1. Présentation de Proxmox

Proxmox Virtual Environment (VE) est une plateforme de virtualisation open source basée sur une distribution Debian avec un noyau modifié. Elle permet de faire fonctionner plusieurs systèmes (Linux, Windows, etc.) sur un seul serveur physique, en utilisant deux méthodes :

- les machines virtuelles (VM) avec KVM : cela permet la virtualisation complète d'un ordinateur ;
- les conteneurs LXC (CT) : plus légers, ils partagent le noyau Linux du système hôte.

1.1. Les machines virtuelles (VM) avec KVM

1.1.1. Qu'est-ce que c'est ?

KVM (Kernel-based Virtual Machine) est une technologie de virtualisation intégrée directement dans le noyau Linux. Elle transforme Linux en un hyperviseur de type 1, capable d'exécuter plusieurs systèmes d'exploitation isolés.

1.1.2. Fonctionnement

Chaque VM fonctionne comme un processus Linux classique, mais avec un accès dédié à des ressources virtualisées (CPU, RAM, disque, carte réseau, etc.)

1.1.3. Avantages

- Performances des machines virtuelles proches de celles directement installées sur un matériel physique.
- Support de nombreux système d'exploitation, Linux et non Linux (Windows, FreeBSD, etc.).
- Support de la **virtualisation imbriquée** (nested virtualization).

1.1.4. Intégration dans Proxmox

Proxmox utilise KVM en arrière-plan, mais fournit :

- une interface web pour créer, configurer et gérer les VM ;
- des outils en ligne de commande (qm) ;
- des fonctionnalités avancées : sauvegardes (complète et snapshot), migration à chaud, haute disponibilité.

1.2. Les conteneurs (LXC)

1.2.1. Qu'est-ce que LXC ?

LXC (Linux Containers) est une technologie de virtualisation légère qui permet d'exécuter plusieurs systèmes Linux isolés sur un même noyau Linux. Contrairement aux machines virtuelles, les conteneurs ne virtualisent pas le matériel, mais utilisent les fonctionnalités du noyau pour isoler les processus.

1.2.2. Principes de fonctionnement

LXC repose sur plusieurs fonctionnalités du noyau Linux :

- **namespaces** : cela permet d'isoler les processus, le réseau, etc. ;
- **cgroups** (Control Groups) : cela permet de limiter et de surveiller l'utilisation des ressources (CPU, RAM, etc.) ;
- **chroot** : renforce la sécurité et l'isolation des conteneurs.

1.2.3. Caractéristiques principales

- **Partage du noyau** : tous les conteneurs utilisent le même noyau Linux Debian de l'hôte.
- **Solution légère** car l'utilisation du noyau de l'ordinateur hôte entraine peu de surcharge.
- **Démarrage** en quelque secondes des conteneurs.
- Ne peut **exécuter que des distributions Linux** : Debian, Ubuntu, etc.
- **Facile à cloner** : idéal pour déployer rapidement plusieurs instances de conteneurs.

1.2.4. Gestion dans Proxmox

Proxmox offre :

- une interface web pour créer, configurer et gérer les conteneurs ;
- des outils en ligne de commande (pct) ;
- des fonctionnalités avancées : sauvegardes (complète et snapshot), migration à chaud, haute disponibilité.

1.2.5. Cas d'usage typiques

- Hébergement de services Linux (web, base de données, DNS...).
- Environnements de développement ou de test.
- Déploiement rapide d'applications.
- Infrastructure DevOps (CI/CD, microservices).

