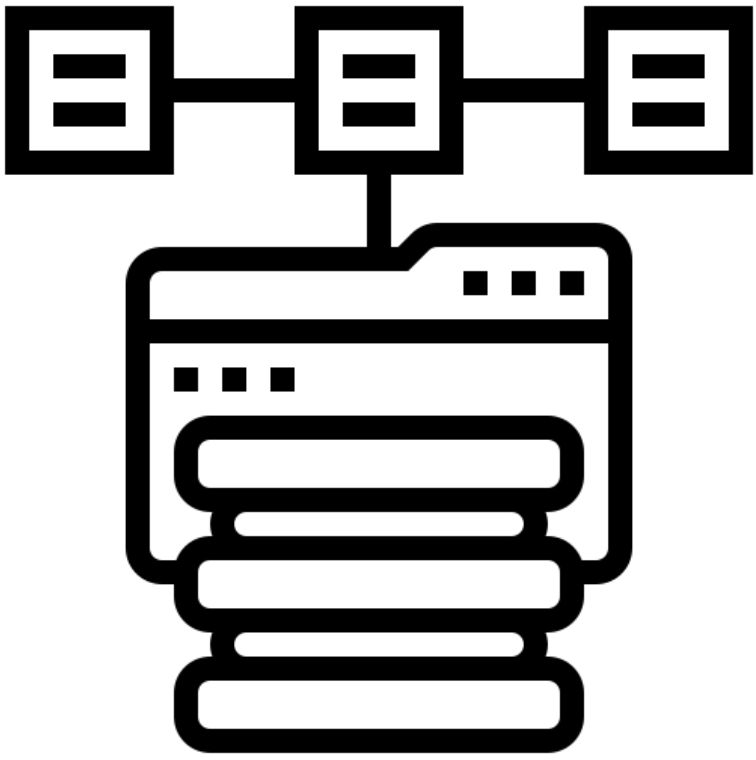
Les systèmes de fichier distribués Ceph et JuiceFS

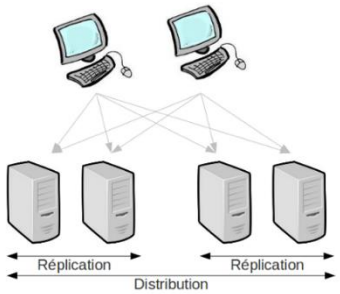
Article Big data



Mr HERMAND

Table des matières

Système de fichier Distribué



Un Système de fichiers distribué (DFS) permet le partage de fichiers à travers le réseau, l'accès à un espace de stockage virtuel unique. L'outils DFS, répartit les données sur les postes clients de façon transparente. Chaque DFS dispose d'une solution afin de garantir une disponibilité des données et une non-perte de ces dernières, pour ce faire le système assure une redondance et une réplication des données. Il est inenvisageable pour un DSF de permettre la perte de données, regardez ce qu'il s'est passé avec OVH l'année dernière et le nombre d'entreprise impactées. Un DFS repose sur l'accessibilité de données par tous de n'importe où. Ils ne peuvent se permettre de perdre leurs données et d'handicaper l'entreprise qui fait appel à leurs services.

Les systèmes répondent aux besoins des entreprises afin de gérer et faciliter l'exploitation de données volumineuses. Ils doivent être puissants afin d'extraire et renseigner des informations à partir des données mais également performent pour gagner du temps lors des recherches et ainsi diminuer le coût généré par ces recherches.

Parmi les solutions DFS existantes, la plus répandue est CEPH et nous décidons de la comparer avec JuiceFS, un DFS récent et encore peu rependu afin de comprendre leurs différences.

Ainsi nous nous demanderons quels sont les différences qui les caractérisent ?

CEPH

Qu'est-ce que c'est ?

CEPH est une solution Open Source de stockage disposant de son propre système de fichiers (CephFS) compatible POSIX (Portable Operating System Interface). CephFS permet le stockage d'un grand nombre de données sur plusieurs composants de son réseau mais aussi les sauvegarder à des emplacements de stockage physiquement différents.

Ce système offre une sécurité et Fiabilité des données. Ceph permet une sauvegarde redondante des donnés sur plusieurs systèmes. Il est possible d'utiliser une ou plusieurs pools de stockage RADOS (Reliable Autonomic Distributed Object Store) avec différentes configurations.

