

# Introduction à la virtualisation système

G. Urvoy-Keller

R320

# Les 2 types de virtualisation

## ➤ **Virtualisation lourde**

- Virtualbox en salle de TP
- OS + applications

## ➤ **Virtualisation légère**

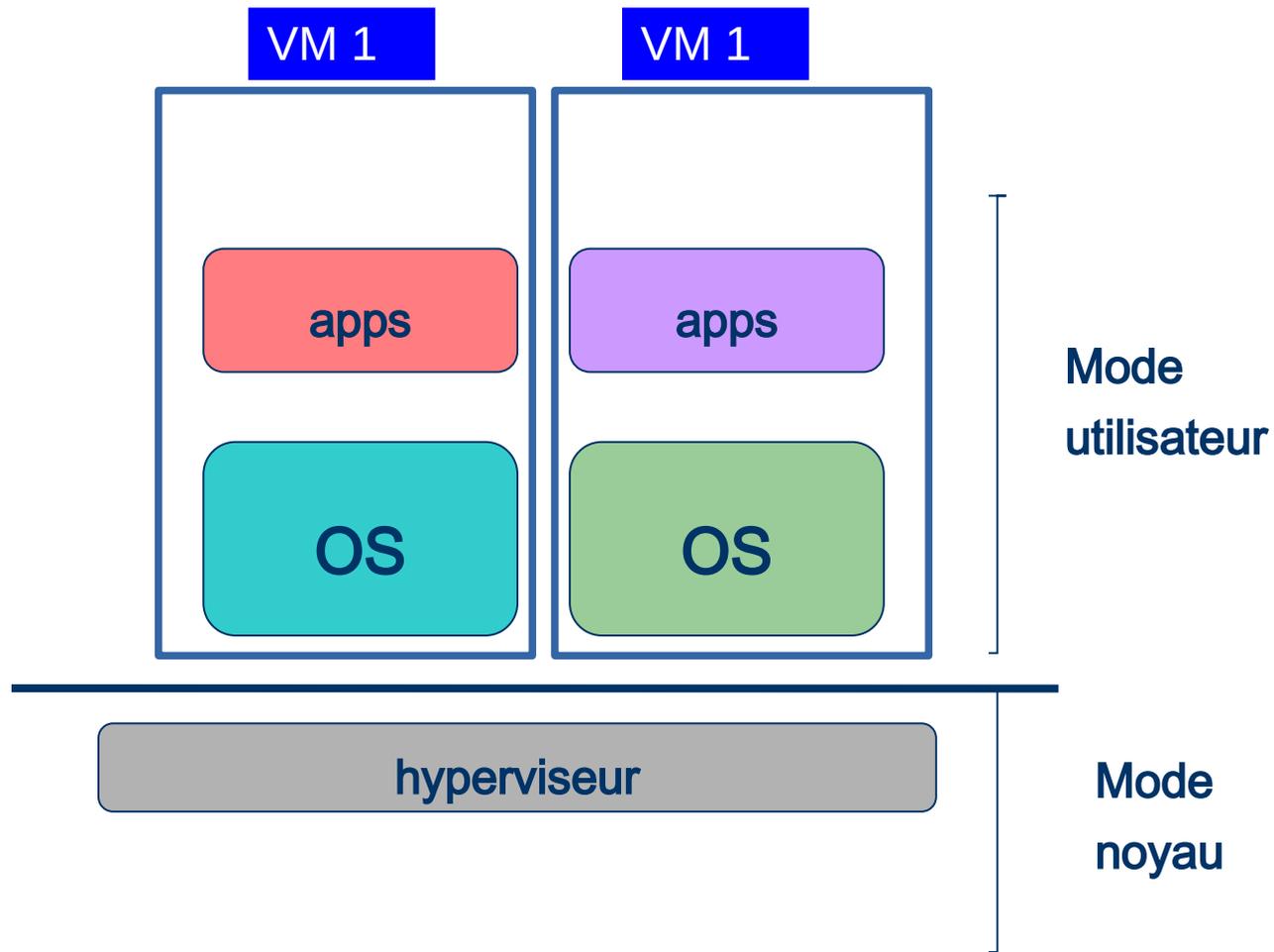
- Containers
- Ce que vous ferez en TP
- Application seule

