Résumé CLD - TE1

CLD

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

Résumé du document

Table des matières

1. AWS 3
1.1. EC2 3
1.2. Security Group 3
1.3. NAT 3
1.4. Gestion des coûts 3
1.5. CLI 3
1.6. Load Balancer 3
2. Cloud introduction 4
2.1. Définir une instance 4
2.2. Etats d'une instance 4
3. Cloud concepts 5
3.1. Modèle de service cloud 5
3.2. Définition du cloud (NIST) 5
3.2.1. Caraactéristiques du cloud 5
3.2.1.1. Service à la demande en libre-service 5
3.2.1.2. Accès réseau large 5
3.2.1.3. Mutualisation des ressources 5
3.2.1.4. Élasticité rapide (rapid elasticity) 5
3.2.1.4.1. Scalable vs Elastique 6
3.2.1.5. Service mesuré 6
3.2.2. Modèles de service 6
3.2.3. Modèles de déploiement 6
3.2.3.1. Public cloud 6
3.2.3.2. Private cloud 7
3.2.3.3. Community cloud 7
3.2.3.4. Hybrid cloud 7
3.3. Barrières à l'adoption du cloud 7
4. IaaS 8
4.1. Regions 8
4.1.1. Définition AWS 8
4.2. Virtualisation 8
4.2.1. Propriétés 8
4.2.2. Hypervisor type 1 8
5. Scaling App 9
5.1. Scaling type 9
5.1.1. Vertical scaling 9
5.1.2. Horizontal scaling 9
5.2. HTTP Proxy 9
5.3. Load Balancer 9
5.3.1. TCP vs HTTP 10

5.3.2. Forward policies 10
5.4. Health Check 10
5.4.1. Default health check 10

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
- Désallouer les adresses IPv4 quand elles ne sont pas utilisées

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

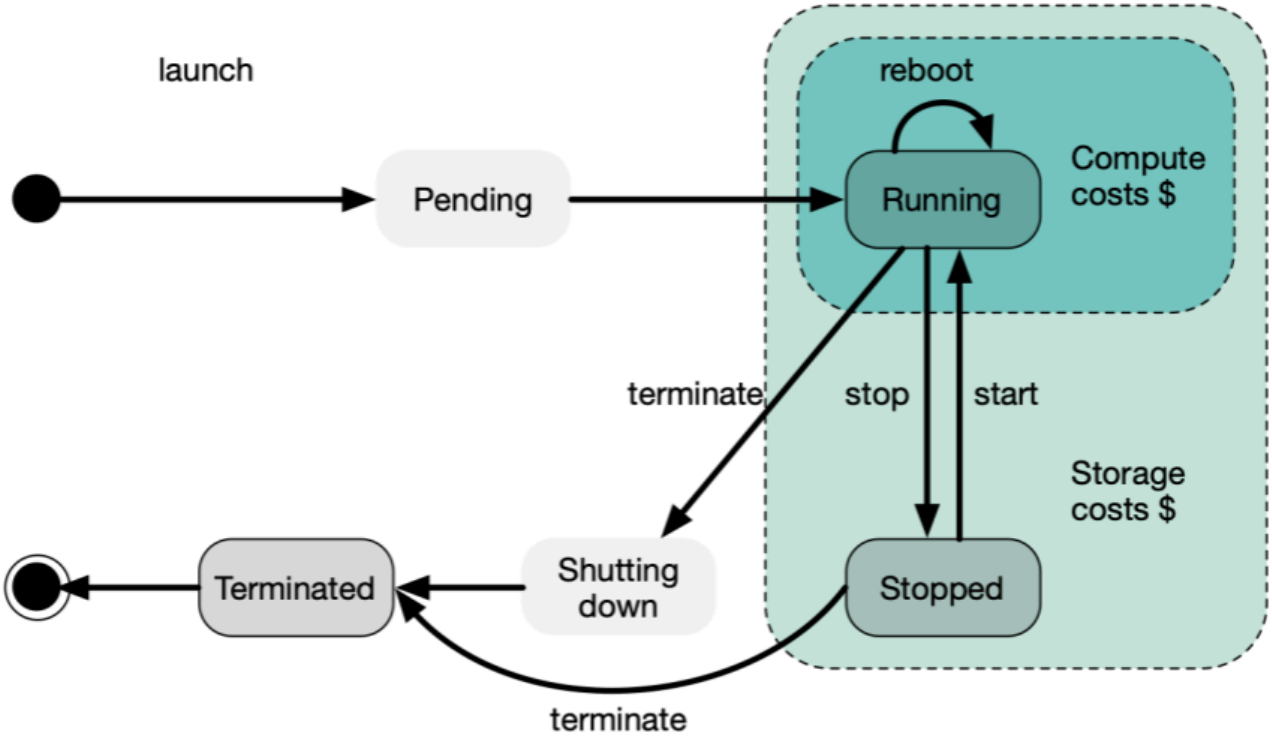
Le Load Balancer permet de répartir la charge entre plusieurs instances. De plus, peu importe le nombre d'instances, on doit exposer au client qu'un seul point d'entrée.