RADOS est un service de stockage d'objet open source. Le système Ceph RADOS se compose généralement d'un grand nombre d'ordinateurs/serveurs, également appelés nœuds de stockage. Un nœud est l'appellation d'un ordinateur appartenant à un Cluster, des ordinateurs connectés ensemble. Un système Ceph RADOS sert principalement de système de stockage autonome.

Ceph dispose de plusieurs modules dont :

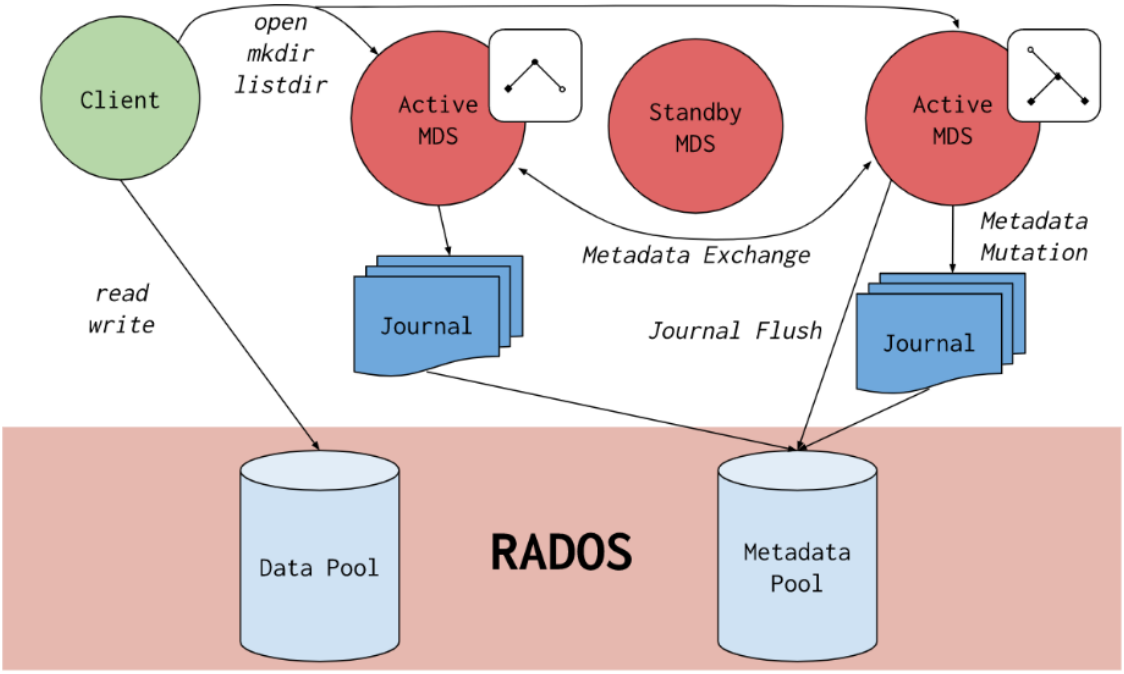
- Ceph-mon, permet la supervision du cluster
- Ceph-mds, qui est un serveur de metadonnées
- Ceph-osd, un périphérique de stockage. Il stock le contenu des fichiers sur un systeme de fichier local.

Le serveur de métadonnées (MDS) Ceph stocke des métadonnées pour CephFS. Ils permettent aux utilisateurs du système de fichiers d'exécuter des commandes tels que la recherche de fichiers (commandes ls, find, like), sans imposer une charge importante sur la grappe de stockage Ceph.

Ces métadonnées de fichier sont donc stockées dans le pool RADOS afin de séparer des données de fichier dans un cluster redimensionnable de MDS afin de pouvoir effectuer des charges de travail plus élevé.

Pour fonctionner Ceph a besoin d'au moins deux réserves RADOS, une pour les données et une pour les métadonnées. Les données sont stockées sous formes d'objets et sont répliqués n fois par réserves.

Les clients (applications utilisant CephFS) ont un accès direct en lecture et écriture de bloc de données fichier RADOS. Ils envoient des demandes de métadonnées au MDS actif. En retour, les serveurs de métadonnées leurs fournissent ces informations avant de les mettre en cache afin de réduire les demandes au pool de métadonnées de sauvegarde. Les métadonnées sont répliquées entre les MDS actifs et fusionnent les mutations de métadonnées dans un journal puis sont envoyés vers un pool de sauvegarde.