# Virtualisation lourde : dé-priviléger un OS

- 2 modes de fonctionnement processeur : mode noyau et mode utilisateur
- Mode noyau : accès à tous les périphériques → OS
- Mode Utilisateur : pour les applications. Accès aux données chargées en mémoire et opération sur ces données



# Virtualiser : dé-priviléger un OS



# Le zoo des hyperviseurs

## ➤ **Entreprise :**

- Vsphere de VmWare - 60% du marché
- Hyper-V de Microsoft - 20 %
- Xen Server de Citrix – 4 % du marché
- QEMU/KVM natif en Linux

## ➤ **Pour les tests :**

- Virtualbox d'Oracle
- VMWare Player

# Pourquoi virtualiser?

- **Dans les années 1990-2000, le coût des serveurs décroît**
  - En valeur absolue du fait de la concurrence
  - Par rapport au coût des mainframes encore très présents
- **Les éditeurs (Microsoft, distribution Linux) recommandent une application/service par système d'exploitation** →
  - Une machine pour le DNS
  - Une machine pour le mail
  - Une machine pour NFS, etc.

# Pourquoi virtualiser?

## ➤ **Au final:**

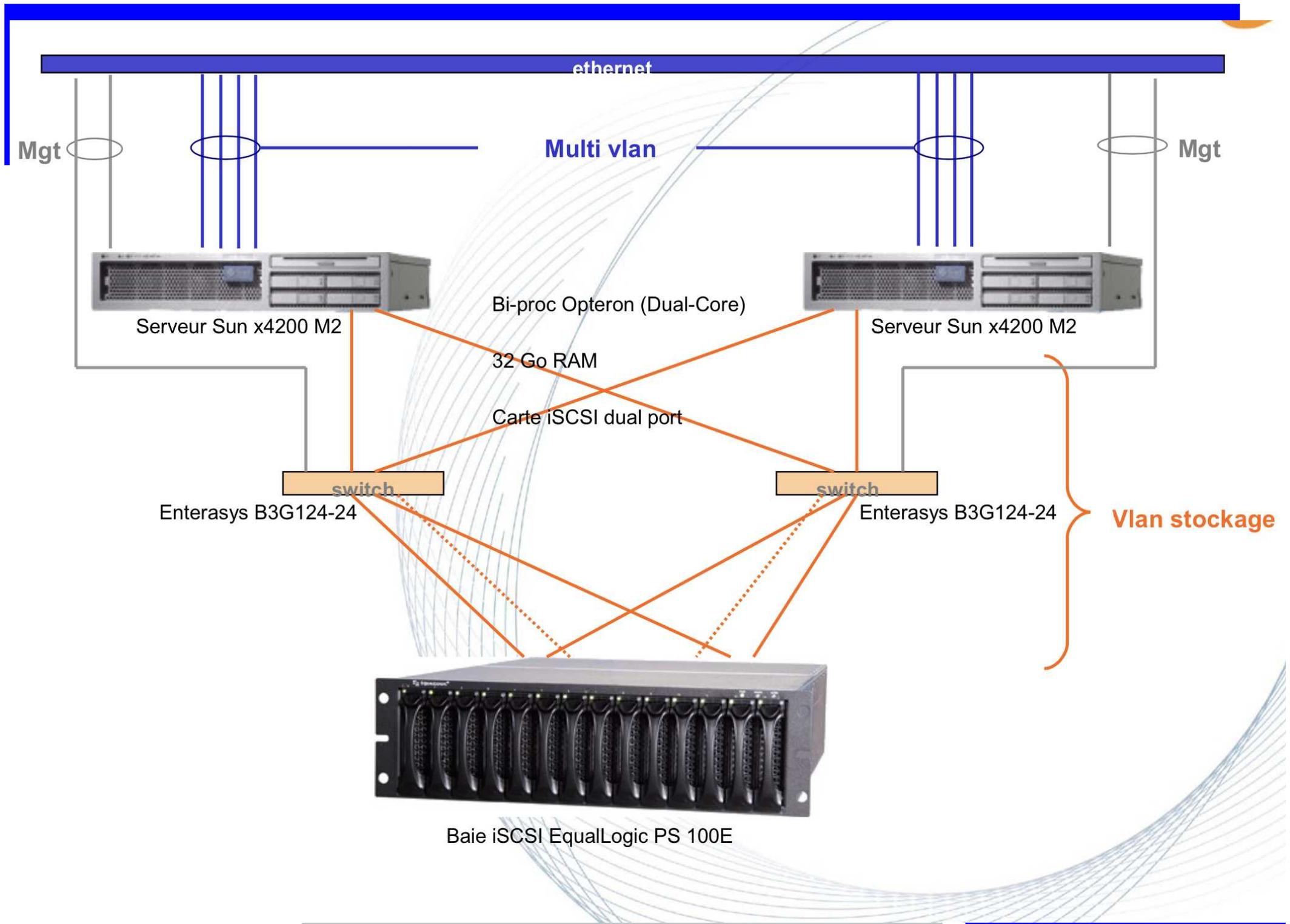
- Une multiplicité de serveurs dans les datacenters des entreprises
- 80% ont une utilisation moyenne inférieure à 10%
- Des coûts d'exploitation/maintenance (personnel du service informatique) qui croissent avec le nombre de serveurs
- Des coûts en place : les salles serveurs ne sont pas indéfiniment extensibles
- Des coûts en climatisation/électricité élevés

