TP1 - Cloud Infrastructures

First contact with OpenNebula

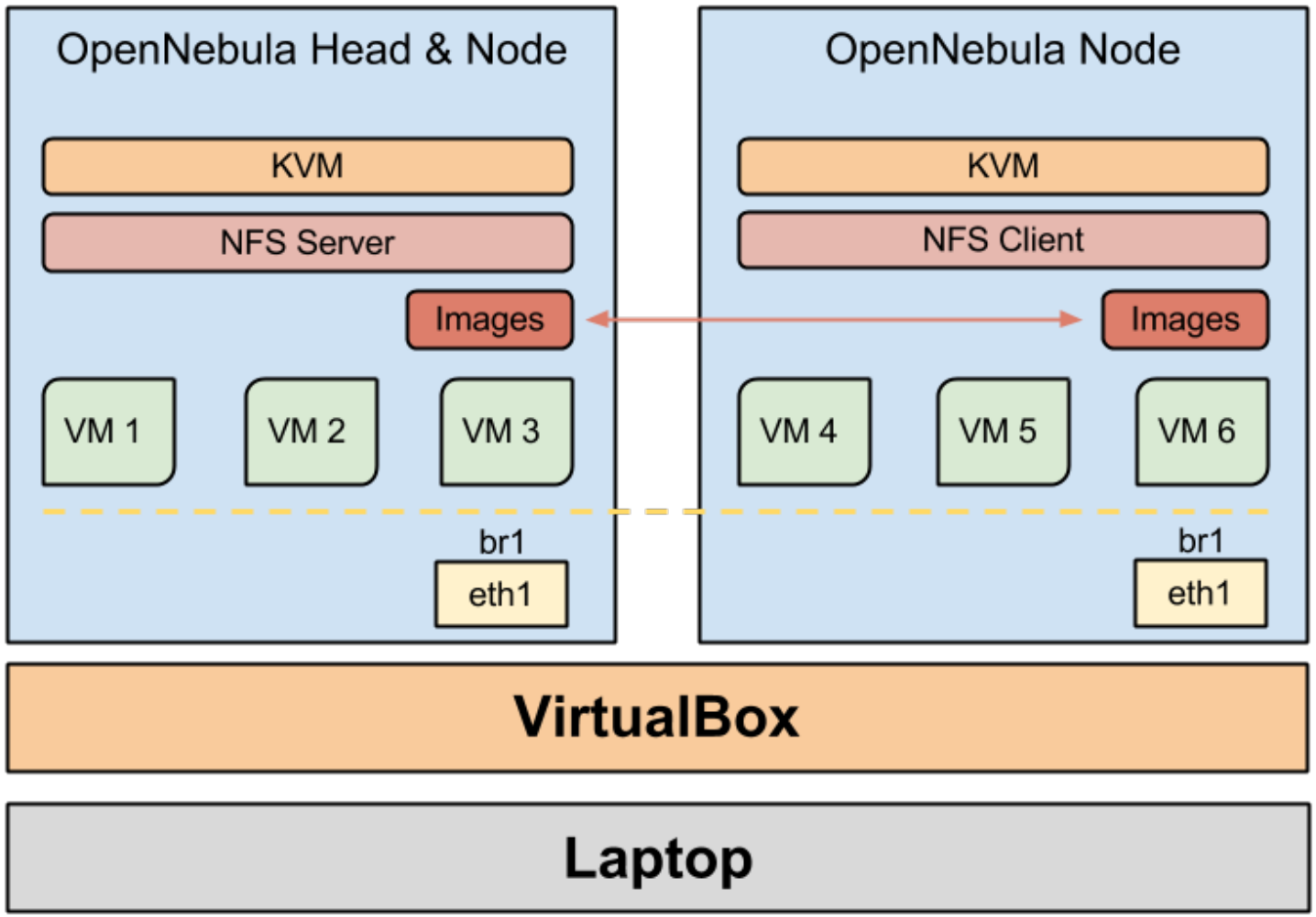
Juan Ángel Lorenzo del Castillo

ICC Cloud Infrastructures

EISTI Pau

2020-2021

Virtual Lab


OpenNebula Head & Node
KVM
NFS Server
Images
VM 1
VM 2
VM 3
br1
eth1
OpenNebula Node
KVM
NFS Client
Images
VM 4
VM 5
VM 6
br1
eth1
VirtualBox
Laptop

TP1 - Running OpenNebula

Hands on!

