OpenNebula

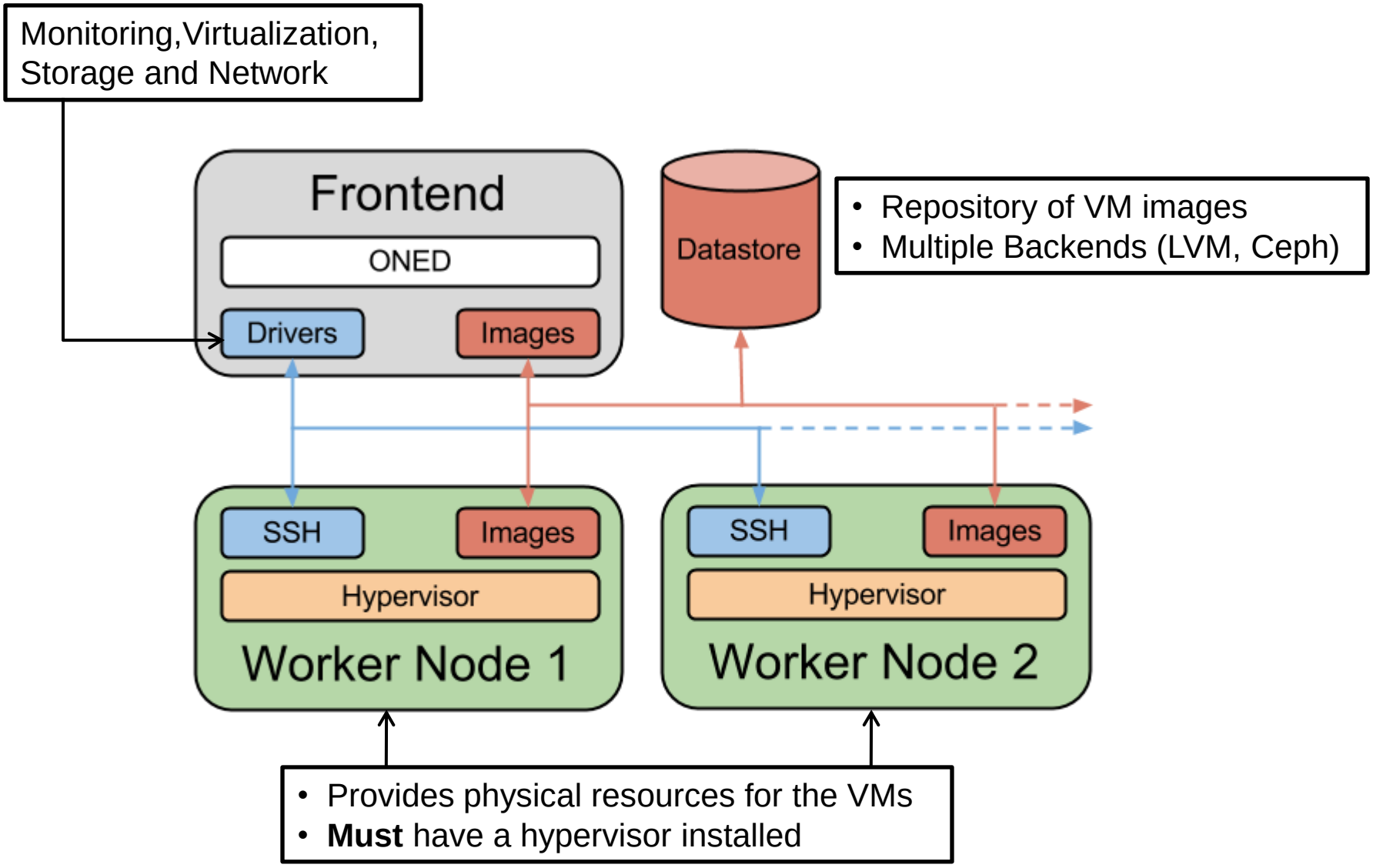
Tutorial

OpenNebula.org

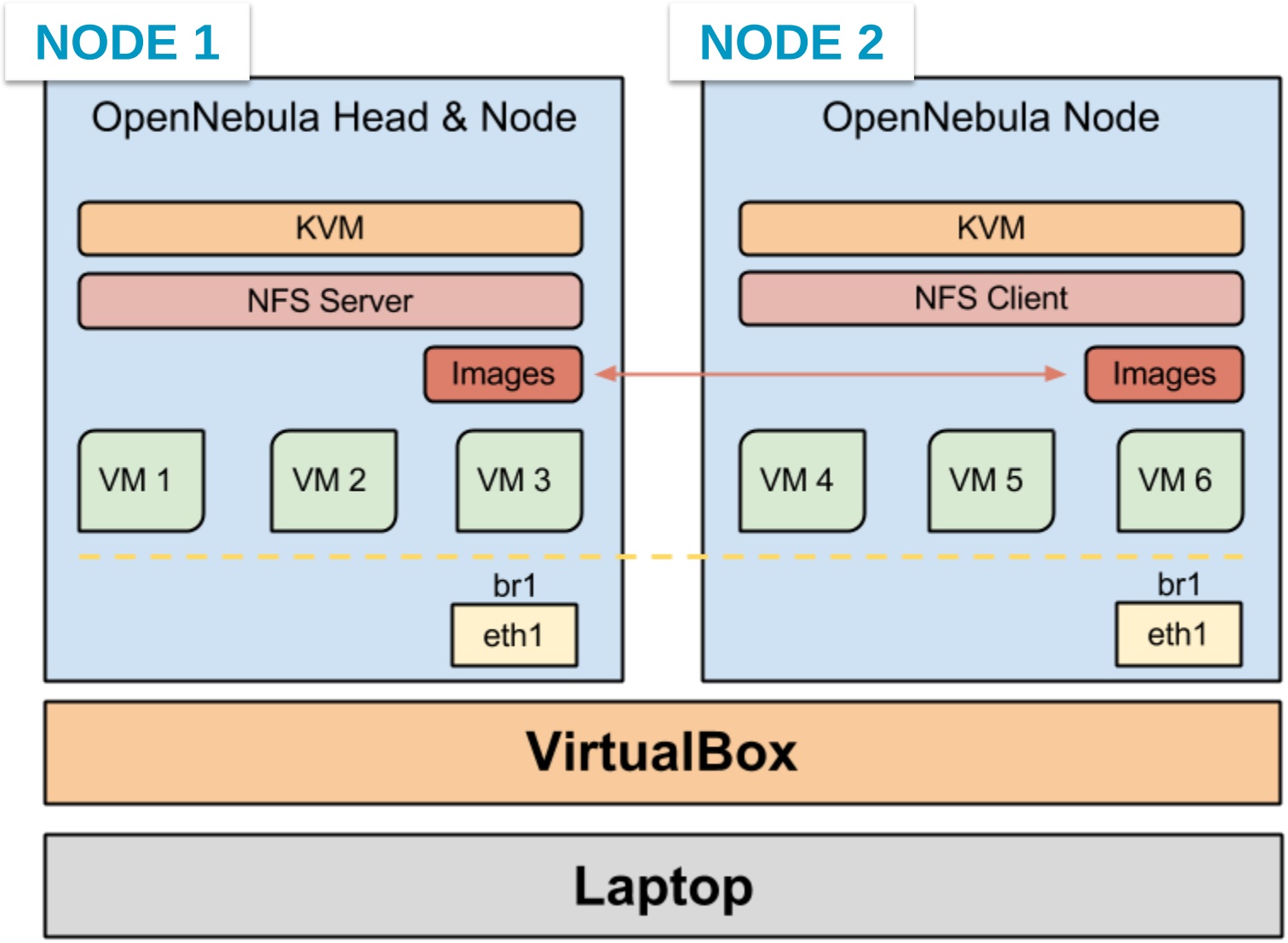
© OpenNebula Project. Creative Commons Attribution-NonCommercial-ShareAlike License

A Typical OpenNebula Environment

Planning the Installation



Planning the Installation



Hands on (node1)

Install Packages

```
# yum install opennebula-server opennebula-sunstone
  opennebula-node-kvm opennebula-flow opennebula-gate

# /var/local/tutorial/configure_tutorial.sh

# echo oneadmin:opennebula > /var/lib/one/.one/one_auth
```

Hands on (node1)

Start the services

```
# service opennebula start

# service opennebula-sunstone start

# service libvirtd restart
```

Switch to oneadmin

```
# su - oneadmin

$ oneflow-server start

$ onegate-server start
```

Hands on (node1)

Overview of the CLI (as oneadmin)

```
# su - oneadmin

$ oneuser show

$ oneuser -h

$ one[TAB]
```

Hands on (node1) !

OpenNebula CLI Commands

```
$ 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		

Hands on (node2)

Configure the hypervisor node

```
# ssh root@node2

# yum install opennebula-node-kvm

# service libvirtd restart
```

Hands on! (always node1 from now on)

OpenNebula needs passwordless ssh access to all the nodes from all the nodes

```
# (as oneadmin)

