

Résumé CLD - TE1

CLD

Cloud Introduction, Cloud concepts, IaaS, Scaling App and labs

Résumé du document

Table des matières

1. AWS	2
1.1. EC2	2
1.2. Security Group	2
1.3. NAT	2
1.4. Gestion des coûts	2
1.5. CLI	2
1.6. Load Balancer	2
2. Notion de réseaux	3
2.1. Load Balancer	3
2.1.1. Load Balancer applicatif	3
2.1.2. TCP vs HTTP	3
2.2. Forward policies	3
3. Cloud introduction	4
3.1. Définir une instance	4
3.2. Etats d'une instance	4
4. Cloud concepts	5
4.1. Modèle de service cloud	5
4.2. Définition du cloud (NIST)	5
4.2.1. Caractéristiques du cloud	5
4.2.1.1. Service à la demande en libre-service	5
4.2.1.2. Accès réseau large	5
4.2.1.3. Mutualisation des ressources	5
4.2.1.4. Élasticité rapide (rapid elasticity)	5
4.2.1.4.1. Scalable vs Elastique	6
4.2.1.5. Service mesuré	6
4.2.2. Modèles de service	6
4.2.3. Modèles de déploiement	6
4.2.3.1. Public cloud	6
4.2.3.2. Private cloud	7
4.2.3.3. Community cloud	7
4.2.3.4. Hybrid cloud	7

1. AWS

1.1. EC2

Statut d'instance

- Pending : démarrage de l'instance
- Running : instance en cours d'exécution, coût sur la RAM, CPU, espace mémoire et IP
- Stopping : arrêt de l'instance
- Stopped : instance arrêtée, coût sur l'espace mémoire et IP
- Terminated : instance va être supprimée définitivement

1.2. Security Group

Les security groups sont des pare-feu virtuels qui contrôlent le trafic entrant et sortant d'une instance. Ceux-ci peuvent être attribués à une ou plusieurs instances.

Ils permettent de contrôler avec des **inbounds** et **outbounds** rules le trafic entrant et sortant d'une instance. Pour cela, nous pouvons définir quel **port** ou **protocole** doit être autorisé ou non.

1.3. NAT

Le NAT permet de faire une translation d'adresse IP privée vers une adresse IP publique. Il permet de faire communiquer des instances privées avec l'extérieur.

1.4. Gestion des coûts

- Prendre des instances moins performantes et adaptées à nos besoins
- Limiter le temps d'utilisation des instances
- Utiliser un mode de facturation par paiement en avance

1.5. CLI

Le CLI d'AWS utilise des commandes pour interagir avec les services AWS. Il utilise le protocole HTTP/S pour dialoguer avec les API d'AWS.

1.6. Load Balancer

Le Load Balancer permet de répartir la charge entre plusieurs instances. De plus, peu importe le nombre d'instances, on doit exposer au client qu'un seul point d'entrée.

2. Notion de réseaux

2.1. Load Balancer

2.1.1. Load Balancer applicatif

Attention! pour que LB puisse être appliqué à une application il faut absolument que celle-ci soit **stateless**.

2.1.2. TCP vs HTTP

Un load balancer TCP sera plus rapide qu'un load balancer HTTP car celui-ci utilise seulement la couche TCP alors que le load balancer HTTP devra attendre une requête avec de pouvoir faire sa décision de redirection. Un load balancer TCP sera donc plus rapide en termes de latence et de performances.

Un load balancer TCP sera moins précis dans ses choix de redirections comparé à un load balancer HTTP.

