pas appliquée automatiquement en outbound, il faut l'appliquer manuellement
- **Stockage Objet (S3) :** on ne peut pas modifier juste une partie d'un fichier donc si modification alors tout l'objet est remplacé
- **Stockage Bloc (EBS) :** permet de modifier uniquement les parties modifiées

Géographie

- **Datacenter :** plus petite unité AWS
- **AZ (Availability Zone) :** ensemble de datacenter dans la même zone
- **Région :** plusieurs AZ
- **Edge location :** POP (point of presence) pour la mise en cache (CloudFront) et DNS (Route 53)
- **Regional Edge :** décharge les EL lorsque ressources trop volumineuses, c'est un cache L2 utilisé par EL.

S3

Les classes

Classe	Quand le choisir ?
S3 Standard	• Téléchargement rapide • Données fréquemment consultées • Appels API moins cher • Réplication sur 3 AZ
Amazon S3 Intelligent-Tiering	• AWS choisi automatiquement la meilleure classe pour nous
S3 IA	• Téléchargement rapide • Données consultées peu fréquemment • La requête est + cher que S3 standard • Réplication sur 3 AZ
S3 One Zone-IA	• Même chose que S3 IA mais redondance sur 1 AZ
S3 Glacier	• Téléchargement très lent (3 à 5h) • La requête est + cher que S3 standard • Réplication sur 3 AZ
S3 Glacier Deep Archive	• Le moins cher mais le + lent que S3 Glacier

- 11*9 (99,999999999%) de **durabilité** et 99,99% de **disponibilité** (sauf 99,5% pour One Zone-IA)
- Récupération plus rapide d'un objet dans S3 Glacier (+ c'est rapide + c'est cher) :
  - **Expedited retrieval** : 1 -> 5 mins
  - **Standard retrieval** : 3 -> 5h
  - **Bulk retrival** : 5 -> 12h

Cohérence des données

- Read after Write pour le PUTS
- Attente de quelques ms si écrasement d'objet (PUTS and DELETES)

Chiffrement

- Données en Transit :
  - SSL/TLS par défaut
- Chiffrage (Encryptions) in REST :
  - **Keys-SSE-S3** : AWS gère pour nous la clé de chiffrage
  - **Keys-SSE-KMS** : Client + AWS qui gère la clé
  - **Keys-SSE-C** : le client fournit à AWS la clé de chiffrement
  - **Client Side Encryption** : chiffré sur le pc du client puis ensuite renvoyé à S3

Paiement

- Paiement au mois selon la taille de stockage + appels API (PUT, GET, LIST, gratuit pour DELETE)

Sécurité

- Contrôle d'accès :
    - **Policies** : permission attachée à un bucket et sont très granulaires
    - **ACL (Access Control Lists)** : Limite l'accès aux objets aux user/group AWS.

Services en +

- Possible de déployer des **sites web statique** (serverless)
- **URL signée** : url d'autorisation d'accès pour un ou plusieurs utilisateurs de confiance
- **Cross Version Replication :** Réplication des données sur une autre région
- **Amazon S3 Transfer Acceleration** : un CloudFront automatique, facturation que si accélération
- **Lifecycle policies** : durée pour déplacer les objets vers une autre classe de stockage et de pouvoir ensuite la supprimer
- **Versioning** : git de S3, l'objet reste présent même si suppression
- Le Versioning et le Lifecycle policies sont complémentaires
- Si versioning activé avec le Cross Version Replication :
    - Les DELETES ne sont pas répliqués :
    - Les objets déjà existant ne sont pas répliqués, il faut le faire depuis le cli

Les limites

- Aucune limite de stockage ou d'objets
- **Key design** = **key hashing** = **prefix** = 3500 PUT et 5500 GET /s/prefix
    - + de préfixes = + de performance
    - + de performance si le nom du préfix est random
- 100 buckets par compte AWS
- On peut attacher jusqu'à 20 policies
- Taille de l'objet :
    - 5TB max
    - 0 byte min
- **Multipart upload** :
    - Possible si objet >= 5MB
    - Recommandé si objet >= 100MB
    - Obligatoire si objet >= de 5G