$ ssh-keyscan node1 node2 > ~/.ssh/known_hosts

# test it!

$ ssh node2
$ exit
$ ssh node1
$ exit
```

Try out Sunstone!

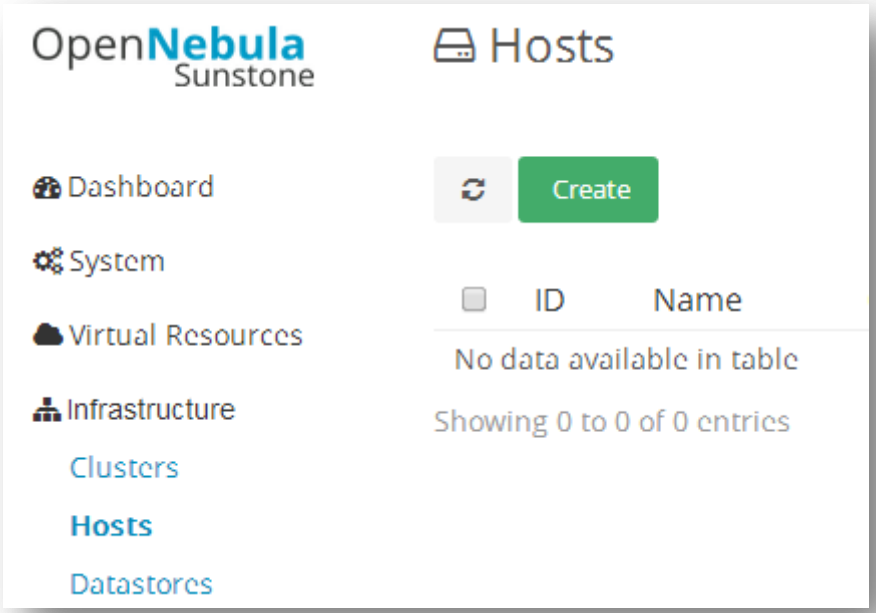
http://localhost:9869

Login: oneadmin

Password: opennebula

Hands on! (Sunstone)

- **Create** one host in Sunstone: **node1**
  - Type: KVM
  - Network: Default (dummy)
  - Cluster: Default (none)
  - Hostname: node1
- Watch transition **INIT** => **ON**



- **Click on the row** for more information
  - Automatic gathering of monitoring data
  - Take a look at the graphs

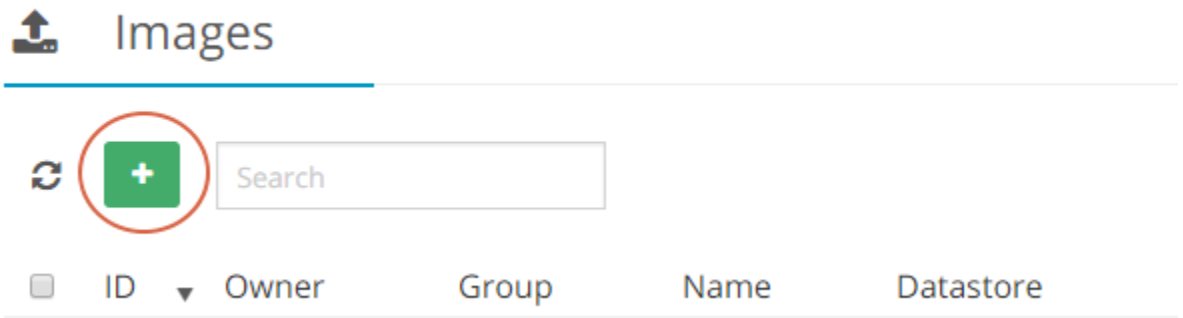
Hands on! (CLI)

(always as oneadmin in the Frontend – node 1)

```
$ onehost -help
$ onehost create –help

$ onehost create node2 –i kvm -v kvm -n dummy
$ onehost list
$ onehost top

# Wait for ON ... and then CTRL-C

$ onehost show node2
$ onehost show 1
$ onehost show -x 1
```

Hands on! (Sunstone)

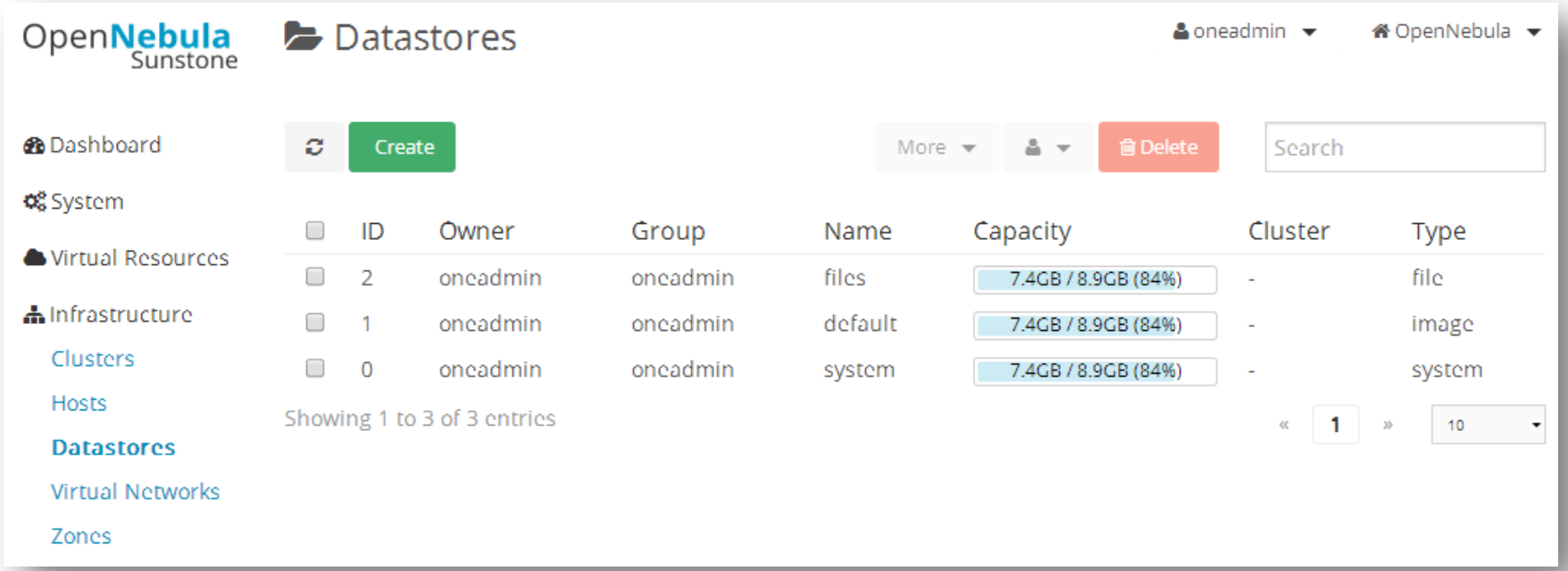
- Create a new Image



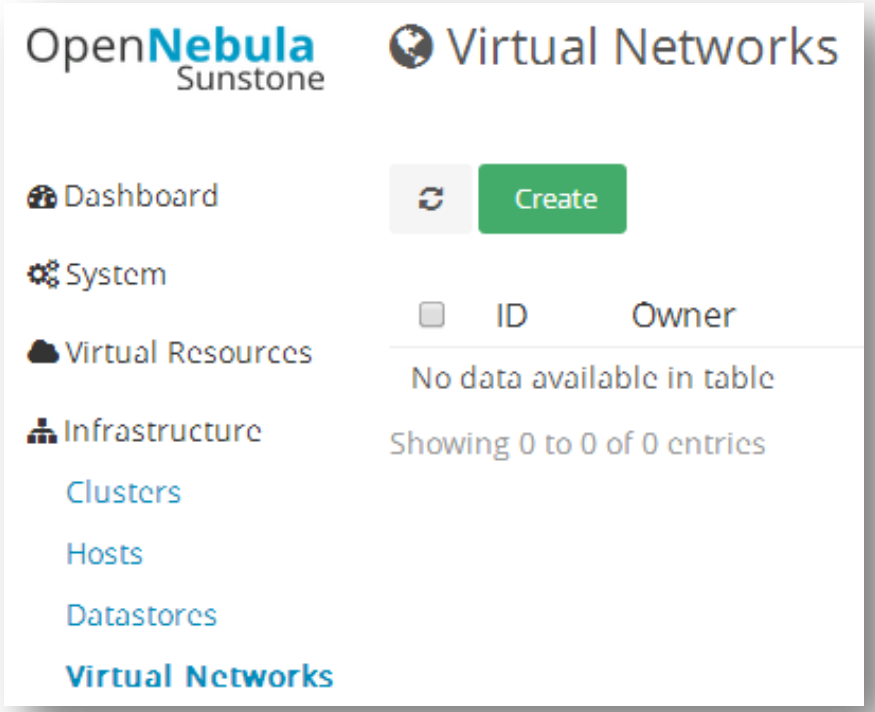
- **Name**: `ttylinux`
- **Path**: `/var/local/tutorial/ttyvd-context.qcow2`
- **Device Prefix**: `vd`
- **Driver**: `qcow2`

Basic Usage - Datastores

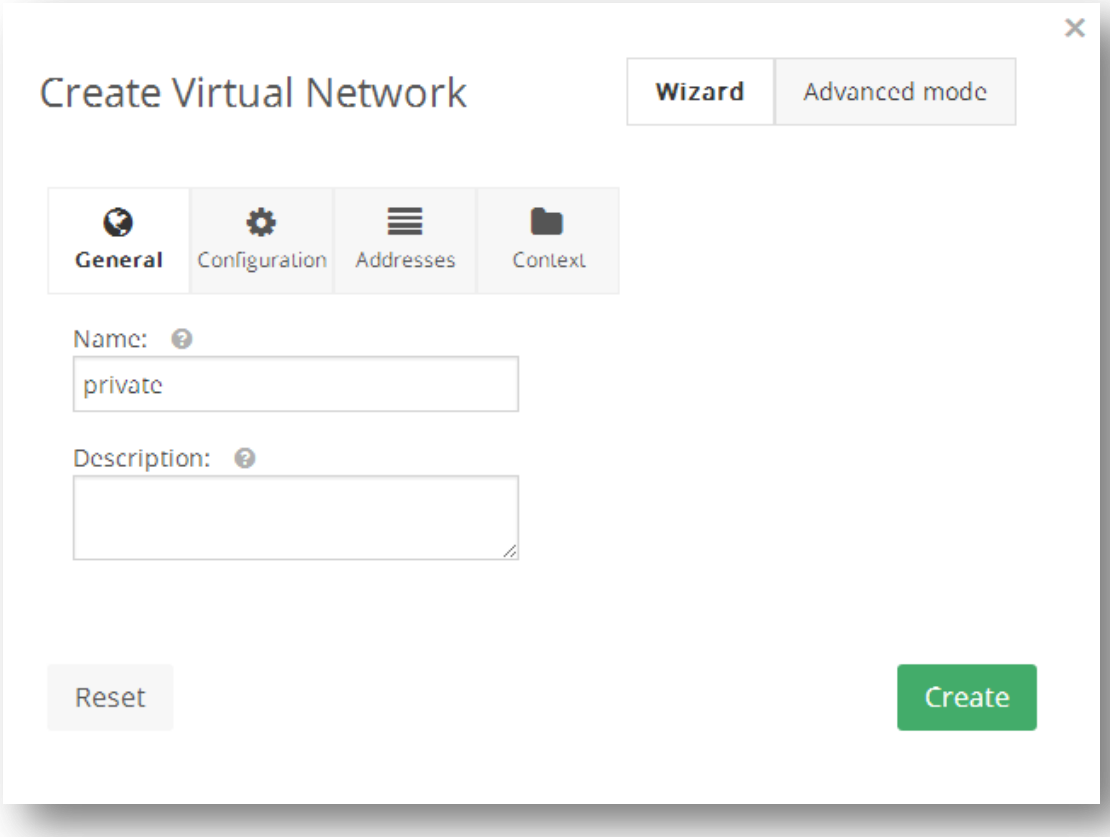
Hands on! (Sunstone)



Hands on! (Sunstone)



Hands on! (Sunstone)



General

- **Name**: *private*

Configuration

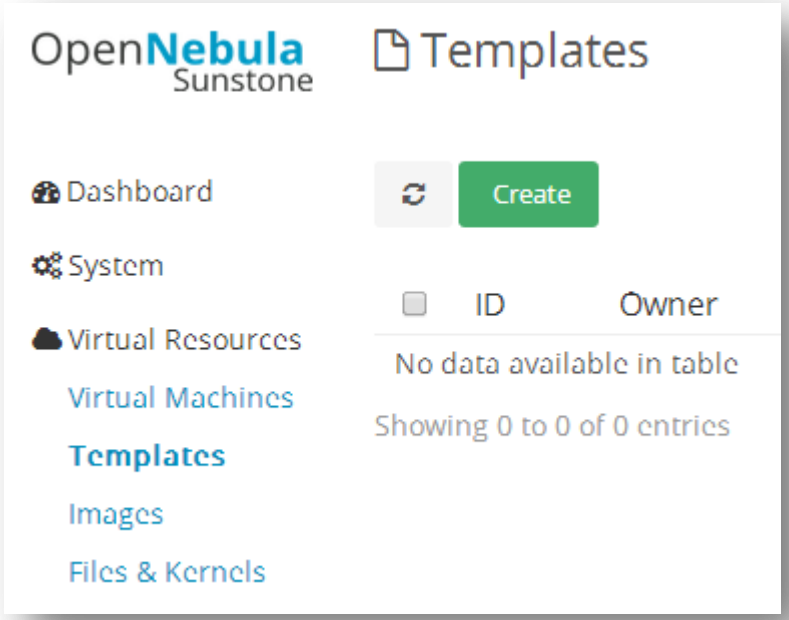
- **Bridge**: *br1*

Addresses

- **IP start**: *192.168.0.100*
- **Size**: *100*

Hands on! (Sunstone)

- A template is a Virtual Machine definition ready to be **instantiated**
- It has CPU, Memory, Disks, NIC, Graphical Ports, etc...



General

- Name: ttylinux
- Logo: Linux
- Description: Testing VM
- **CPU**: 0.1
- **Memory**: 64

Storage

- Click **ttylinux**

Network

- Click **private**

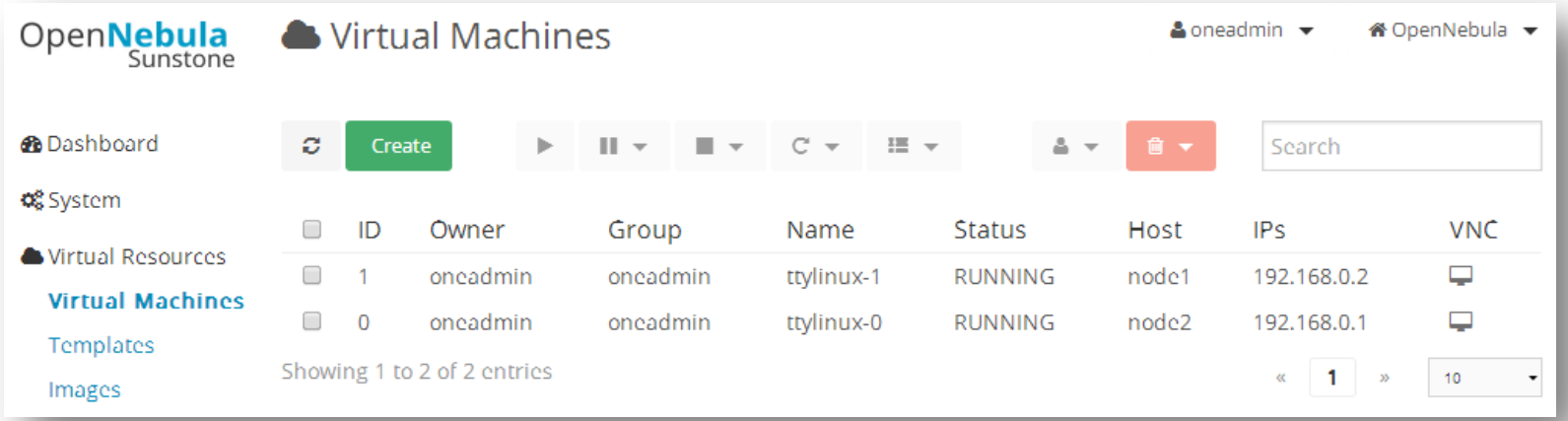
Input/Output

- Click **VNC** and add **Keymap**

Context

- Add OneGate token

Hands on! (Sunstone)



- **Instantiate** the template
- Deploy 2 VMs
- Leave the name blank
- Open **Virtual Machines**
- Watch the transition **PENDING** => **RUNNING**
- Deployed in different hosts

- **VNC** (*root* / *password*)
- ifconfig: configured using **context**
- **migrate**
- **live-migrate**
- **ping** the other machine

Hands on!

Login to the first VM and look at contextualization

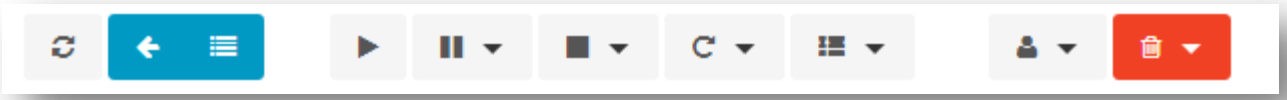
```
ttylinux ver 16.1 [bricolage]
x86_64 class Linux kernel 3.7.1  (/dev/tty1)
The initial "root" and "user" password is "password".
yuki login: root
Password:

Chop wood, carry water.

# cat /mnt/context.sh
# Context variables generated by OpenNebula
DISK_ID='1'
ETH0_IP='192.168.0.1'
ETH0_MAC='02:00:c0:a8:00:01'
NETWORK='YES'
TARGET='hda'
# _
```

Take a look at the contextualization files:

```
# cat /etc/rc.d/init.d/onecontext
# ls /etc/one-context.d/
00-network          01-dns              02-ssh_public_key  03-hostname
# _                                     _ _
```

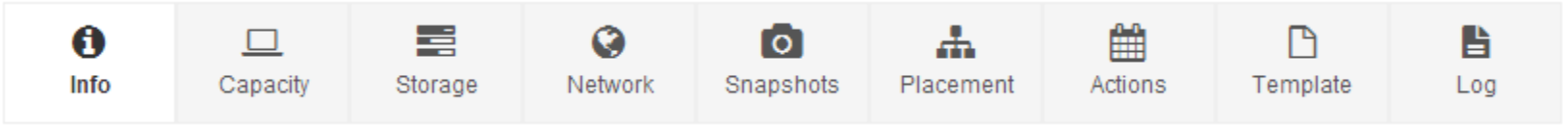
Main VM actions

Basic Usage – VMs



suspend	VM state saved. Kept in the host.
power off (--hard)	Powers off a VM. Kept in the host.
stop	VM state saved. Taken to the system datastore.
undeploy (--hard)	Powers off a VM. Taken to the system datastore.
reboot (--hard)	Reboots the VM.
delete --recreate	Cleans the VM and moves it to PENDING.
shutdown (--hard)	Powers off a VM, cleans host and VM is removed from OpenNebula.
delete	VM is immediately destroyed regardless of state. Recommended only for **oneadmin**.

Hands on! (Sunstone)



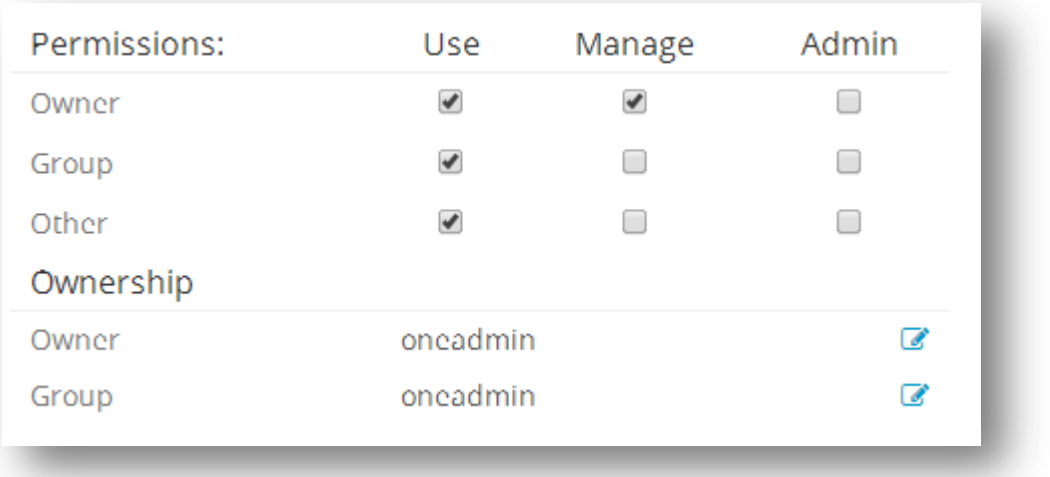
- **Storage**: Attach new disk
  - Create new Image => Type: "*Datablock*"; "*Empty Datablock*"; Size: *100*; FS Type: *"qcow2"* ; Device Prefix: "*vd"; Driver: "qcow2"*

