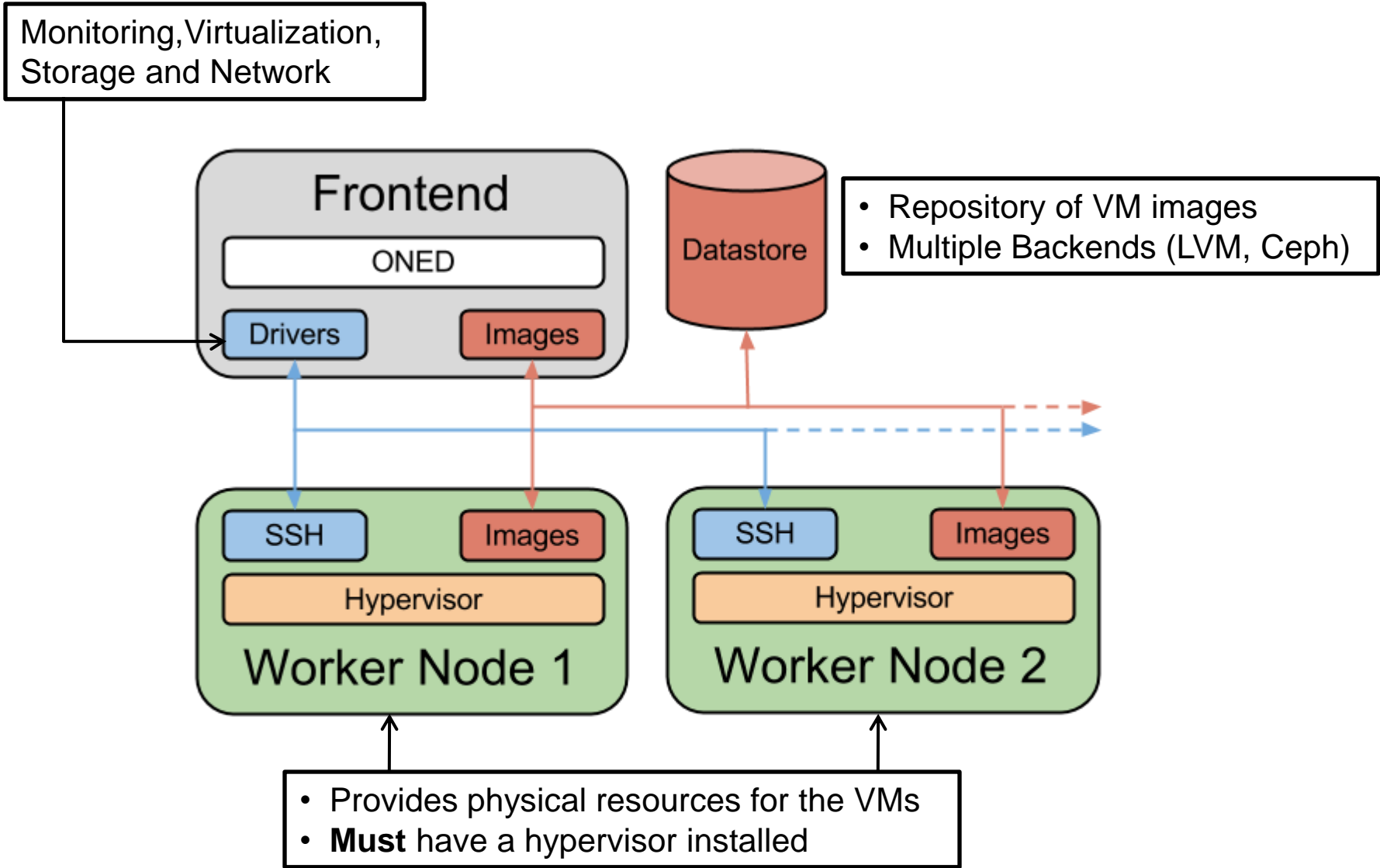


OpenNebula

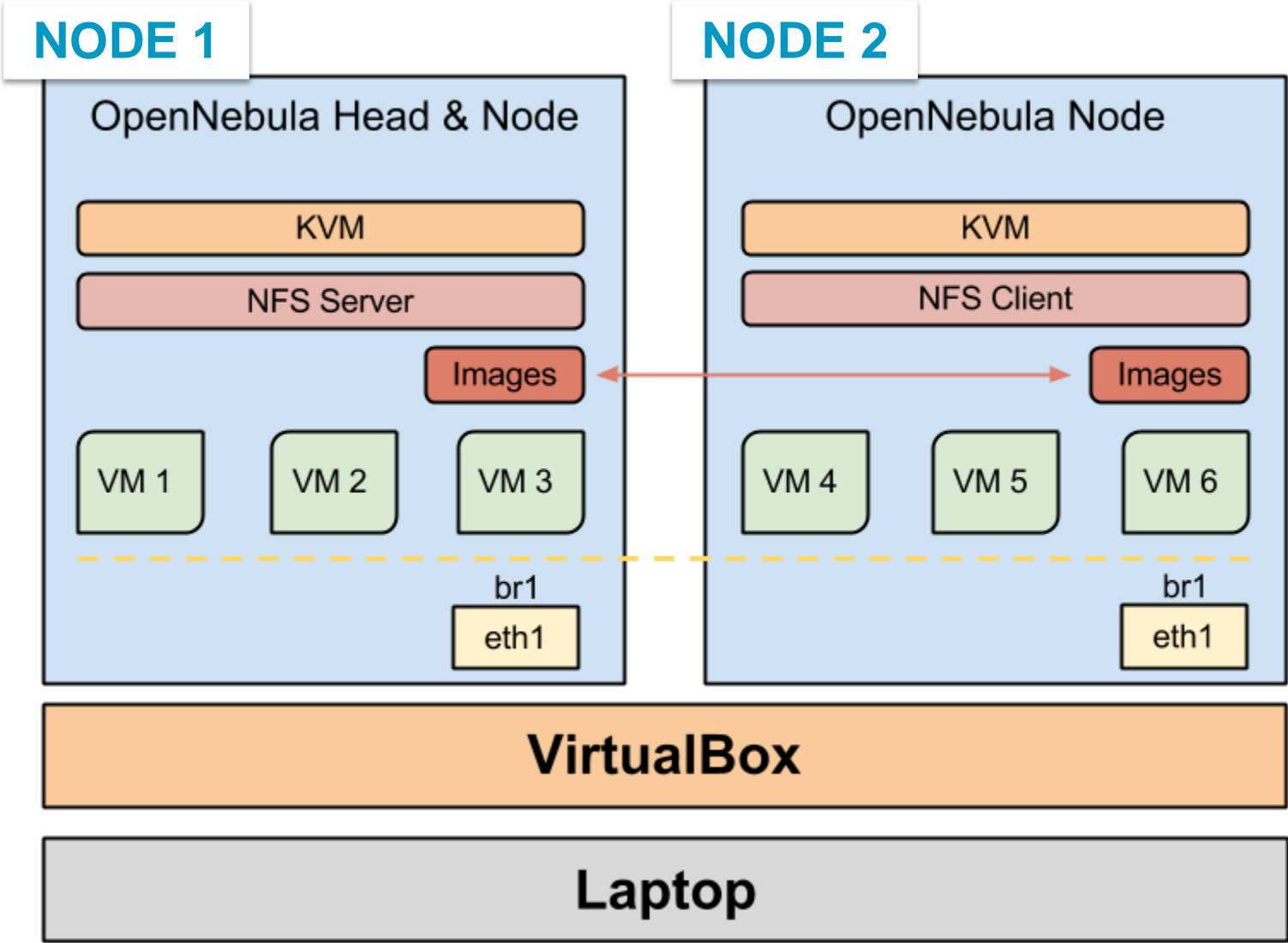
Tutorial

OpenNebula.org

Planning the Installation



Planning the Installation



Installing

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
```

Installing

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
```

Installing

Hands on (node1)

Overview of the CLI (as oneadmin)

```
# su - oneadmin
```

```
$ oneuser show
```

```
$ oneuser -h
```

```
$ one[TAB]
```

Installing

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		

Installing

Hands on (node2)

Configure the hypervisor node

```
# ssh root@node2
```

```
# yum install opennebula-node-kvm
```

```
# service libvirtd restart
```