Solution Cloud hybride

- **AWS Storage Gateway :** Service de stockage hybride (ce n'est pas une liaison directe privée)
    - Les types qu'on peut stocker :
        - **File Gateway** (pour les fichiers)
        - **Volume Gateway** (pour les VM)

        - **Tape Gateway** : bande magnétique (ancien modèle)
    - Type de stockage
        - **Stored volume** : Réplication des données sur votre S3
        - **Cached volme (moins cher)** : garde seulement les données fréquemment utilisées en on premise

Autres infos

- Stockage de type objet
- Service managé
- Le nom du bucket doit être unique car il forme l'url d'accès au buket
- Objets privés par défaut (possible de les rendre public de laisser le bucket en privé)
- Possible d'utiliser le MFA pour la suppression intentionnelle
- Code HTTP 200 si upload OK
- Si on n'arrive pas à détecter une image c'est un **problème CORS** (solution : créer une config CORS)

EC2

Model d'achat :

Type d'achat	Description
On-Demand	• Paiement à la second Linux • Paiement à l'heure Windows
Spot Instances	• Le prix suit les flux du marché (comme à la bourse) • Il nous laisse 120 secondes avant la suppression de l'instance
Reserved Instances	• Paiement sur l'année (coute moins cher que le On-Demand)
Schedule Reserved Instances	• Reserved instance sur une période choisie (ex : tous les samedis entre 00h à 6h)
Dedicated hosts	• Rack réservé (on sait quel est l'host) • On le choisit pour être compliant avec les licences
Dedicated instances	• Instances lancées dans le même hardware isolées des autres clients

- On ne paie que les EC2 à l'état running
- L'**Etat hibernate** permet de freezer notre machine (On ne paie pas), cependant notre EBS doit être chiffré.

Stockage

- 2 types de stockage disponible dans une AMI :
    - **Instance Storage :**
        - Vient avec une instance téléchargée depuis une AMI
        - Données effacées si l'instance est arrêtée (sauf si que reboot)
    - **EBS :**
        - Les données sont persistantes
        - Ne peut être monté que sur une seule instance
        - Ne peuvent être attachés à une instance que si l'instance est sur le même AZ.
- Pour un volume partagé entre différentes instances EC2 :
    - **EFS :** permet un partage entre différentes AZ, régions, et même entre comptes
    - **FSX** (pour NTFS windows) : permet un partage uniquement entre différentes AZ

Limites

- 20 EC2 max / region
- 5 **ENI** par EC2

Pour information ENI (**Elastic network interface)**, c'est une carte réseau virtuelle qu'on associe à une EC2

Classe des EBS

	Volume SSD Provisioned IOPS	Volume SSD General Purpose	VolumeHDD Throughput Optimized	Volume Cold HDD
Taille mini	1	4	500 Gio	500 Gio
Débit max IO/s	64000	16000	500	250
Débit max	1000 Moi/s	250 Moi/s	500 Mo/s	500 Mo/s
Se mesure en	IOPS	IOPS	MB/s	MB/s
Quand le choisir ?	Beaucoup de lecture écriture	Utilisation normal (non intensive)	Bande passante élevée et Optimalisation des couts	Le moins cher et données très peu consultées

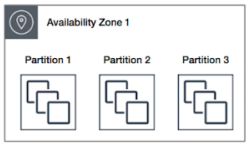
Snapshot

- On ne peut pas supprimer un snapshot déjà utilisé en tant que root EBS
- Les snapshots EBS sont stockés sur S3, et sont stockés incrémentalement
- On peut créer un snapshot d'une instance sans la stopper mais ce n'est pas recommandé, car on risque de perdre des données
- Il est possible de créer une AMI depuis un snapshot ou un volume
- On peut partager un snapshot mais seulement s'il est déchiffré
- Les snapshots sont chiffrés automatiquement si le volume l'est aussi