Objective

Getting to know OpenNebula's administration environment and its components.

1. Discovering Sunstone

1. Download and start the *OpenNebula VirtualBox Sandbox*:
   - VirtualBox image in
     http://vms.pau.eisti.fr/torrent/OpenNebula-5.2-Sandbox-VBOX.ova.torrent
   - Follow the steps from section 3.2 (*Create a new Virtual Machine*) on
     https://archives.opennebula.org/cloud:sandboxvirtualbox but make sure that your virtual machine has 1 GB of Memory size (at least) instead of the 512 MB recommended.
2. In a browser on your local machine, open the **Sunstone** GUI: http://localhost:9869
   - login: `oneadmin`
   - pasword: `opennebula`
3. Create a user account.
4. Exit the administrator account and open a session with the user account that you have just created.

Hands on!

1. Deploy two VMs, **vm1** et **vm2** with the default parameters:
   - Template `ttylinux`.
   - Capacity 0.1cpu, 64MB memory.
   - Hard drive 40 MB.
   - `Cloud` network interface.
2. Open the VMs console. Try to con connect from one VM to the other one (ssh, ping) to explain their network configuration:
   - Interfaces and networks of each VM.
   - Interfaces and networks of the frontend (sandbox VirtualBox) (login: `root`, password: `opennebula`)
   - Connection between the three of them.

Hands on!

2. Basic administration with Sunstone

1. Open an administrator session.
2. Create a Virtual Network called `external` in the address range `192.168.1.0`.
3. Start a new VM exclusively with the `external` interface attached. Explain the steps followed.
4. Start a new VM with the `Cloud` and `external` interfaces attached. Explain the steps followed.

Hands on!

3. Use of the command line

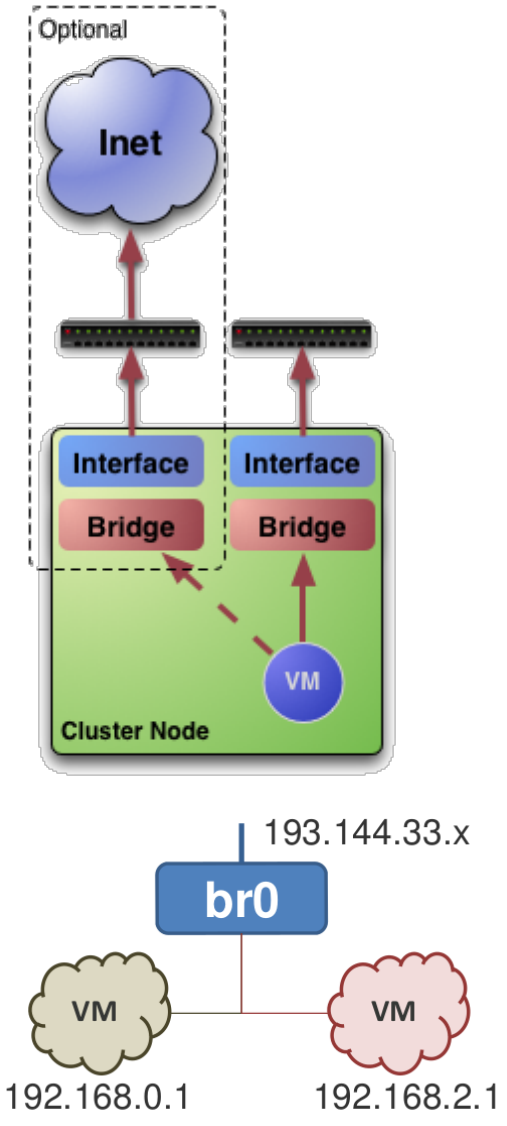
1. Repeat the steps from points 1 and 2 using (when possible) OpenNebula's administrator command line
   - Using the command console in the frontend.
   - `$ su - oneadmin`

```
$ one[TAB]
```

