

# TP Proxmox

Pour ce TP vous allez découvrir et explorer Proxmox, une plateforme de virtualisation open-source. Vous allez apprendre à configurer Proxmox VE (Virtual Environment), à créer et gérer des machines virtuelles (VM ou QEMU) et des conteneurs (LXC), ainsi qu'à utiliser les fonctionnalités avancées de gestion des ressources et de haute disponibilité.

Pour ce TP vous pouvez vous aider de vos amis ; Google, ChatGPT et du site de Proxmox. Proxmox est doté d'une très grande communauté, ce qui en fait un vrai point fort, profitez-en !

Et bien entendu n'hésitez pas à poser des questions et à demander de l'aide à votre enseignant 😊

## Qu'est-ce que Proxmox ?

- Proxmox Virtual Environnement (PVE) est une solution de virtualisation open source de type « bare metal » (pour ce TP il est installé sur une VM mais il est généralement installé directement sur un serveur physique)
- Basé sur l'hyperviseur Linux KVM
- Proxmox VE est fourni sur une ISO (mais possible de l'installer sur Debian)
- Gratuit mais support payant

- Connaissez-vous d'autres hyperviseurs ?
- Quels sont les avantages de Proxmox sur ses concurrents ?
- Proxmox propose 2 types de virtualisation, rappelez les deux types vues en cours.

## Let's go !

Depuis le catalogue sur <https://im2ag-ecloud-vra.u-ga.fr/>, instanciez une VM Proxmox. Puis connectez vous à l'interface web de la VM, nouvellement créée, via **https://@ip:4206** en tant que root.

PS : normalement le port par défaut pour Proxmox est le 8006 mais il a été modifié pour les accès aux VM d'eCloud.

**Vous y êtes !**

Explorez le GUI (Graphical User Interface).

- Où se trouvent :
  - Les caractéristiques de votre hyperviseur ?
  - Les infos de stockage ?
  - Les sauvegardes de vos futurs VM/CT ?
  - La gestion des réseaux ?

## Le réseau

Comme vous pouvez le constater, il y a 3 réseaux affichés. Regardez attentivement leur configuration et comprendre leur utilité à chacun. Aidez vous aussi du shell (etc/network/interfaces)

Expliquez brièvement chaque réseau.

Pour information, la gestion sur réseau à l'UFR IM2AG se fait « au-dessus ». C'est à dire qu'on fait passer plusieurs VLAN sur l'interface réseau physique de nos serveurs. Ensuite nous taguons les VLAN ID sur la conf de nos machines virtuelles en fonction du réseau choisi.

## Le stockage

Comme vous pouvez le constater, il existe deux stockages ; local et local-lvm.

Expliquez la différence entre ces deux stockages.

## Création conteneur

Dans le cadre de ce TP, nous ne manipulerons pas des VM mais des CT, question de rapidité et de simplicité...

Avant de lancer la création de votre premier conteneur, il faut que vous téléchargiez un **template** existant, sur lequel vous vous appuyerez pour créer les conteneurs. Les templates sont des images pré-configurées de systèmes d'exploitation. En utilisant un template, vous pouvez créer un conteneur fonctionnel en quelques minutes, sans avoir à installer et configurer manuellement le système d'exploitation à partir de zéro.

Téléchargez un template Debian 12 depuis le stockage local, puis créez votre premier conteneur.

- Un CT « non-privilégié » ne pourra pas monter de stockage externe (NFS, CIFS..). Décochez cette case. Et dessous... la case « Nesting », qu'est-ce que c'est ?
- Mettre le CT dans le réseau privé 192.168.42.0/24
- Laissez le reste par défaut.

Vous avez maintenant votre premier conteneur créé. Connectez vous dessus via la console et vérifiez que vous avez bien le net.

Tapez la commande **uname -r** et la commande **cat /etc/os-release**.

- Que constatez-vous ?

Faites un clic droit sur le CT nouvellement créé et constatez les boutons.

- Quelle est la différence entre **Stop** et **Shutdown** ?

## Les sauvegardes

Sur le CT vous avez un onglet Backup et un onglet Snapshots.

- Faites un backup puis un restore
- Faites un snapshot puis un restore

- Que constatez-vous ?