Installing

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

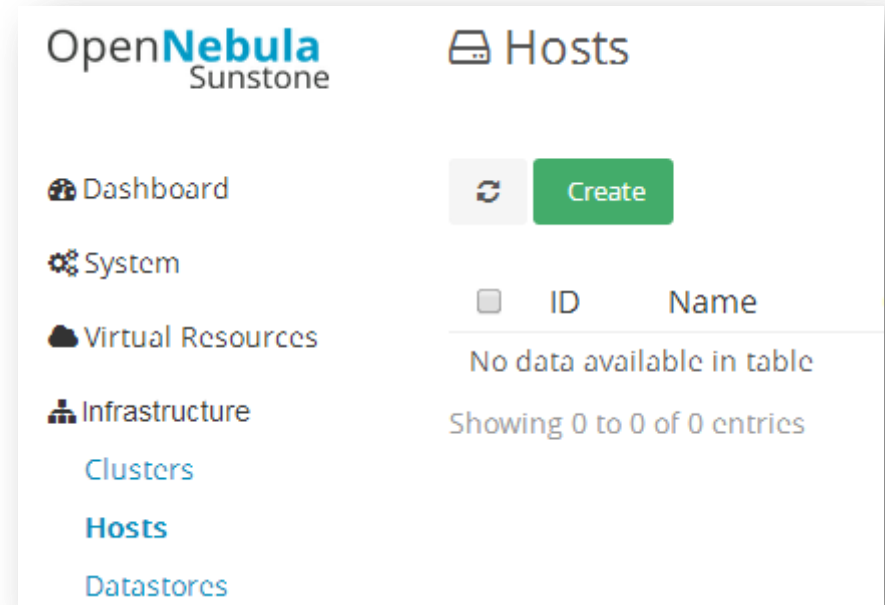
Password: opennebula

Adding Hosts - Sunstone

Basic Usage – Hosts

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



Adding Hosts - CLI

Basic Usage – Hosts

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
```

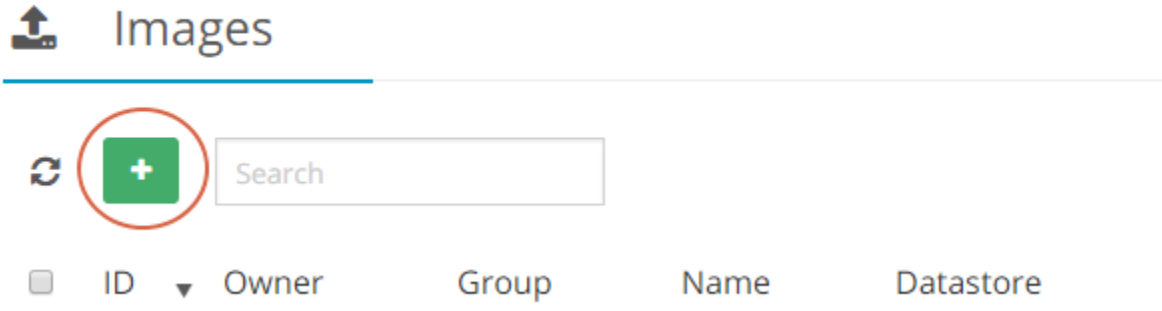
```
$ onehost show -x 1
```

Adding Images

Basic Usage – Images

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)

OpenNebula
Sunstone

Dashboard

System

Virtual Resources

Infrastructure

Clusters

Hosts

Datastores

Virtual Networks

Zones

Datastores

Refresh

Create

More

Delete

Search

<input type="checkbox"/>	ID	Owner	Group	Name	Capacity	Cluster	Type
<input type="checkbox"/>	2	oneadmin	oneadmin	files	7.4GB / 8.9GB (84%)	-	file
<input type="checkbox"/>	1	oneadmin	oneadmin	default	7.4GB / 8.9GB (84%)	-	image
<input type="checkbox"/>	0	oneadmin	oneadmin	system	7.4GB / 8.9GB (84%)	-	system

Showing 1 to 3 of 3 entries

«

1

»

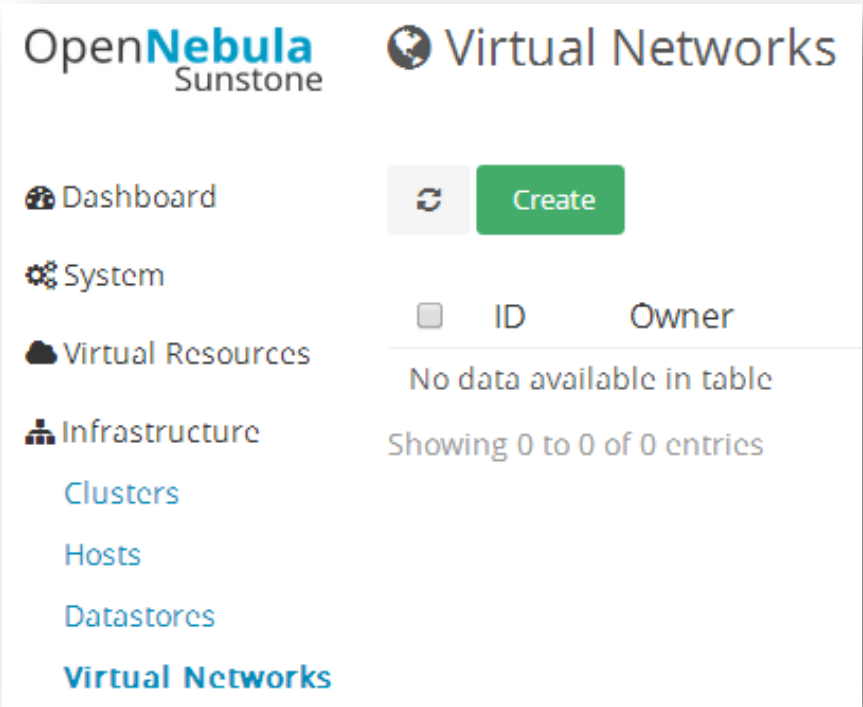
10

oneadmin

OpenNebula

Basic Usage – Networks

Hands on! (Sunstone)



Adding Networks

Basic Usage – Networks

Hands on! (Sunstone)

General

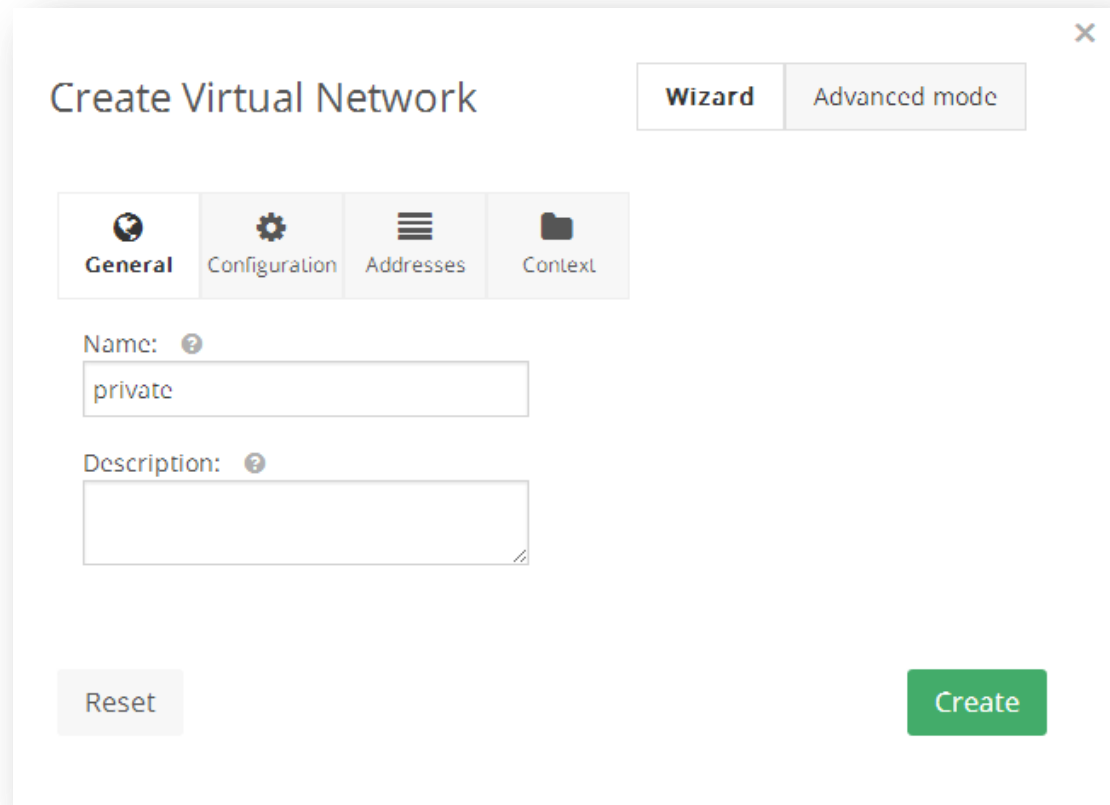
- **Name:** *private*

Configuration

- **Bridge:** *br1*

Addresses

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



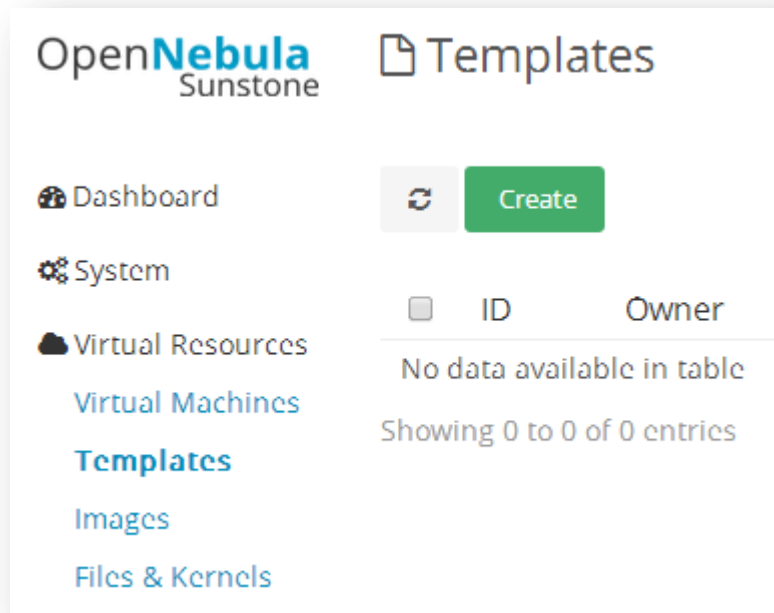