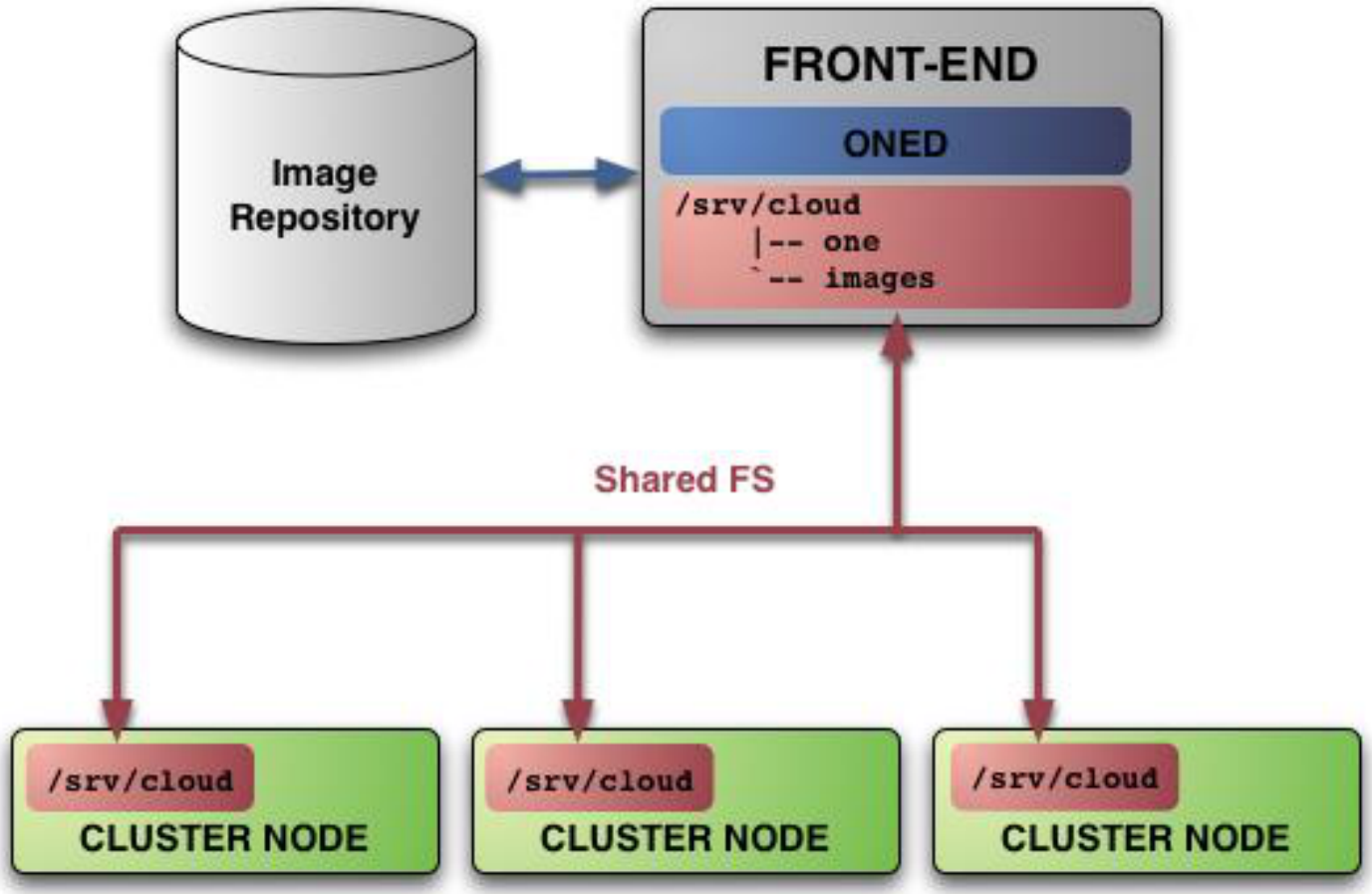
oneuser	Manage Users	**oneimage**	Manage Images
onegroup	Manage Groups	**onetemplate**	Manage Templates
oneacl	Manage ACLs	**onevm**	Manage VMs
onehost	Manage Hosts	**oneacct**	Accounting Tool
onecluster	Manage Clusters	**onemarket**	Marketplace Tool
onevnet	Manage Networks	**onedb**	DB Tool
onedatastore	Manage Datastores		