2. Cloud introduction

2.1. Définir une instance

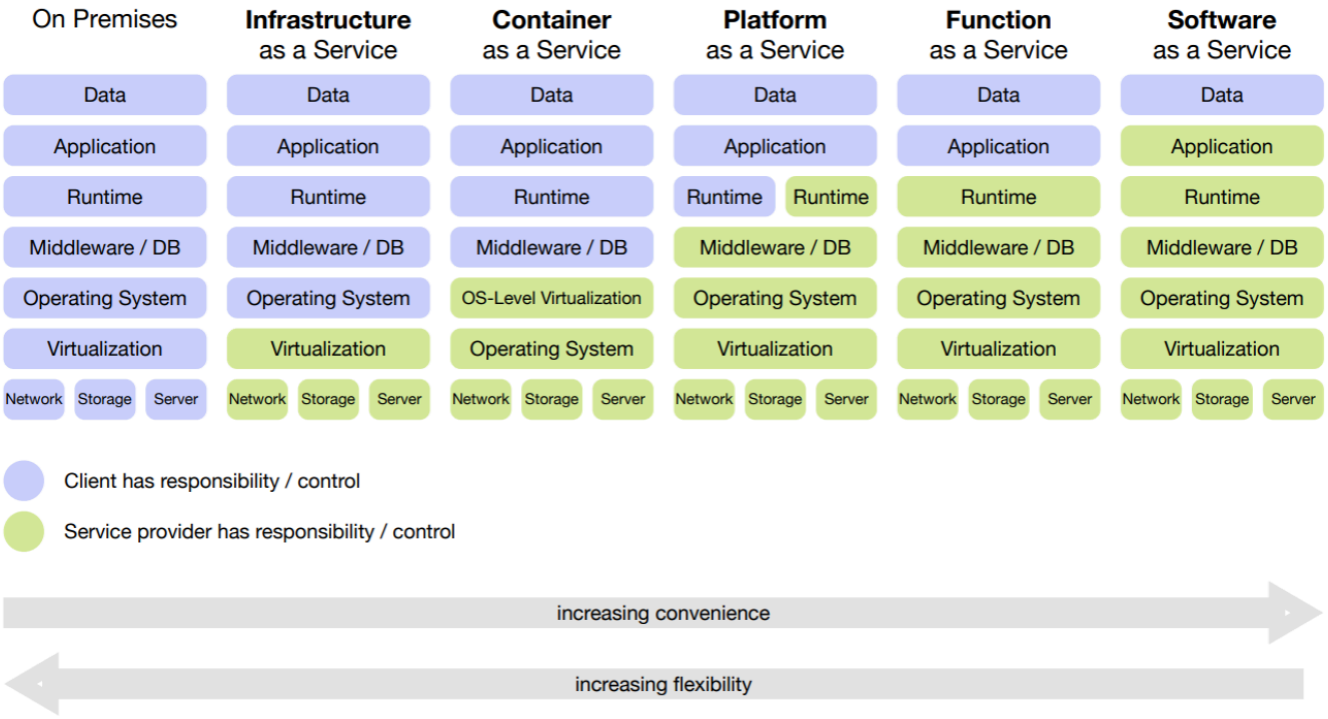
- Ou créer l'instance (region / availability zone)
- Quel OS choisir : machine image
- Quelle quantité de CPU et RAM : type d'instance
- Quelle quantité de stockage : volume de stockage
- Quelles règles de sécurité : security group (AWS)
- Comment se connecter à l'instance : clé publique/privée (SSH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

2.2. Etats d'une instance



3. Cloud concepts

3.1. Modèle de service cloud



3.2. 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

3.2.1. Caraactéristiques du cloud

3.2.1.1. 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.