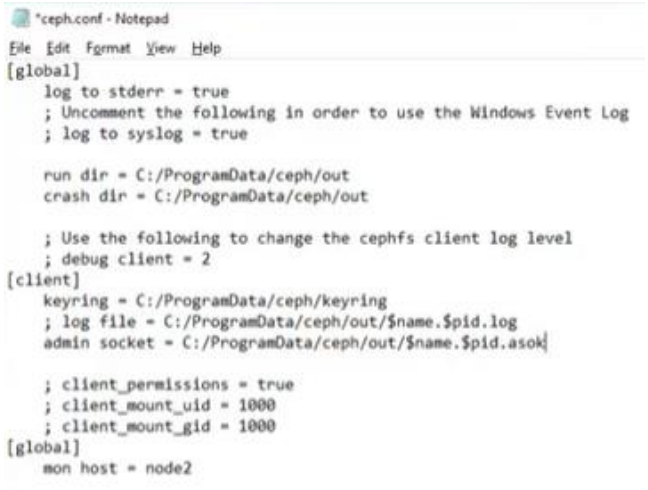


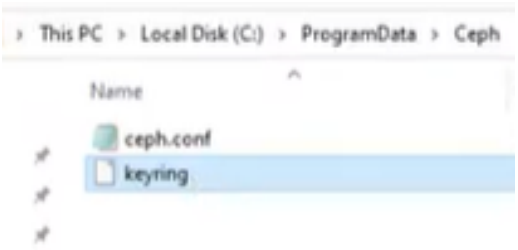
Démonstration

En termes de prérequis, il est nécessaire d'avoir au moins 3 machines pour la création d'un cluster Ceph : une machine avec un minimum de puissance (8GB RAM) pour faire office d'OSD (Object Store), deux autres machines moins puissante (2GB RAM) pour faire office de Monitor et de MDS (MetaData Store).

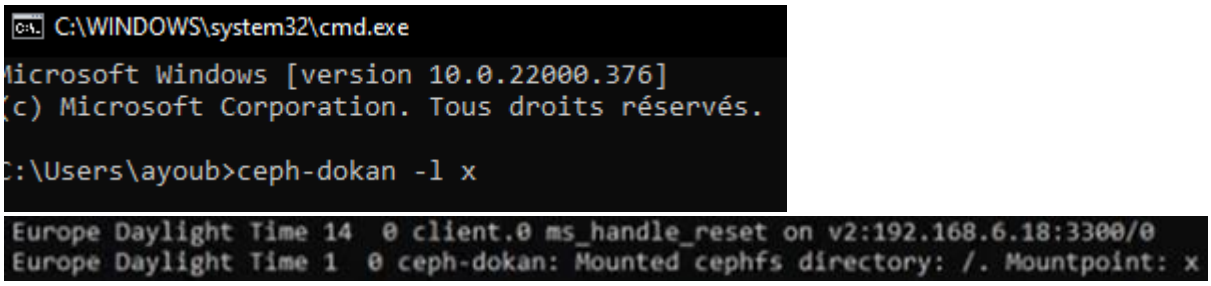
Pour ce qui concerne Windows, il existe un installeur simple pour Ceph téléchargeable depuis la plateforme CloudBase.

Après installation il sera nécessaire de créer son propre fichier de configuration et son fichier keyring contenant les clés issues de son cluster, comme tel et le stocker dans le dossier C:/ProgramData/Ceph :





Par la suite pour permettre l'accès au cluster depuis notre machine Windows 10 il suffit d'exécuter cette commande dans un cmd :