```
# bash
# echo 1 > /sys/bus/pci/rescan
```

- **Snapshot**
  - Take (system) snapshot
  - Modify the VM
  - Revert
- **Capacity**: Resize VM capacity

Hands on! (Sunstone)

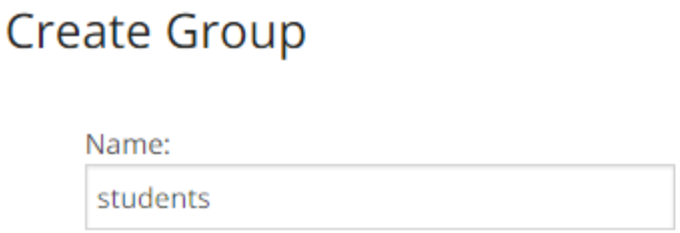
Make the all the resources previously created usable by everybody

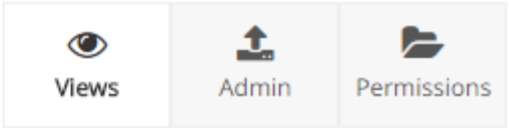


Apply to all the images, templates and virtual networks.

Owner could be changed too => **chown** and **chmod**.

Hands on!





Admin: Click **Create an administrator user**
Username: students-admin
Password: …

Check the **Permissions** tab (no modifications)

Associated ACLs

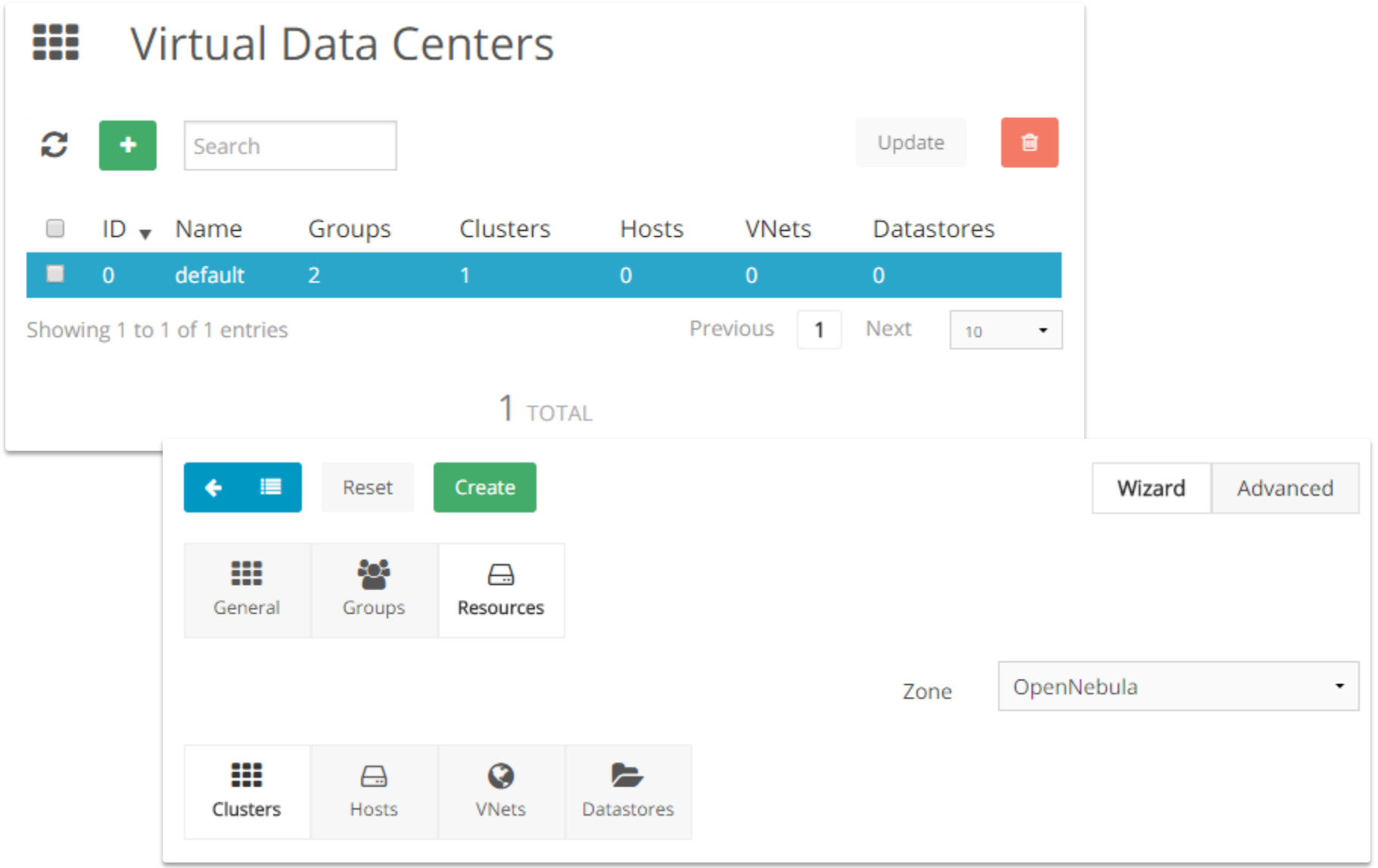
Groups

Hands on!

When a group is created, a new set of **ACLs** are introduced

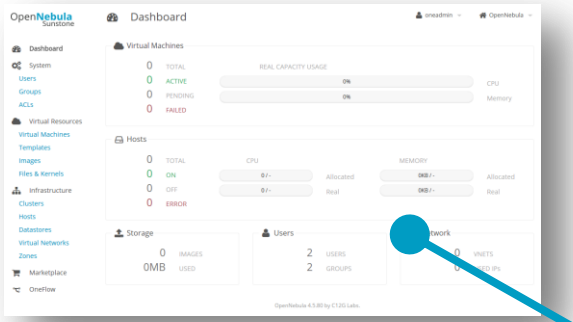
ID	Applies to	Affected resources	Resource ID / Owned by	Allowed operations	Zone
8	User 2	Virtual Machines, Images, VM Templates, Documents	Group students	use, manage, create	All
7	User 2	Users	Group students	use, manage, admin, create	All
6	Group students	Virtual Machines, Images, VM Templates, Documents	All	create	All
5	Group students	Virtual Networks, Datastores	All	use	OpenNebula