3.2.1.2. 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.

3.2.1.3. 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.

3.2.1.4. Élasticité rapide (rapid elasticity)

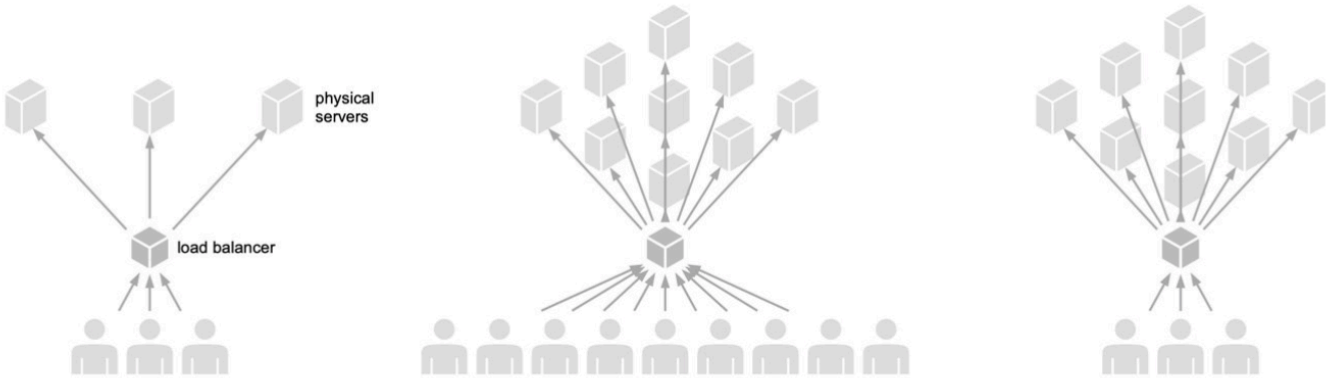
Provisionnement/déprovisionnement rapide pour s'étendre/réduire, capacité apparemment illimitée : Lorsque le nombre d'utilisateurs augmente soudainement et donc la charge sur les ressources, les ingénieurs clients du cloud peuvent rapidement obtenir (provisionner) des ressources supplémentaires. De même,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

3.2.1.4.1. Scalable vs Elastique

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.

3.2.1.5. 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.

3.2.2. Modèles de service

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

3.2.3. Modèles de déploiement

3.2.3.1. 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.

3.2.3.2. Private cloud

L'infrastructure cloud est utilisée par une seule entreprise cliente.
- L'entreprise héberge et exploite elle-même le cloud, sans délégation à une autre entreprise.
- Ou l'entreprise demande à un fournisseur de cloud de créer un cloud exclusivement pour elle.

Utilisé par des entreprises avec des données extrêmement sensibles.

Les "clients" du cloud sont maintenant les départements et les employés de l'entreprise. Ils bénéficient toujours de :
- Service en libre-service à la demande
- Mise en commun des ressources entre les départements de l'entreprise
- Élasticité rapide (partielle)

Exemples :
- Banques privées
- Défense nationale

3.2.3.3. Community cloud

Un nombre très limité de clients ayant des intérêts similaires (par exemple, hôpitaux, constructeurs automobiles, universités) se réunissent et créent un cloud uniquement pour eux-mêmes.

Utilisé lorsque les données sont sensibles, mais pas extrêmement, ou pour des besoins de calcul spéciaux.

Exemples :
- Cloud de santé
- Cloud de simulation de crash de voitures
- Cloud de formation en intelligence artificielle

3.2.3.4. Hybrid cloud

Une entreprise utilise simultanément un ou plusieurs clouds publics, son propre cloud privé et éventuellement un cloud communautaire.

Exemples : C'est le cas courant pour les grandes et moyennes entreprises.

3.3. Barrières à l'adoption du cloud

- Risques de sécurité / confiance envers le fournisseur de cloud
- Déplacements de données relativement coûteux
- Verrouillage du fournisseur
- Accord de niveau de service (SLA)
- Aspect légal
- Aspect politique

4. IaaS

4.1. Regions

Une région est une zone géographique. Cela permet de placer des instances dans des zones géographiques différentes pour éviter les pannes.

