
vOneCloud Documentation

Release 4.10-beta

OpenNebula Systems

November 05, 2014

CONTENTS

1	Release Notes vOneCloud 4.10 Beta	3
1.1	What Is?	3
1.2	What's New vOneCloud 4.10 Beta	4
1.3	vOneCloud Features	4
1.4	System Requirements	5
1.5	Known Issues	6
1.6	Resolved Issues	6
1.7	vOneCloud Limitations	6
2	Simple Cloud Deployment	7
2.1	All About Simplicity	7
2.2	Download and Deploy	7
2.3	Import Existing vCenter	14
2.4	Create a Virtual Datacenter	17
2.5	vOneCloud Interfaces	19
2.6	vCenter View	20
2.7	VDC Admin View	21
2.8	Cloud View	22
3	Security and Resource Consumption Control	25
3.1	Introduction	25
3.2	Users, Groups and ACLs	25
3.3	Resource Quotas	27
3.4	Accounting & Monitoring	29
4	Infrastructure Configuration	31
4.1	Introduction	31
4.2	Add New vCenters And VM Templates	31
4.3	Hybrid Clouds	32
4.4	Multi VM Applications	34
4.5	Authentication	36

State development vOneCloud 4.10-Beta

RELEASE NOTES VONECLOUD 4.10 BETA

1.1 What Is?

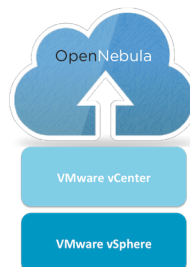
vOneCloud

The Open Replacement for vCloud

vOneCloud extends vCenter with cloud features such as provisioning, elasticity and multi-tenancy. vOneCloud is designed for companies that want to create a self-service cloud environment on top of their VMware infrastructure without having to abandon their investment in VMware and retool the entire stack.

vOneCloud is an OpenNebula distribution optimized to work on existing VMware vCenter deployments. It deploys an enterprise-ready OpenNebula cloud just in a few minutes where the infrastructure is managed by already familiar VMware tools, such as vSphere and vCenter Operations Manager, and the provisioning, elasticity and multi-tenancy cloud features are offered by OpenNebula. It inherits all the [benefits](#) from the open source cloud management platform, adding an easy to deploy, easy to use aspect due to pre configuration of the OpenNebula install contained within the appliance.

vOneCloud is distributed as a virtual appliance in OVA format for vSphere. It contains all required OpenNebula services within a single CentOS Linux appliance. All components are fully open-source and have been certified to work in enterprise environments. It leverages advanced features such as vMotion, HA or DRS scheduling provided by the VMware vSphere product family.



Powerful	Virtual data centers, self-service, datacenter federation, hybrid cloud on VMware environments
Cost Effective	Free, there are no license costs, all components are fully open-source software
Flexible	Completely open, customizable and modular, so it can be adapted to your needs
No Lock-in	Platform independent, gradually migrate to other virtualization platforms
Simple	Very easy to install, upgrade, and maintain, with easy-to-use graphical interfaces
Enterprise-ready	Certified, production-ready with commercial support subscriptions and professional services

1.2 What's New vOneCloud 4.10 Beta

This is the first beta release of the vOneCloud appliance. As such, everything is new, check out the documentation ahead.

1.3 vOneCloud Features

vOneCloud leverages the [functionality of OpenNebula](#). The following features come preconfigured and can be used **out-of-the-box** with vOneCloud:

- **Cloud User Interfaces**
 - AWS EC2 and EBS APIs
 - Simple, clean, intuitive portals for cloud consumers and VDC admins
- **On-demand Provision of Virtual Data Centers**
 - Dynamic creation of Virtual Data Centers (VDCs) as fully-isolated virtual infrastructure environments where a group of users, under the control of the VDC administrator, can create and manage compute capacity
 - Placement of VDCs to multiple vCenters

- **Hybrid Cloud**
 - Cloud-bursting of VMs to public clouds
- **Fast Provisioning**
 - VM and service (multi-VM apps) templates
 - Automatic provision of VM and service (multi-VM apps) from a catalog
 - Automatic execution and scaling of multi-tiered applications
 - Snapshotting
- **Security and Resource Consumption Control**
 - Resource Quota Management to track and limit computing resource utilization
 - Integration with user management services like LDAP, Active Directory...
 - Fine-grained accounting and monitoring
 - Complete isolated VDCs and organizations
 - Fine-grained ACLs and user quotas
 - Powerful user, group and role management
- **Cloud Admin Interfaces**
 - Powerful CLI that resembles typical UNIX commands applications
 - SunStone Portal for administrators and advanced users
- **Reliability, Efficiency and Massive Scalability**
 - Leverage years of testing and production use
 - Be sure that your Cloud Management Platform will be up to the task

vOneCloud can leverage all the functionality that OpenNebula delivers, but some of it needs **additional configuration** steps:

- **Centralized Management of Multiple Zones.** Federate different datacenters by joining several vOneCloud instances.
- **Community Virtual Appliance Marketplace.** Create your own marketplace or benefit from community contributions with an online catalog of ready-to-run virtual appliances.
- **Broad Commodity and Enterprise Platform Support.** Underlying OpenNebula software features an amazingly flexible and plugin oriented architecture that eases the integration with existing datacenter components. Do no reinvent your datacenter, evolve it!.
- **Virtual & Physical Infrastructure Control.** Manage all aspects of your physical (hypervisors, storage backends, etc) & virtualized (VM lifecycle, VM images, virtual networks, etc) from a centralized web interface (Sunstone).

