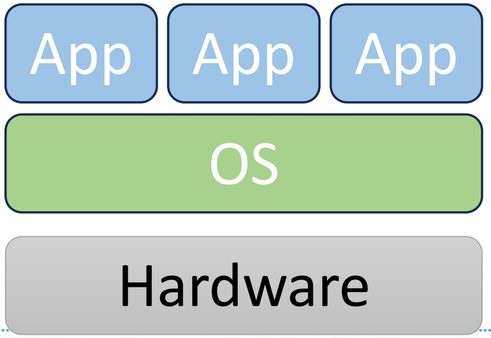
Introduction to virtualization

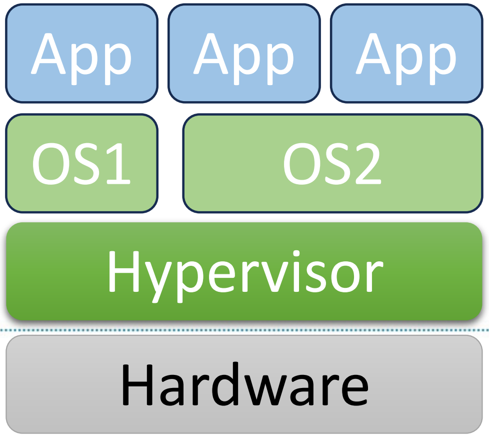
UNIVERSITÉ
CÔTE D'AZUR

Heavy virtualization

Introduction

- A set of techniques to enable several Operating Systems (guests) on a physical machine (host)
  - Not to be confused with multi-boot
- Use of a specific program called Hypervisor or Virtual Machine Monitor (VMM)





Hypervisor Zoo

- For servers
  - VMware ESXI
  - Nutanix AHV
  - Microsoft Hyper-V
  - Citrix Hypervisor (IBM)
  - QEMU/KVM
- For Clients
  - Oracle VirtualBox
  - VMWare player
  - Parallels Desktop

Why virtualization

- In the 90s, cost of servers decreased gradually
- Software editors (Microsoft, distribution Linux) advocate one application/service per server for mission critical services
  - One DNS server
  - One mail server
  - One NFS server
- Each server with specific OS version and libraries
- Servers isolation

Why virtualization

- Consequences
  - Plenty of servers in datacenters
  - But low usage
    - 80% have utilization < 10%
  - Maintenance/operation costs increase with number of servers
  - Lack of space in datacenters
  - Energy costs skyrocket

