Université de Bretagne Occidentale
Faculté de Sciences et Techniques
Département Informatique

Thales Brest
Département Discipline Logicielle et Innovation

Rapport de Projet
Étude d'une Solution de Stockage Distribué

Lucas Videlaine

Étudiant Licence Informatique
Parcours Conception et Développement d'Applications - UBO

2020

Université de Bretagne Occidentale
Faculté de Sciences et Techniques
Département Informatique

Thales Brest
Département Discipline Logicielle et Innovation

Rapport de Projet
Étude d'une Solution de Stockage Distribué

Lucas Videlaine

Étudiant Licence Informatique
Parcours Conception et Développement d'Applications - UBO

Remerciements

Je souhaite remercier Thales Brest, et sa directrice Mme Marie-José Vairon d'avoir été une structure d'accueil aussi enrichissante.

Je remercie également l'Université de Bretagne Occidentale de Brest, et son président Mr Matthieu Gallou pour m'avoir permis de réaliser un stage dans le domaine qui me passionne.

Je voudrais également remercier Mr Alexandre Skrzyniarz de m'avoir conseillé en dirigeant mon choix d'étude.

Je tiens à remercier Mr Laurent Lemarchand, toute l'équipe enseignante, et mes collègues de promotion pour cette année de licence qui a été riche d'expériences.

Enfin je remercie Mr Emmanuel Braux et Mr Yvon Kermarrec pour le soutien qu'ils me fournissent, et sans qui je n'aurais pas cette passion pour le DevOps.

Sommaire

Introduction

Depuis sa création en 1968, Thales est un groupe spécialisé en électronique dans les domaines de la sécurité, de la défense, du transport et de l'aérospatiale. Thales est un des leaders mondiaux dans l'équipement aéronautique, de l'espace, de la défense et des transports.

Dans le cadre de la guerre électronique, les différents bâtiments militaires, tels que les sous-marins ou les frégates, sont équipés de systèmes de détection de mines et de torpilles. Ces systèmes reposent sur des infrastructures embarquées ainsi que sur des solutions de virtualisation.

Dans le but d'améliorer l'efficacité des systèmes embarqués, le département Discipline Logicielle et Innovation et Thales Brest ont décidé d'évaluer une nouvelle solution.

Mon travail dans le cadre du stage de Licence Informatique est donc d'analyser et synthétiser les capacités de cette solution dans des cas d'usage spécifiques afin de valider, ou non, son exploitation sur le terrain.

Du fait des événements sanitaires de 2020, mon stage a été reporté, et afin de ne pas perdre trop de temps sur le projet, on m'a demandé de m'autoformer sur Ceph, une solution de stockage distribué, grandement utilisé dans le domaine système.

Nous présenterons donc le groupe qu'est Thales, ainsi que son activité dans le monde. Au chapitre 2 nous présenterons le principe de stockage distribué et la technologie utilisée. Il présentera la solution Ceph. Et il s'achèvera par l'exposé des résultats au chapitre 3, en plus d'explications sur la manière de déployer et d'administrer la solution.

1 - Présentation de l'entreprise

1.1 - Situation géographique

Thales Brest est implanté près du technopôle, en face de la gare du tramway, aux portes de Plouzané.

Son siège social se situe dans le quartier de la Défense à Paris. Mais Thales dispose de beaucoup d'autres sites, puisque le groupe est présent dans près de 68 pays à travers le monde.

Vue du ciel des locaux brestois extrait du site actu.fr

1.2 - Historique

En 1968, la société Thomson-CSF est créée dans le but de poursuivre l'exploitation des brevets, en France, de la maison mère américaine Thomson-Houston Electric Corp.

Alors qu'à la fin des années 1980 une croissance s'opère dans le domaine de la défense en Europe, le gouvernement français demandent aux sociétés Aerospatiale, Alcatel et Dassault Industries de conclurent un accord de coopération avec Thomson-CSF. Cet accord, signé en 1998, permet de renforcer les actifs de Thomson-CSF et de consolider son périmètre d'activité dans la défense et l'électronique tout en améliorant son implantation dans les pays d'Europe.

C'est enfin en 2000, que Thomson-CSF devient Thales et axe son organisation autour de la défense, de l'aéronautique et des technologies de l'information. Thales se voit grandir vers les métiers de la sécurité jusqu'en 2007, où la société acquit les activités de transport, de sécurité et d'aéronautique d'Alcatel-Lucent, ce qui propulse la société en tant qu'acteur mondial. Toujours en 2007, Thales signe un accord avec DCNS qui lui confère une participation de 25 %

dans cet acteur français de l'industrie navale et lui permet de devenir son partenaire industriel.

Finalement, en 2019, Thales fait l'acquisition de Gemalto, une société de renommée internationale dans le domaine de la sécurité numérique au service des entreprises et des gouvernements. Thales devient ainsi le leader mondial de la sécurité.

1.3 - L'activité de Thales

Thales est présent dans 5 grands secteurs.

- l'Aéronautique : gestion du trafic aérien, simulation, services embarqués.
- l'Espace : télécommunications, exploration, infrastructures orbitales.
- le Transport Terrestre : signalisation, communications, sécurité.
- la Sécurité Numérique : biométrie, cloud, chiffrement, authentification.
- la Défense : protection des États, systèmes de surveillance/détection/renseignements.

1.4 - Les ressources de Thales

A l'image de son empreinte mondiale, Thales regroupe un total impressionnant de 83 000 collaborateurs répartis dans 68 pays. Le groupe représente un chiffre d'affaires de 19 milliards d'euros en 2019, dont 1 milliard est consacré à la recherche et au développement.

Au cours des années, Thales a déposé de nombreux brevets, en 2020 son portefeuille de propriété intellectuelle s'élève à 20 500 brevets, dont un prix Nobel de physique en 2007.

1.5 - Le projet

Le département Discipline Logicielle et Innovation souhaite [illegible] N'ayant pas pu commencer cette analyse pour le moment, du fait du confinement en relation avec le Covid-19 j'ai pour mission de m'autoformer sur Ceph. L'objectif est que je puisse à la fois étoffer mes compétences tout en intégrant des connaissances me permettant d'échanger sur le sujet avec un certain avis d'expertise.

2 – Présentation du principe de stockage distribué

2.1 – Les systèmes de stockage « classique »

Aujourd'hui les entreprises possèdent différentes méthodes pour stocker leurs données. En fonction du type de donnée, de son importance, de sa confidentialité, les manières dont les entreprises stockent ces données diffèrent grandement.

