Openstack et Ceph : des atomes crochus

David Grimbichler
david.grimbichler@uca.fr

Université Clermont Auvergne, Direction Opérationnelle des Systèmes d'Information, Mésocentre

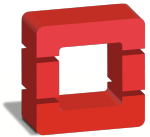
Mercredi 8 juin 2022

Sommaire

Sommaire

Openstack (1)

- lancé en 2010 (NASA + Rackspace Hosting)
- ensemble de logiciels *open source*
- permettant de déployer des infrastructures de *cloud computing* (IaaS)
- contrôle les ressources des machines virtuelles (calcul, stockage, réseau)
- porté par la Fondation OpenInfra formée de grands acteurs (Canonical, Red Hat, SUSE, Dell, IBM, Yahoo, Blizzard Entertainment, ...)



Openstack : Fonctionnement (2)

Fonctionnement :

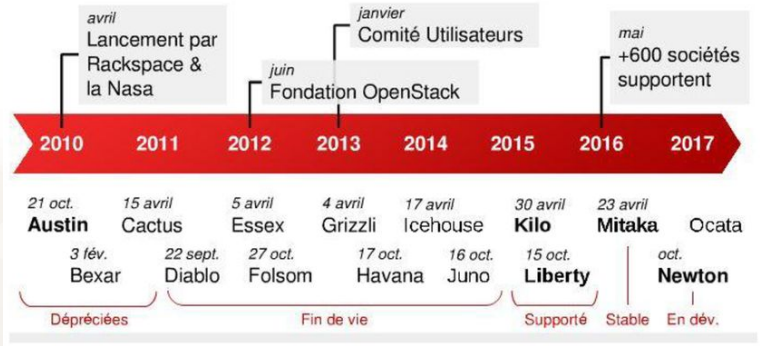
- architecture modulaire composée de plusieurs projets (Nova, Keystone, Neutron, ...)
- les modules ne sont pas tous nécessaires
- chaque module peut être configuré selon les besoins
- abstraction des ressources (virtuelles)
- forte utilisation d'interfaces de programmation d'applications (API)

OpenStack ne virtualise par les ressources, mais utilise ces dernières pour construire des clouds. (`https://www.redhat.com/fr/topics/openstack`)

Openstack : Cycle de vie (3)

Cycle de vie :

- 2 releases par an (peut-être difficile à suivre...)
- 1 version stable
- 2 versions sous support



Openstack : Packaging (4)

Packaging :

- Disponible en RPM, DEB, ...
- Ubuntu : packaging suit de près le développement ; Canonical fournit la Ubuntu Cloud Archive (Openstack stable sur Ubuntu LTS)
- Debian : Openstack intégré dans les dépôts officiels
- Red Hat : très actif dans la communauté, propose une distribution avec support (RHOSP)
- SUSE : propose une distribution avec support (SUSE OpenStack Cloud)

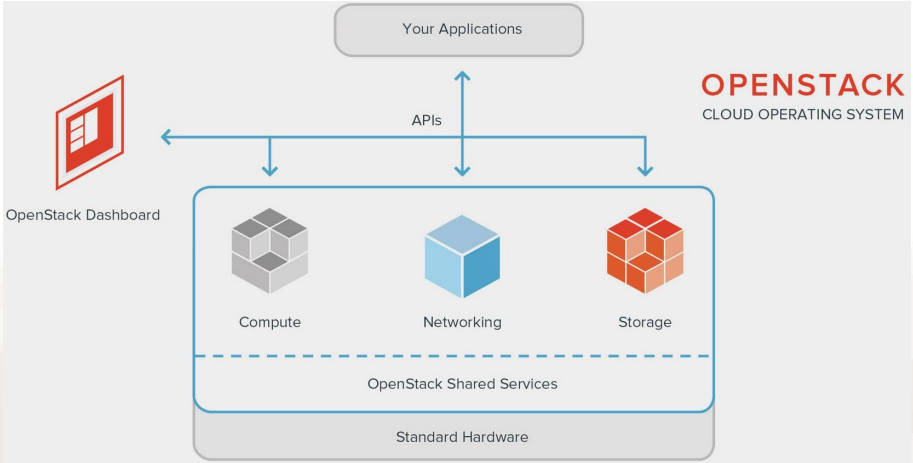
Openstack : Technologies (5)