OpenNebula Networks

- A Virtual Network, or Vnet, in OpenNebula:
  - is used to define a MAC/IP address range that can be used by the VMs.
  - A Vnet is associated to a physical network through a bridge.
  - Vnets can be isolated (at *layer 2* level)
- How to define a Virtual Network
  - **Name** of the network
  - **Type**
    - **Fixed:** a set of IP/MAC *leases*
    - **Ranged:** to define an address range
  - **Bridge:** Name of the physical bridge in the physical host where the VM must connect its network interface.
- The Virtual networks can be managed also via the CLI: `onevnet`



OpenNebula Image Repository



Images

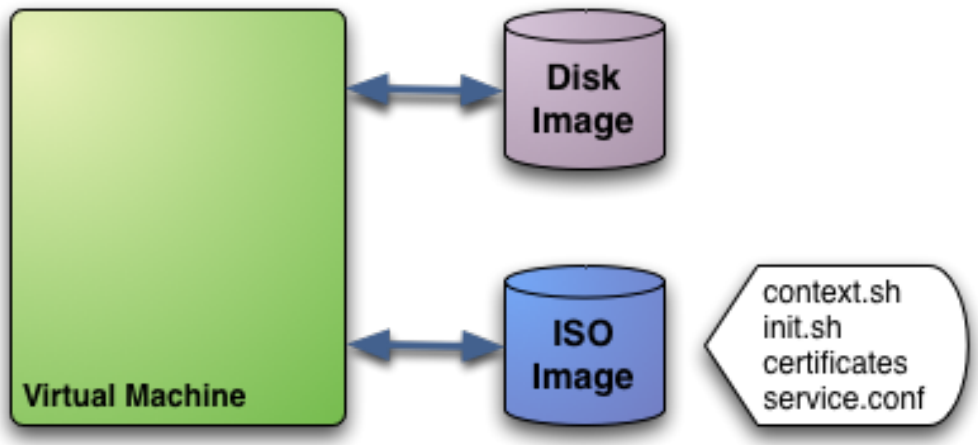
- In OpenNebula repository
  - A VM hard disk can be used as a device to store either an operating system or data.
  - Images can be **persistent** and/or public
  - The modifications performed to an image can be saved as a new image.
- Image types:
  - **OS** : Contains a working operating system
  - **CDROM** : Read-only data
  - **DATABLOCK** : Data storage (hard drive empty or with already existing data)
  - **RAMDISK** : File to use as a ramdisk
  - **KERNEL** : File to use as a kernel
  - **CONTEXT** : File to use in a CDROM drive as a contextualisation file

Templates

- Models to define VMs
- The same template can be used to create multiple VMs
- Templates comprise:
  - **Capacity** : Memory and CPU
  - **NICs** (Network interfaces) attached to one or several vLANs
  - A set of **hard drives**
  - An optional file to store a running VM's memory status plus information about the hypervisor.

Contextualisation

- Gives configuration information to a VM that has just been created.
- F.ex : Boot scripts, connexion ids, network configuration, etc.



Some practical information

- The configuration file is on `/etc/one/oned.conf`
- Logs of OpenNebula's daemon (`oned`) : `/var/log/one/oned.log`
- Logs of the scheduler : `/var/log/one/sched.log`
- ▸ opennebula_5.2_operation_guide.pdf

Hands on!

4. Create a new Template...

1. ... that boots from an existing hard disk containing an operating system
2. with two network interfaces
3. with an additional disk in ext3 format
4. that, when creating the VM, asks the user for a welcome message to show in the VMs console at boot time.
5. Once the VM has started make sure that you can access (`mount`) the additional hard disk
6. Using iptables, make sure you give the VMs access to the internet through the frontend (Virtualbox OpenNebula VM)
7. Repeat the process without using Sunstone (OpenNebula's GUI)

TP1 - Running OpenNebula

5. Virtual machine Images

1. Start a new VM with an empty hard disk. Install an operating system of your choice.Explain the steps followed
2. Export the new VM image to make it available for all users

6. Contextualisation of virtual machines

- Create a user SSH key
- Create a contextualisation file to apply the following actions to each of your user VMs:
  1. Start the sshd service.
  2. Disable the access by login/password
  3. Insert the user SSH key and enable access by SSH key