The screenshot shows a 'Create Virtual Network' dialog box with a close button (X) in the top right corner. It has two tabs: 'Wizard' (selected) and 'Advanced mode'. Below the tabs are four icons representing different configuration sections: General (selected), Configuration, Addresses, and Context. The 'General' section contains two text input fields: 'Name' with a help icon and the value 'private', and 'Description' with a help icon and an empty field. At the bottom left is a 'Reset' button, and at the bottom right is a green 'Create' button.

Adding Templates

Basic Usage – Template

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

Instantiating

Basic Usage – VMs

Hands on! (Sunstone)

OpenNebula Sunstone Virtual Machines

oneadmin OpenNebula

Dashboard System Virtual Resources **Virtual Machines** Templates Images

Create [play] [pause] [stop] [refresh] [grid] [user] [trash] Search

ID	Owner	Group	Name	Status	Host	IPs	VNC
1	oneadmin	oneadmin	ttylinux-1	RUNNING	node1	192.168.0.2	[VNC icon]
0	oneadmin	oneadmin	ttylinux-0	RUNNING	node2	192.168.0.1	[VNC icon]

Showing 1 to 2 of 2 entries

« 1 » 10

- **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

Contextualization

Basic Usage – VMs

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:

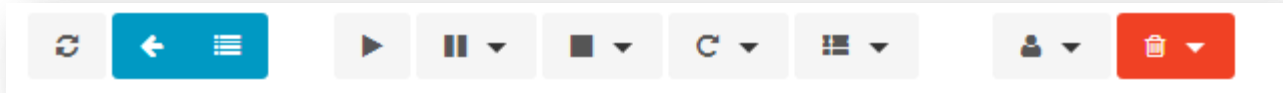
Chop wood, carry water.

# 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
# _
```