Technologies :

- Approche micro-services
- Une base de données par service
- Services sans états (state-less)
- Utilisation privilégiée de commandes par messages (AMQP) : composants indépendants qui dialoguent entre eux via une file d'attente
- API synchrone utilisée (presque) seulement pour accéder aux composants

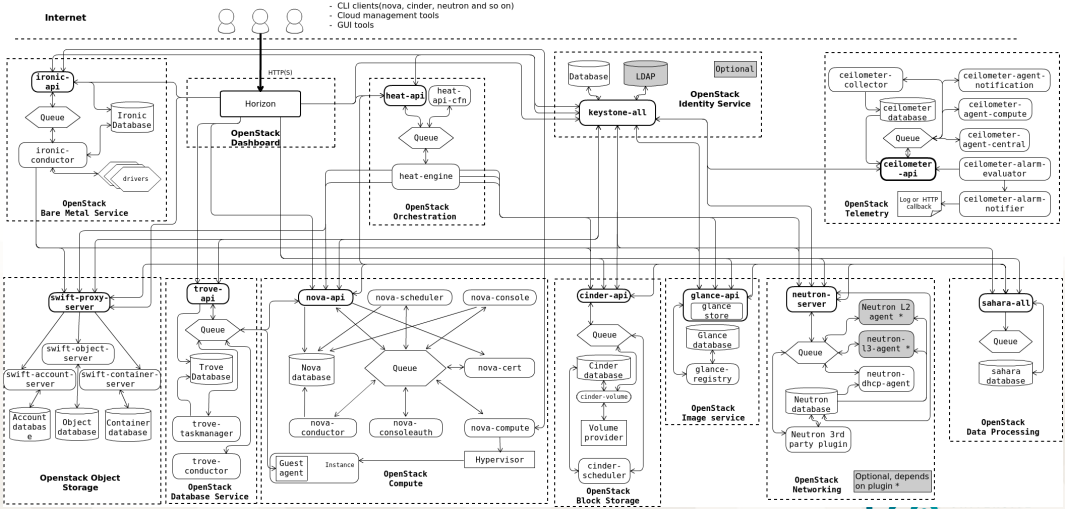
Openstack : Composants principaux (6)

Composants individuels et autonomes



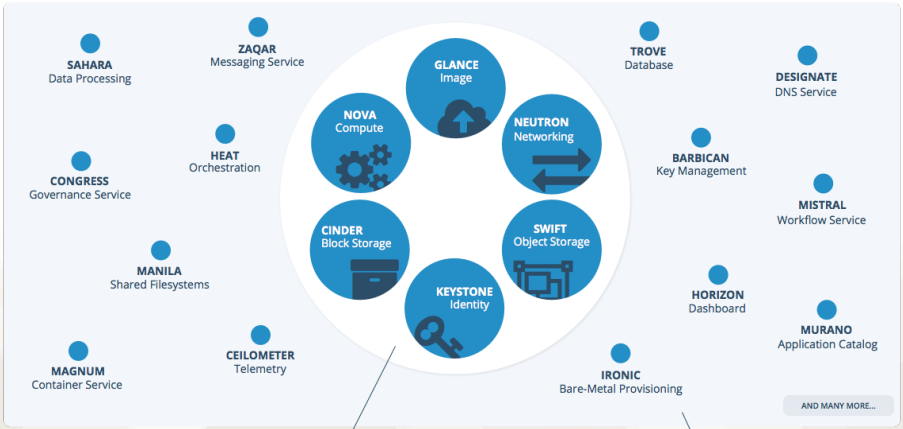
Openstack : Composants détaillés (7)