1.3. Comparatif entre KVM et les conteneurs LXC

Critère	KVM (Machines virtuelles)	LXC (Conteneurs Linux)
Type de virtualisation	Virtualisation complète (hyperviseur type 1)	Virtualisation légère (niveau système)
Isolation	Très forte (chaque VM a son propre noyau)	Moins forte (partage le noyau de l'hôte)
Systèmes d'exploitation supportés	Linux, Windows, FreeBSD, etc.	Uniquement Linux
Performances	Légèrement inférieures à un ordinateur dédié	Très proches d'un ordinateur dédié
Consommation de ressources	Plus élevée (RAM, CPU, disque)	Très faible
Démarrage	Plus lent (comme un vrai ordinateur)	Très rapide (quelques secondes)
Cas d'usage typiques	Serveurs Windows, Windows 11, OPNSense	Services Linux légers, microservices (Web, DNS, DHCP), DevOps

1.4. La gestion du réseau avec Proxmox VE

Proxmox VE permet de gérer l'accès au réseau :

- en connectant les VM et les conteneurs à un **bridge**, avec ou sans VLAN ;
- en agrégeant les interfaces réseaux physiques de l'hyperviseur Proxmox en définissant un **bond**.

1.4.1. Qu'est-ce qu'un bond dans Proxmox ?

Un bond (ou agrégation de liens) est une technique qui permet de regrouper plusieurs interfaces réseau physiques en une seule interface logique. Cela permet :

- la **tolérance de panne** (si un câble tombe, le trafic passe par l'autre) ;
- d'**augmenter la bande passante** (selon le mode utilisé) ;
- de **répartir la charge** des flux réseau (load balancing).

1.4.2. Qu'est-ce qu'un bridge dans Proxmox

Un bridge réseau (ou pont réseau) est une interface virtuelle qui agit comme un commutateur (switch) logiciel qui permet de :

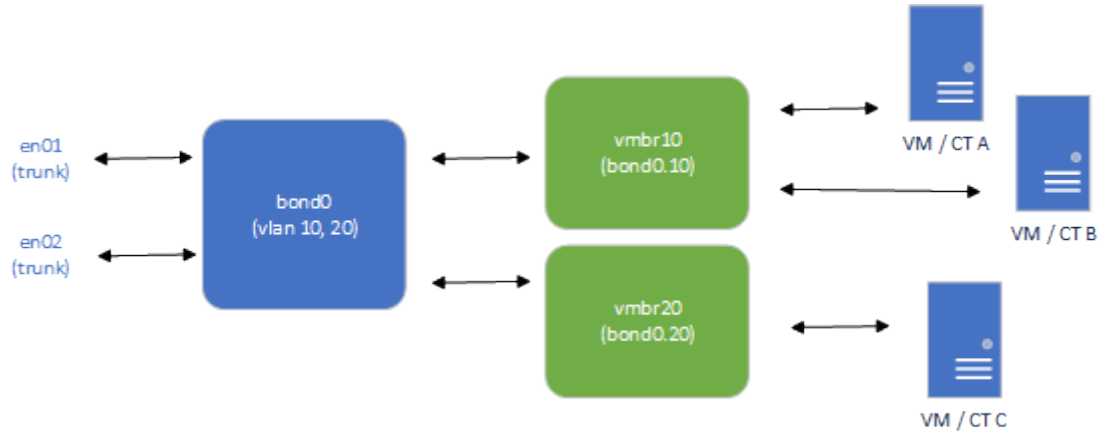
- relier les VM/CT au réseau local comme s'ils étaient des machines physiques ;
- partager une ou plusieurs interfaces réseaux physiques, ou bien un bond, entre plusieurs VM/CT ;
- gérer un bridge par VLAN (méthode classique) : les VMs et les conteneurs sont connectées à des bridges distincts selon leur VLAN ;
- gérer les VLANs directement dans un même bridge puis en associant un tag (étiquelle) de VLAN dans la configuration réseau des VM et des conteneurs LXC.

1.4.3. Exemple d'utilisation d'un bond et d'un bridge distinct par VLAN

Voici un schéma montrant :

- les **interfaces réseaux physiques** de l'hyperviseur appelées **en01 et en02** et configurées en **mode trunk** pour les deux VLAN 10 et 20 nécessaires. Sur le switch où sont connectées les deux interfaces physiques de l'hyperviseur, deux VLAN 10 et 20 doivent être créés au préalable et être autorisés dans le trunk ;
- un **bond** appelé bond0 pour agréger deux interfaces réseau physiques de l'hyperviseur en01 en02 (tolérance de panne ou