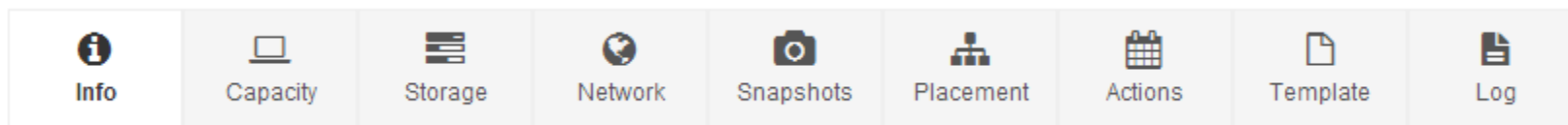
Basic Usage – VMs



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 .

Basic Usage – VMs

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

Permissions

Hands on! (Sunstone)

Make the all the resources previously created usable by everybody

Permissions:	Use	Manage	Admin
Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Group	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ownership			
Owner	oneadmin		
Group	oneadmin		

Apply to all the images, templates and virtual networks.
Owner could be changed too => **chown** and **chmod**.

Groups

Hands on!

Create Group

Name:

students



Views



Admin



Permissions

Admin: Click **Create an administrator user**

Username: students-admin

Password: ...

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

Groups

Hands on!

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

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

The Power of VDCs

Virtual Data Centers

+

Search

Update

<input type="checkbox"/>	ID ▾	Name	Groups	Clusters	Hosts	VNets	Datastores
<input type="checkbox"/>	0	default	2	1	0	0	0