4	Group students	Hosts	All	manage	OpenNebula

The Power of VDCs

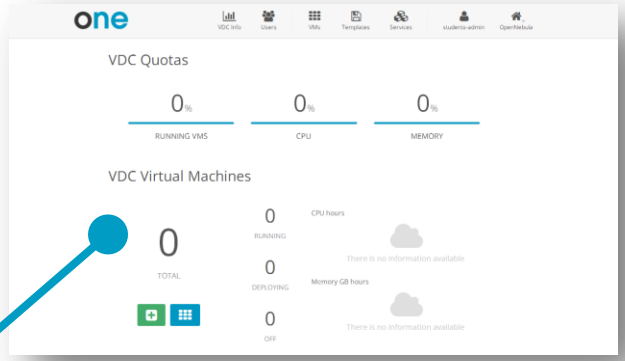


OpenNebula Views

Admin

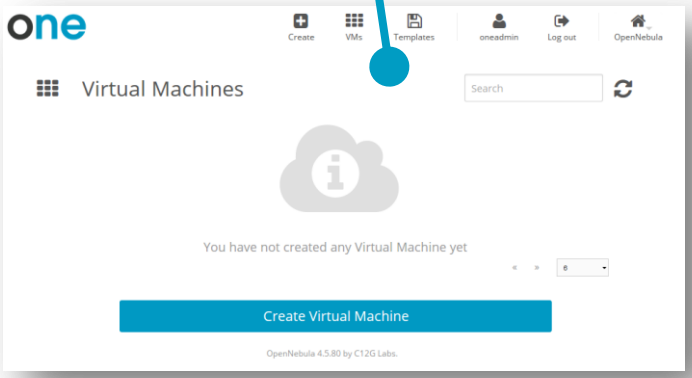


Group Admin



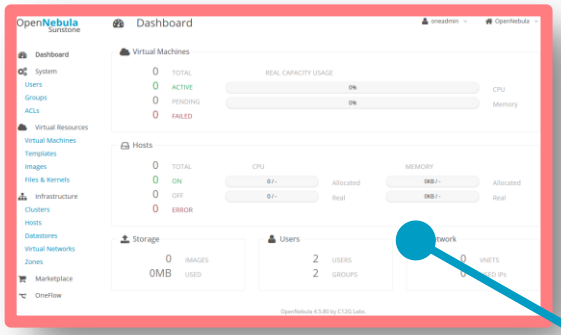
OpenNebula

Cloud User

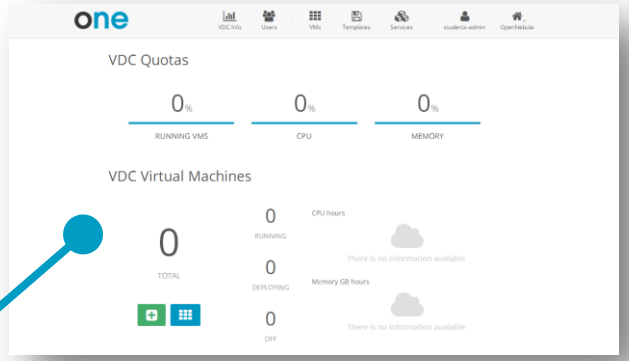


OpenNebula Views

Admin

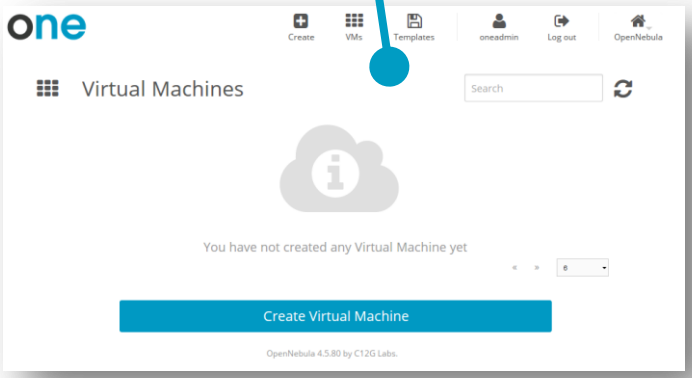


Group Admin



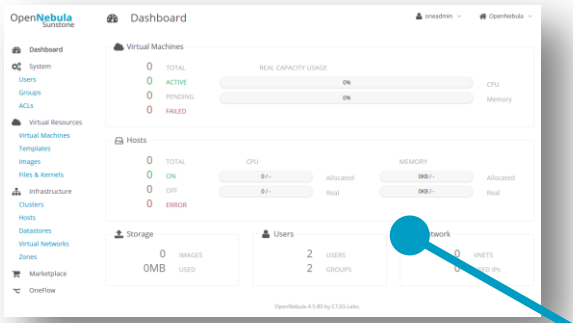
OpenNebula

Cloud User

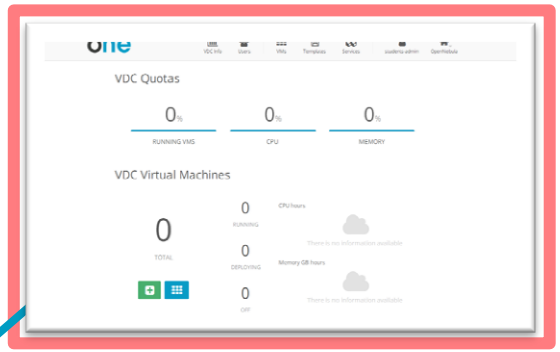


Configure your Cloud from the Admin perspective

Admin

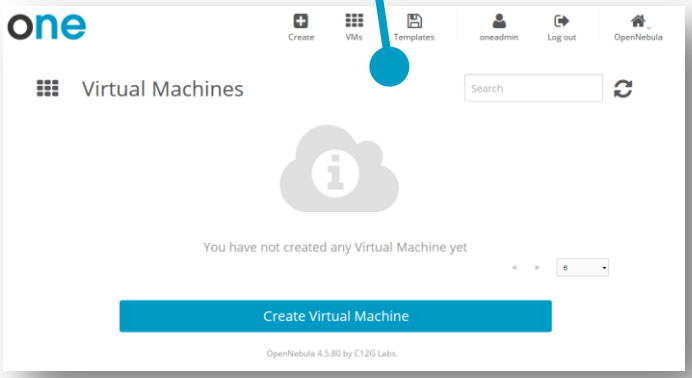


Group Admin



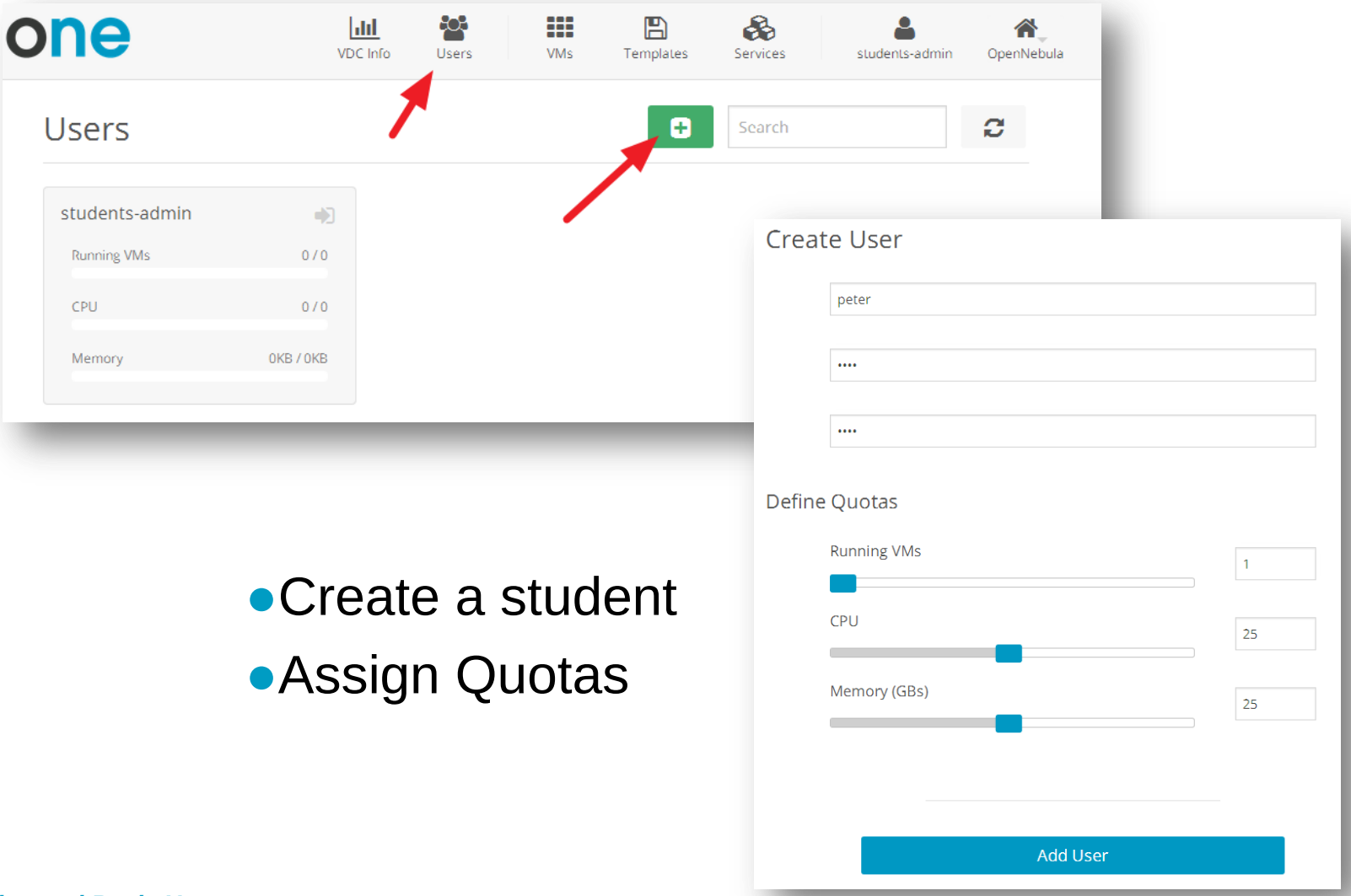
OpenNebula

Cloud User



Hands on!

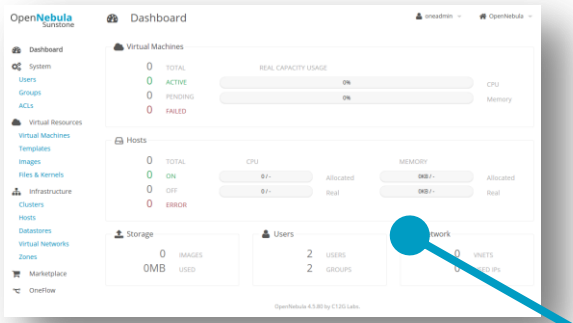
Login as students-admin



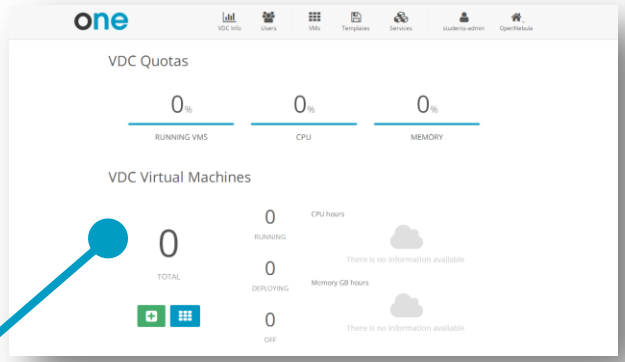
- Create a student
- Assign Quotas

OpenNebula Views

Admin

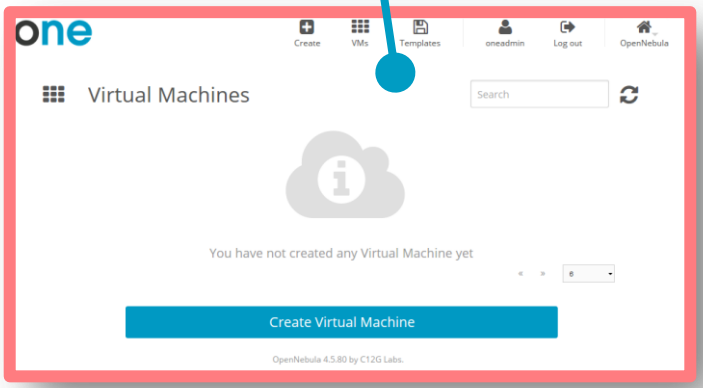


Group Admin



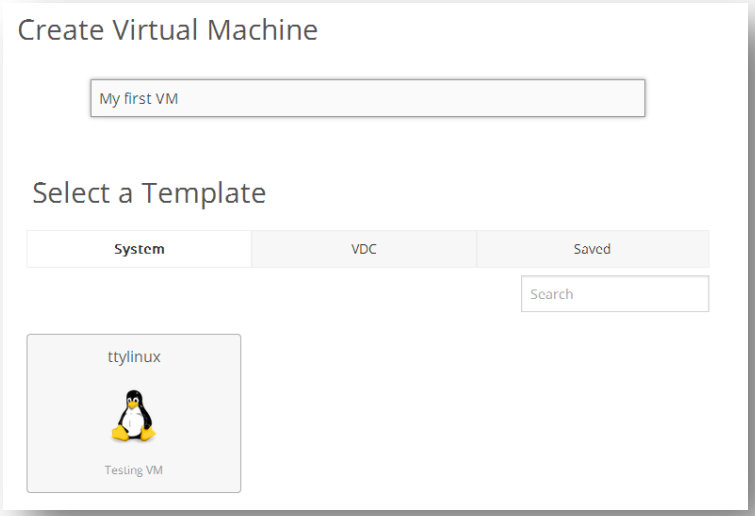
OpenNebula

Cloud User



Hands on!

Login as the student and instantiate a new VM

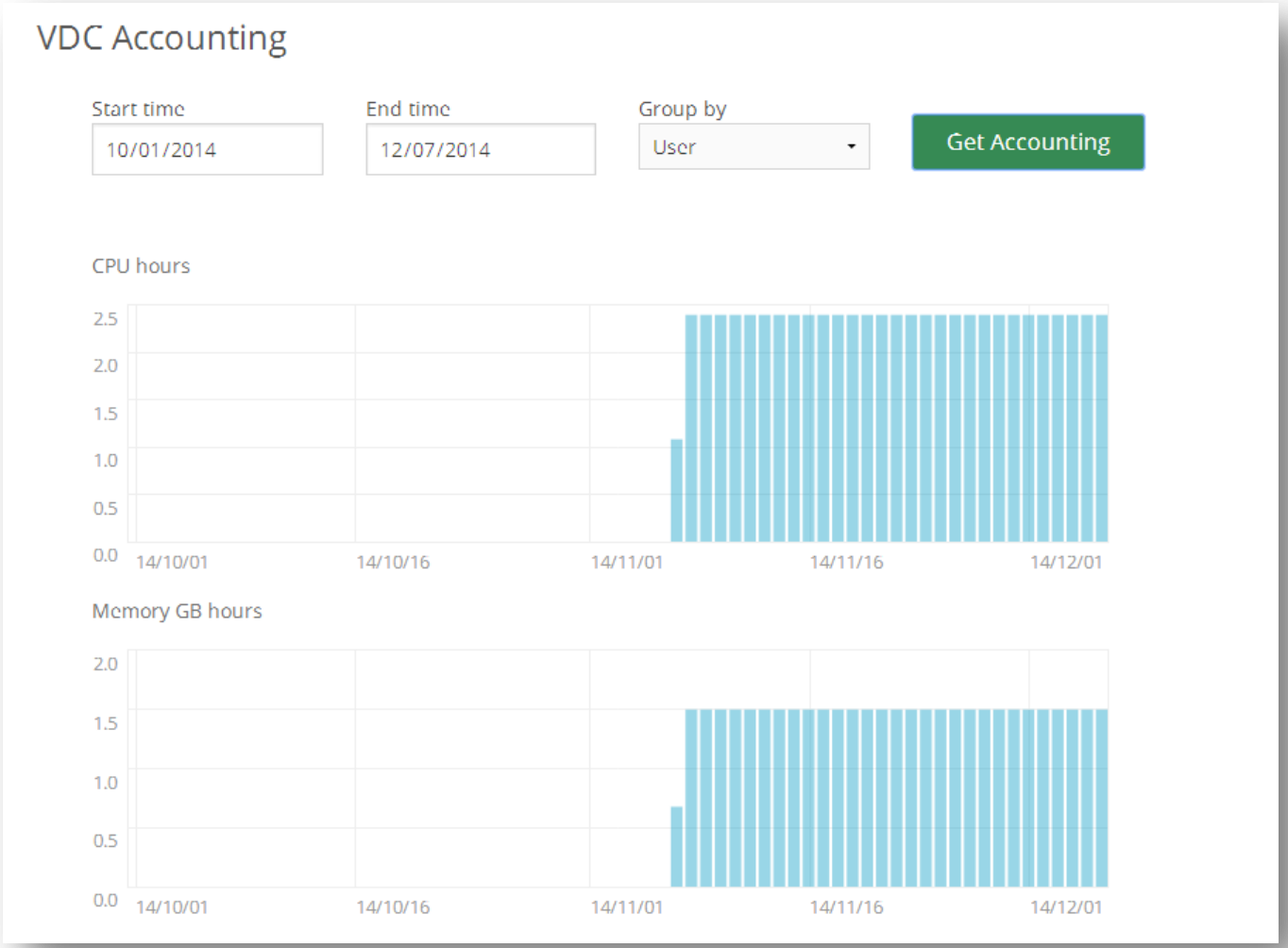


- Select the *ttylinux* template
- Network *private*