# Pourquoi virtualiser?

- **Les serveurs deviennent si puissants qu'une seule application par serveur n'est plus justifiable**
  - Processeurs 64 bits multi-coeurs
  - Un serveur en 2009 est estimé, en moyenne, 10 à 12 fois plus puissant qu'un serveur en 2004
- **Remplacer les serveurs à raison de un pour un n'est plus possible**

# Avantages de la virtualisation

- **Plaçons-nous dans le cas où la virtualisation est effective**
- **Vous avez acheté :**
  - Deux gros serveurs
  - Un espace de stockage commun, SAN : storage area network
  - Des licences d'un outil de virtualisation, par exemple VMWARE
  - Les formations pour votre personnel



ethernet

Mgt

Multi vlan

Mgt

Serveur Sun x4200 M2

Bi-proc Opteron (Dual-Core)

32 Go RAM

Carte iSCSI dual port

Serveur Sun x4200 M2

switch

Enterasys B3G124-24

switch

Enterasys B3G124-24

Vlan stockage

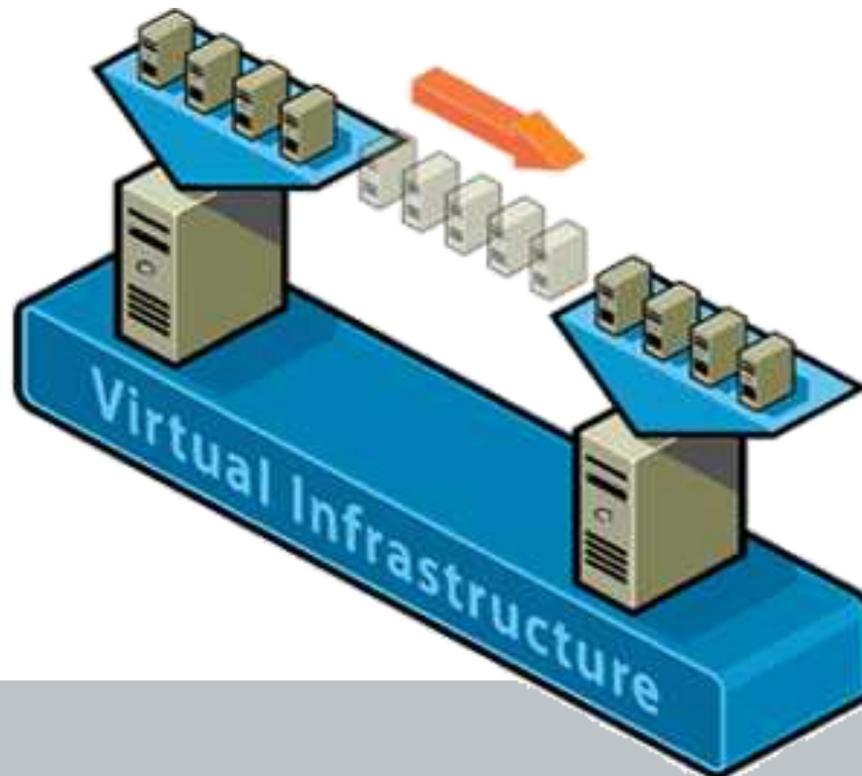
Baie iSCSI EqualLogic PS 100E

# Avantages de la virtualisation

- **Réduction des coûts**
  - 20 à 40% en général
- **Avantages additionnels :**
  - Gain de place
  - De nouvelles fonctionnalités

# Avantages de la virtualisation

- **Migration des machines virtuelles d'un serveur physique à l'autre**
  - Utile si panne, mais pannes rares sur les serveurs
  - Très utile pour maintenance des serveurs physiques, typiquement mise à jour de sécurité de l'OS (plusieurs fois par an)



# Avantages de la virtualisation

- **Mise en service quasi-instantanée d'une nouvelle machine, grâce aux templates**
  - une machine quasi-finalisée qui permet de démarrer en 1 minute une nouvelle machine
  - Createvm en salle TP qui trouve les images dans */home/VBox/vdi*
  - Amazon Web Service, VMware, Openstack

# Avantages de la virtualisation

- **Possibilité de faire des instantanés (snapshots) des machines**
- **Scénario typique :**
  - Vous voulez installer une nouvelle fonctionnalité sur une machine, faire une mise à jour
  - Vous n'êtes pas sûr du résultat.
  - Vous :
    - \* Faites un instantané
    - \* Effectuez le test
    - \* Si l'installation échoue, vous ... revenez dans le temps!!!

# Virtualisation légère : les containers Linux

# Container versus hyperviseur

- **Partage du noyau entre toutes les VMs (ou containers)**
- **Sur un serveur typique:**
  - 10-100 machines virtuelles
  - 100-1000 containers
- **Un containter est un groupe de processus dans une machine Linux, isolés des autres processus qui tournent sur la machine**
  - Utilisation des **namespaces** (du noyau Linux) pour assigner à un groupe de processus : isolation, leur propre pile réseau (interfaces, sockets, routage), volumes
  - Utilisation du **cgroups** (du noyau Linux) pour assigner des ressources à ces processus par exemple de la CPU, de la mémoire
    - Similaire à ce que vous faites sous Virtualbox