Showing 1 to 1 of 1 entries

Previous

1

Next

10 ▾

1 TOTAL

Reset

Create

Wizard

Advanced

General

Groups

Resources

Clusters

Hosts

VNets

Datastores

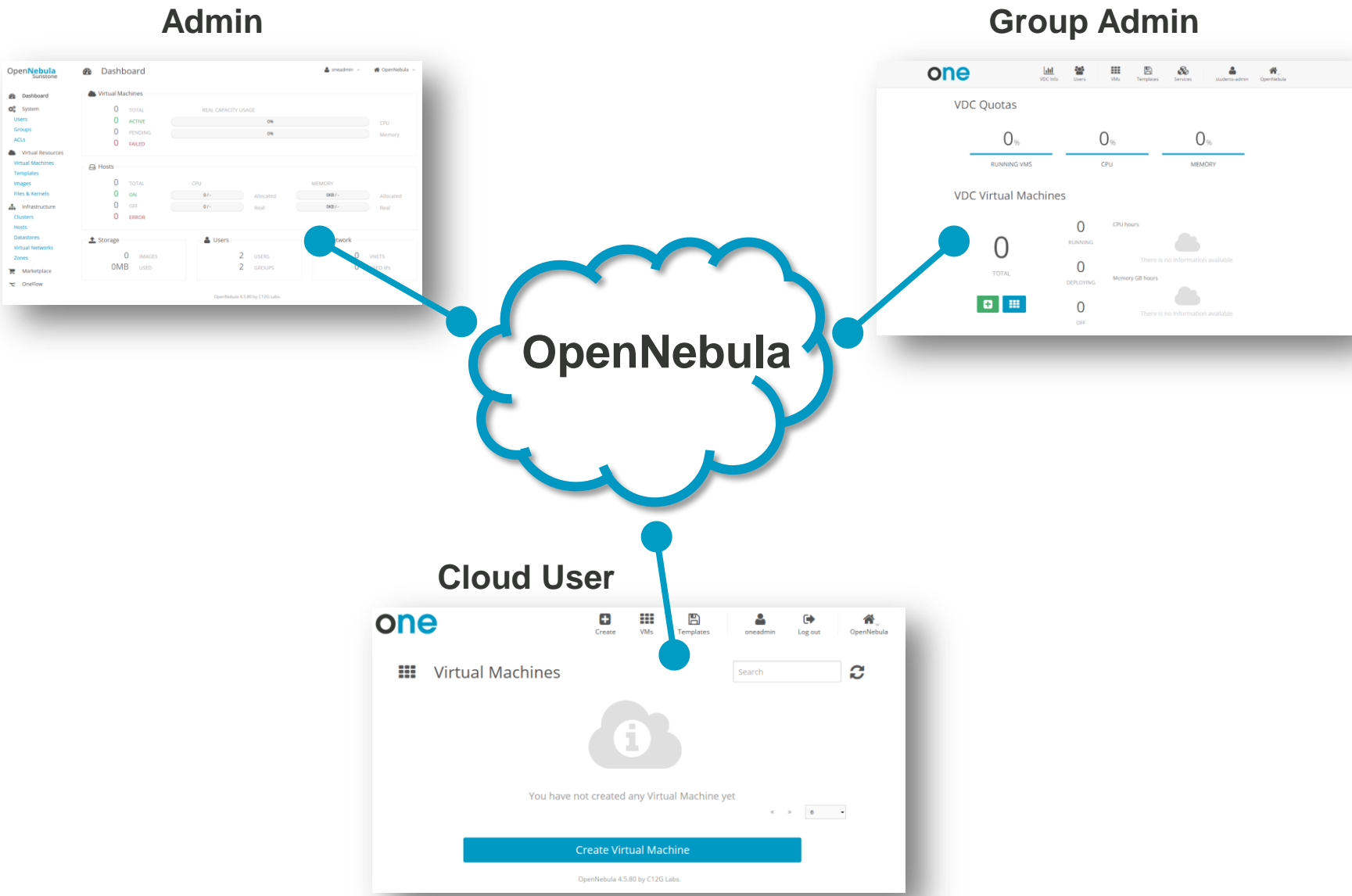
Zone

OpenNebula

Installing and Basic Usage

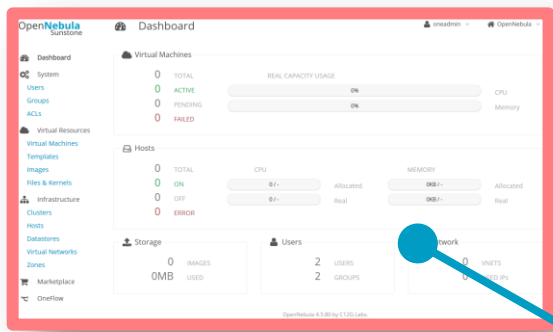
25

OpenNebula Views

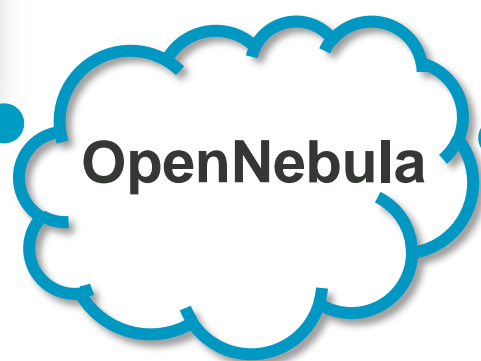
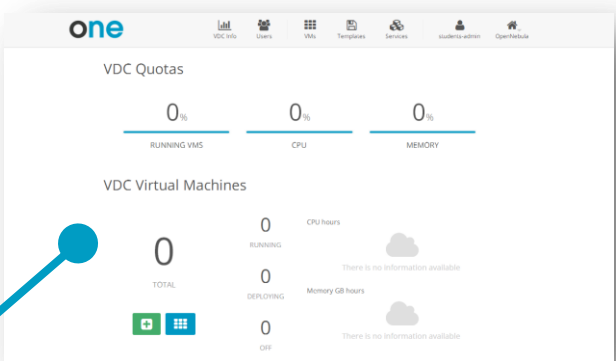


OpenNebula Views

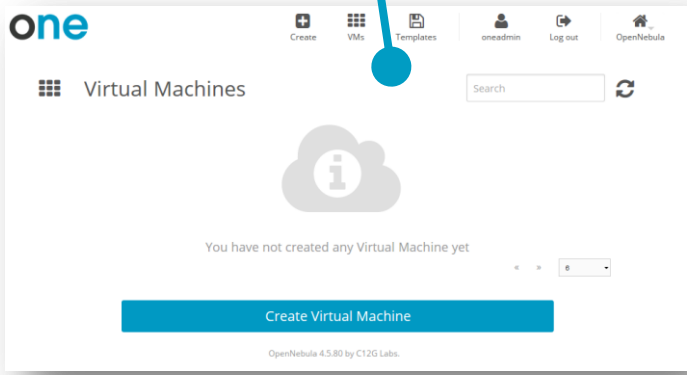
Admin



Group Admin

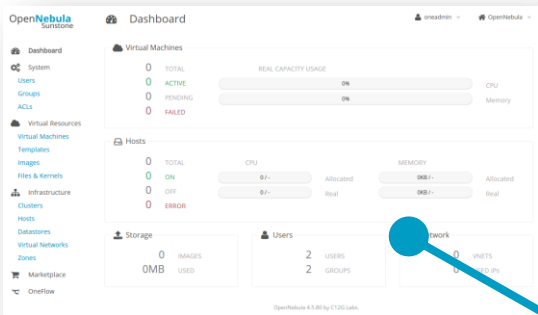


Cloud User

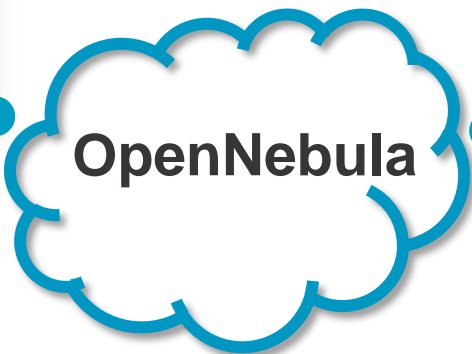
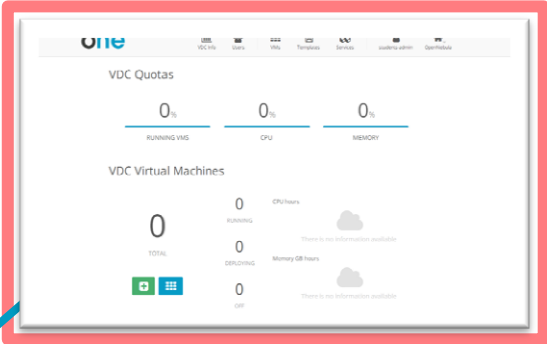