Chiffrement

- Par défaut le root volume n'est pas chiffré seul les volumes rajoutés le sont.
- Comment chiffrer un volume root ?
    1. Créer un snapshot depuis le volume non chiffré
    2. Créer une copie de ce snapshot et sélectionner l'option chiffrement
    3. Créer une AMI depuis la copie de ce snapshot
    4. Lancer une instance depuis cette AMI

Cluster

	Clustered Placement group	Partition Placement Groups	Spread Placement group
Description	Même rack et même hardware	Même rack mais différents hardware	Différents racks et différents hardwares
Quand l'utiliser ?	• Traffic réseau élevé • Très peu de latence • Puissance de calcul ++	Réduire la probabilité de défaillances de matériel corrélé pour une application	Applications ayant un petit nombre d'instances critiques, qui doivent être séparées les unes des autres
Schéma			

On ne peut pas bouger une instance qui est déjà dans une placement group, il faut créer une AMI depuis cette instance.

Autres infos

- Moteur **Hyperviseur type 1**, plutôt que type 2 (VirtualBox)
  - Nitro (KVM), AWS vise 100% nitro
  - Xen
- On crée une instance à partir d'une **AMI** (Amazon Machine Image)
- Possible d'importer une VM (ex : Vmware) dans EC2, il suffit d'exporter la VM dans S3 et la convertir en AMI
- **user-data** peut être sous la forme de :
  - Script bash (que EC2 va exécuter au démarrage)
  - Fichier texte (pour un partage d'infos)
- Par défaut le root volume est supprimé (pour éviter cela mettre **DeleteOnTermination** à False)
- On ne peut pas partager un volume situé dans une AZ dans une autre AZ, en revanche on peut faire un export dans S3 et le réimporter dans l'autre AZ
- **VM Import/Export** permet d'importer des VM (ex : VMWare) sur EC2 et de les réexporter en on primise.
- Le Scaling vertical est possible
- On ne peut pas modifier une Reserved instance d'une région à l'autre mais en AZ oui
- Récupérer les data (user-data, public IP, private IP, etc …) depuis une url
  - http://169.254.169.254/latest/meta-data/public-hostname
  - http://169.254.169.254/latest/meta-data/public-ipv4
  - http://169.254.169.254/latest/meta-data/(xxx)
  - http://169.254.169.254/latest/user-data/

Database

RDS (BDD Relationnelle)

- 2 types :
  - **OLTP ➔ RDS** : dédiée aux données transactionnelles
  - **OLAP ➔ Amazon RedShift :** dédiée à l'analyse de données
- Pas 100% managé mais le reste beaucoup
- Rétention **35 jours** max (1 défaut) donc par défaut le backup est activé
- Tourne sur une instance EC2
- Utiliser **KMS** pour chiffrer les données
- Parmi RDS, il existe **Aurora** :
  - Bdd OLTP d'AWS
  - Plus puissant que les autres bdd OLTP
  - Le seul à être **serverless** (scaling automatique on définit juste les **ACUs aurora capacity unit**)
  - Donne beaucoup plus de possibilités que les autres bdd OLTP
  - Aurora peut permettre de lancer du Postgresql et Mysql

Les services supplémentaires de Redshift

- **Redshift Enhanced VPC Routing** : permet à une ressource AWS d'accéder directement à Redshift depuis un VPC, sans passer par internet
- **Redshift Cross region replication :** Réplication des données Redshift dans une autre région.