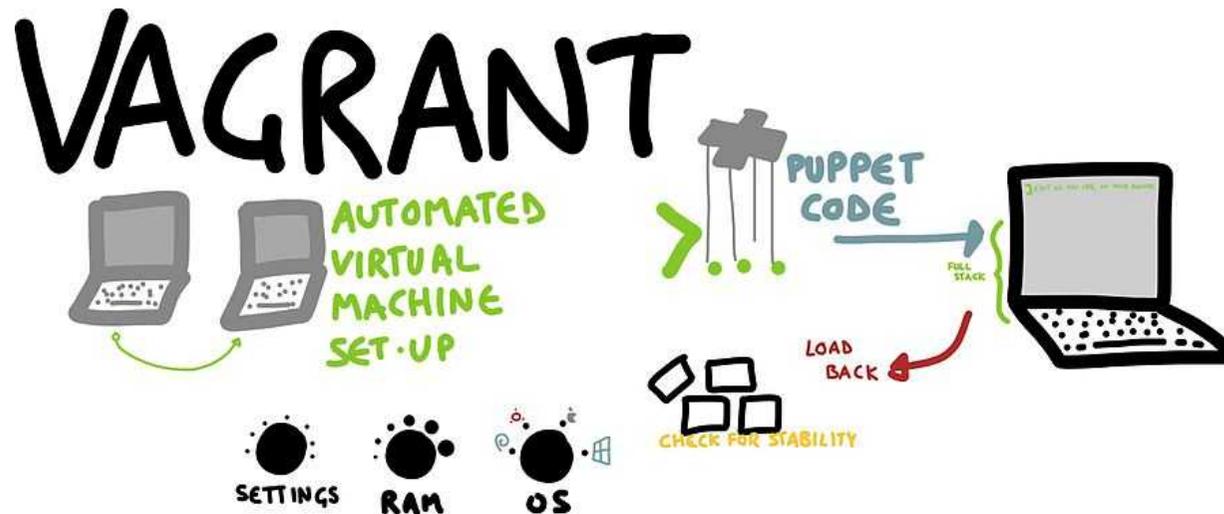
# Container versus Hyperviseur

- **De l'intérieur, ressemble à une VM**
- **De l'extérieur, ressemble à des processus normaux**
- **Un container peut être une VM complète ou un groupe de processus par exemple un serveur Apache ou MySQL**
  
- **Moteurs de gestion de containers:**
  - LXC (Linux Containers – August 2008)
  - Docker (started in March 2013)
  - Openvz (started in 2005)

# Management des VMs et containers

# Gestion de VMs

- VMware Vsphere, Citrix Xen permettent de gérer quelques serveurs physiques
- **Vagrant: Gestion de VMs indépendamment des hyperviseurs**
  - Notion d'images (boxes de Vagrant)
  - Configuration automatique de VM: support de Puppet, Chef, Ansible
  - Un fichier qui contient toute la configuration



# Fichier de configuration Vagrant

```
## -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure(2) do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://atlas.hashicorp.com/search.
  config.vm.box = "ubuntu/vivid64"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false

  # Create a forwarded port mapping which allows access to a specific port
  # within the machine from a port on the host machine. In the example below,
  # accessing "localhost:8080" will access port 80 on the guest machine.
  config.vm.network "forwarded_port", guest: 5001, host: 5001

  # Create a private network, which allows host-only access to the machine
  # using a specific IP.
  # config.vm.network "private_network", ip: "192.168.33.10"

  # Create a public network, which generally matched to bridged network.
  # Bridged networks make the machine appear as another physical device on
  # your network.
  config.vm.network "public_network"
```

# Gestion de VMs

## ➤ Plateforme de gestion de clouds

### ➤ Openstack

➤ Chaque fonction (management de VM, réseaux, volumes, identités) est un composant (au final un service Linux)