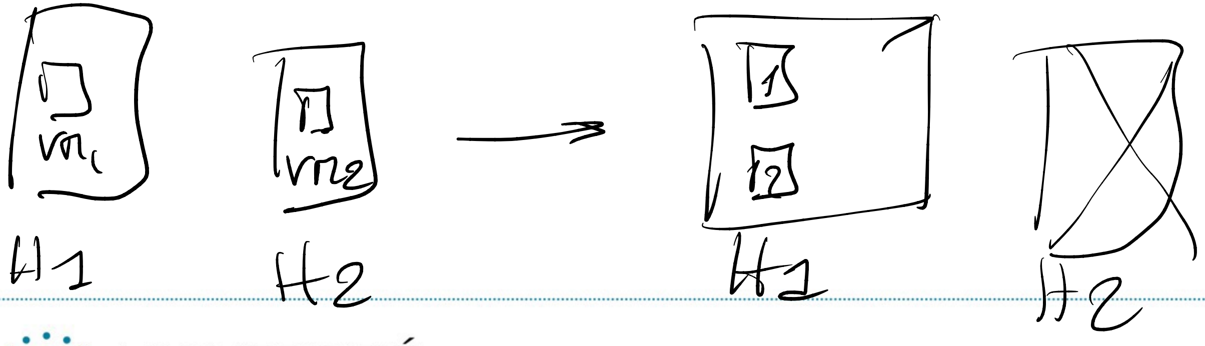
Why virtualization

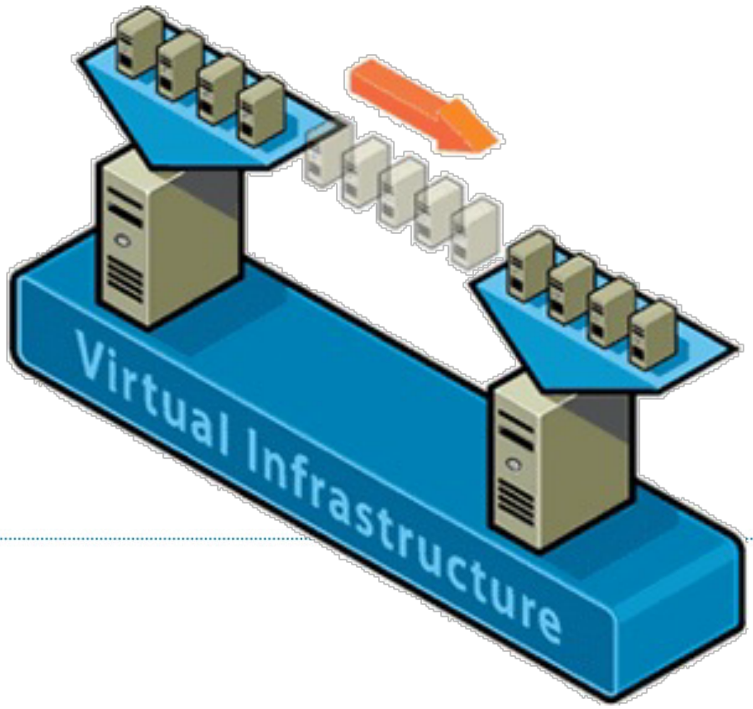
- Compared to 90s, servers are very cheap and more powerful
  - 64 bits multi-core with 100s of GB of RAM
- Using one server per service is a waste of resources
- Placing multiple services on the same server
  - Makes economical/ecological sense
  - But must maintain isolation
- Virtualization ensures
  - Cost reduction (20 to 40%) by reducing number of servers
  - More space in server room
- And virtualization brings new functionalities

New functionalities

Migrating VMs

- Migration of VM from one host to another
- Useful for
  - Planed downtime (physical server upgrade)
  - VM consolidation





1) Pause VM → {migrate → unpause
{destroy original

⊖ downtime

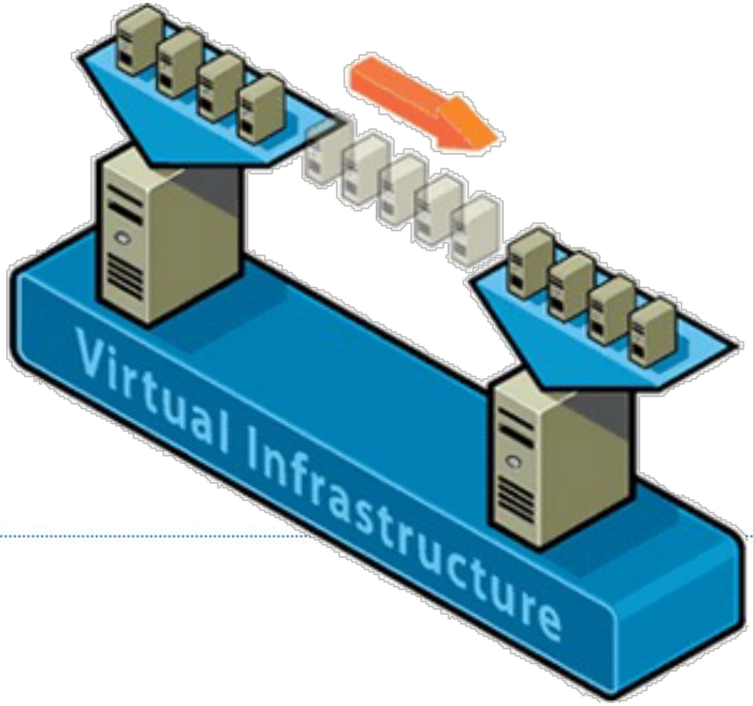
2) Copy VM while running → destroy original one

⊖ inconsistent memory

3) Copy VM while running
pause VM
copy modified memory

Snapshoting VMs

- Save the complete state of the VM
  - OS and apps
- Useful for
  - Testing updates
  - Backups



Vertical scaling

- On the fly reconfiguration of VMs
  - Add CPU, memory, disks to the VM
  - Need support from the guest OS and application
- Aka Hot Add
  - Supported by RedHat Enterprise Linux, Windows Server