https://docs.openstack.org/install-guide/get-started-logical-architecture.html

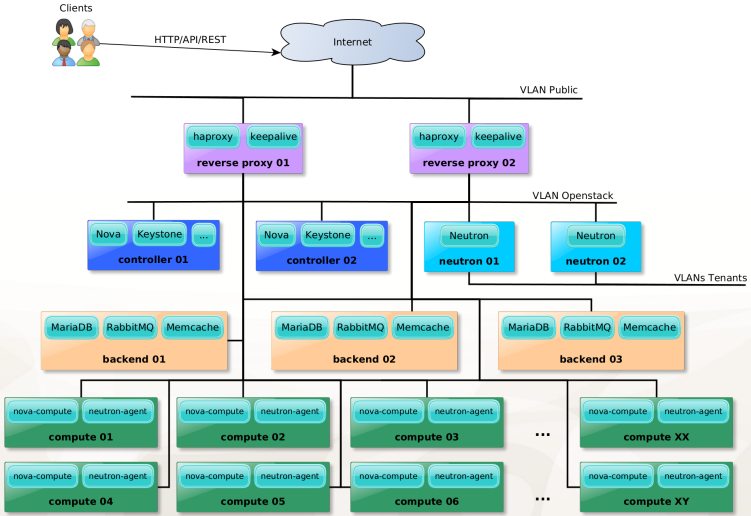


Openstack : Composants essentiels (8)

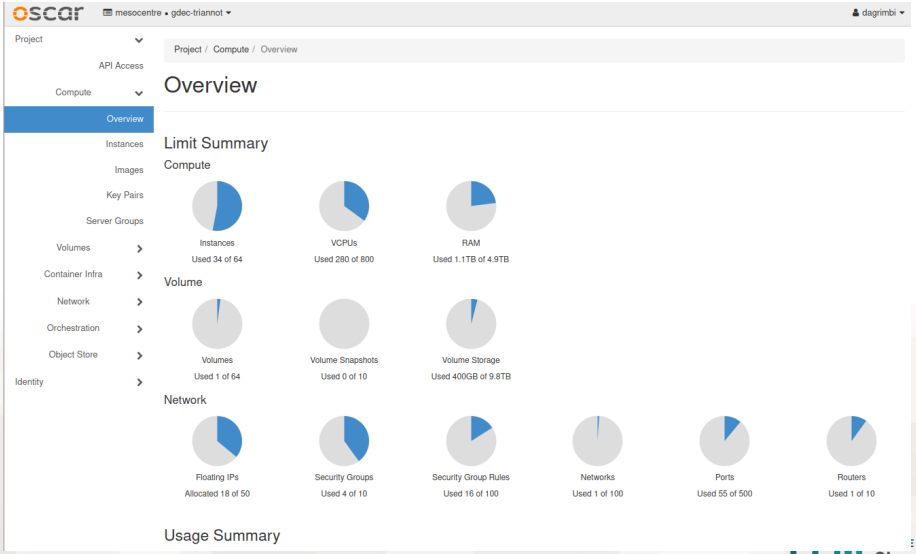
https://docs.openstack.org/upstream-training/



Openstack : Exemple d'architecture (9)



Openstack : Exemple Dashboard utilisateur (10)



OpenStack vs Proxmox

Openstack et Proxmox,
même combat ?

	Virtualisation (Proxmox)	Cloud computing (OpenStack)
Définition	Technologie	Méthodologie
Objet	Créer plusieurs environnements simulés à partir d'un même système physique	Regrouper et automatiser des ressources virtuelles pour une utilisation à la demande
Utilisation	Fournir des ressources en paquets à des utilisateurs spécifiques pour une tâche spécifique	Fournir des ressources variables à des groupes d'utilisateurs pour diverses tâches
Configuration	À partir d'une image	À partir d'un modèle
Durée de vie	Années (long terme)	Heures ou mois (court terme)
Coût	Dépenses d'investissement élevées, dépenses d'exploitation faibles	Cloud privé : dépenses d'investissement élevées, dépenses d'exploitation faibles.
Type d'architecture	Client unique	Multi-client

Sommaire

Ceph (1)