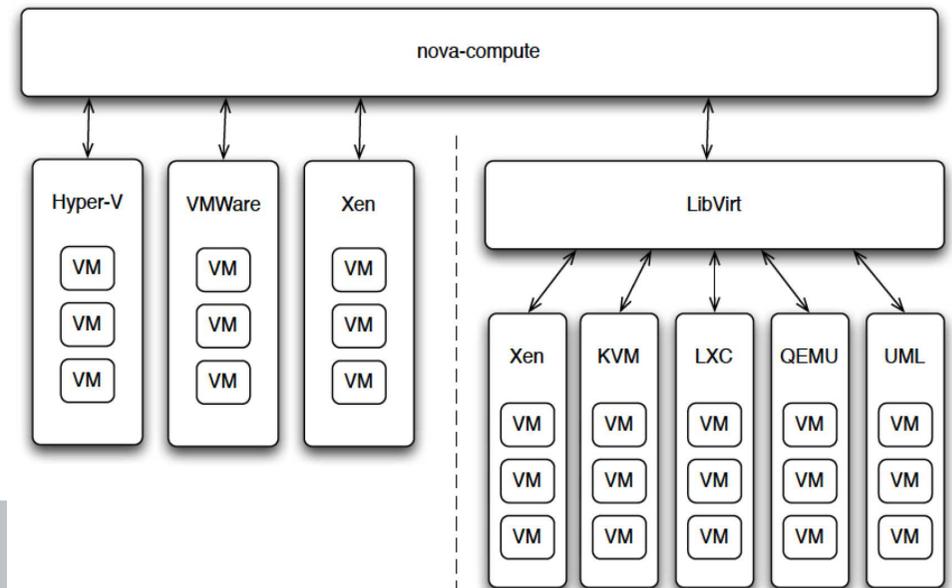
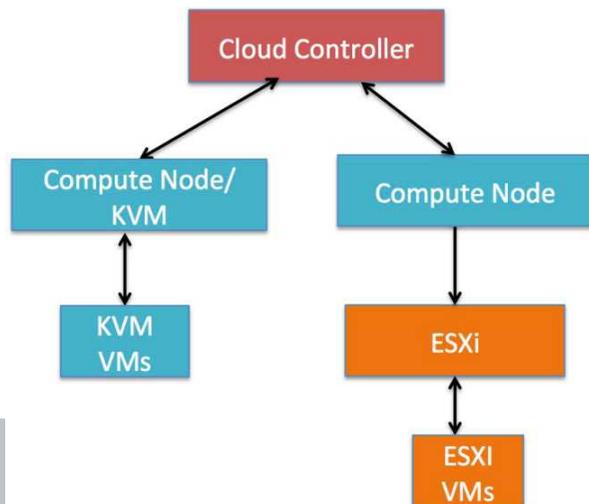
➤ Nova: compute nodes (hyperviseurs)

➤ Cinder : volumes

➤ Neutron : réseaux

➤ Les composants interagissent via une API REST (~HTTP)

➤ Les nœuds Compute (serveurs physiques) peuvent faire tourner des hyperviseurs différents : KVM, Xen, Citrix, etc



# Gestion de containers

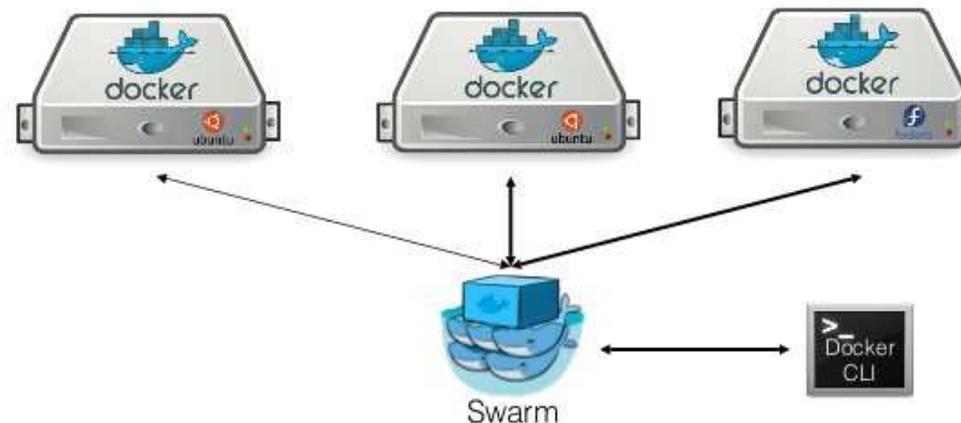


## ➤ Orchestration de containers

- Serveur unique: Docker, LXC
- Several multiples: Docker Swarm

## ➤ Orchestration avancés : Kubernetes, Swarm, Mesos

### With Docker Swarm



# Ce dont on a pas le temps de parler....

## ➤ **SDN : Software Defined Network**

- Modèle de réseau centralisé et non distribué comme on fait classiquement (avec algorithme de spanning tree niveau 2 ou routage niveau
  - des switches en mode esclave
  - un contrôleur unique qui envoie les règles aux switches

## ➤ **NFV : network virtualisation function**

- fonctions réseaux du type NAT, répartiteur de charge, pare-feu dans des VMs

## ➤ **La virtualisation dans les réseaux mobiles**

- CloudRAN (RAN = Radio Access Network) : virtualisation des fonctions de base (transmission)