BDD non relationnelle (NoSQL)

- Nom du service ➔ **DynamoDB**
- Format JSON (format clé valeur)
- Autoscalé par défaut
- Parfait pour stocker uniquement des **métadonnées** mais aussi des **sessions**
- **DynamoDB Streams** permet de s'associer à une fonction Lambda et de l'exécuter immédiatement après la modification d'un élément de la table
- Le code d'erreur 400 de DynamoDB signifie :
  - **ThrottlingException**
  - **LimitExceededException**

ElasticCache (mise en cache)

Gain de performance en mettant en cache les requêtes fréquemment utilisées et possible de l'utiliser comme une base de données indépendante :

- **Memcached :** Très simple à mettre en place et permet de faire du scaling horizontal
- **Redis :** Plus riche en fonctionnalité comme le Multi AZ, ou la création de backups

High Availability, Fault Tolerance et Puissance

- **Multi-AZ** ➔ pour la **HA :** failover automatique (réplication **synchrone**)
- **Read Replica** ➔ pour la **Puissance :** augmente la capacité de lecture (réplication **asynchrone**)
- **Multi-AZ + Read Replica** = **Fault Tolerance:** faire un Read Replica avec du MultiAZ pour que notre BDD soit répliquée sur une autre région avec le failover

Types de scaling

- **scaling horizontal** : avec read replicas (15 max)
- **scaling vertical** : changement type/taille instance
- **aurora servless**
- **shunking/sharding**: partitionnement de sa base (amélioration de la performance)

Autres infos

- SDD Storage (on choisit le nombre d'IOPS)
- Répartition sur 3 AZ
- AWS effectue un snapshot du volume de stockage de l'instance de la bdd **une fois par jour**
- **AWS Database Migration** (DMS) peut être utilisé pour migrer une base de données vers une BDD open source sur une RDS ou une EC2.
- **Neptune** : bdd orientée graph
- Pour câbler redshift avec un endpoint, il faut activer **Redshift Enhanced VPC Routing**
- Aurora stocke **6 copies** de mes données RDS par défaut

VPC (Amazon Virtual Private Cloud)

Les limites

- 5 VPC par région par compte
- 200 Subnets par VPC
- Elastic IP 5 par région
- 1 Internet Gateway par VPC

Communication avec internet

- Par défaut un VPC ne communique pas avec internet
- Pour un **Public Subnet**, il faut :
    - Créer un **Internet Gateway** (IGW)
    - Dans la table de routage du Public Subnet :
        - Destination : 0.0.0.0/0
        - Cible : IGW
- Pour un **Private subnet** (ex d'utilité : télécharger des maj sur internet), il faut :
    - Un public subnet ou on peut retrouver (à placer dans un Public Subnet) :
        - **Nat Gateway** : entièrement managé, automatiquement scalable
        - **Nat Instance** : c'est une EC2 qu'on peut personnaliser
        - **Bastion Host** : instance EC2 pour une connexion privée public en ssh et rdp

VPC et on premise

- Pour créer une communication entre notre Datacenter et notre VPC on peut :
    - Passer directement par internet (moins de sécurité)
    - Ligne directement connecté, DX (**AWS Direct Connect**)
- Peu importe la méthode il faut au moins deux composants :
    - VPG (**Virtual Private Gateway**) : attaché au VPC où on rajoute l'IP du datacenter on premise
    - **Customer Gateway** : attaché au datacenter on premise
- BGP (**Border Gateway Protocol**) **:** permet de connecter un VPG pour plusieurs Customer Gateway

Attention : Les données ne sont pas chiffrées par défaut dans un DX à l'inverse d'un VPN qui chiffre généralement les données en Transit mais les données passent tout de même par internet

Communication privé (ne passe pas par internet)

- Pour faire communiquer 2 VPC de manière privée, il faut créer une **VPC Peering**
- Le **VPC peering** n'est pas transitive
    - Ex : prod <-> test <-> PREPROD
    - Il faudra rajouter un lien prod <->PREPROD si on veut les faire tous communiquer
- Pour la rendre transitive, on peut créer une **Transit Gateway**
- **VPC endpoint** garantie que le trafic ne sort pas sur Internet, il existe deux types :
    - **VPC Gateway Endpoint** : utilisé principalement pour l'accès S3 et DynamoDB
    - **VPC Interface Endpoint** : utilisé pour le reste des services