2.2. Forward policies

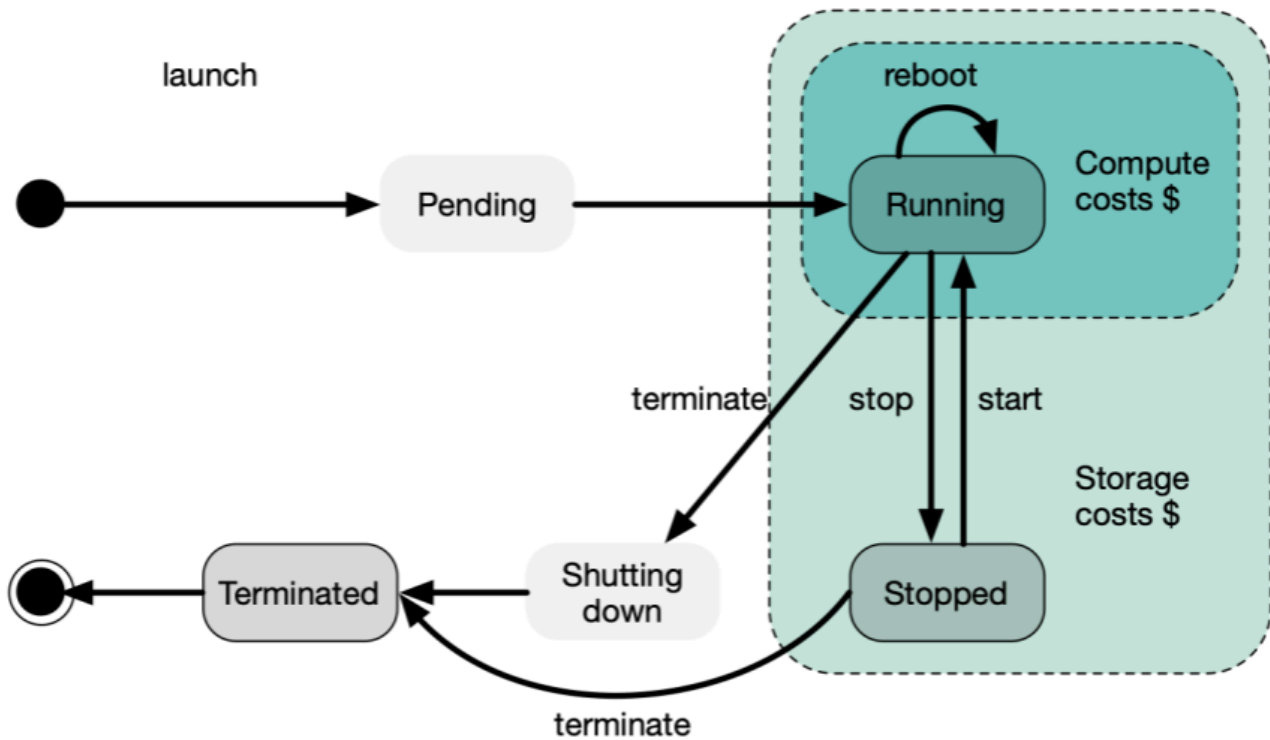
- Round Robin : Aller d'un serveur à un autre pour chaque requêtes et ceci dans un ordre précis et boucler comme cela.
- IP Hash : Hashé l'IP du client pour choisir un serveur.
- Least Connection : Aller sur les serveurs avec le moins de connexions.