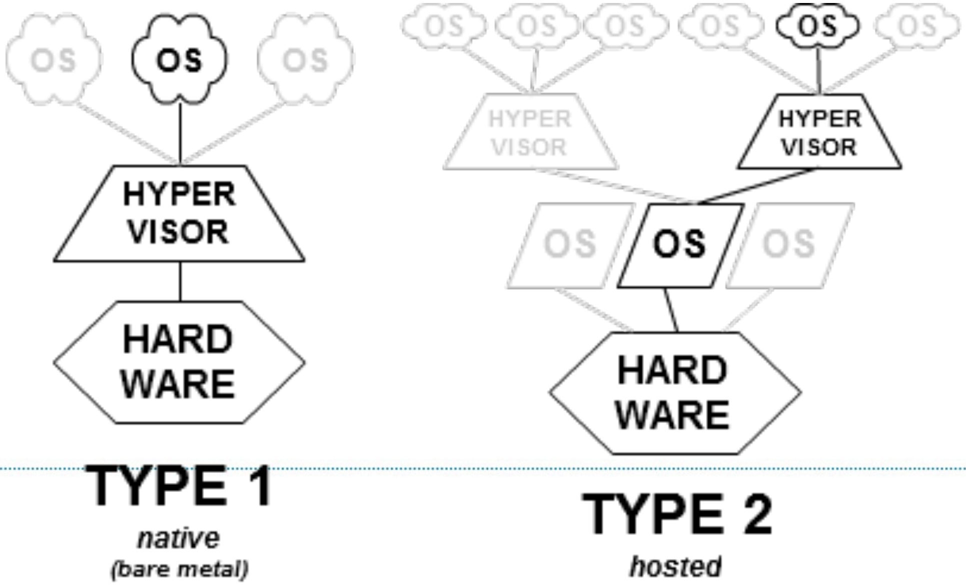
Hypervisors

Bare metal hypervisor

- Hypervisor comes in 2 types
- Bare Metal (Type 1)
- Run directly on top of hardware
  - Machine boots on hypervisor
  - Hypervisor starts OS
- Used in production servers and data centers
- Examples
  - VMware ESXI
  - Nutanix AHV
  - Microsoft Hyper-V
  - Citrix Hypervisor (IBM)

Host-based

- Host-based (type 2)
- Hypervisor is an application running inside an existing OS
- Typically deployed on end-user-machines

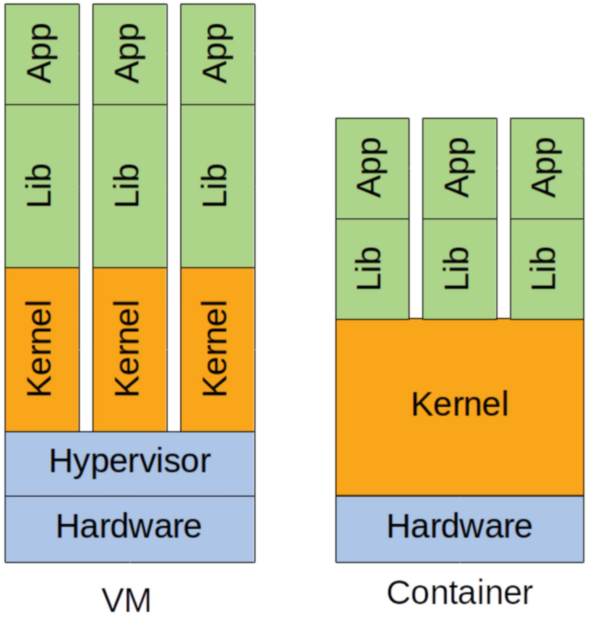


From wikipedia

Light virtualization

Container-based

- Rather than using an hypervisor, the container approach shares the Kernel among VM
- Lightweight
- On a typical server
  - 10-100 VMs
  - 100-1000 Containers
- Containers engines
  - LXC (Linux Container, 2008)
  - Docker (2013)