Autres infos

- Fonctionne avec le **CIDR** (Classless Inter Domain Routing)
- Supporte de l'IPV6 ou l'IPV4
- Se déploie dans une région donc c'est une isolation par Région
- 1 Subnet = 1 AZ
- 5 IPs sont réservés par AWS pour chaque subnet
- Par défaut ne sont créés que :
    - **Route table**
    - **NACL (Network Access Control List)**
    - **Security Group**
- Pour NAT en IPV6 il faut utiliser EGRESS car le NAT classique ne fonctionne pas
- **Isolation au niveau du réseau** (pour les accès utilisateurs on va utiliser le service IAM)
- Une adresse IP privée et publique est assignée automatiquement à une instance. Si on stop/termine une instance alors on perd notre adresse IP publique à l'instar d'une nouvelle (sauf si reboot), pour éviter ce problème on utilise le service **Elastic IP**, qu'on assigne soit à une EC2 ou une ENI

Sécurité (Firewall)

Security Group

- **Firewall Statefull au niveau des instances EC2** (layer 7)
- Un Security Group peut autoriser le traffic provenant d'un autre Security Group
- Permet de bloquer ders ports mais non une IP (voir NACL pour faire ça)
- **Par défaut :**
  - Bloque tout le trafic entrant (donc on ne peut que ALLOW et non DENY)
  - Autorise tout le trafic sortant

NACL (Network Access Control Lists)

- **Firewall StateLess au niveau des Subnets** (layer 4)
- Permet de bloquer les adresses IP
- 1 Subnet par NACL cependant un NACL peut être associé à plusieurs subnet
- 1ere règle qui match sera la première règle qui sera appliquée
- **Par défaut :**
  - Autorise tout le trafic entrant
  - Autorise tout le trafic sortant

De façon générale, privilégiez Security Group à NACL car NACL est stateless, autorisation trop large et la règle est activée automatiquement pour chaque nouvelle instance créée dans le Subnet.

ELB (Elastic Load Balancing)

Lexique

Pour équilibrer la charge un ELB permet de faire du :

- **InService :** Rajout d'instances
- **OutofService :** Suppression d'instances

On peut équilibrer la charge selon un :

  - **Time-based elasticity :** planification (on connait déjà le temps)
  - **Volume-based elasticity :** en fonction de la charge

Les types

- Les classes ELB :
  - **Classic Load Balancers :** load balancer web classique non intelligent (Layer 7)
  - **Application Load Balancers : :** load balancer web intelligent (Layer 7) permet la redirection selon url source et prend en compte le multi port
  - **Network Load Balancers :** pour le Traffic TCP (Layer 4)
- 2 types de ELB :

- **External ou Internet Facing** : load balancer avec une IP publique
- **Internal facing** : laod balancer avec une IP privée

Autres informations

- Idéal pour la HA
- Il faut un **Auto Scaling Group** pour qu'il fonctionne
- Supporte-le :
  - **stickiness** : permet de garder les sessions permanentes sur la même instance EC2)
  - **X-forwarded-for :** permet de récupérer la vrai source IP sur notre instance et non celle du ELB
- **Cross Zone Load Balancing** : faire de la répartition de charge dans différentes AZ.
- **Path patters** : permet de diriger le trafic vers différentes instances EC2 en fonction de l'URL contenue dans la demande
- Erreur **504** quand il y a une erreur
- Possède toujours un nom de domaine et jamais une IP
- Distribue le trafic entre différentes AZ mais NON sur les différentes régions (il faut utiliser le service Route53 pour ce service)
- Doit être lancé sur au moins 2 Subnets différents

Group Auto Scaling

- Il est possible de combiner différents types de modèles d'achat EC2
- 20 instances maximum par région
- Scale in et out très rapidement que faire ?? il faut surveiller sur le timing du lancement de l'alarme
- **Cooldown timers** permet d'indiquer une estimation de temps que prendra notre machine pour démarrer.
- **Lifecycle Hooks :** temporiser la création ou la suppression d'une instance (ex : attendre 5 mins avant de la supprimer) avant de recevoir des requêtes.