3. Cloud introduction

3.1. Définir une instance

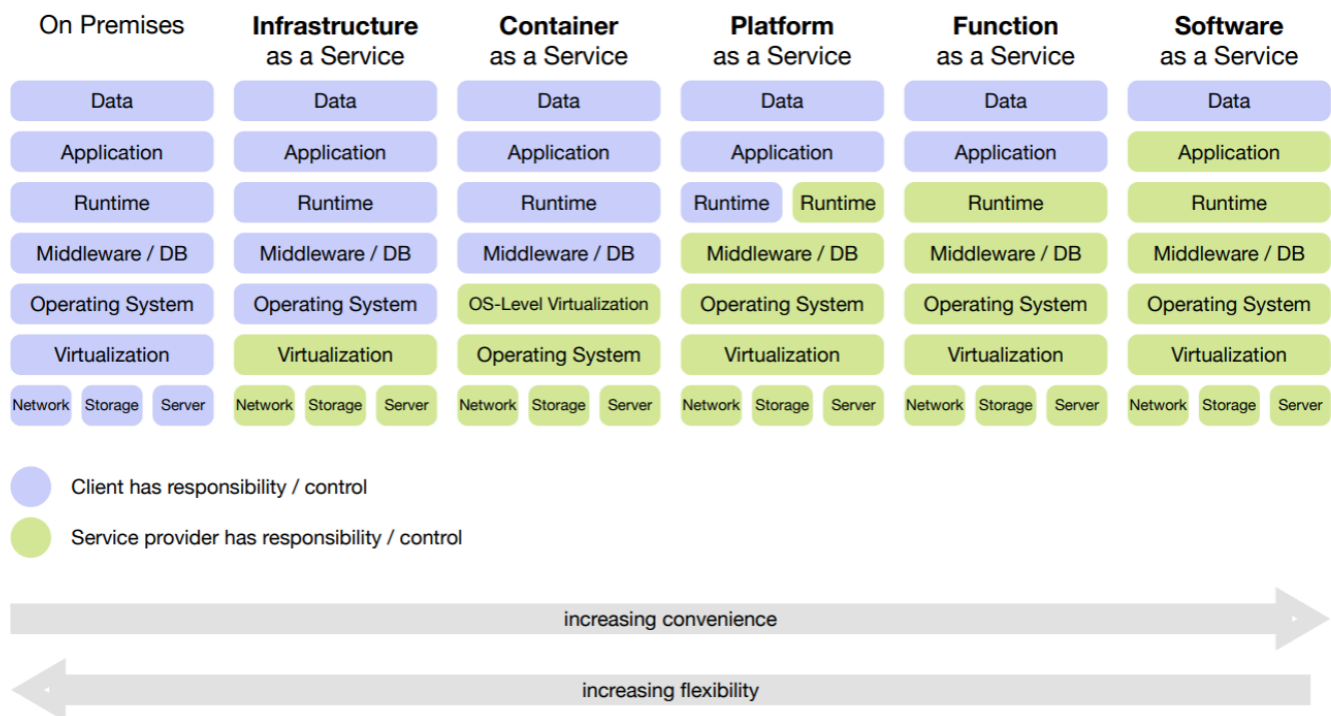
- Ou créer l'instance (region / availability zone)
- Quel OS choisir : machine image
- Quelle quantité de CPU et RAM : type d'instance
- Quelle quantité de stockage : volume de stockage
- Quelles règles de sécurité : security group (AWS)
- Comment se connecter à l'instance : clé publique/privée (SSH port 22)

3.2. Etats d'une instance



4. Cloud concepts

4.1. Modèle de service cloud



4.2. Définition du cloud (NIST)

- L'informatique en nuage est un modèle permettant
 - un accès réseau omniprésent, pratique et à la demande
 - à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services)
 - qui peuvent être rapidement provisionnées et libérées avec un minimum d'effort de gestion ou d'interaction avec le fournisseur de services

4.2.1. Caractéristiques du cloud

4.2.1.1. Service à la demande en libre-service

Provisionnement automatique **sans** interaction humaine : Le client du cloud est capable d'obtenir des ressources cloud presque instantanément (en quelques minutes) à tout moment. Le fournisseur a complètement automatisé le processus et il n'y a aucune interaction humaine.

4.2.1.2. Accès réseau large

Accès via des protocoles standardisés depuis une variété de clients : Les ingénieurs du client cloud et les utilisateurs peuvent accéder aux ressources cloud en utilisant Internet et des protocoles standardisés.

4.2.1.3. Mutualisation des ressources

Servir plusieurs clients dans un modèle multi-locataire, attribution dynamique des ressources à partir d'un pool, indépendance de l'emplacement : Le fournisseur de cloud fournit des ressources cloud à plusieurs clients cloud (= locataires). Le fournisseur de cloud utilise un logiciel spécial qui fournit des espaces séparés et isolés pour chaque locataire. Le fournisseur de cloud dispose de grands clusters de serveurs et de disques (pools) qui servent des milliers de clients. Pour le client, il n'a pas d'importance de savoir où exactement sa machine virtuelle fonctionne ou où ses données sont stockées.