Containers in Linux

OS

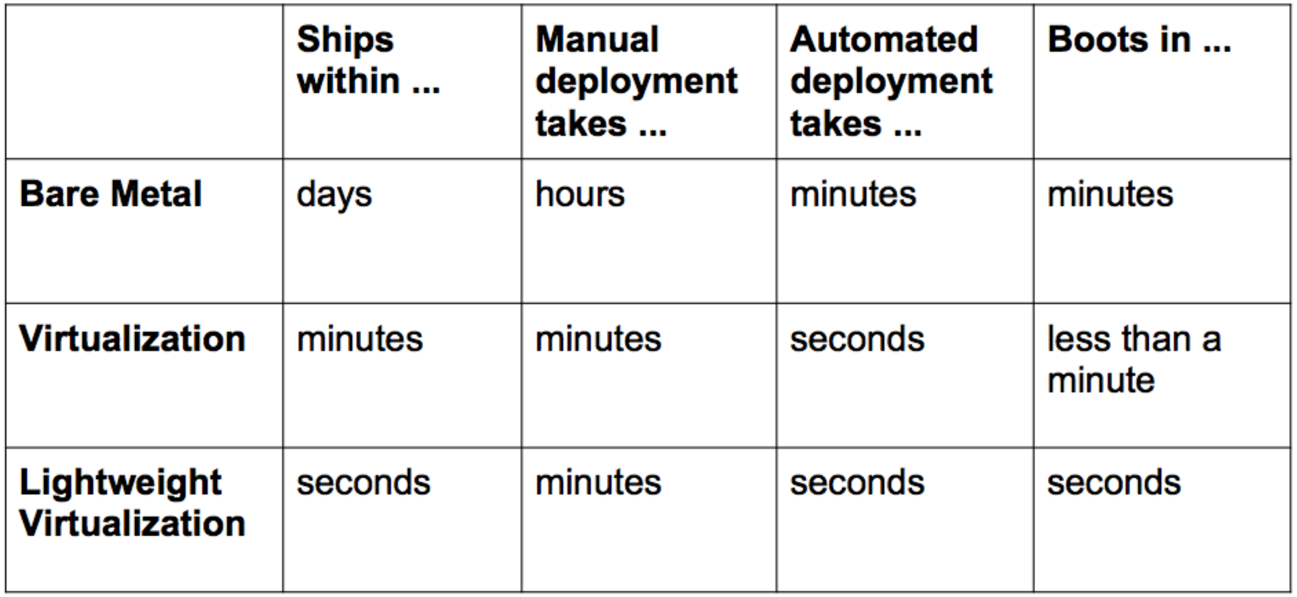
- Dockers and LXC relies on kernel support
- A container is a group of process
  - Running on the host
  - In an isolated environment
- Specific mecanisms
  - namespaces to isolate processes
  - cgroups to assign resources (CPU share, memory limit…)

Containers vs VMs

- Containers
  - Look like a VM from inside
  - Look like a normal process from outside
- VM
  - Contains a complete operating system
- On the host
  - Heavy virtualization will hide processes with “top” or “ps”
  - With containers, application processes will be visible

Containers vs VM

- Source : *lost …*
- Ships withing means physical delivery or download



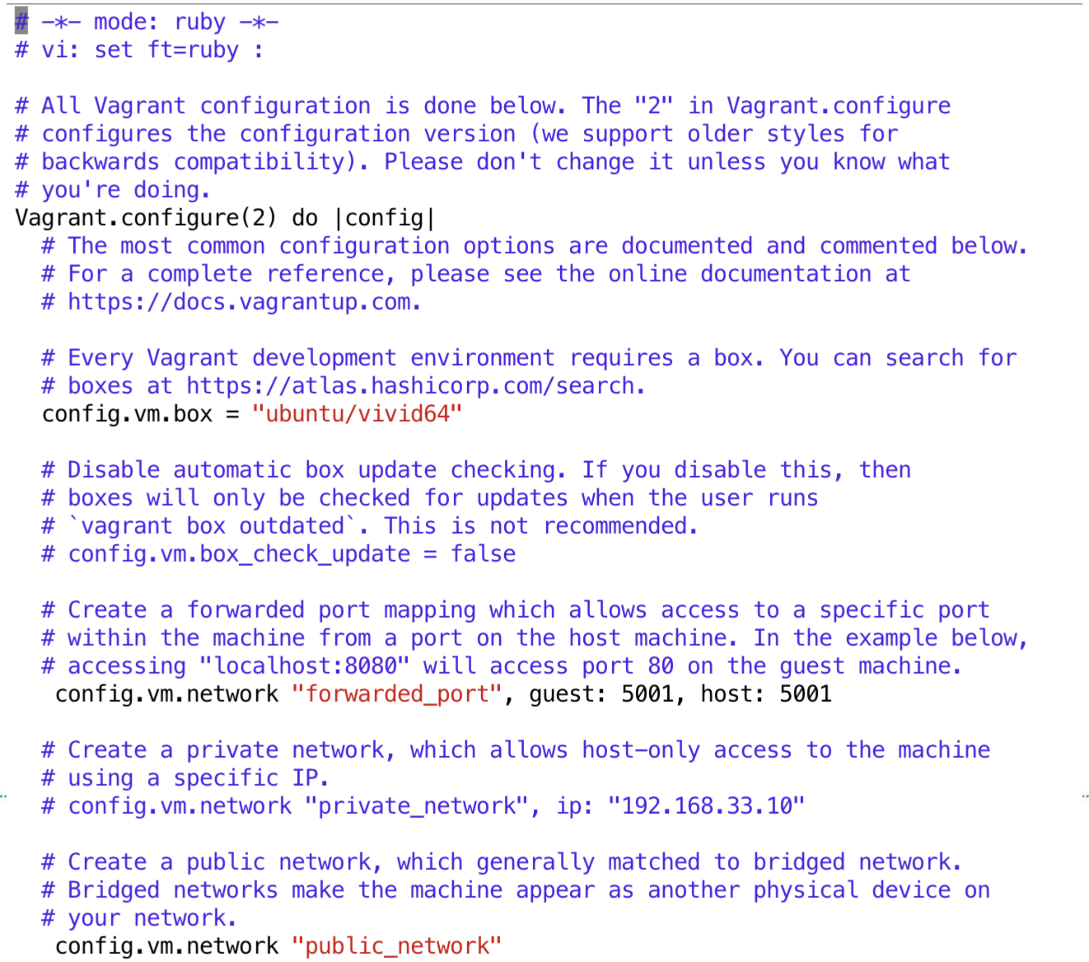
	Ships within …	Manual deployment takes …	Automated deployment takes …	Boots in …
Bare Metal	days	hours	minutes	minutes
Virtualization	minutes	minutes	seconds	less than a minute
Lightweight Virtualization	seconds	minutes	seconds	seconds