Route53

- **Simple routing policy :** sélectionne par ordre
- **Failover routing policy :** sélectionne par health
- **Geolocation routing policy :** Rediriger vers une géolocalisation
- **Geoproximity routing policy :** choisir toujours la géolocalisation la plus proche
- **Latency routing policy :** choisir selon le TTL
- **Multivalue answer routing policy :** sélectionnèrent de manière aléatoire.
- **Weighted routing policy :** par % de charge (ex : 20% sur la 1ere route, puis 30% sur la 2eme route, puis 50% sur la 3eme route)

Il existe une limite par défaut de 50 noms de domaine

IAM

Les entités

- **Users** : les utilisateurs finaux tels que les personnes, les employés d'une organisation
- **Groups** : une collection d'utilisateurs. Chaque utilisateur du groupe héritera des autorisations du groupe.
- **Policies** : Fichier JSON qui donne des autorisations sur ce qu'un user/group/role. Il peut être soit un :
    - **Identity-Based policiy :** attaché au user/group/role
    - **Ressource-Based policy :** attaché à certaines ressources (S3, SQL)
- **Roles** : à ajouter à une ressource AWS (ex : une EC2 qui a accès à S3 sans besoin d'authentification)

Authentification

- 2 types d'authentification :
    - Login + mdp : pour la console
    - **Key ID et Secret Access Key** : pour la cli et SDK

Autres informations

- Pour la communication inter-service AWS qui demande des authentifications, il faut toujours privilégier les rôles car il n'enregistre pas nos credentials.
- Pas de système de région, les IAM sont universel
- Par défaut un utilisateur crée n'a aucun droit (il n'y a pas de permission par défaut)
- C'est toujours le **DENY** policy qui l'emporte
- **Cross Account Iam role** : accorde aux clients l'accès à des ressources AWS de leur compte à un tiers (ex : partenaire APN)
- **Amazon Cognito** : prend en charge la connexion avec les fournisseurs d'identité sociale tels que Facebook, Google et Amazon, et les fournisseurs d'identité d'entreprise via SAML 2.0.
- **MA (Multrifactor authentication)** est non activé par défaut, et il est attaché aussi au compte root

- Possible de créer ses propres règles de mot de passe.

Analyse et Monitoring

CloudWatch

- Monitoring pour de la **PERFROMANCE**
- Permet d'avoir plus d'élasticité
- Possible de monitorer les :
    - Metrics : les metrics différent selon les services monitorés, **en EC2 il n'est pas possible de monitorer la mémoire** (il faut utiliser un custom **metric** pour ça)
    - Logs : hébergés dans S3 et ensuite on peut créer des Dashboards, des alarmes, etc …
    - Alarms : permet de lancer des alarmes)
    - Events : événements lors d'un changement d'état
    - Dashboards

CloudTrail

- Monitorer les appels API et les stockent dans S3
- Possible de l'activer dans toutes les régions ou une seule région**.**
- Un seul CloudTrail est capable de monitorer plusieurs régions

VPC Flow Logs

Activable dans les VPC, subnets et les ENI (carte réseau), il capture le trafic entrant et sortant dans notre VPC, les logs sont publiés dans CloudWatch

Athena

Analyse les données non structurées, semi-structurées et structurées, stockées dans Amazon S3. Par exemple, des formats de données **CSV** ou **JSON**. Il fonctionne sans serveur. Il n'existe aucune infrastructure à gérer et vous ne payez que pour les requêtes que vous exécutez.