Facteur de séléction d'une région

- Proximité géographique des clients
- Gouvernance des données, respect des lois
- Quels services sont disponibles dans la région
- Les couts peuvent varier d'une région à l'autre

Quand on parle d'**availability zone**, on parle de **datacenter**.

4.1.1. Définition AWS

- Les Zones de Disponibilité sont constituées de centres de données distincts (généralement au nombre de 3).
- Les Zones de Disponibilité sont connectées entre elles par un réseau privé à haute vitesse.
- Il est garanti que deux Zones de Disponibilité sont séparées par plusieurs kilomètres afin que les catastrophes telles que les incendies et les inondations soient contenues dans l'une et n'affectent pas l'autre.
- Il est garanti que deux Zones de Disponibilité sont situées à moins de 100 km l'une de l'autre afin que la latence du réseau soit faible.

4.2. Virtualisation

La virtualisation permet de créer des machines virtuelles sur un serveur physique. Cela permet de faire tourner plusieurs systèmes d'exploitation sur un seul serveur.

4.2.1. Propriétés

Compatibilité binaire

- Le code (système d'exploitation et applications) s'exécutant dans une machine virtuelle "pense" qu'il fonctionne sur une machine normale. Le système d'exploitation pense qu'il contrôle complètement la machine.
- Aucune modification du système d'exploitation ou des applications.

Interposition

- Toutes les actions d'une machine virtuelle doivent passer par l'hyperviseur.

Isolation

- Un programme s'exécutant dans une machine virtuelle ne peut pas accéder aux données d'une autre machine virtuelle.
  - Isolation logicielle
  - Isolation des pannes
  - Une machine virtuelle avec une charge de traitement élevée ne peut pas affecter les performances d'une autre machine virtuelle.
  - Isolation des performances

Encapsulation

- L'état complet d'une machine virtuelle peut être capturé dans un fichier : image de machine virtuelle.
- Le fichier peut être manipulé comme n'importe quel autre fichier : transféré, dupliqué, supprimé, …

4.2.2. Hypervisor type 1

L'hyperviseur de type 1 est installé directement sur le matériel. Il permet de gérer les machines virtuelles et de les faire tourner sur le matériel. Il est plus performant que l'hyperviseur de type 2 car n'a pas de couches intermédiaires. C'est ce type de machines qui sont proposés par les fournisseurs de cloud.

5. Scaling App

5.1. Scaling type

5.1.1. Vertical scaling

Le vertical scaling consiste à ajouter des ressources à des machines existantes.

- Facile à faire en cloud computing
- Peut devenir vite cher
- Limité par l'architecture de la machine

5.1.2. Horizontal scaling

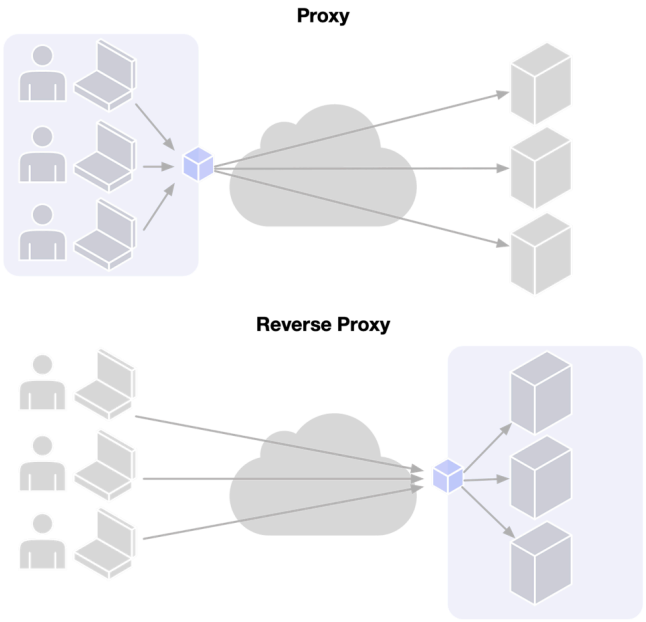
Le horizontal scaling consiste à ajouter des machines pour supporter la charge.