- Explore the VM actions. **Power Off** and **Save VM**
- With the *oneadmin* account see the new template and image
- Other options: SSH Key, Quotas, Settings

Hands on!

Login as students-admin



Other Features

Federation

OpenNebula can scale by federation many OpenNebula instances.

Scheduler

The OpenNebula Scheduler is extremely flexible. Write your own rules you want to guarantee that your vms end up wherever you need them.

OneGate

Send custom Metrics to OpenNebula

AppMarket

Deliver appliances ready to be consumed

CloudBursting

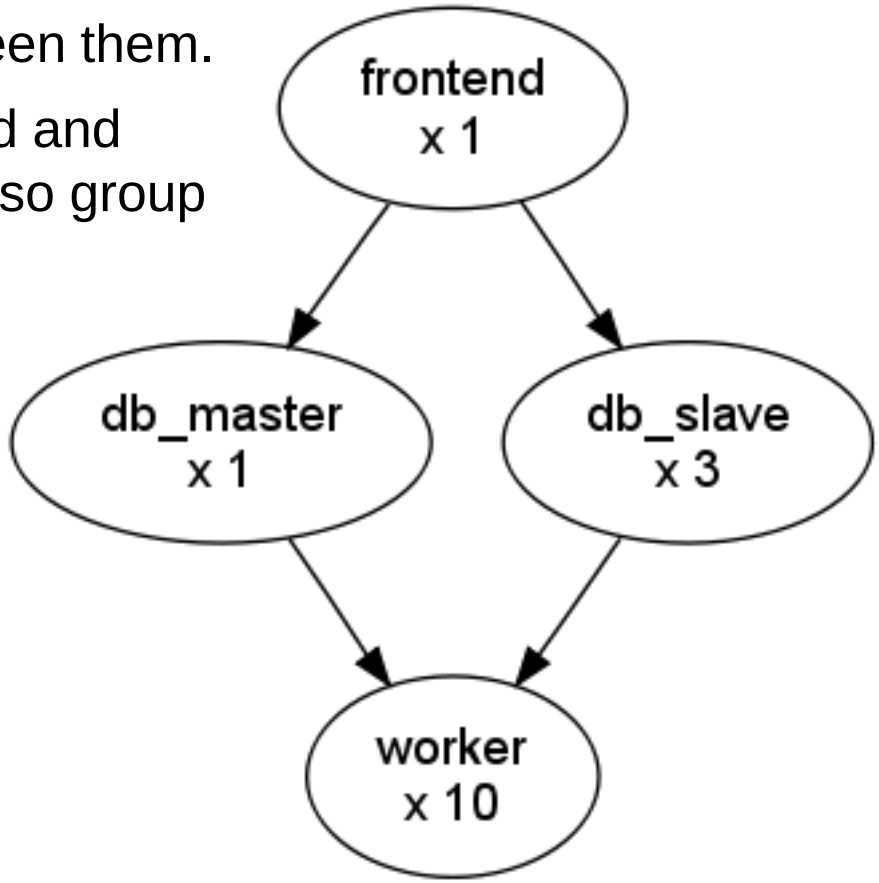
Scale out your private resources to EC2

vOneCloud

vCloud Director Replacement with al the OpenNebula Cloud benefits

Managing Multi-tier services

- OneFlow allows users and administrators to define, execute and manage multi-tiered applications
  - services composed of interconnected VMs with dependencies between them.
  - each group of VMs is deployed and managed as a single entity -> so group and ACL management apply



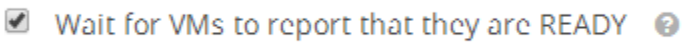
Deployment Strategies

- **none**: All roles are deployed at the same time