Jusqu'à présent, les entreprises utilisent des supports de stockage tels que des clés USB (même si l'aspect sécurité de cette solution est très mauvais sur bien des points), des disques durs partagés sur le réseau connu sous le nom de NAS dont certains embarquent des technologies de RAID permettant d'améliorer les performances, la sécurité, ou la tolérance aux pannes. Aussi, on voit certains groupes externaliser sur le Cloud.

Par exemple, une entreprise va permettre à ses salariés d'utiliser des clés USB entre leur station de travail et leur machine personnelle afin de transférer des fichiers. D'autres centralisent sur des NAS afin de prévenir les pertes de données à l'aide du RAID. On voit aussi maintenant des entreprises utilisant la Google Suite (pour ne citer qu'elle) afin de gérer leurs traitements de texte ou leur tableurs.

Cette logique reste vraie pour la plupart des situations, mais pour des cas particuliers dans le domaine système, on ne peut se permettre de travailler avec des montages de lecteur réseau comme vu ci-dessus. En effet, il paraît insensé de déployer des serveurs avec comme solution de stockage un simple NAS.

2.2 – Les systèmes distribués

Dans le monde de l'informatique, l'architecture distribuée est maintenant une technologie répandue, et incontournable.

Ce procédé consiste à partager, entre un ensemble de machine, les ressources disponibles à l'aide de messages à travers le réseau.

Cette architecture très flexible est reconnue pour sa disponibilité, car dans le cas de panne l'architecture ne s'effondre pas et seul le service touché est indisponible.

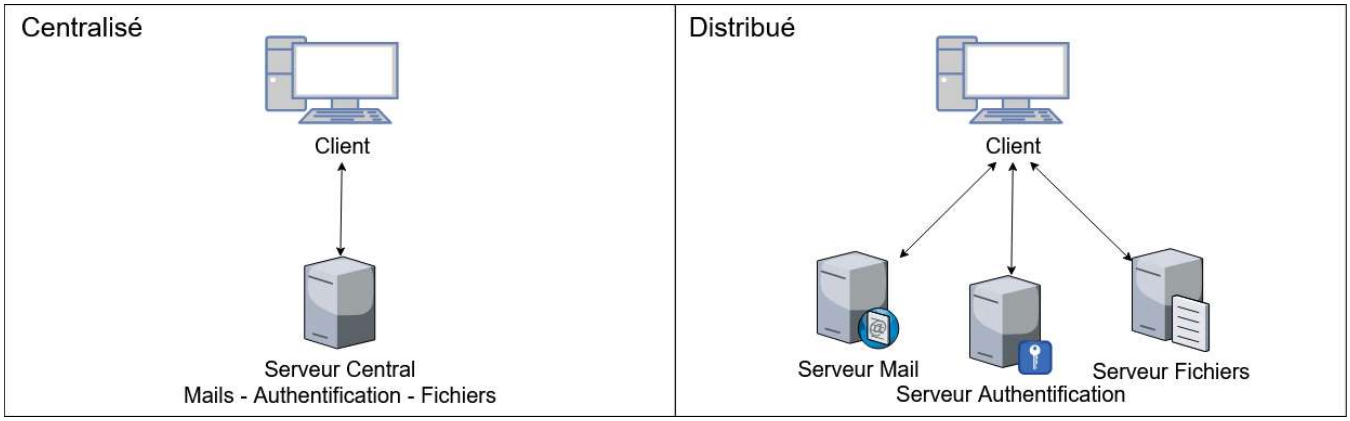


Figure 1 – Illustration Système Distribué

La figure 1 ci-dessus présente de façon simplifiée le principe de système centralisé et de système distribué.

On comprend que si dans le cas d'un système centralisé le serveur central tombe en panne, alors l'ensemble des services qu'il gère seront indisponibles. A contrario, dans le contexte d'un système distribué, si un serveur tombe en panne il entraînera l'indisponibilité du seul service qu'il gère.

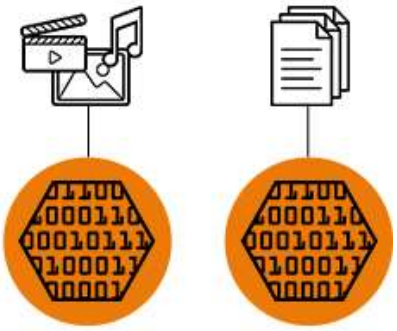
2.3 – L'architecture Ceph

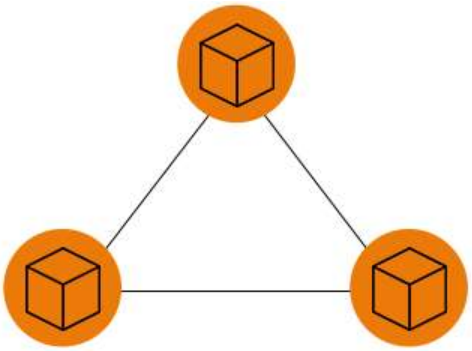
Ceph est une solution libre de stockage distribué, initialement créé par Sage Weil lors de sa thèse de doctorat qu'il a effectué à l'Université de Californie à Santa Cruz entre 2004 et 2007. Une fois diplômé, Sage Weil a commencé à travailler à plein temps sur Ceph en fondant son entreprise « InkTank ». Puis, en 2014, RedHat décide de racheter « InkTank » afin de poursuivre le développement de Ceph en interne. Finalement, en 2015, un commité est formé afin d'assister et de conseiller les équipes sur la direction à prendre lors du développement de la solution. Ce commité comprend des grandes organisations telles que Intel, Cisco, SUSE ou encore Canonical.

Les deux objectifs principaux de Ceph sont d'être complètement distribué sans point unique de défaillance, tout en étant extensible (Ceph permet de gérer des quantités de données jusqu'à l'exaoctet, 10^{18} octets). Les données sont répliquées, permettant au système d'être tolérant aux pannes : ainsi même si l'un de vos serveurs subit une panne, l'architecture est capable de reconstruire les données et d'éviter toute perte.

A la base de Ceph on trouve RADOS, un système de stockage d'objets répartis. On trouve ensuite LIBRADOS, l'API et Librairie qui permet aux applications d'échanger avec les objets stockés. Cette solide base permet à Ceph de proposer 3 services de stockage à ses utilisateurs que voici.