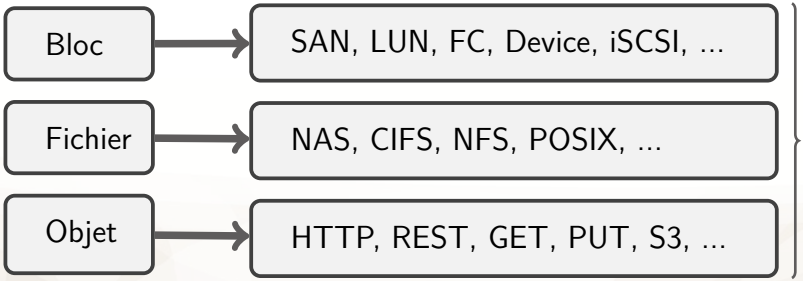
- Solution open source
- plateforme unifiée
- de stockage distribué
  - pas de SPOF, éléments redondés en mode multi-actif
  - extensible jusqu'à plusieurs exaoctets
  - conçu pour s'auto réparer
  - conçu pour réduire les coûts d'exploitation
  - évolution dynamique (scale-out)
- objets, blocs, fichiers
- tolérante aux pannes
- fonctionnant sur du matériel standard
- création : thèse de Sage Weil en 2007 (algorithme CRUSH)

Ceph : stockage distribué (2)

1 baie SAN
croissance verticale

Serveurs *commodity hardware*
croissance horizontale

Ceph : bloc, fichier, objet (3)



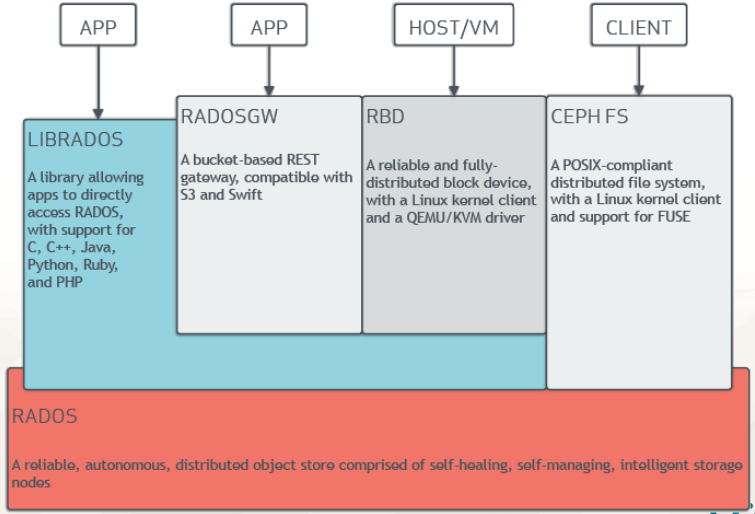
RADOS

Reliable
Autonomic
Distributed
Object
Store

moteur
unifié de
Ceph

Ceph : RADOS (4)

https://docs.ceph.com/en/latest/architecture/

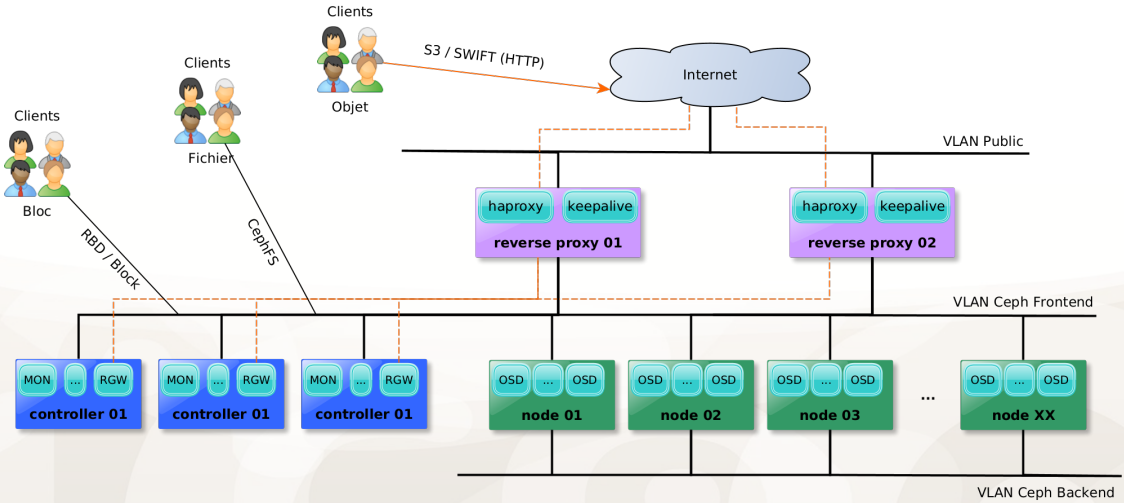


Ceph : services de base (5)