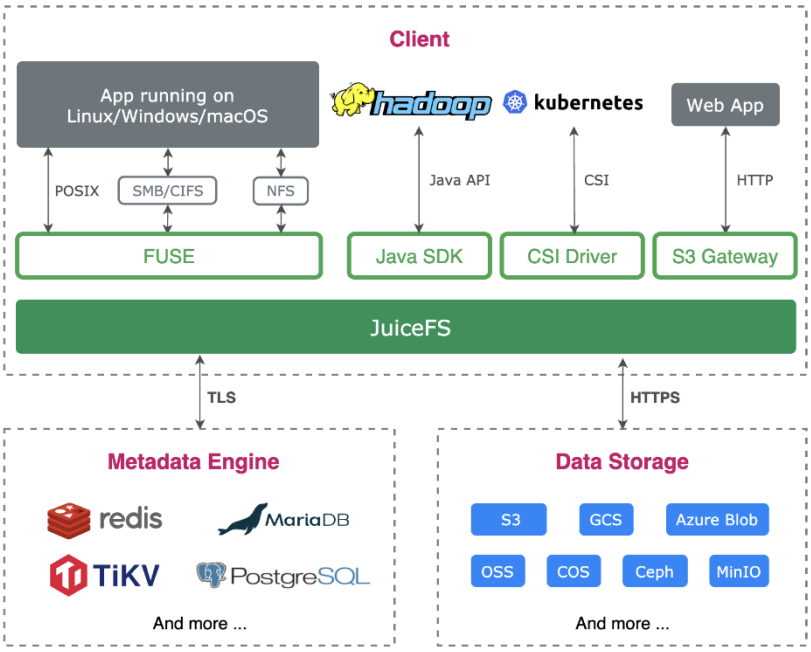


JuiceFS

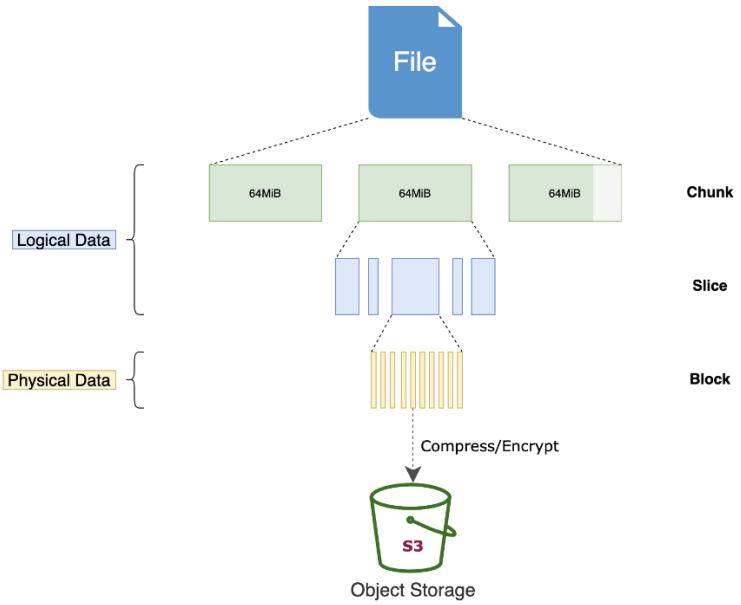
Qu'est-ce que c'est ?

JUICEFS est un système de fichiers Open Source de stockage qui se compose de trois parties distinctes :

- JuiceFS Client qui coordonne le stockage et le moteur de métadonnées. Il implémente également l'interface système de fichiers telles que Hadoop
- Data Storage qui prend en charge le stockage local ou bien le stockage d'objets dans le cloud, HDFS …
- Metadata Engine, les métadonnées sont des données permettant de décrire le fichier nous y retrouvons le nom, la taille, l'heure de création et bien d'autres prenant en charge redis, MySQL et bien d'autres.



Les données stockées sous JuiceFS sont divisées en morceaux de taille fixe selon des règles, les morceaux sont ensuite stockés dans votre stockage d'objets et les métadonnées dans une base de données définie. Chaque donnée est divisée en morceau de 64Mo qui sont eux-mêmes divisé par la suite en tranches logiques en fonction de la situation. Ces tranches sont à leurs tour divisé en un ou des blocs logiques lors de l'écriture dans le magasin d'objet, un bloc est égal à un objet.



Cette solution peut être utiliser dans le domaine du Big Data Analytics grâce à sa compatibilité avec HDFS ce qui permet une intégration transparente avec les moteurs informatiques Hive ou Spark, c'est un espace de stockage infinie avec des coûts d'exploitation et de maintenance minime.