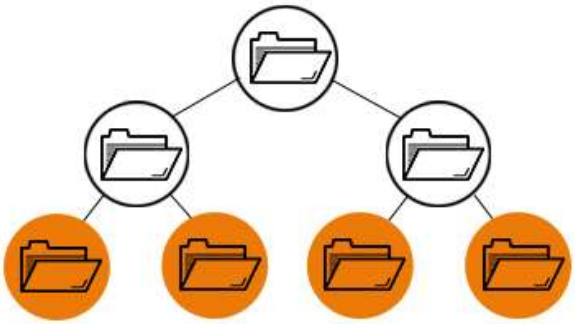
Le Stockage en mode Objet avec RADOSGW : C'est une structure de données dans laquelle les fichiers sont décomposés en éléments unitaires distincts que l'on appelle objet. Ces objets sont ensuite conservés dans un référentiel et n'apparaissent pas comme fichiers ou dossiers au sein de la zone de stockage. Ce mode est recommandé pour les mails, photos, vidéos, et documents texte.





Le Stockage en mode Bloc avec RBD : C'est une structure de données dans laquelle on regroupe les données par bloc comportant un identifiant unique. Chaque bloc est stocké dans l'emplacement qui lui a été attribué. On peut ainsi optimiser le stockage en plaçant les blocs dans les environnements qui conviennent le mieux (avec par exemple une infrastructure hétérogène embarquant des machines Windows, Linux et Mac). Ce mode est recommandé pour supporter les serveurs puisque permet les snapshots, le cache tiering (explications *Figure 3*) et la compression des données.

Le Stockage en mode Fichier avec CephFS : C'est le mode de stockage le plus commun puisque c'est celui qu'utilisent nos ordinateurs de tous les jours. Ce système est basé sur des chemins d'accès, tout ce qu'il y a de plus classique. Ce mode est utilisé comme système de fichiers (compatible Posix), intégrant des Listes de Contrôle d'Accès (ACLs).



Illustrations des modes de stockage tirées du site RedHat.com - https://www.redhat.com/fr/topics/data-storage/file-block-object-storage

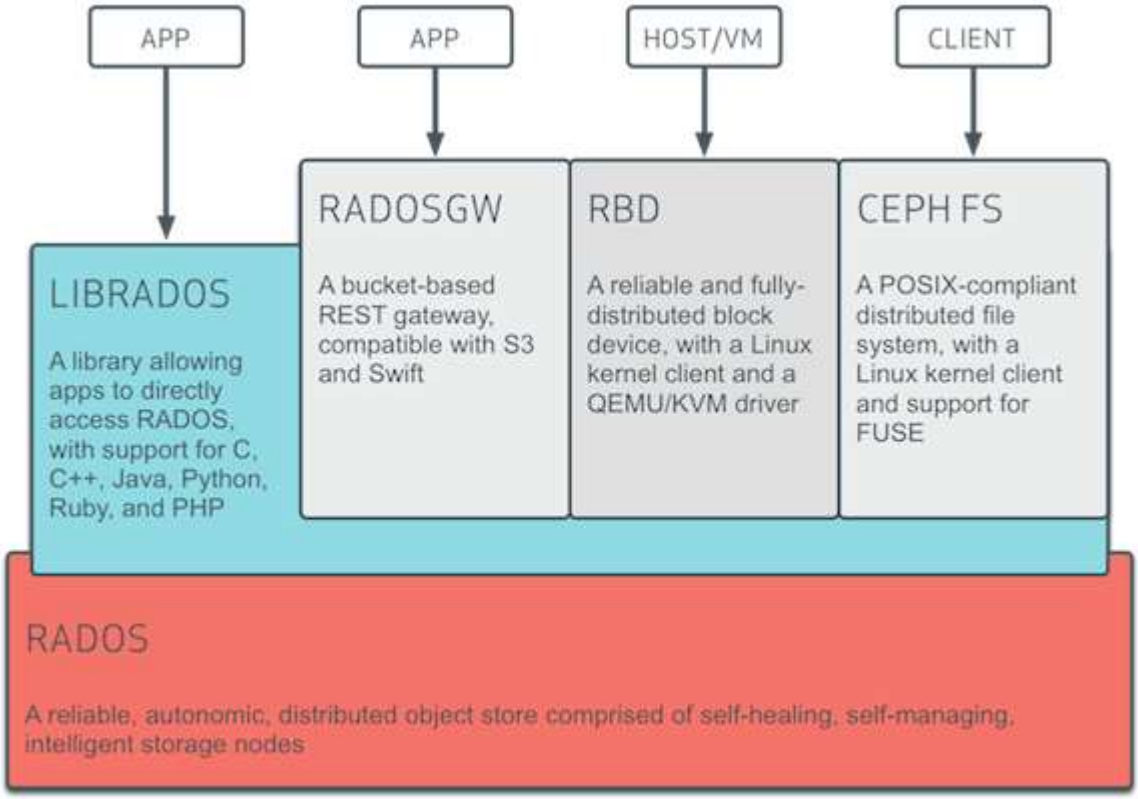


Figure 2 – Illustration de l'architecture de Ceph
https://docs.ceph.com/docs/hammer/architecture/

La figure 2 ci-dessus présente l'architecture globale de Ceph. On peut observer les trois services de stockage précédemment cités ainsi que la couche LIBRADOS qui permet de gérer les accès bas niveau à RADOS. RADOS (Reliable Autonomic Distributed Object Store) est le logiciel sur lequel repose l'ensemble de la solution Ceph, lorsqu'une requête est envoyée par un client, Ceph lit et écrit les données via RADOS. RADOS embarque deux types de démons, qui sont la fondation du cluster. Le premier, Object Storage Daemons (Ceph OSD), stocke les données sous forme d'objet sur les nœuds de stockage du cluster. Le second, Monitor, sert au maintient de la carte du cluster : cette carte nommée « cluster map » permet d'enregistrer la composition du cluster en terme de machines afin de gérer les données.

La figure 3 ci-dessous présente de façon simplifiée le principe de « Cache Tiering » que l'on peut traduire par Cache classifié ou Cache par niveau/étage.