Principalement deux services de base :

- OSD : Object Storage Device
  - Service de stockage des objets
  - Plusieurs dizaines dans un cluster
  - 1 OSD par disque (1 par SSD, 1 par HDD, ...) local du serveur
  - Sert les données **directement** aux clients
  - Effecttue les tâches de réplication et de récupération par peering intelligent
- MON : Monitor
  - Quelques uns, nombre impair (quorum)
  - Ne sert pas de données aux clients
  - Gère les membres et l'état du cluster
  - Maintient une copie des cartes du cluster
  - Fournit un consensus pour les décisions distribuées

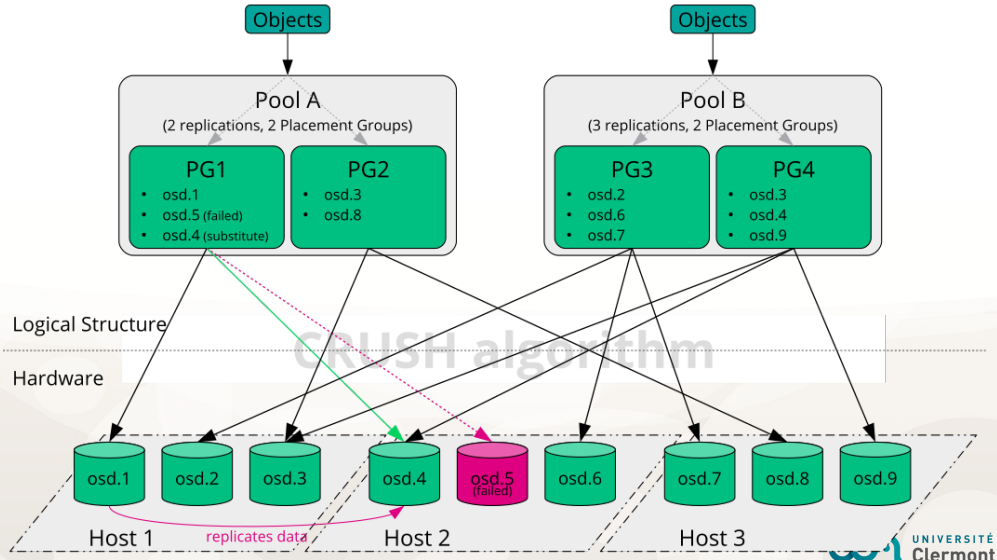
Ceph : exemple d'infrastructure (6)



Ceph : placement des objets dans RADOS (7)

- Stockage :
  - sur du matériel standard, sans contrôleur RAID
  - donc meilleure évolution du matériel (différentes capacités, vitesses, technologies)
  - pas besoin de disques *hot-spare*
- Pool : groupe logique pour stocker les objets
  - résilience : type de réplication et nombre de réplicats
  - contient les *Placement Group* (PG)
  - applique une règle CRUSH : détermine la distribution des objets en fonction de l'infra (OSD, serveur, chassis, rack, PDU, allée, pièce, datacenter, région)
- Placement Group : fragments d'un pool
  - agrégat d'objets qui permet de déterminer rapidement leurs états
  - monitorent le placement d'objets (données, métadonnées)
  - composés d'un groupe d'OSD qui se surveillent (nombre d'OSD = nombre réplicats)
  - chaque PG a un OSD primaire désigné ; un objet RADOS atteint en 1er cet OSD, puis est répliqué ver les autres OSD du PG.