- **straight**: Each Role is deployed when all its parent Roles are RUNNING

Service is running when all the Roles are RUNNING.

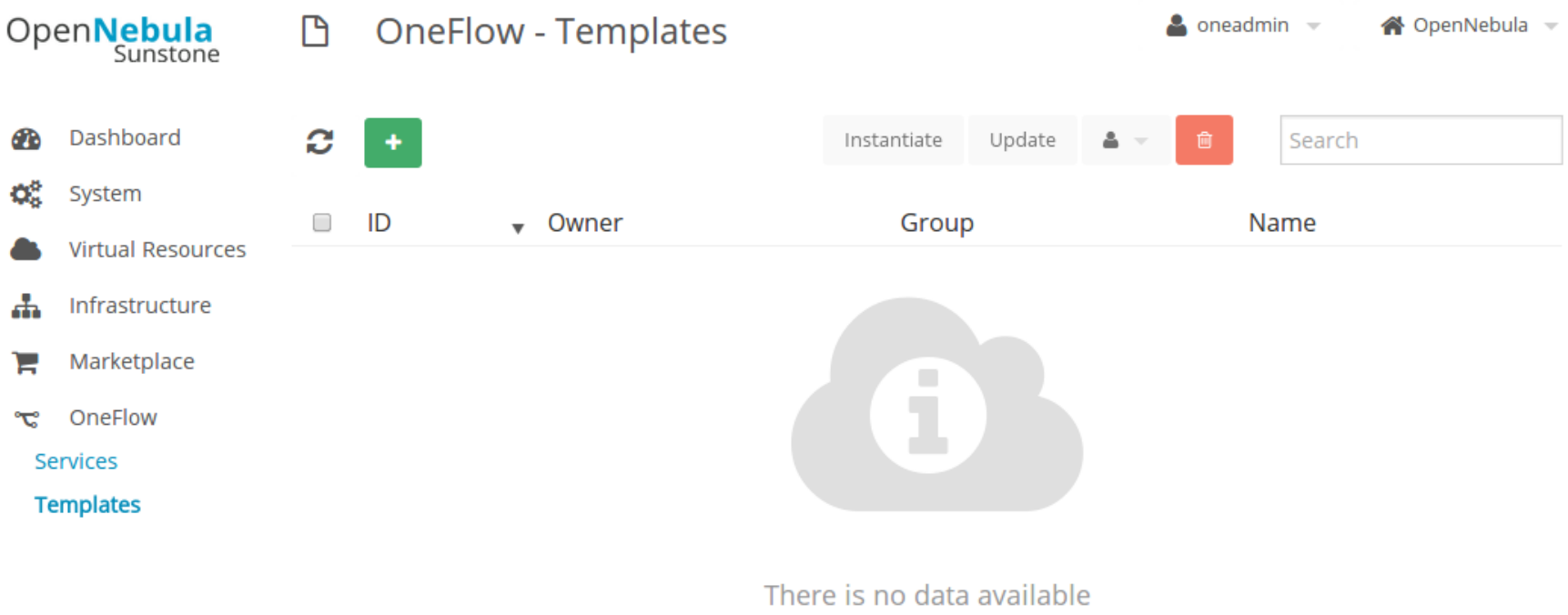
Running State

A role will not be considered to be running unless all the VMs in that role report to OpenNebula that thy are running, if this checkbox is enabled:

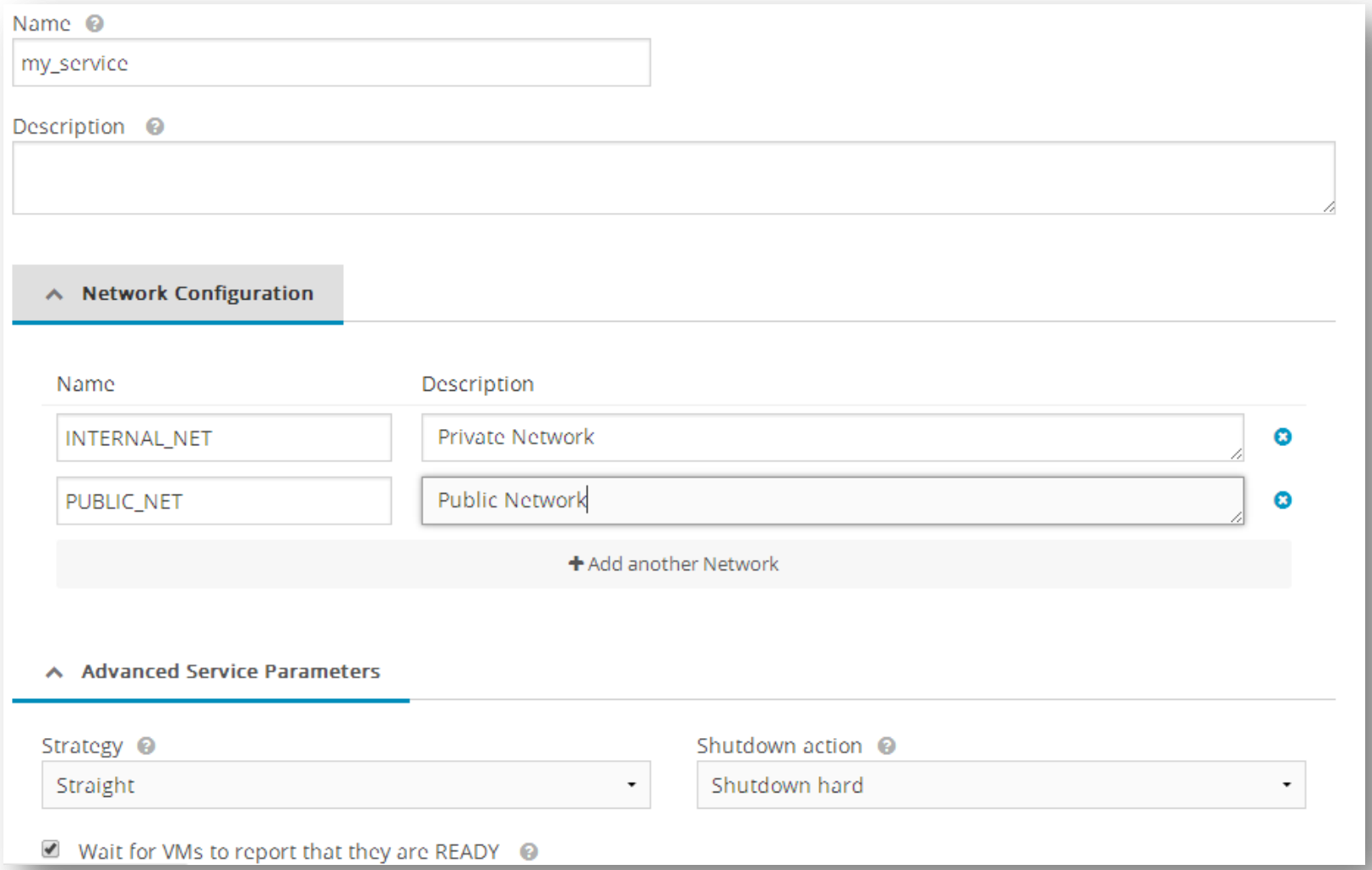


To report it, the VM will use a very simple HTTP request (curl, wget, …) and the OneGate token to authenticate.

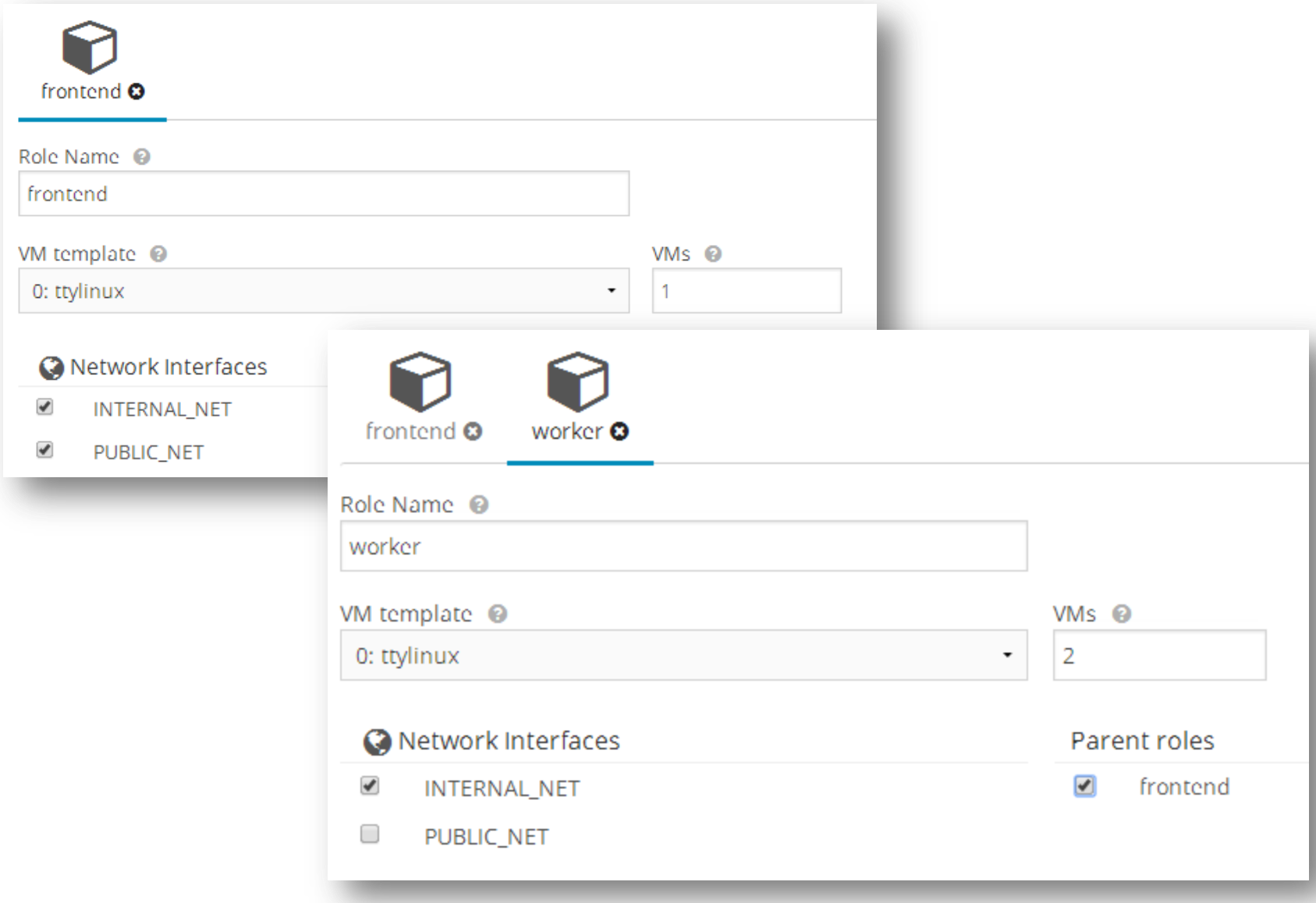
Hands On – Template Creation



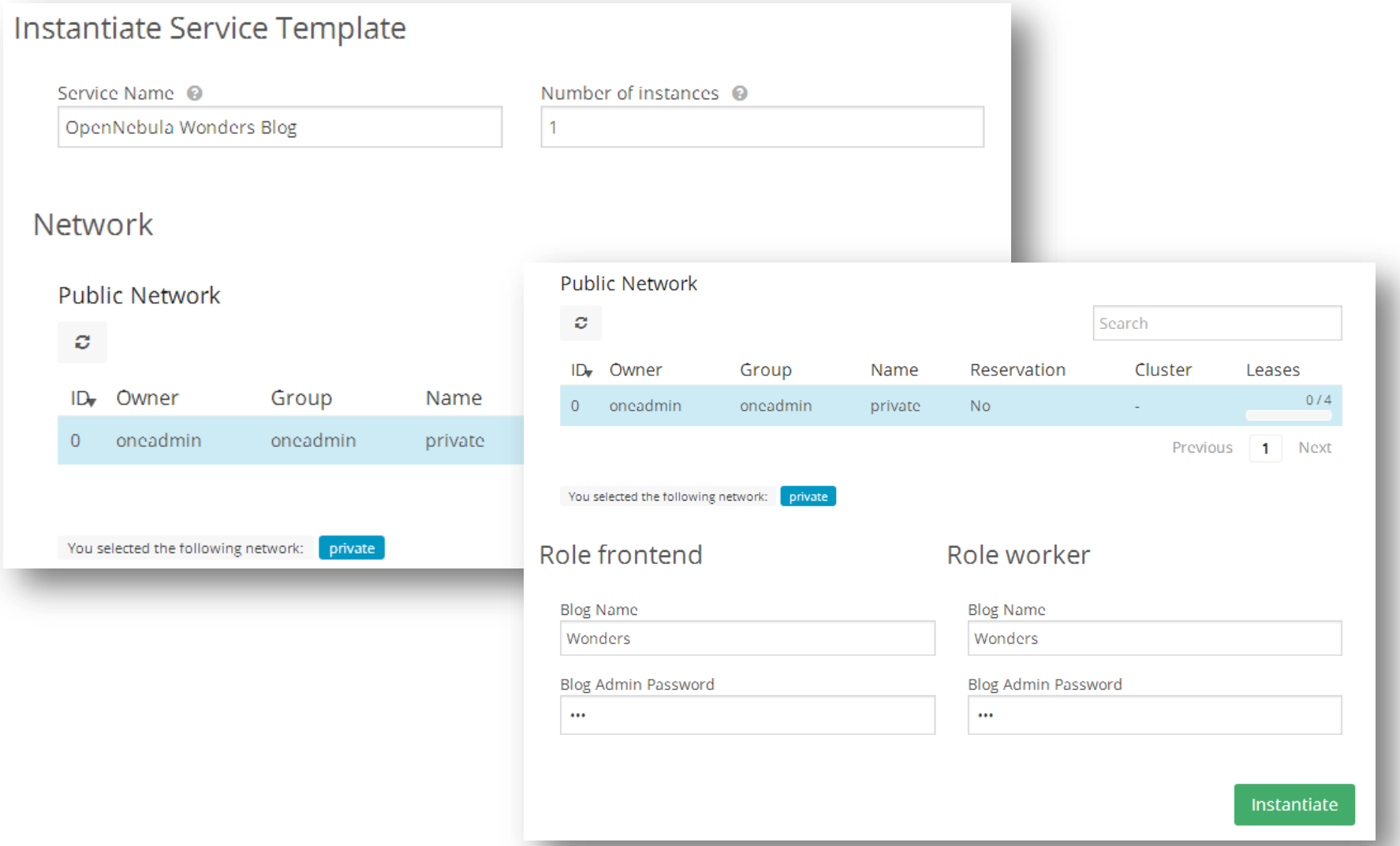
Hands On – Service Template and Frontend Role



Hands On – Service Template Database Master Role

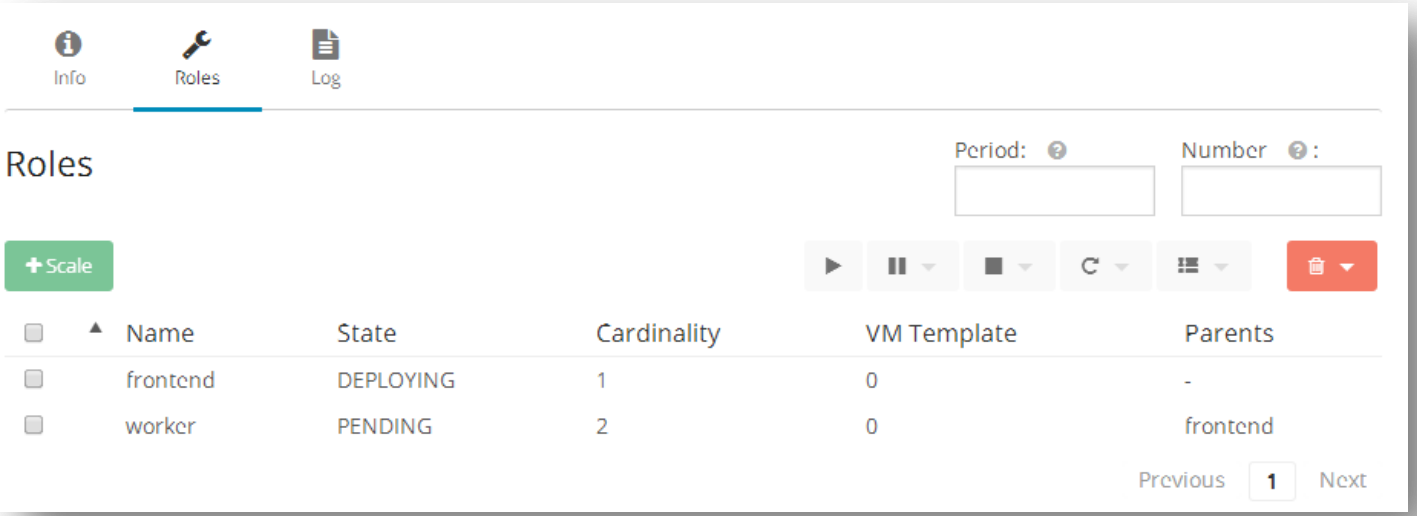


Hands On – Instantiate the Template



Hands On – Service Information

worker role will not be deployed until the parent role (***frontend***) reports that it's ready.



Update the ***frontend*** role using ONEGATE or updating the template

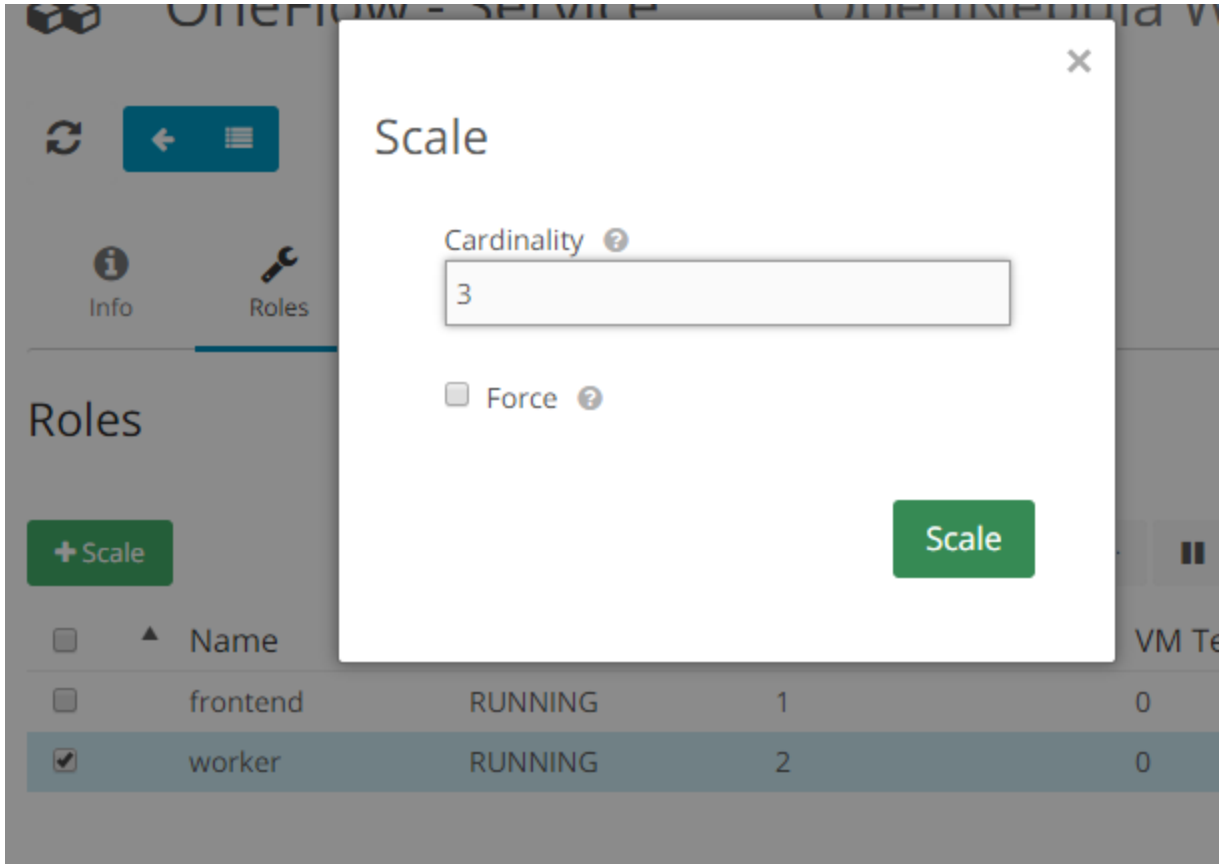
```
curl -X "PUT" http://<onegate>/vm \
  --header "X-ONEGATE-TOKEN: ..." \
  --header "X-ONEGATE-VMID: ..." \
  -d "READY = YES"
```