Management

Management of VMs

- Management of VMs
  - Vmware, Nutanix, IBM can offer management of a handful nodes of the same vendor
- Vagrant: Management of VMs a hypervisor independent approach
  - Notion of images (boxes in Vagrant)
  - Provisioning of VM: Puppet, Chef, Ansible to configure automatically the VMs
  - A single file that includes everything

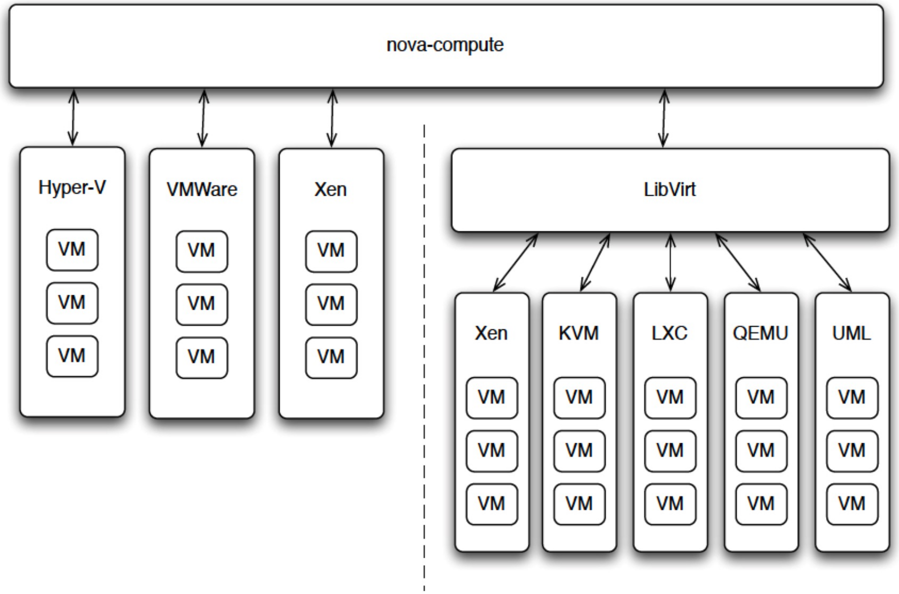
Vagrantfile (excerpt)



```
# -*- 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"
```

Cloud platforms

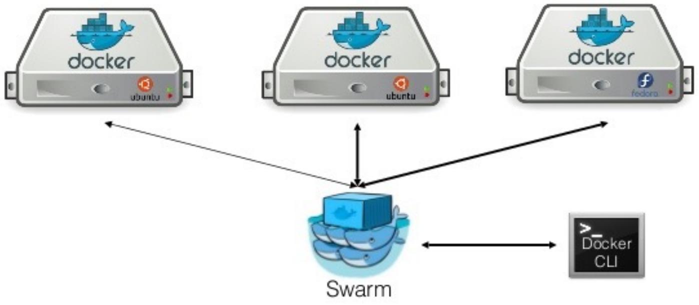
- Manage VMs at larger scale
- Openstack
  - Each function (management of VM, network, volumes…) is a component
  - Components interact through REST API
  - Can manage mixed hypervisors



Orchestration of containers

- Manage containers
  - On a single host : Docker, LXC
  - On several hosts : Docker Swarm, Kubernetes

With Docker Swarm


docker
ubuntu
docker
ubuntu
docker
fedora
Swarm
>_
Docker
CLI