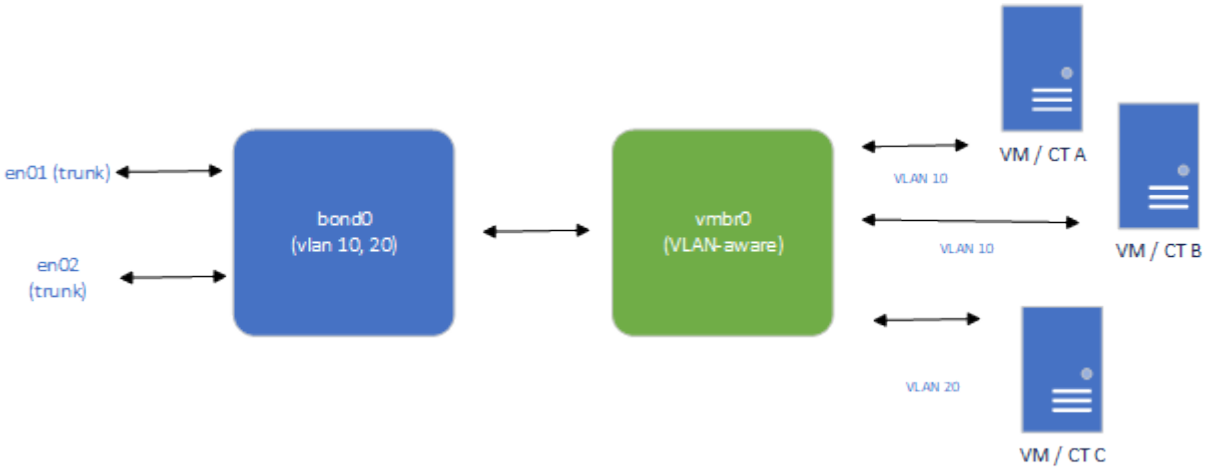
répartition de charge) ;
- un bridge virtuel appelé **vmbr10** connecté à **bond0** en précisant l'**ID 10 du VLAN** utilisé pour relier les VM/CT A et B au réseau ;
- un deuxième bridge virtuel appelé **vmbr20** connecté également à **bond0** en précisant l'**ID 20 du VLAN** utilisé pour relier la VM/CT C au réseau.



1.4.4. Exemple d'utilisation d'un bond et d'un seul bridge (VLAN-aware) pour gérer 2 VLAN

Voici un schéma montrant :

- les **interfaces réseaux physiques** de l'hyperviseur appelées **en01 et en02** et configurées en **mode trunk** pour les deux VLAN 10 et 20 nécessaires. Sur le switch où sont connectées les deux interfaces physiques de l'hyperviseur, deux VLAN 10 et 20 doivent être créés au préalable et être autorisés dans le trunk ;
- un seul bridge virtuel vmbr0 pour lequel VLAN-aware est activé.
- lors de configuration d'une VM ou d'un conteneur LXC, il faut indiquer l'utilisation du bridge **vmbr0** en précisant le tag du VLAN (10 ou 11).



1.4.5. Exemple d'utilisation d'un bond et d'un bridge distinct par VLAN avec Proxmox dans WSL

Voici un schéma montrant :

- la seule **interface réseau physique** du PC Windows appelée **eth0**. Il n'y a pas de trunk configuré ;
- il n'y a pas de bond car il n'y a qu'une seule interface réseau physique ;
- un bridge virtuel appelé **vmbr10** connecté à **eth0** sans ID de VLAN utilisé pour relier les VM/CT A et B en gérant un sous-réseau IP distinct)? Si aucune adresse IP n'est associée au bridge, ce réseau isolé des autres, ce qu est équivalent au réseau interne de VirtualBox ;
- un deuxième bridge virtuel appelé **vmbr20** connecté également à **bond0** en précisant l'**ID 20 du VLAN** utilisé pour relier la VM/CT C au réseau.

pour gérer

From:
https://siocours.lycees.nouvelle-aquitaine.pro/ - **Les cours du BTS SIO**

Permanent link:
https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/cloud/proxmox/description?rev=1749405614

Last update: **2025/06/08 18:00**