Configure your Cloud from the Admin perspective

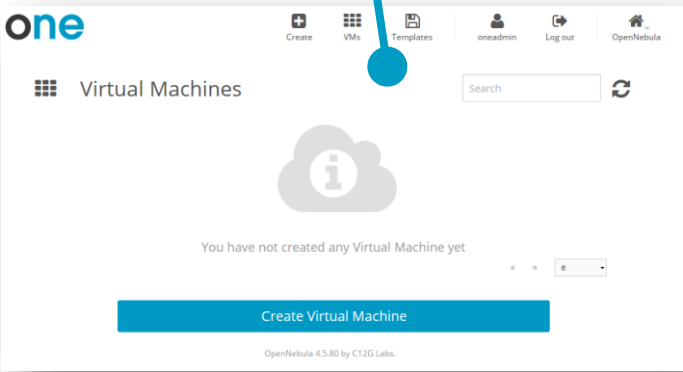
Admin



Group Admin



Cloud User

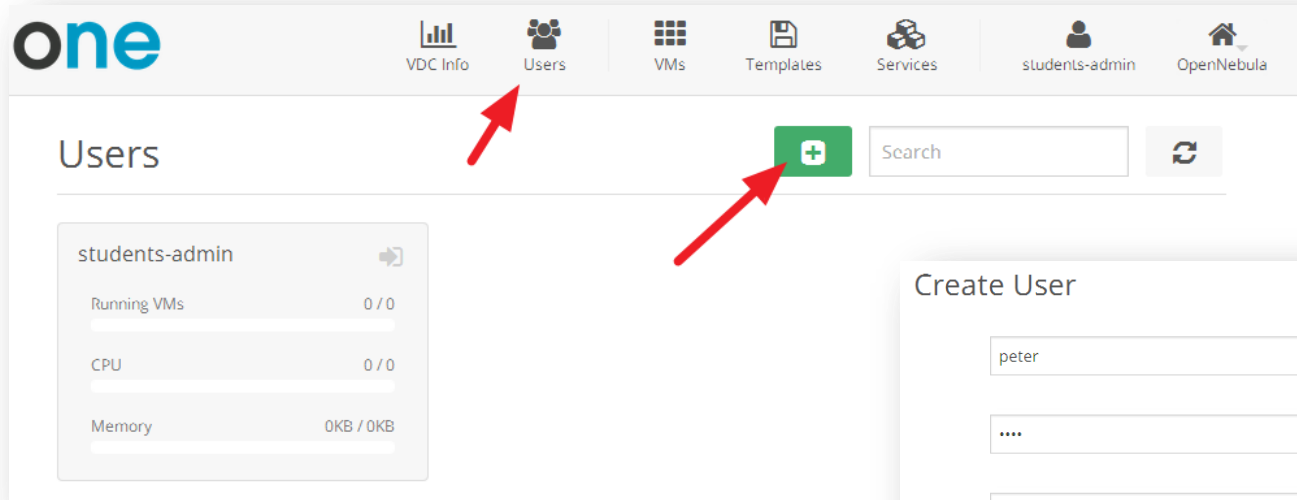


Group Admin View

Using Groups

Hands on!

Login as students-admin

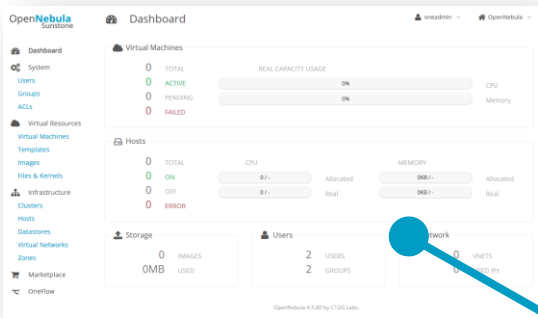


- Create a student
- Assign Quotas

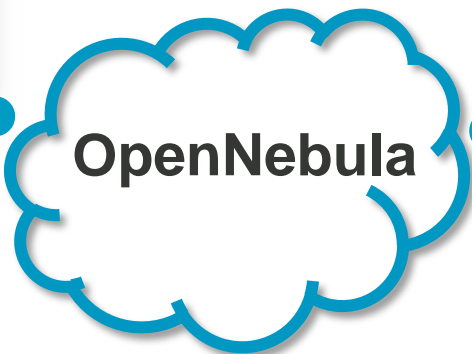
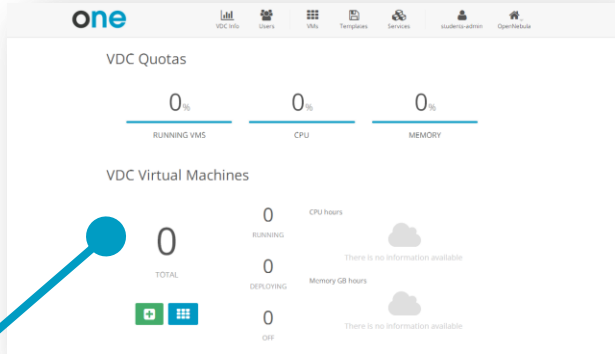
The 'Create User' dialog box is shown. It has three input fields for the user details: a text field for the username (containing 'peter'), and two password fields (both containing '....'). Below these fields is the 'Define Quotas' section, which includes three rows of sliders and input boxes for setting limits: Running VMs (slider at 1, input box '1'), CPU (slider at 25, input box '25'), and Memory (GBs) (slider at 25, input box '25'). At the bottom of the dialog is a blue button labeled 'Add User'.

OpenNebula Views

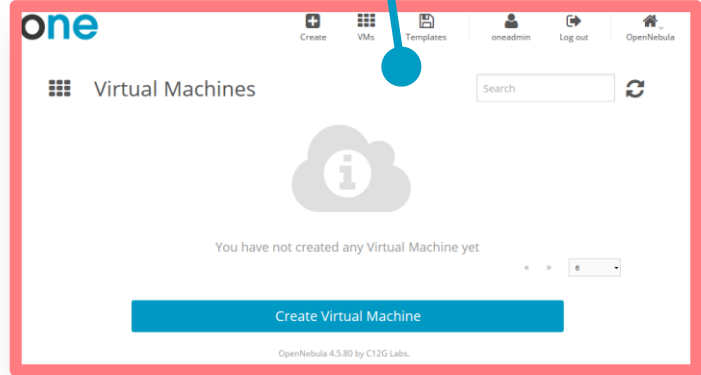
Admin



Group Admin

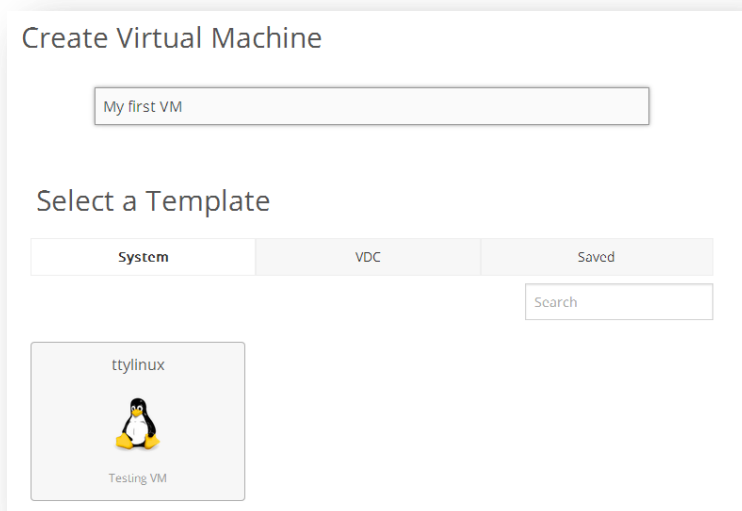


Cloud User



Hands on!

Login as the student and instantiate a new VM



The screenshot shows the 'Create Virtual Machine' window. At the top, there is a text input field containing 'My first VM'. Below this is the 'Select a Template' section, which includes three tabs: 'System', 'VDC', and 'Saved'. The 'System' tab is active. To the right of the tabs is a search input field with the placeholder text 'Search'. Below the tabs, a template card is displayed for 'ttylinux'. The card features a penguin icon and the text 'Testing VM' at the bottom.