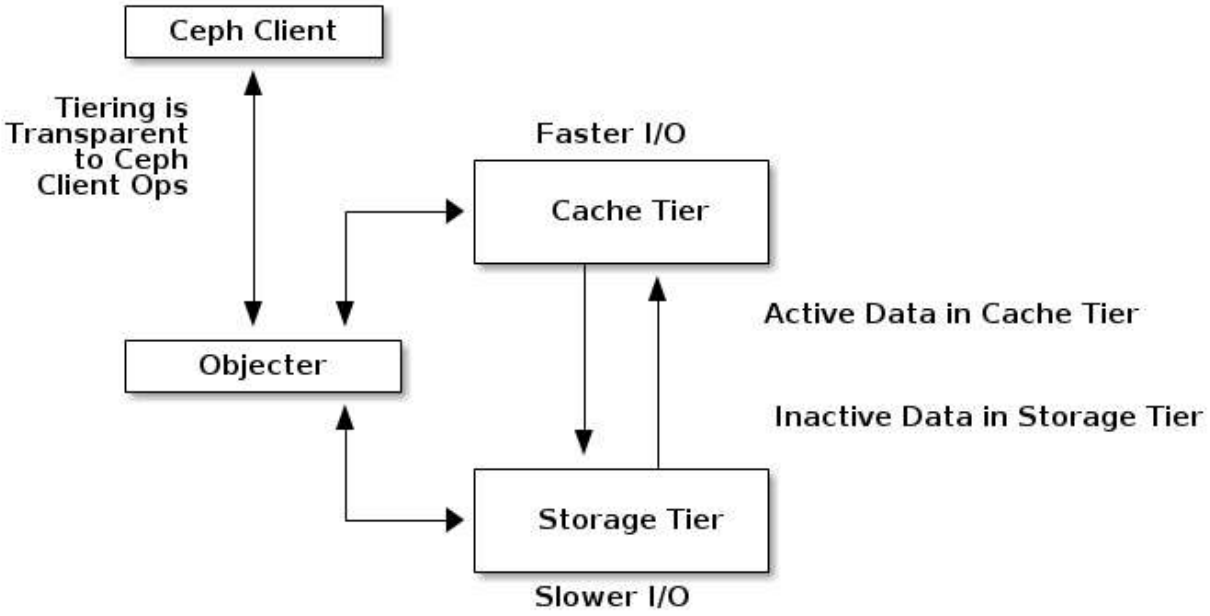


Figure 3 – Illustration du principe de « Cache Tiering »
https://docs.ceph.com/docs/hammer/architecture/#cache-tiering

La fonction de « Cache Tiering » permet aux utilisateurs du cluster Ceph de disposer de meilleures performances en lecture/écriture pour certaines données. Cette fonction se base sur une plage de stockage rapide, employant le plus souvent des disques à état solide (SSDs). En effet, l'objectif est de réserver les disques les plus performants du cluster afin qu'ils agissent comme cache : ainsi les données les plus régulièrement demandées par les utilisateurs se situeront dans la zone la plus rapide. Lorsque Ceph détecte qu'une donnée n'a plus été exploitée depuis un certain temps, il la déplace sur la zone de stockage classique (utilisant des disques durs). Une donnée n'est donc jamais présente dans les deux zones, puisqu'elle est déplacée et non copiée. Cette fonctionnalité, au delà des performances, est totalement transparente pour l'utilisateur final.

Maintenant que nous avons vu l'architecture système sur laquelle repose le concept de Ceph, nous allons nous intéresser à l'architecture réseau. En effet, évoluant sur le principe de système distribué, Ceph se doit d'avoir une structure réseau très solide afin de ne pas s'écrouler lors de montée en charge. Le placement des données est un point névralgique dans le fonctionnement de Ceph, il consiste en différents points :

- Le stockage des objets sur le matériel physique générique, c'est à dire sur des disques classiques n'ayant pas tous les mêmes performances en terme de vitesse de lecture/écriture et de capacités.
- La reconstruction en parallèle des données indisponibles sans nécessité d'ajout de disque, c'est à dire que si un disque du cluster tombe en panne, Ceph est capable de reconstruire les données présentent sur ce disque et de les replacer sans intervention externe.

Pour répondre à ses services, il faut donc une condition réseau optimale pour ne pas ralentir le fonctionnement de Ceph.

La figure 4 ci-dessous représente la configuration réseau permettant de déployer un cluster Ceph dans les meilleures conditions. La principale difficulté, en particulier pour les petites organisations, est de disposer de deux réseaux physiques distincts. Également, Ceph fonctionne uniquement sous forme de cluster, c'est à dire qu'il est impossible de le faire fonctionner correctement sans un minimum de 2 machines physiques (principe du système distribué).

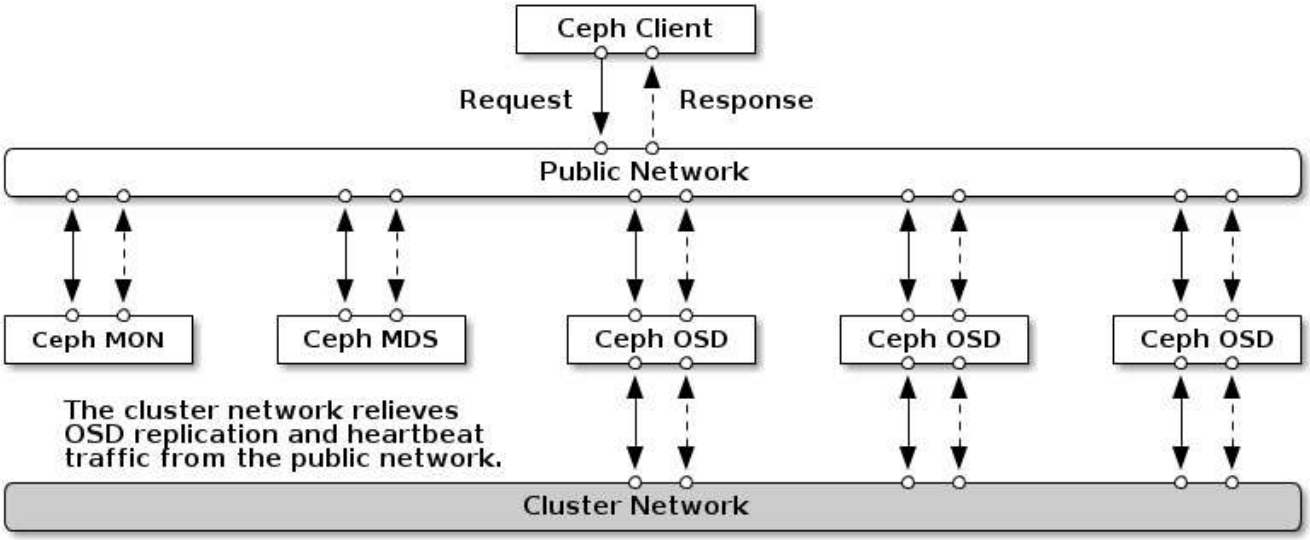


Figure 4 – Illustration de la construction réseau de Ceph
https://docs.ceph.com/docs/hammer/rados/configuration/network-config-ref/

- MON : Service maintenant une copie des cartes du cluster.
- MDS (MetaData Server) : Service utile à Ceph FS. Assure l'enregistrement des métadonnées Posix.
- OSD (Object Storage Device) : Service de stockage des objets employant les disques locaux des machines sur lequel il s'exécute.