Vous venez sûrement de découvrir le terme **snapshot**.

Les snapshots (instantanés) sont très répandus dans le monde du stockage mais aussi dans la gestion des bases de données, dans le versionning de code (comme avec Git). Ils permettent de capturer l'état (sauvegarde incrémentielle) d'un système, d'un fichier, d'un projet ou autre à un moment donné, facilitant ainsi la restauration et la gestion des versions.

Contrairement à une copie de fichier complète, un snapshot ne duplique pas l'ensemble des données. Il capture uniquement les métadonnées nécessaires pour enregistrer l'état d'un système à un moment précis. Par conséquent, les snapshots utilisent des techniques de gestion des différences, en ne sauvegardant que les modifications effectuées depuis le dernier snapshot. Cela permet de prendre **très peu de place sur le disque**, car seuls les changements incrémentiels sont enregistrés, et non les fichiers entiers.

## Exploration par la CLI de Proxmox

Les fichiers de configuration se trouve dans **/etc/pve**.

- Où se trouvent les fichiers de conf de :
  - Vos conteneurs ?
  - Vos VMs ?
  - Vos nœuds/serveurs ?
  - Vos stockages ?

Différentes commandes pour manipuler des containers via la CLI :

- `pct create <ctid> <template> [--hostname <hostname>] [--storage <storage>] [--net0 <options>]`
- `pct start <ctid> / pct stop <ctid> / pct shutdown <ctid>`
- `pct status <ctid>`
- `pct enter <ctid>`
- `pct clone <ctid> <newctid>`

**!!! TOUT ce qui est faisable depuis l'interface web l'est depuis la CLI !!!** En réalité lors de bug assez complexe, la CLI est nécessaire pour résoudre les problèmes. **Tout n'est pas faisable depuis l'interface web.** Proxmox bénéficie d'une grande communauté, vous trouverez beaucoup d'information sur le net.

## Les templates

Vous allez maintenant créer votre propre template sur lequel vous pourrez vous appuyer pour vos prochains CT. Cette manipulation permet d'avoir une pré-configuration commune pour plusieurs CT. Par exemple les mises à jours, des paquets nécessaire à tous, des alias de commande, etc.

Ici nous ferons simple. Sur votre CT, testez la commande `l` (L en minuscule) et constatez qu'elle n'existe pas. Allez dans votre `~/bashrc` et décommentez la ligne concernée puis faites « **source** `~/bashrc` » puis testez de nouveau la commande `l` (L en minuscule).

- Faites une sauvegarde de ce conteneur
- Grâce au chapitre précédent, retrouvez où se trouve la sauvegarde via la CLI
- Faites de cette sauvegarde, un template
- Créez un second CT s'appuyant sur ce template
- Vérifiez avec la commande `l` (L en minuscule) depuis le nouveau CT

## Cluster

Se mettre en trinôme et mettez vos trois nœuds Proxmox en cluster.

Il est recommandé d'avoir un cluster Proxmox avec un minimum de trois nœuds plutôt que deux car Proxmox fait appel à un mécanisme appelé **Quorum**, qui permet de prendre des décisions consensuelles. Avec trois nœuds, le quorum garantit une majorité. Pour faire simple, si l'un des nœuds dit faux, on sait que les deux autres disent vrai. Tandis que s'il y a seulement deux nœuds, comment savoir qui dit faux, qui dit vrai ?

Faites la migration d'un CT d'un nœud à un autre et constatez le temps que ça prend et que vous avez bien toujours le réseau sur le CT une fois celui-ci migré.

- Pourquoi le CT a toujours le réseau ?

## Cluster - Le stockage

Vos 3 nœuds sont « clusterisés » mais ils n'ont pas de stockage en commun, alors qu'il est essentiel d'accéder à des données communes dans un cluster, comme les Backups, les ISO, les templates.