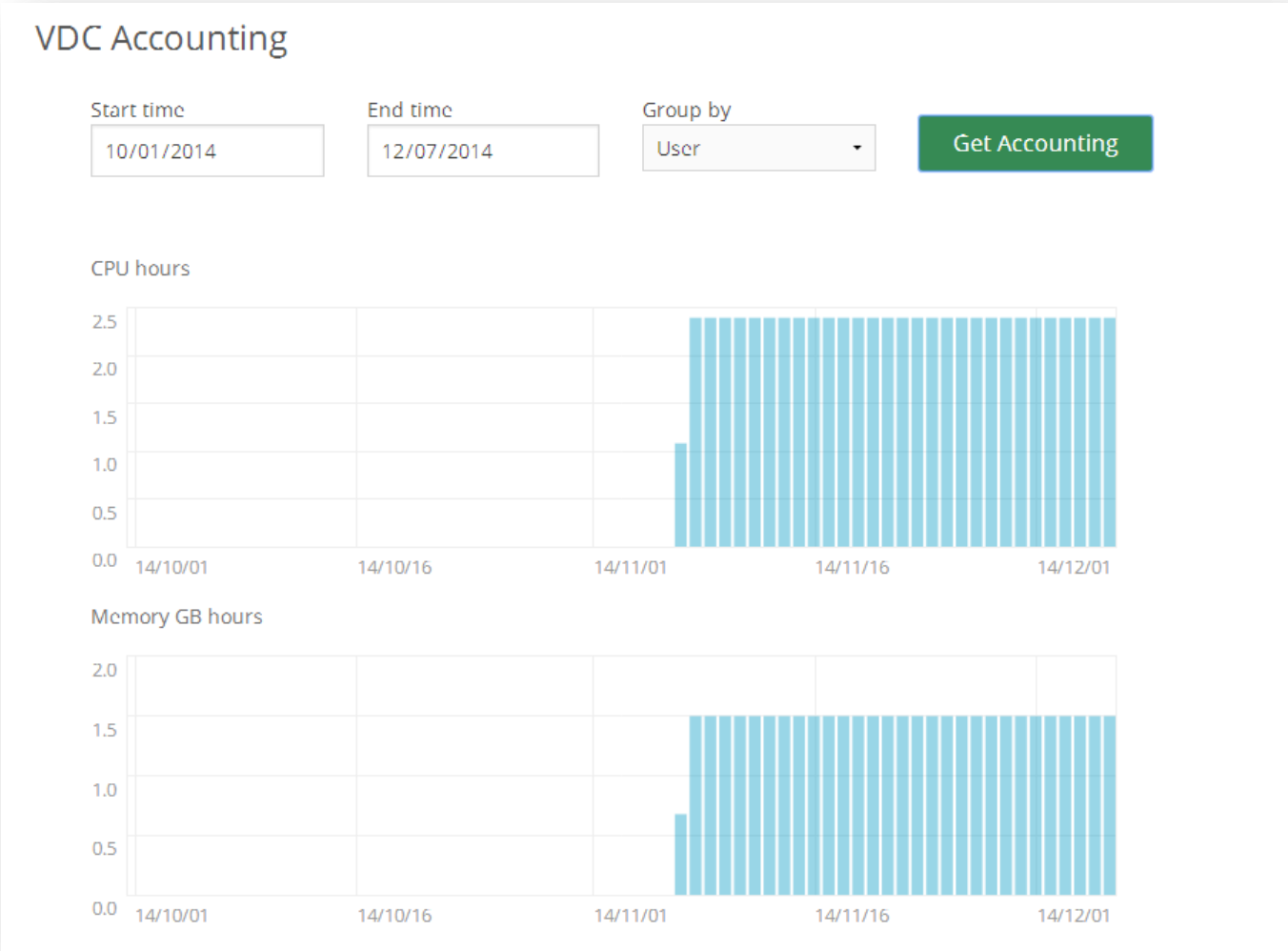
- Select the *ttylinux* template
- Network *private*

- 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

Using Groups

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

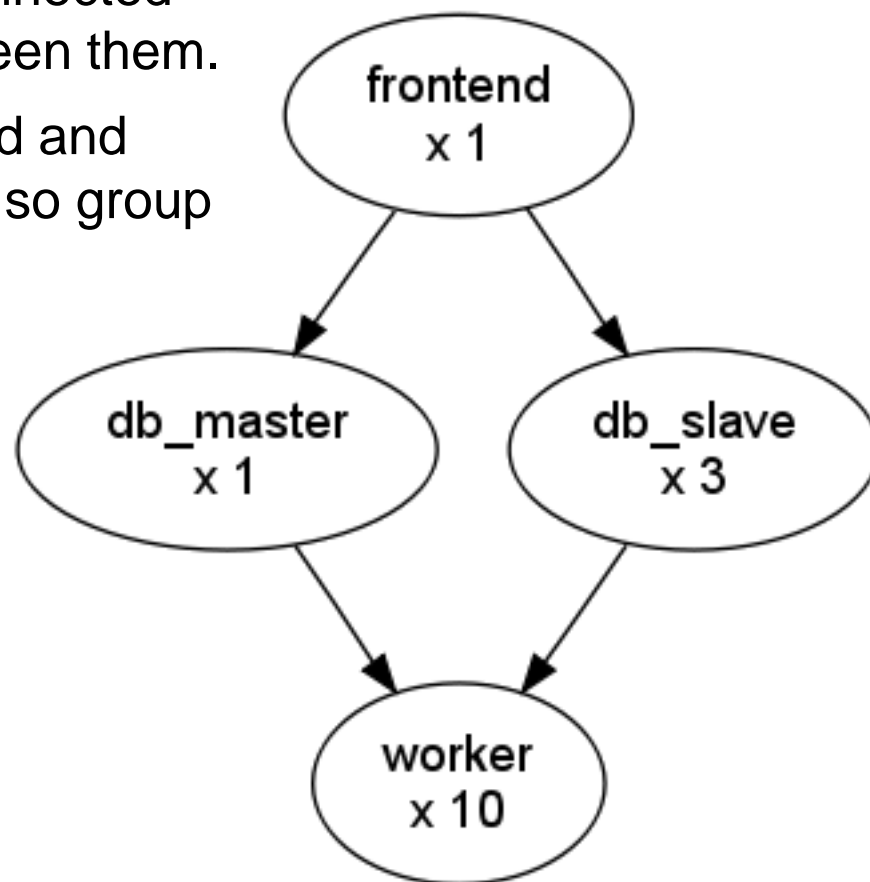
Scale out your private resources to EC2

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



Strategies

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 they are running, if this checkbox is enabled:

☒ Wait for VMs to report that they are READY ⓘ

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

Hands On – Template Creation

OpenNebula
Sunstone

Dashboard

System

Virtual Resources

Infrastructure

Marketplace

OneFlow

Services

Templates

OneFlow - Templates

oneadmin

OpenNebula

Instantiate

Update

Search

ID	Owner	Group	Name
<div><div></div><div>There is no data available</div></div>			

Hands On – Service Template and Frontend Role

Name ?