```
$ onevm update <id>
READY=YES
...
```

Repeat for the ***worker*** role

Hands On – Manual Scaling

Scale the ***worker*** node to 3



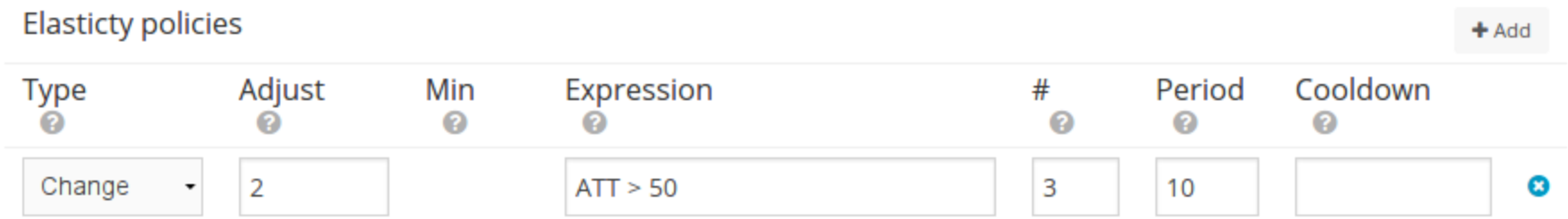
Auto-Scaling based on Metrics

Each role can have an array of elasticity_policies

- Define an expression that will trigger a cardinality adjustment

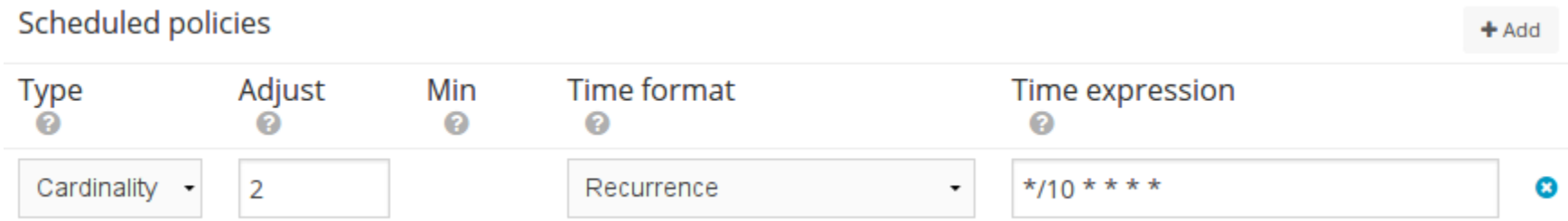
These expressions can use performance data from

- The VM guest. Using the OneGate server, applications can send custom monitoring metrics to OpenNebula.
- The VM, at hypervisor level (CPU, MEMORY, NET_{TX,RX})



Auto-Scaling based on Schedule

Combined with the elasticity policies, each role can have an array of scheduled_policies. These policies define a time, or a time recurrence, and a cardinality adjustment



Questions?

We Will Be Happy to Clarify Any Question