Ceph : exemple stockage objet (8)



Ceph : parenthèse stockage S3 (9)

L'accès S3 au stockage capacitif Ceph : requêtes HTTP, REST, donc idéal pour les applis web. Exemples où le **stockage objet** a tout son sens :

- registry type *Harbor* ou *Artifact* : stockage des images Docker
- système de gestion des cours type *Moodle* : stockage des données
- plate-forme de fouille de données / de gestion de données type *Galaxy*
- sites web permettant l'accès à des grosses quantités de données : bucket avec accès public, intégration directe (ex : site Conservatoire Botanique National Massif Central, 15To d'images de planches d'herbiers scannées, liens S3 directs ; Netflix)
- bibliothèques python : pandas, boto, s3fs, Dask, ...
- entraînements type *deep learning* avec TensorFlow : utilisation lib python `S3fs`, accès en mode stream au dataset sans copie locale. TensorFlow accepte directement

```
dataset = tf.data.TFRecordDataset(['s3://bucketname/42/file1.tfrecord'])
```

... mais usage compliqué dans le monde HPC (souvent besoin de POSIX - cf `goofys`).

Sommaire

Affinités (1)

Besoins d'OpenStack niveau stockage :

- possibilité de passer à l'échelle, comme OpenStack lui-même
- le passage à l'échelle doit se faire indépendamment de Cinder (volumes), de Manila (fichiers), de Swift (objets)

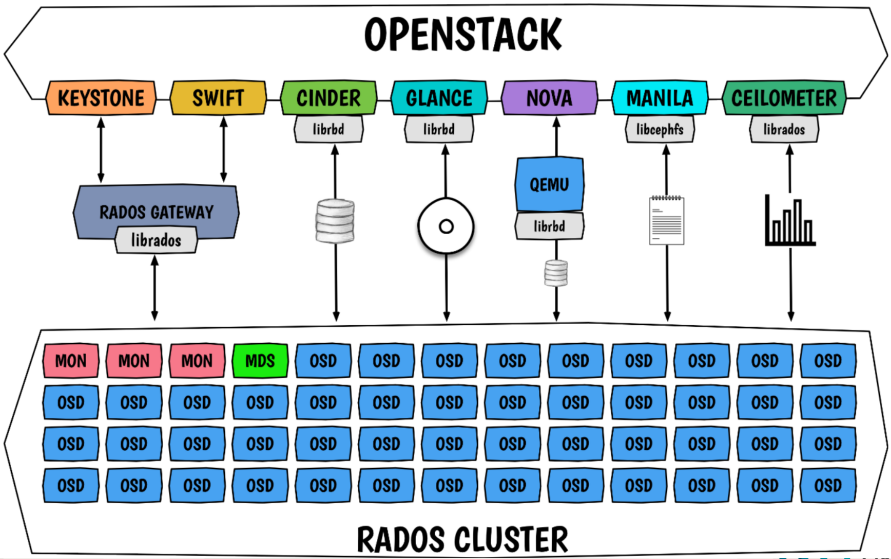
Ceph, coûts réduits :

- tourne sous Linux et pas sur un système propriétaire
- fonctionne sur du *commodity hardware*
- OSD sur compute nodes + MON sur contrôleurs OpenStack (hyperconvergé)

Projets *open source* :

- Ceph et OpenStack sont open source
- rend possible des intégrations fines et des développements inter-projets
- Les constructeurs de solutions propriétaires sont moins enclins à partager leurs informations (secrets industriels à garder), et leur influence est assez limitée dans les communautés open source.

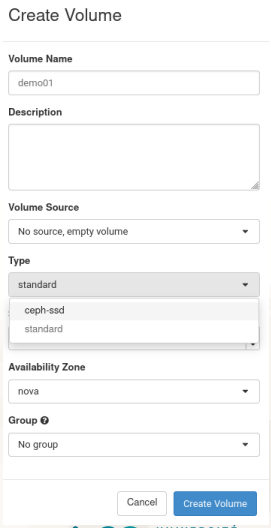
Affinités (2)


OPENSTACK
KEYSTONE
SWIFT
CINDER
GLANCE
NOVA
MANILA
CEILOMETER
librbd
librbd
libcephfs
librados
QEMU
librbd
RADOS GATEWAY
librados
MON
MON
MON
MDS
OSD
RADOS CLUSTER

Affinités : Cinder (3)

Cinder (volumes) :

- mode block, utilisation `librbd`
- mécanisme snapshots : instantané grâce à Ceph
- réplication (sécurisation) : instantané grâce à Ceph
- multi-site : config dans Ceph, indépendant de Cinder
- plusieurs types de volumes, liés à des pools Ceph, visibles par projet
- règles spécifiques sur chaque pool ceph
- volumes non liés à un contrôleur openstack en particulier (haute disponibilité)
- `cinder-backup` : peut envoyer dans du S3, potentiellement multi-site, avec mécanisme deversioning propre au S3.