Description ?

^ Network Configuration

Name	Description
<input type="text" value="INTERNAL_NET"/>	<input type="text" value="Private Network"/> ✕
<input type="text" value="PUBLIC_NET"/>	<input type="text" value="Public Network"/> ✕

+ Add another Network

^ Advanced Service Parameters

Strategy ?


Straight ▾

Shutdown action ?

Shutdown hard ▾

☒ Wait for VMs to report that they are READY ?

Hands On – Service Template Database Master Role


frontend ✕

Role Name ?

frontend

VM template ?

0: ttylinux


VMs ?


1

Network Interfaces

☒ INTERNAL_NET

☒ PUBLIC_NET


frontend ✕


worker ✕

Role Name ?

worker

VM template ?

0: ttylinux

VMs ?

2

Network Interfaces

☒ INTERNAL_NET

☐ PUBLIC_NET

Parent roles

☒ frontend

Hands On – Instantiate the Template

Instantiate Service Template

Service Name ?
OpenNebula Wonders Blog

Number of instances ?
1

Network

Public Network

ID	Owner	Group	Name
0	oncadmin	oncadmin	private

You selected the following network: private

Public Network

Search

ID	Owner	Group	Name	Reservation	Cluster	Leases
0	oncadmin	oncadmin	private	No	-	0 / 4

Previous 1 Next

You selected the following network: private

Role frontend

Blog Name
Wonders

Blog Admin Password
...

Role worker

Blog Name
Wonders

Blog Admin Password
...

Instantiate

Installing and Basic Usage

39

Managing Services

Hands On – Service Information

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

Roles

Period: Number:

[+ Scale](#)

<input type="checkbox"/>	Name	State	Cardinality	VM Template	Parents
<input type="checkbox"/>	frontend	DEPLOYING	1	0	-
<input type="checkbox"/>	worker	PENDING	2	0	frontend

Previous **1** Next

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

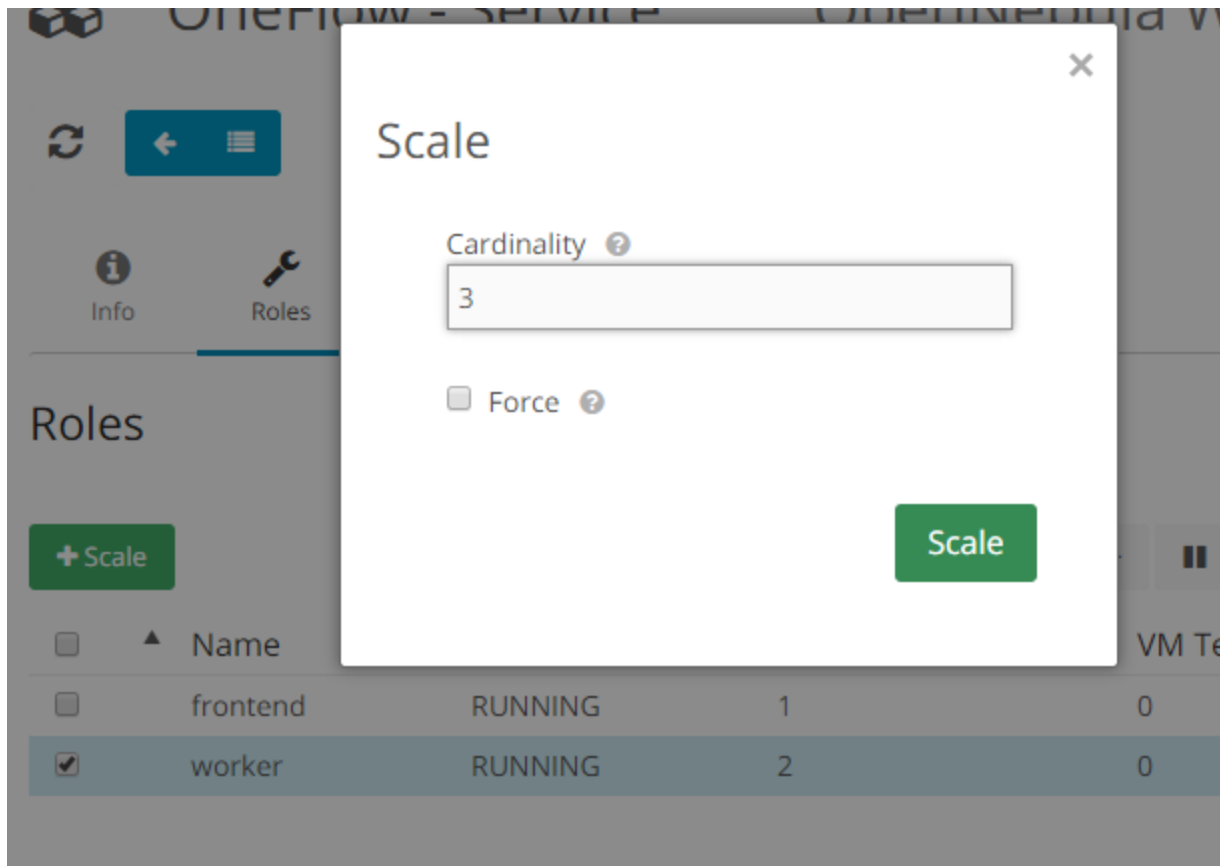
```
curl -X "PUT" http://<onegate>/vm \
  --header "X-ONEGATE-TOKEN: ..." \
  --header "X-ONEGATE-VMID: ..." \
  -d "READY = YES"
```

```
$ 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})

Elasticity policies + Add

Type ?	Adjust ?	Min ?	Expression ?	# ?	Period ?	Cooldown ?
Change ▾	2		ATT > 50	3	10	

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

Scheduled policies + Add

Type ?	Adjust ?	Min ?	Time format ?	Time expression ?
Cardinality ▾	2		Recurrence ▾	<input type="text" value="*/10 * * * *"/> ✕

We Will Be Happy to Clarify Any Question