4.2.1.4. Élasticité rapide (rapid elasticity)

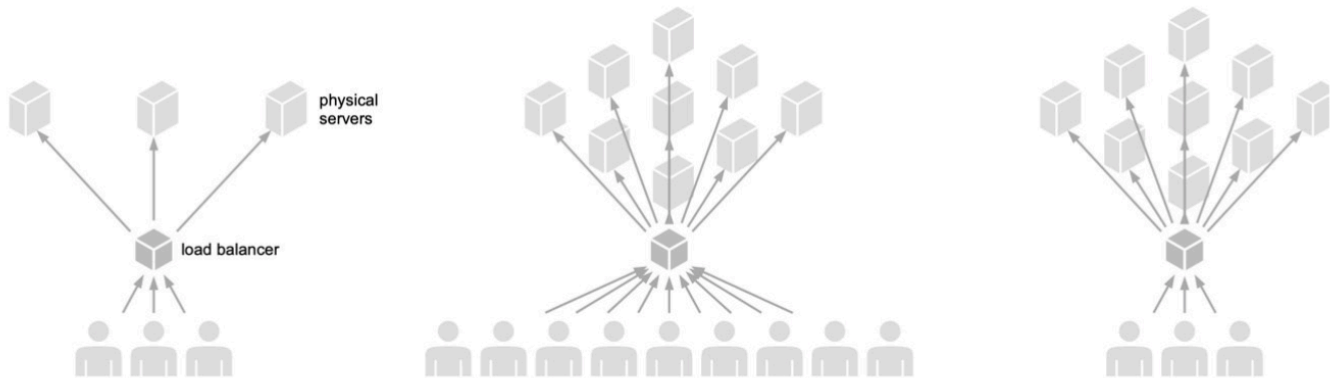
Provisionnement/déprovisionnement rapide pour s'étendre/réduire, capacité apparemment illimitée : Lorsque le nombre d'utilisateurs augmente soudainement et donc la charge sur les ressources, les ingénieurs clients du cloud peuvent rapidement obtenir (provisionner) des ressources supplémentaires. De même, lorsque la charge diminue à

nouveau, ils peuvent libérer (déprovisionner) des ressources et cesser de les payer. Le fournisseur de cloud dispose de tant de ressources que le client ne se voit jamais refuser une demande de nouvelle ressource.

4.2.1.4.1. Scalable vs Élastique

Un système **scalable** permet de supporter une **montée** en charge en ajoutant des ressources. Cependant il est **difficile de profiter d'une diminution** de la charge.

- Trop de friction pour reconfigurer les serveurs pour une autre tâche
- Le système est scalable, mais pas élastique



Un système **élastique** permet de supporter une **montée** en charge en ajoutant des ressources, mais aussi de profiter d'une **diminution** de la charge en retirant des ressources. Cela permet de faire beaucoup d'économies en stoppant les machines non-utilisées.



4.2.1.5. Service mesuré

L'utilisation est surveillée et contrôlée, offrant une transparence dans la facturation : Le fournisseur de cloud mesure précisément quand une machine virtuelle est provisionnée pour un client et quand elle est déprovisionnée, combien de données un client stocke dans une base de données, etc. (surveillance). À la fin du mois, le fournisseur de cloud envoie une facture détaillée au client, répertoriant chaque ressource cloud et son coût.

4.2.2. Modèles de service

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

4.2.3. Modèles de déploiement

4.2.3.1. Public cloud

- Un fournisseur de cloud héberge et fournit un cloud et permet à quiconque possède une carte de crédit de l'utiliser.
- L'infrastructure est partagée par tout le monde. Par exemple, sur le même serveur fonctionnent des machines virtuelles appartenant à différents clients.

4.2.3.2. Private cloud

4.2.3.3. Community cloud

4.2.3.4. Hybrid cloud