Affinités : Cinder configuration (4)

Quelle configuration pour Cinder et Ceph ?

- Côté Ceph :
  - Créer un pool Ceph qui va accueillir les volumes (`ceph osd pool create ...`)
  - Définir sur ce pool les règles que l'on choisit (réplication, multi-site, CRUSH, ...)
  - Créer un keyring d'authentification pour Cinder (`ceph auth ...`)
- Côté Cinder :
  - Récupérer un fichier `ceph.conf` minimaliste
  - Récupérer le keyring créé pour Cinder
  - Adapter le fichier `cinder.conf` :

```
enabled_backends = rbd01
[rbd01]
volume_driver = cinder.volume.drivers.rbd.RBDDriver
rbd_pool = oscar-volumes
rbd_user = <nom indiqué dans le keyring>
rbd_ceph_conf = /etc/ceph/ceph.conf
rbd_secret_uuid = ...
...
```

Affinités : Cinder en action (5)

```
$ openstack volume create --size=64 --image=centos-7-20220127 volume01
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| attachments         | []                                   |
| availability_zone   | nova                                 |
| bootable            | false                                |
| consistencygroup_id | None                                 |
| created_at          | 2022-06-07T12:43:02.190444           |
| description         | None                                 |
| encrypted           | False                                |
| id                  | fe582529-fd49-4555-9acf-b2bdefc7cb78 |
| size                | 64                                   |
| multiattach         | False                                |
| name                | volume01                             |
[...]
```

```
# rbd info oscar-volumes/volume-fe582529-fd49-4555-9acf-b2bdefc7cb78
rbd image 'volume-fe582529-fd49-4555-9acf-b2bdefc7cb78':
size 64 GiB in 8192 objects
order 23 (8 MiB objects)
snapshot_count: 0
id: 8cb8145283c881
block_name_prefix: rbd_data.8cb8145283c881
format: 2
features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
parent: oscar-images/f85e0272-8e27-43a4-be8c-9d997d067732@snap
overlap: 8 GiB
[...]

# rbd du oscar-volumes/volume-fe582529-fd49-4555-9acf-b2bdefc7cb78
NAME                                        PROVISIONED  USED
volume-fe582529-fd49-4555-9acf-b2bdefc7cb78       64 GiB   0 B
```

Affinités : Et pour les autres composants de OpenStack ? (6)

Glance :

- même remarques que pour Cinder
- même type de configuration :

```
enabled_backends = standard:rbd
[standard]
store_description = "ceph backend"
rbd_store_pool = oscar-images
rbd_store_user = <keyring user>
rbd_store_ceph_conf = /etc/ceph/ceph.conf
```

Keystone et Swift (objets) :

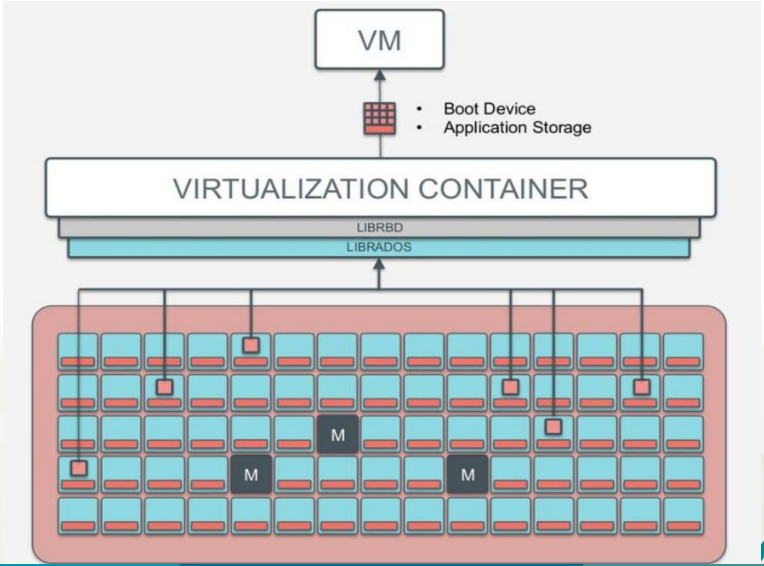
- besoin mode objet
- radosgw pour swift et S3
- utilisation librados