Démonstration

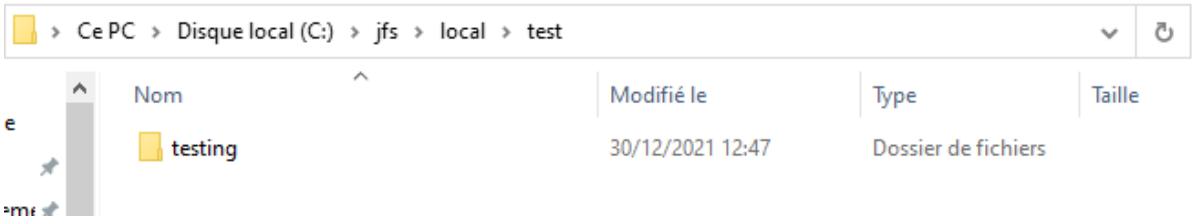
Prérequis choisir une base de données compatible avec JuiceFS. Dans notre cas nous avons choisi de faire un docker redis pour éviter de devoir installer directement sur nos machines une base de données. Nous avons aussi choisi cette option pour des raison de performance, nos ordinateurs. Dans une optique de faire une démonstration simple nous avons choisi de faire un object store local mais il existe plusieurs options pour avoir accès par tout aux données (voir la présentation ci-dessus).

Tout d'abord il faut créer un container docker de la base :

```
docker run -d --name redis
    -v redis-data:/data
    -p 6379:6379
    --restart unless-stopped
    redis redis-server --appendonly yes
```

Nous créons ensuite le système de fichier en informant uniquement la base de données, le disque local sera pris comme stockage :

```
juicefs format redis://localhost:6379/1 test
```



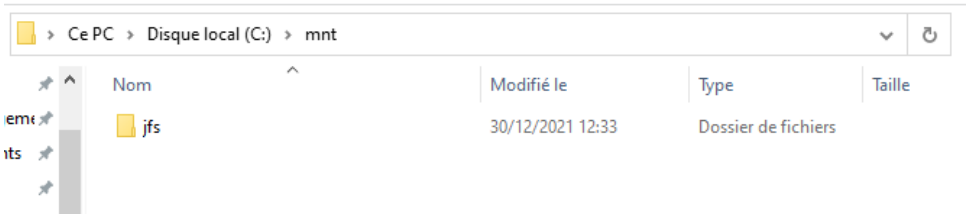
Voici la ligne de commande à faire si vous utilisé un object storage :

```
juicefs format
    --storage nameStorage
    --bucket lienStorage
    --access-key user
    --secret-key password
    redis://urlRedis:6379/1
    myjfs
```

Et ensuite monter le système de fichier avec cette base :

```
juicefs mount redis://0.0.0.0:6379/1 /mnt/jfs
```

Pour finir dans votre dossier C:/ va apparaitre un dossier mnt (le nom donné au système) où vous pourrait y mettre toutes vos données.



Comparaison

Tout d'abord les deux DFS utilisent la même architecture qui est de séparer les données et les métadonnées.

Un des aspects qui nous plaît le plus chez JuiceFS c'est son ouverture aux autres, nous choisissons où vont être stocké nos métadonnées et nos données, nous ne sommes pas obligés d'utiliser Ceph-mds et Ceph-osd. Si nous le souhaitons en utilisant JuiceFs on peut utiliser Ceph pour stocker nos objets mais l'inverse est impossible. Nous pensons que cela est dû à l'époque de création du DFS, il y a encore quelques années nous n'avions pas cette richesse d'outils et ils n'étaient pas toujours viable. De plus les mentalités n'était pas pareil, aujourd'hui nous voulons avoir le choix de personnalisé et utiliser un outil plutôt qu'un autre.

Les deux systèmes divisent les fichiers en morceau, CephFS divise en object_size (4Mio), chaque morceau correspond à un RADOS. Alors que Juice fait un sur découpage des données qui entraine donc une baisse des performances.

Le chiffrement des données, Juice crypte et décrypte les données au moment du téléchargement alors que CEPH le fait sur la couche transport du réseau. Tous deux crypte les données pour éviter un vol de données ou une écoute de celle-ci par des personnes malveillantes.

Conclusion

Ce projet nous as permis de découvrir systèmes de fichier distribué, cette façon de partager les fichiers est super intéressant pour toute entreprises qui traitent ou de stocker des montagnes de données. Il est donc possible de se demander pourquoi ce type solution reste un marché de niche alors qu'avec OneDrive ou Google vous pouvez créer vous-même votre propre espace de partage.

La réponse est simple : sa mise en place. Pour une petite entreprise ou des particuliers comme nous c'est long est compliqué. C'est pourquoi nous comprenons totalement les entreprises qui préfèrent une solution comme CEPH qui regroupe tout dans un seul endroit, pas besoin de choisir et de configurer sois même le stockage objet ou des métadonnées, ça facilite grandement le processus.