- Public Network : Réseau sur lequel les utilisateurs de Ceph se trouve afin d’avoir accès aux ressources.
- Cluster Network : Réseau entièrement dédié à Ceph afin que les OSD du cluster répliquent les informations entre eux. Ce réseau ne doit surtout pas être accessible aux utilisateurs classiques afin de ne pas subir de surcharge, ce qui entraînerait une diminution des performances globales de la solution.

Maintenant que nous avons les notions sur le fonctionnement global (système & réseau) de Ceph, nous pouvons nous intéresser plus particulièrement à son déploiement.

3 - L'Étude

3.1 - Déploiement de Ceph

Nous allons maintenant nous intéresser à un déploiement de Ceph sur sa dernière version stable : « Ceph Octopus ». Nous allons pour cela utiliser une méthode automatisée, nommée Ceph Deploy. Il existe une méthode de déploiement se basant sur Ansible, et également une méthode de déploiement totalement manuelle.

Dans mon cas, je possède deux serveurs physiques IBM System x3650, tournant sur Ubuntu 18.04LTS, c'est pourquoi j'ai décidé de déployer le cluster dans ce contexte.

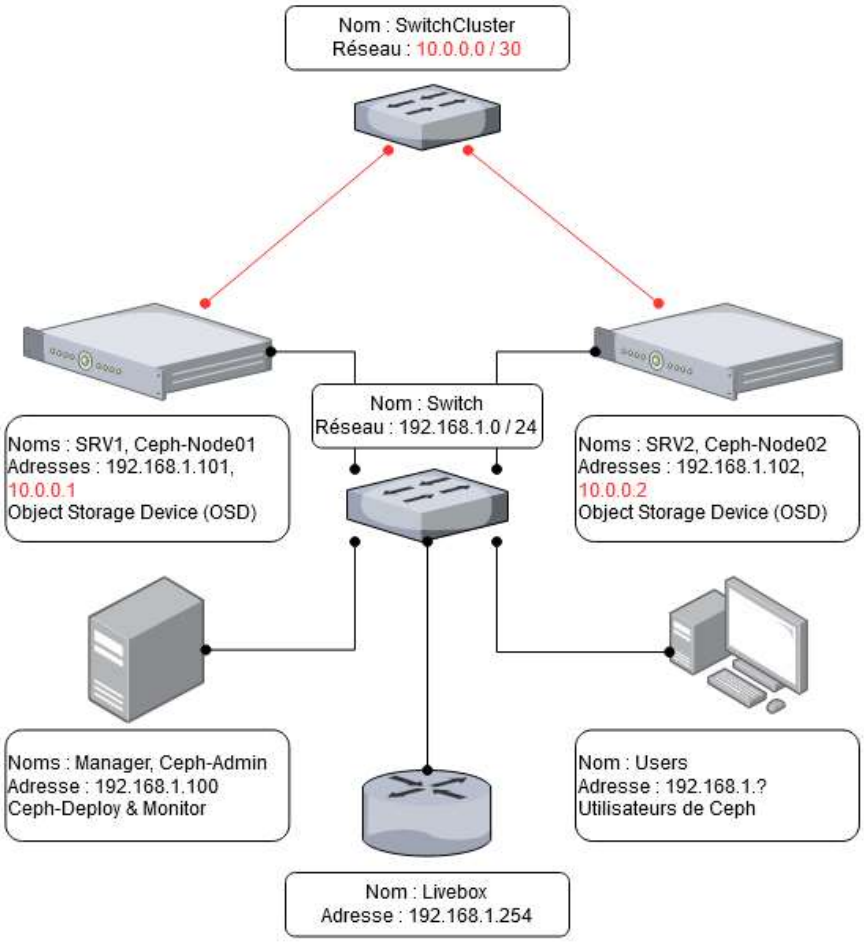


Figure 5 – Schéma de l'infrastructure déployée

La figure 5 ci-dessus présente l'infrastructure que nous allons déployer physiquement. Elle est donc composée de deux nœuds de stockage « Ceph-Node01 » et « Ceph-Node02 », pour ces deux machines j'utiliserai les deux

serveurs IBM System x3650 dont je dispose. Ils comportent chacun 3 disques durs de 500Go fonctionnant à 10K tours/min, mais je n'ai pas de SSD pour ces serveurs donc je ne pourrai pas mettre en place de Cache Tiering efficace. Ensuite j'utilise un troisième serveur « Ceph-Admin » moins puissant afin de déployer et monitorer le cluster. Enfin j'ai mis en place un nouveau réseau, en plus de mon LAN, pour que les deux OSD puissent échanger des données le plus efficacement possible.

Maintenant que nous avons une infrastructure prête à héberger notre cluster, nous pouvons commencer à installer les différents services dont nous avons besoin. *L'objectif est de comprendre la méthode de déploiement du cluster, de ce fait certains détails sont omis pour plus de clarté.*

Sur toutes les machines du cluster, nous mettons à jour les dépendances et nous installons les outils que nous allons utiliser par la suite :

```
lucas@manager:~$ sudo apt-get update
lucas@manager:~$ sudo apt-get install openssh-server python-minimal --yes
```

Ensuite, il faut que nous définissions les adresses et les noms des machines dans le fichier /etc/hosts afin que l'ensemble puisse communiquer correctement à travers le réseau. Cette manipulation permet de faire une résolution de nom (DNS) locale sans avoir à envoyer de requête au serveur (ici ma box Internet). Ce fichier doit être à jour sur l'ensemble des machines également.

```
lucas@manager:~$ less /etc/hosts
...
192.168.1.100 ceph-admin
192.168.1.101 ceph-node01
192.168.1.102 ceph-node02
```

Afin de déployer Ceph sur tous les nœuds, il faut que notre machine « Ceph-Admin » puisse se connecter en SSH sur toutes les machines. Pour ce faire, nous allons créer le même utilisateur sur chacune des machines. Nous allons configurer cet utilisateur pour qu'il puisse exécuter des commandes avec des droits d'administrateur sans devoir utiliser l'attribut « sudo » ni entrer de mot de passe supplémentaire.

```
lucas@manager:~$ sudo useradd -m -s /bin/bash cephadmin
lucas@manager:~$ echo "cephadmin ALL=(ALL:ALL) NOPASSWD:ALL" >> /etc/sudoers.d/
cephadmin
lucas@manager:~$ chmod 0440 /etc/sudoers.d/cephadmin
```

Lorsque notre utilisateur est prêt sur toutes les machines du cluster, nous pouvons créer une paire de clé RSA qui lui sera attribuée. Ainsi notre manager

pourra se connecter entre toutes les machines, en ayant les droits d'administration, et sans devoir entrer de mot de passe. Cette situation va nous permettre de déployer très rapidement l'ensemble des paquets requis pour Ceph dans notre cluster.

```
lucas@manager:~$ su – cephadmin
cephadmin@manager:~$ ssh-keygen
...
```

Nous ne devons pas paramétrer de « passphrase » lors de la génération des clés afin que la connexion aux machines soit directe. On sauvegarde la paire dans le répertoire par défaut qui est dans mon cas « /home/cephadmin/.ssh/ ».