- Permet d'utiliser des machines moins chères
- Permet une scalabilité quasi infinie

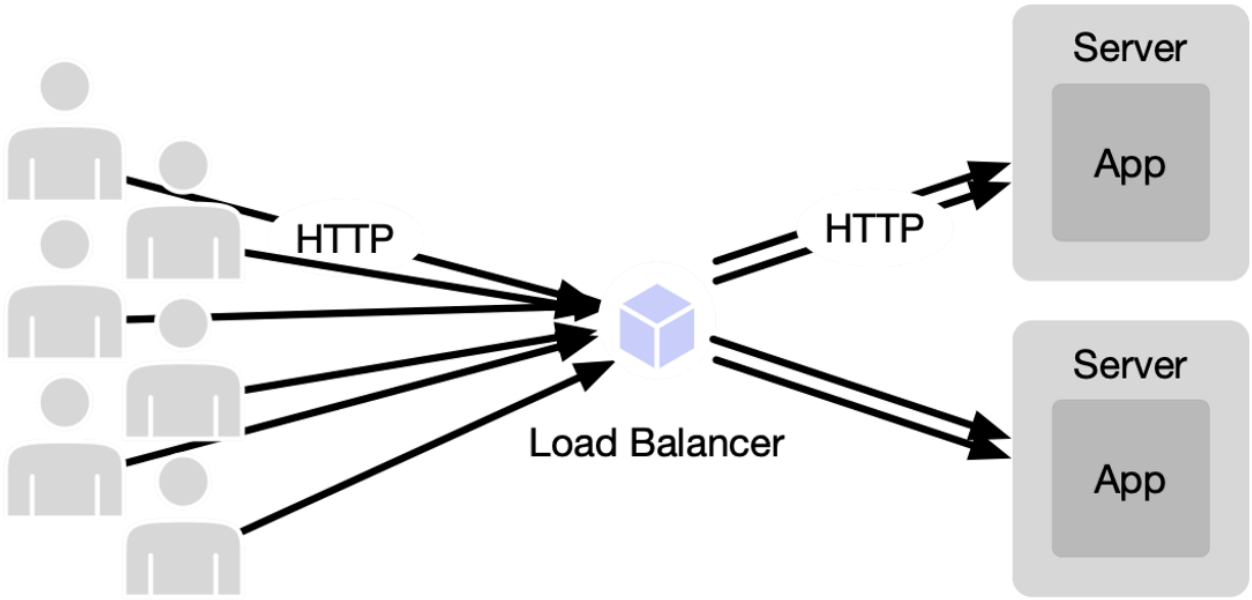
5.2. HTTP Proxy

Un proxy HTTP est un serveur qui sert d'intermédiaire entre les clients HTTP et les serveurs HTTP. Il a pour but d'être transparent pour le client et le serveur. Cela s'appelle un proxy quand il est deployé côté client et un reverse proxy quand il est deployé côté serveur.

Important: un load balancer est un reverse proxy.



5.3. Load Balancer



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

5.3.1. 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.

5.3.2. 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.

5.4. Health Check

- Les serveurs peuvent rencontrer des problèmes et renvoyer des erreurs ou devenir totalement non réactifs (panne matérielle, bug logiciel, disque plein, etc.).
- Lorsque le répartiteur de charge envoie une requête utilisateur à un serveur non réactif, l'utilisateur percevra le site web comme non réactif.
- Pour éviter cette situation, le répartiteur de charge vérifie en continu que les serveurs sont "en bonne santé", c'est-à-dire qu'ils fonctionnent correctement.
- Le répartiteur de charge envoie régulièrement une requête HTTP de test au point de terminaison de l'application, appelée vérification de santé.
- Lorsque l'application répond par 200 OK, le répartiteur de charge considère le serveur comme sain.
- Lorsque l'application répond par une erreur ou ne répond pas du tout, le serveur est considéré comme non sain.
- Le répartiteur de charge exclut immédiatement les serveurs non sains du routage des requêtes.

5.4.1. Default health check

Les causes suivantes peuvent mettre un serveur hors service, donc mettre en defaut un health check :

- Port et protocole malconfigurés
- Service down ou surchargé
- Problème de routage