Cloud QuickSight

Permet aux entreprises de créer et analyser des visualisations de leurs données client (Demande un certain temps de configuration)

Amazon Kinesis

Vous pouvez utiliser Amazon Kinesis Data Streams pour collecter et traiter des flux volumineux d'enregistrements de données en **temps réel sur vos instances EC2.**

AWS Config

Permet de déterminer, de contrôler et d'évaluer les configurations de vos ressources AWS

AWS WAF

Pare-feu d'application Web qui aide à protéger les applications Web contre les failles Web les plus courantes.

Amazon Inspector

Description courte : Analyse les vulnérabilités sur vos EC2 et les listes avec un score de gravité.

Description longue : Teste l'accessibilité de réseau des instances EC2 et l'état de sécurité de vos applications qui s'exécutent sur ces instances. Il évalue les demandes pour leur exposition, leurs vulnérabilités et leurs écarts par rapport aux bonnes pratiques. Après avoir effectué une évaluation, Amazon Inspector produit une liste détaillée de constatations en matière de sécurité qui sont classées par niveau de gravité.

Amazon EMR

Plate-forme de cluster gérée qui simplifie l'exécution des infrastructures de données massives, telles qu'Apache Hadoop et Apache Spark, sur AWS pour traiter et analyser de grandes quantités de données

Déploiement rapide

Cloudformation

- C'est du IAS (Information as code)
- Permet d'automatiser la création de nos ressources AWS depuis un template JSON ou YAML.
- Il existe un outil visuel **Quick starts** qui permet de créer notre template
- **AWS Drift** permet de détecter les changements entre l'infra actuelle et l'infra décrite dans le templates

AWS Systems Manager

Fonctionne comme Ansible et permet de configurer et de gérer vos instances Amazon EC2 et vos serveurs on-premise.

AWS OpsWorks

Automatise les opérations sur les instances EC2 grâce à **Chef** et **Puppet**

AWS Elastic Beanstalk

Permet de déployer et de mettre à l'échelle des applications et services Web développés avec (Java, .NET, PHP, Node.js, Python, Ruby, Go et Docker) sur des serveurs familiers, tels qu'Apache, Nginx, Passenger et IIS.

Codepipline

C'est le Gitlab-ci d'AWS (intégration continue)

Il automatise les phases de développement, de test et de déploiement à chaque fois qu'un changement de code a lieu, en fonction du modèle de diffusion que vous avez défini. Il vous permet de diffuser des fonctionnalités et des **mises à jour de manière rapide** et fiable.

AWS Codedeploy

Comme Ansible, il permet d'automatiser les déploiements de logiciels vers divers EC2, vm On premise.

AWS CodeBuild

Service d'intégration continue entièrement qui compile votre code source, exécute des tests et produit des packages logiciels prêts à être déployés.

Mise en cache

Cloudfront

- C'est un CDN
- Son emplacement est dans les Edge Location.
- Il n'est pas fait pour héberger du contenu dynamique
- 2 types :
    - **Web distribution :** Pour les sites webs
    - **RTMP :** Media streaming
- Durée des objets S3 selon le **TTL**
- On peut recharger le cache soit :
    - Avec le TTL
    - En changeant le nom de l'objet

Pour les bases de données

Pour les bdd, on va **éviter d'utiliser Cloudfront**, car il existe des services qui sont déjà dédiés à ça.

Le cache sur une bdd, permet d'accéder à des requêtes plus rapidement, car les données sont stockées la mémoire et non plus dans le disque.