Nova (compute) :

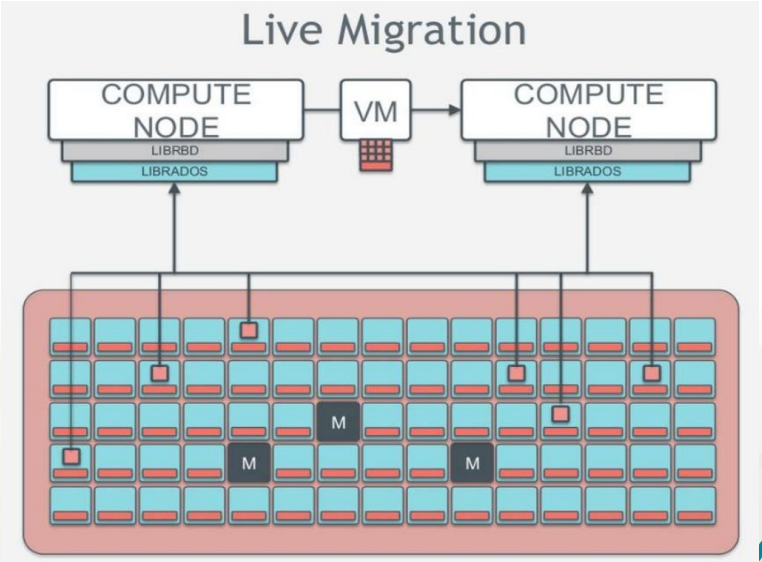
- configure Qemu via libvirt
- driver rbd qemu (librbd)
- disque de l'instance dérive de l'image : copy-on-write, pas besoin de copie sur l'hyperviseur, instanciation très rapide (démo ?)

```
qemu-kvm -name guest=instance-00001fe7
  -blockdev { "driver":"rbd",
    "pool":"oscar-instances",
    "image": "ff2df0c7-2d50-4020-841e-c236
    ... } -netdev tap,fd=89,id=...
```

Affinités : Nova (7)



Affinités : Nova live migration (8)



Affinités : Nova en action (9)

```
$ openstack server create --image=centos8 --flavor=c1.large demo01
+---------+--------------------------------------+
| flavor  | c1.large (200)                       |
| id      | 77b8afdc-74a5-4256-90b3-682f0b6c460a |
| image   | centos8                              |
| name    | demo01                               |
[...]

=====================================================================
# rbd info oscar-instances/77b8afdc-74a5-4256-90b3-682f0b6c460a_disk
rbd image '77b8afdc-74a5-4256-90b3-682f0b6c460a_disk':
        size 64 GiB in 8192 objects
        block_name_prefix: rbd_data.8d692ca93d9338
        features: layering, exclusive-lock, object-map, fast-diff, deep-flatten
        parent: oscar-images/bac55d9d-a76f-4abd-9cab-5e7535202a9b
        overlap: 10 GiB
  [...]
=====================================================================
# rbd du oscar-instances/77b8afdc-74a5-4256-90b3-682f0b6c460a_disk
  NAME                                       PROVISIONED  USED
  77b8afdc-74a5-4256-90b3-682f0b6c460a_disk       64 GiB  880 MiB
```

Merci !

MÉSOCENTRE
UNIVERSITÉ
Clermont Auvergne

DIRECTION OPÉRATIONNELLE
DES SYSTÈMES D'INFORMATION
UNIVERSITÉ
Clermont Auvergne

Merci de votre attention ☺

Annexe : Séquence création VM