Pour finir, il suffit de téléverser notre clé publique dans toutes les machines du cluster.

```
cephadmin@manager:~$ nano /home/cephadmin/.ssh/config
Host Ceph-Node01
      Hostname Ceph-Node01
      User cephadmin
Host Ceph-Node02
      Hostname Ceph-Node02
      User cephadmin
cephadmin@manager:~$ for i in ceph-node01 ceph-node02 ; do ssh-copy-id $i ;
done
```

On renseigne les mots de passe demandés lors de la connexion aux différentes machines, et une fois que le script a fini son exécution, la configuration initiale des serveurs est terminée.

Pour valider, vous pouvez essayer de vous connecter d'une machine à une autre simplement avec la commande SSH et l'utilisateur cephadmin.

```
cephadmin@manager:~$ ssh ceph-node01
cephadmin@ceph-node01:~$ exit
cephadmin@manager:~$ ssh ceph-node02
cephadmin@ceph-node02:~$ exit
```

Nous pouvons maintenant nous lancer dans la partie sérieuse de l'installation : le déploiement de Ceph en lui-même. Grâce à la configuration de l'utilisateur et de notre paire de clé nous allons pouvoir tout mettre en place depuis « Ceph-Admin ».

Nous commençons par ajouter les dépôts permettant d'installer l'outil « ceph-deploy », puis nous l'installons.

```
root@manager:~# wget -q -O- 'https://download.ceph.com/keys/release.asc' | apt-
key add -
root@manager:~# echo deb https://download.ceph.com/debian-octopus/ $
(lsb_release -sc) main | tee /etc/apt/sources.list.d/ceph.list
root@manager:~# apt-get update
root@manager:~# apt-get install ceph-deploy --yes
```

Nous pouvons maintenant utiliser cet outil pour déployer notre cluster, en commençant par la machine de monitoring, dans mon cas c'est « Ceph-Admin » :

```
root@manager:~# su – cephadmin
cephadmin@manager:~$ mkdir mon-cluster-ceph
cephadmin@manager:~$ cd mon-cluster-ceph
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy new ceph-admin
...
[ceph_deploy.new][DEBUG ] Creating new cluster named ceph
[ceph_deploy.new][INFO  ] making sure passwordless SSH succeeds
[ceph-admin][DEBUG ] connected to host: ceph-admin
[ceph-admin][INFO  ] Running command: ssh -CT -o BatchMode=yes ceph-admin
[ceph-admin][DEBUG ] connection detected need for sudo
[ceph-admin][DEBUG ] connected to host: ceph-admin
[ceph-admin][DEBUG ] detect platform information from remote host
[ceph-admin][DEBUG ] detect machine type
[ceph-admin][DEBUG ] find the location of an executable
[ceph-admin][INFO  ] Running command: sudo /bin/ip link show
[ceph-admin][INFO  ] Running command: sudo /bin/ip addr show
[ceph-admin][DEBUG ] IP addresses found: [u'192.168.1.100']
[ceph_deploy.new][DEBUG ] Resolving host ceph-admin
[ceph_deploy.new][DEBUG ] Monitor ceph-admin at 192.168.1.100
[ceph_deploy.new][DEBUG ] Monitor initial members are ['ceph-admin']
[ceph_deploy.new][DEBUG ] Monitor addrs are ['192.168.1.100']
[ceph_deploy.new][DEBUG ] Creating a random mon key...
[ceph_deploy.new][DEBUG ] Writing monitor keyring to ceph.mon.keyring...
[ceph_deploy.new][DEBUG ] Writing initial config to ceph.conf
```

On continue ensuite avec l'installation de tous les nœuds en spécifiant leurs noms :

```
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy install ceph-node01 ceph-
node02
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy mon create-initial
```

La commande « ceph-deploy install » installe la dernière version stable de Ceph Octopus sur les machines spécifiées, tandis que la commande « ceph-deploy mon » va initialiser le monitoring du cluster que nous avons préparé plus haut lors de la commande « ceph-deploy new ».

```
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy admin ceph-node01 ceph-node02
```

Enfin la commande « ceph-admin » permet de donner à des serveurs le droit d'exécuter des commandes Ceph avec les droits d'administration. Ainsi nous avons le plein accès au CLI depuis n'importe quelle machine.

Enfin nous devons déployer le cluster manager, dans notre cas il se trouvera sur la même machine que le cluster monitor, c'est-à-dire sur « Ceph-Admin ».

```
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy mgr create ceph-admin
```

Nous avons maintenant terminé le déploiement de notre cluster. Nous pouvons vérifier l'état en temps réel de chaque nœud avec la commande suivante :

```
cephadmin@manager:~$ ceph health
HEALTH_OK
cephadmin@manager:~$ ssh ceph-node01 sudo ceph health
HEALTH_OK
cephadmin@manager:~$ ssh ceph-node02 sudo ceph health
HEALTH_OK
```

Pour voir l'état de santé global du cluster, nous devons utiliser la commande suivante depuis la machine de monitoring :

```
cephadmin@manager:~$ ceph status
cluster:
  id:     478e46f1-ae41-5d43-9c8f-72c458ab0a18
  health: HEALTH_OK

services:
  mon: 1 daemons, quorum ceph-admin
  mgr: ceph-admin(active)
  mds: 0 up  {0=a=up:active}, 0 up:standby
  osd: 2 osds: 2 up, 2 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage:   3 GiB used, 2997 GiB / 3000 GiB avail
  pgs:
```

Notre infrastructure est donc opérationnelle. Pour que les utilisateurs puissent exploiter ce cluster, il suffit d'installer des services comme par exemple un serveur de fichier, un serveur de mail, un hyperviseur, etc. L'ensemble des données traitées par ces services seront répliquées dans le cluster. Nous avons donc une infrastructure permettant de rendre hautement disponible l'accès à n'importe quelle donnée.