Il existe 3 types de stockage fondamentaux à connaître :

**Mode fichier** > pour stocker des fichiers de manière organisée et hiérarchisée.

**Mode bloc** > pour avoir de l'accès concurrentiel. Utilisé pour les datastores au niveau des hyperviseurs & pour les bases de données (iSCSI : comme si tu monter un disque sur ta machine en local)

**Stockage objet** > énormément utilisé dans les domaines du cloud, du Big Data, mais aussi sauvegarde et archive. C'est un type de stockage idéal pour **stocker des données qui sont rarement modifiées**. Par exemple pour n'importe quel type de médias et tout contenu fixe. Système de données non structurées, où un objet contient la donnée, les métadonnées et un identifiant. L'objet sera dupliqué et dispersé dans plusieurs pool de stockage. Ça permet une très grande évolutivité, une forte résilience et une meilleure performance.

Les protocoles d'accès :

**NFS** (Network File System) > NFS est un protocole de stockage en **mode fichier** qui permet à un ordinateur client d'accéder à des fichiers distants via le réseau, de manière similaire à l'accès local. Il est principalement utilisé sur des systèmes Unix. Vos HOMES sont stockés sur un volume NFS distant (dans un datacentre du campus) et ce volume est monté sur chaque machine de l'UFR IM2AG.

**CIFS** (Common Internet File System) > **Mode fichier**, Globalement la même chose que le NFS mais côté Windows. Quand vous vous connectez à ebureau, vous retrouvez vos HOMES via le CIFS.

**iSCSI** (Internet Small Computer Systems Interface) > iSCSI est un protocole de stockage en **mode bloc**. Principalement pour accéder à des stockages distants, de type base de données et datastore d'hyperviseur.

**S3** (Simple Storage Service) est un service de **stockage objet** en ligne largement utilisé pour stocker et récupérer des données, que ce soit pour des applications web, des sauvegardes, des données statiques ou des contenus multimédias.

Pour ce TP nous ne verrons pas ces différents protocoles. Nous verrons en revanche le système de stockage **CEPH**, qui permet une grande scalabilité, une grande tolérance et de la haute disponibilité. CEPH prend en charge les trois types de stockage vus plus haut.

## Cluster - Le stockage - CEPH

Sur vos nœuds il y a deux disques durs virtuels, sda et sdb.

Updates	Device	Type	Usage	Size	GPT	Model
Repositories	+ /dev/sda	SSD	partitions	68.72 GB	Yes	Virtual_disk
Firewall	+ /dev/sdb	SSD	partitions	34.36 GB	Yes	Virtual_disk
Disks						

Le sdb n'est pour le moment pas utilisé, c'est sur celui-ci que vous allez installer puis configurer CEPH. Il faut créer une partition CEPH sur sdb :

- Depuis la console **cfdisk /dev/sdb** > créez une nouvelle partition GPT (bien faire **Write**).

Puis :

- Installez Ceph sur chaque nœud
- Mettre les 3 nœuds en manager et monitor
- OSD > stockage distribué de Ceph
- On assigne le disque sdb sur OSD

- Que constatez-vous sur les deux autres nœuds ?

Les OSD sont créés, le réseau CEPH est prêt, il faut maintenant créer votre pool de stockage en affectant vos OSD au pool de stockage

`Ceph > Pools > Create` (« Size 3 » indique qu'on a 3 nœuds et « Min Size 2 » indique que si on perd un nœud le pool de stockage continu de fonctionner.

Votre pool de stockage Ceph est maintenant créé !

Changez le stockage cible d'un ou plusieurs CT pour le mettre sur CEPH.

Comparez la taille du disque du CT (ou des CT) et l'allocation des ressources de stockage au niveau du CEPH.

- Que constatez-vous ?

Migrez un CT d'un nœud vers un autre.

- Que constatez-vous ?

Sur le datacenter, allez dans HA et ajoutez dans Ressources un CT.  
Éteignez maintenant le nœud contenant un CT sur CEPH.

- Que constatez-vous ? (patientez un peu)

## Conclusion

Vous avez maintenant acquis des compétences essentielles en virtualisation et gestion d'infrastructure. En explorant les fonctionnalités puissantes de Proxmox, vous avez appris à créer et gérer des **containers**, à configurer des **réseaux** virtuels et à intégrer et utiliser CEPH, un système de **stockage** très répandu dans les domaines du cloud. Ces compétences vous préparent à être plus avisé et plus efficace sur des environnements virtuels.