Service de mise en cache de bdd :

- **Amazon DynamoDB Accelerator** Pour mettre à jour le cache, il faut rajouter un TTL (lazy loading)
- **ElasticCache** avec
    - **Memcached**
    - **Redis**

Liste d’attente et notifications

SQS (Simple queue service)

Ordre de traitement

- **Standard Queue :** ne garantit pas l'unicité (donc possible d’avoir deux fois le même message) et le nombre de msg simultané est illimité
- **FIFO queue** : garantit l'unicité et l'ordre de traitement

Les limites

- Le délai de visibilité d'un message :
  - Par défaut 30 secondes.
  - Mini 0 secondes.
  - Max 12 heures
- Conservation des messages
  - Par défaut 4 jours.
  - Mini 60 secondes
  - Maxi 1 209 600 secondes (14 jours).
- Débit des messages
  - Standard Queue : illimité
  - FIFO : 300message/s (3000 m/s pour le traitement par groupe)

Tarif

On paie selon la taille et le nombre de messages.

Autres informations

- **Polling normal :** demande s’il y a un message à chaque fois et la queue lui répond (ça a un cout)
- **Long polling :** permet de réduire les couts car la queue ne répond que jusqu’à réception de message.
- Pour gérer les priorités (message premium en 1er et le reste en 2eme) , on va créer deux queues, il va taper d’abord dans la queue premium et si elle n’a pas de message alors on passe à la queue standard.
- Le service SQL peut aussi être utilisé pour améliorer le temps de calcul d’une EC2 (pour les traitements vidéo) et pour les bdd (en planifiant l'ajout d'entrées multiples dans bdd).

SNS (Amazon Simple Notification Service)

Service de messagerie (SMS, Email) pub/sub entièrement managé.

Fonctionne avec un système de souscription/Publication (comme à l’abonnement à une newsletter)

Conteneurs

ECS (Elastic container service)

Informations sur le service

- Il peut être combiné avec **EKS** (Elastic Kubenetes Service)
- Hautement scalable et rapide
- Permet de déployer des images docker dans cluster d'instances EC2
- On n'a pas accès aux EC2 ou sont hébergés nos conteneurs
- **AWS Fargate** permet d'exécuter des conteneurs sans avoir à gérer des serveurs ou clusters.

Comment rajouter mon image Docker

- **Etape 1 :** on choisit le nombre d'instance EC2
- **Etape 2 :** on crée une task définition
    - Image docker
    - Les ports à ouvrir
    - CPU memory
    - Container Registry
- Services c'est le nombre de tache qu'on déploie dans EC2

Serverless

Lambda

- Fonction qu'on crée avec notre langage préféré et qui peut être lancée par une ressource AWS.
- Supporte: Node.js, Java, Python, C#, Go and Ruby, PHP
- Peut être lancé depuis une Edge Location
- On peut mettre ses librairies persos dans le package de déploiement, ou les ajouter définitivement dans **lambda layers**
- CloudWatch permet de voir les logs Lambda
- Un délai d'exécution jusqu'à **15 min** max
- Pour les fonctions plus lentes, on doit les splitter via **step functions** qui fonctionnent avec **state language**

API Gateway

C'est une Gateway d'appel api pour n'importe quel service AWS. On peut aussi l'utiliser pour créer notre propre api sans forcément que ça soit lié aux ressources AWS.

Disaster planning

- **RPO** (Recovery Point Objective) : si j'accepte un rpo de 12h alors j'accepte de perdre de mes données pendant de 12h
- **RTO** (Recovery Time Objective) : je peux accepter de perdre une heure de service

Il faut considérer que toute une région AWS peut tomber (du style tremblement de terre) :

- **S3** : Cross-region-replication
- **EBS** : un snapshot peut être copier dans une autre région
- **Route53** : Failover (il faut dans ce cas répliquer l'infra sur deux régions différentes)
- **RDS** : Read Réplicas (qui peut être déployer sur une autre région) combiné avec le multi-AZ

Autres infos

- **AWS Server Migration Service** : permet de migrer une infra on premise , style VMware ou Hypver-V vers AWS.
- La CLI donne plus de privilèges que la console
- MFA ne permet pas de se prémunir de la suppression mais permet juste une prévention