3.2 - Administration de Ceph

Au delà du déploiement de Ceph et des vérifications régulières de l'état de santé des nodes et du cluster, il est intéressant de se demander comment procéder à l'administration d'un cluster déjà existant. Étant donné la complexité de ce genre de solution, il est impossible de résumer facilement l'administration d'un tel cluster. L'infrastructure que j'ai déployé chez moi ne me permet pas non plus de créer des scénarios réels, c'est pourquoi je me suis plutôt intéressé à la manière d'agrandir le cluster. Puisque notre cluster est fonctionnel, nous allons améliorer sa fiabilité en ajoutant un démon de monitoring supplémentaire, ce qui nous permettra de garder une surveillance sur le cluster si la machine sur lequel se trouve le premier démon tombe en panne. Nous ferons de même pour ce qui est du démon de management.

On commence donc par déployer le démon de monitoring sur « Ceph-Node01 » ainsi cette machine remplira les fonctions d'OSD et maintenant aussi de Monitoring.

```
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy mon add ceph-node01
```

Pour vérifier que le nouveau nœud de monitoring est bien reconnu comme tel, il suffit de vérifier le statut du cluster.

```
cephadmin@manager:~/mon-cluster-ceph$ ceph status
cluster:
  id:     478e46f1-ae41-5d43-9c8f-72c458ab0a18
  health: HEALTH_OK

services:
  mon: 2 daemons, quorum ceph-admin, ceph-node01
  mgr: ceph-admin(active)
  mds: 0 up  {0=a=up:active}, 0 up:standby
  osd: 2 osds: 2 up, 2 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage:   3 GiB used, 2997 GiB / 3000 GiB avail
  pgs:
```

On s'aperçoit rapidement que « Ceph-Node01 » est maintenant reconnu comme un des hôtes du démon de monitoring par le cluster.

Puisque notre « Ceph-Node01 » remplit aussi le rôle de monitoring en plus du rôle de stockage d'objets, nous allons ajouter le rôle de management à « Ceph-Node02 ». Il est très important d'avoir plusieurs démons remplissant la même fonction puisque la stabilité du cluster dépend d'eux. Si un des démons s'arrête pour une raison, et qu'il n'y en a pas au moins un autre pour le remplacer, alors on risque de subir une interruption de service dans notre infrastructure.

Pour ce faire, on reprend notre machine « Ceph-Admin » et on déploie grâce à elle un nouveau démon de management.

```
cephadmin@manager:~/mon-cluster-ceph$ ceph-deploy mgr create ceph-node02
```

Pour vérifier que le nouveau nœud de management est bien fonctionnel, on regarde l'état du cluster :

```
cephadmin@manager:~/mon-cluster-ceph$ ceph status
cluster:
  id:     478e46f1-ae41-5d43-9c8f-72c458ab0a18
  health: HEALTH_OK

services:
  mon: 2 daemons, quorum ceph-admin, ceph-node01
  mgr: ceph-admin(active), standbys : ceph-node02
  mds: 0 up  {0=a=up:active}, 0 up:standby
  osd: 2 osds: 2 up, 2 in

data:
  pools:   0 pools, 0 pgs
  objects: 0 objects, 0 B
  usage:   3 GiB used, 2997 GiB / 3000 GiB avail
  pgs:
```

On s'aperçoit rapidement que « Ceph-Node02 » est maintenant reconnu comme un des hôtes du démon de management par le cluster. On peut aussi voir que contrairement au démon de monitoring, celui-ci est en « standby ». En effet, alors que plusieurs démons de monitoring peuvent être effectifs simultanément au sein du même cluster, pour ce qui est des démons de management, un seul est en fonction alors que les autres sont en attente. Ainsi si le démon de management actif s'arrête, un des démons en standby prendra immédiatement le relai, et aucune interruption de service ne subviendra.

3.3 - Les alternatives à Ceph

Les solutions de stockage ne sont pas excessivement nombreuses sur le marché, surtout lorsque l'on recherche une solution libre. Nous pouvons donc essayer de comprendre pourquoi Ceph domine tant ce marché en le comparant à ses « alternatives » les plus connues. En effet, il n'y a pas d'alternative à proprement parler à Ceph, puisque les solutions que nous allons rapidement survoler ne remplissent pas exactement les mêmes critères. Pourtant ce sont celles qui sont les plus à même d'être considérées comme telles.

Dans le cadre de cette partie, nous allons comparer deux autres solutions à Ceph. La première est MinIO, une solution de stockage cloud basée sur le mode objet. C'est une solution appréciée pour sa compatibilité avec le service de stockage d'Amazon, et donc son intégration dans les infrastructures cloud Amazon Web Services. MinIO traite donc des données telles que des photos, des vidéos, ou encore des logs et des images serveurs, mais pour cela l'ensemble est déstructuré en objet. Enfin la limite de taille d'un objet est de 5Tb.

La seconde solution, et dernière solution, que nous allons comparer est GlusterFS. GlusterFS est un système de fichiers distribués, qui permet de stocker jusqu'à plusieurs pétaoctets (10^{15} octets). Cette solution est appréciée pour sa simplicité, puisqu'elle se base sur un modèle client-serveur avec des serveurs remplissant la fonction de « briques de stockage ». Il suffit ensuite de monter les systèmes de fichiers voulus grâce à la commande « glusterfs » ou plus naturellement « mount ».

La première différence que l'on observe est donc le type de stockage de chaque solution qui diffère : alors que MinIO fait du stockage en mode objet, GlusterFS lui fait du stockage en mode fichier. Quand à Ceph, comme on l'a vu, il est capable de remplir ces deux fonctions, et même plus puisqu'il propose aussi du stockage en mode bloc. Ceph est donc plus polyvalent quand à ses domaines d'applications.

Ensuite, MinIO est considéré comme un « cloud storage » tandis que GlusterFS et Ceph sont considérés comme « File Storage », ce qui signifie que MinIO ne peut pas être exploité pour stocker des données en dehors de celles traitées par l'infrastructure cloud présente.

Un autre point négatif pour MinIO est sa documentation qui est légère. GlusterFS est correctement documenté, puisque comme Ceph, il est maintenu par RedHat. C'est en 2011 qu'il est acquis par ces derniers, puis il est diffusé sous le nom « RedHat Storage Server » avant d'être renommé en 2015 « RedHat Gluster Storage » au moment où RedHat acquit Ceph.

On peut donc conclure que lorsque notre besoin est rempli par Ceph, il est difficile d'utiliser une autre solution. Dans le cas où on veut supporter nos applications Cloud Amazon, MinIO se révèle être une bonne solution. Lorsque l'on veut un système de fichier simple afin de manager un cluster et que Samba ne remplit plus nos attentes, alors GlusterFS fera l'affaire. Mais quand on veut une solution puissante afin de gérer de nombreux besoins différents, aussi bien pour stocker des documents, des images ou des snapshots, alors Ceph sera le choix idéal. Ceph domine le marché du stockage libre, puisqu'il regroupe tous les bénéfices des « concurrents » en une seule solution parfaitement documentée.

3.4 - Résultats

3.4.1 - Performances de la solution

Les performances des solutions distribuées sont toujours difficiles à mesurer, puisque beaucoup de facteurs entre en jeu et influent les résultats. Dans mon cas, et puisque l'infrastructure que j'ai déployée n'est pas dimensionnée pour de la production, je ne vais pas effectuer de tests approfondis de la solution en terme de performances. Pourtant je suis capable à mon échelle d'analyser les potentielles limites que peut rencontrer Ceph s'il n'est pas déployé dans un environnement optimal.

Tout d'abord chaque client Ceph communique directement avec les OSD, les MON et les MDS, ce qui entraîne une grande charge sur le réseau en direction du cluster. Il faut donc pouvoir répondre en conséquence et fournir une solution réseau optimale (lien 10Gb/s afin d'absorber le flux constant). Au delà de la configuration matérielle de l'infrastructure réseau, il faut aussi analyser les besoins qu'aura le cluster pour définir les VLANs et les principes de routage/filtrage.

Ensuite, d'un point de vue sécurité, on peut définir des contrôles d'accès au niveau des pools ce qui complexifie la gestion des droits. En effet nous avons plutôt l'habitude de gérer les accès à un niveau plus bas, c'est à dire directement auprès des volumes (disques). Cette gestion des pools est donc un point crucial, surtout si pour optimiser les performances réseau il est nécessaire d'ajouter de la QoS. Il pourrait donc être intéressant de baser la création des pools sur un aspect « géographique » et regrouper les disques que l'on souhaite contrôler dans les mêmes pools et sur les mêmes machines.

Enfin si la mise en place du cluster peut être une tâche difficile dans le cas d'une infrastructure complexe (multi-cluster par exemple), l'utilisation de ce dernier reste au contraire très simple. Son administration est aussi plus agréable puisque la sécurité des données est intrinsèque, puisque c'est l'objectif principal de la solution. De ce fait, on jouit d'une tolérance aux pannes très élevée avec une gestion automatique des pertes de données. En cas de panne ou dysfonctionnement de l'un des OSD, une détection par les démons de monitoring ainsi que des autres OSD va exclure temporairement le fautif du cluster. Si l'OSD fautif n'est pas réapparu au delà d'une courte période (5 minutes par défaut), un processus de reconstruction automatique est engagé, consistant à dupliquer les données impactées sur des OSD de substitution.

La solution Ceph permet donc une approche différente du stockage, avec une utilisation ouverte à travers tout le réseau. Les questions de la sécurité et de la disponibilité étant presque totalement remplacées par le débit disponible pour le cluster.

3.4.2 - Conclusion

On peut donc en conclure que la solution Ceph est fonctionnelle, et suffisamment robuste pour répondre à de nombreux besoins. La communauté regroupée autour du projet est importante et réactive, ce qui permet de trouver rapidement des indications au sein des forums qui traitent le sujet. La documentation officielle est également un gros point fort. Néanmoins il faudra s'affranchir d'un certain investissement matériel afin de posséder des machines et des réseaux capables de soutenir la solution et de la faire persister et évoluer dans le temps.

Index

Lexique

ACL : « Access Control List » ou liste de contrôle d'accès permettant la gestion fine des droits d'accès à des ressources informatiques (réseaux, fichiers, serveurs, etc).

Ansible : Plateforme logicielle permettant l'exécution de tâches automatiques telles que des configurations ou des déploiements de machines.

Cache Tiering : Principe de cache s'appliquant à des données en fonction de leurs activités.

Ceph OSD : Démon logiciel de Ceph qui interagit avec les OSDs. A ne pas confondre avec « OSD ».

CLI : « Command Line Interface » ou interface en ligne de commandes est une interface homme-machine permettant la communication entre l'utilisateur et l'ordinateur afin d'effectuer des interactions en mode texte.

Cloud Computing : Expression anglophone utilisée pour désigner l'utilisation de serveurs informatiques distants par l'intermédiaire d'un réseau.

Cluster : Ensemble de serveurs indépendants fonctionnant à l'unisson pour remplir une tâche commune.

CRUSH : Algorithme permettant de déterminer comment stocker et retrouver rapidement une donnée dans un environnement de stockage distribué.

Erasure Coding : Méthode visant à modifier le codage de l'information afin d'apporter une redondance.

Hyperviseur : Logiciel et/ou plateforme de virtualisation.

Infrastructure : Ensemble des équipements économiques ou techniques d'une entité (pays, société, association, etc).

Métadonnée : Type de donnée permettant de définir ou décrire une autre donnée.

NAS : « Network Attached Storage » ou stockage attaché au réseau permettant de stocker des données dans un volume centralisé pour des utilisateurs hétérogènes.

Node : Du français nœud, unité de base d'un cluster, il désigne un des serveurs qui compose le système distribué.

OSD : « Object Storage Device ». Unité de stockage (physique ou logique) correspondant à un nœud de stockage du cluster. A ne pas confondre avec « Ceph OSD ».

POSIX : Famille de normes techniques de programmation.

RAID : « Redundant Array of Independent Disks » ou regroupement redondant de disques indépendants est une technologie permettant de répartir des données sur plusieurs disques afin d'améliorer les performances, la sécurité, et la tolérance aux pannes.

RedHat : Société multinationale d'origine américaine éditant des distributions logicielles Linux. Elle est dédiée aux logiciels open source.

RSA : Algorithme de chiffrement asymétrique permettant d'échanger des données confidentielles à travers des réseaux informatiques.

Samba : Logiciel permettant de partager des imprimantes et des fichiers dans un réseau informatique.

Snapshot : Sauvegarde de l'état d'un système à un instant donné.

UNIX : Famille de systèmes d'exploitation.

Table des figures

Bibliographie

[1] Ceph – https://www.ceph.io/

[2] GlusterFS – https://www.gluster.org/

[3] MinIO – https://min.io/

[4] Stockage distribué : retour d'expérience avec Ceph – https://bit.ly/300l5kC

[5] Setup Ceph Storage Cluster on Ubuntu – https://bit.ly/3gSlnjh

[6] Ceph Solution de Stockage Distribué – https://bit.ly/2z0smG1