Although the configuration is tailored for vCenter infrastructures, all the power of OpenNebula is contained in vOneCloud and it can be unleashed!

1.4 System Requirements

The following components are needed to be present in the infrastructure to implement a cloud infrastructure run by vOneCloud:

vCenter 5.1 / 5.5	1. The IP or DNS needs to be known, as well as the credentials (username and password) of an admin user . 2. DRS needs to be enabled in vCenter. 3. All ESX belonging to the same vCenter cluster to be exposed to vOneCloud need to share at least one datastore among them. 4. VMs that will be instantiated through vOneCloud saved as VMs Templates in vCenter.
ESX 5.1 / 5.5	With at least 2 GB of free RAM and 1 free CPU

1.5 Known Issues

1.6 Resolved Issues

1.7 vOneCloud Limitations

vOneCloud will use pre defined Templates existing in the vCenter to launch Virtual Machines, as such the following limitations apply:

No VM Contextualization	Virtual Machines won't have the ability of being contextualized, thus the CONTEXT sections of VM Templates won't be honored. The following operations are NOT supported on vCenter VMs managed by OpenNebula, although they can be performed through vCenter - Attach/detach disk to a running VM - Attach/detach network interface to a running VM - Migrate VM to different ESX clusters Firewall rules as defined in Security Groups cannot be enforced in vCenter VMs. vOneCloud Templates representing two or more vCenter VM Templates cannot currently be defined.
VM Unsupported Operations	
No Security Groups	
No MultivCenter Templates	

These limitations will be addressed in future versions of vOneCloud.

SIMPLE CLOUD DEPLOYMENT

2.1 All About Simplicity

vOneCloud is preconfigured to be plugged into your existing vCenter infrastructure and quickly start using its cloud features. vOneCloud is the perfect choice for companies that want to create a self-service cloud environment on top of their VMware infrastructure without having to abandon their investment in VMware and retool the entire stack.

Simple to Use	Simple graphical interfaces for cloud consumers, and VDC and cloud administrators.
Simple to Update	New versions can be easily installed with no downtime of the virtual workload.
Simple to Adopt	Add cloud features, do not interfere in existing VMware procedures and workflows.
Simple to Install	CentOS appliance deployable through vSphere, able to import your system

This guide will guide through all the needed steps to deploy vOneCloud and prepare your new cloud to provision your end users.

2.2 Download and Deploy

vOneCloud can be downloaded as an appliance using this link: <http://appliances.opennebula.systems/vOneCloud/vOneCloud-4.10-beta.ova>.

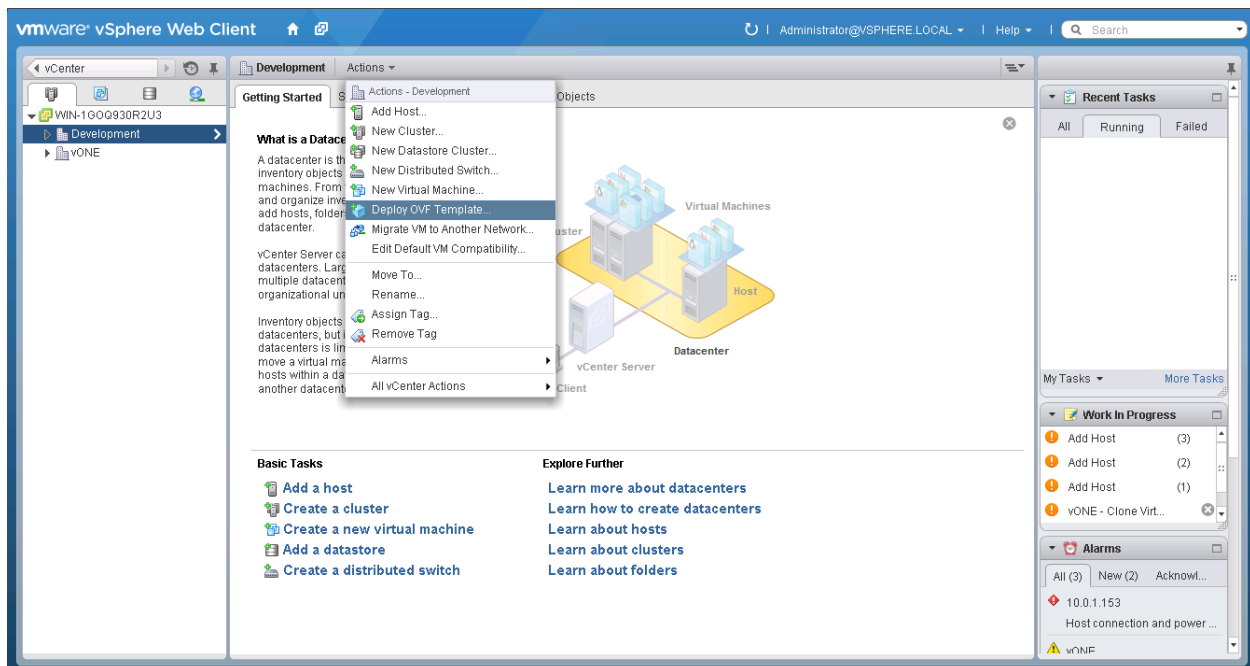
You can import this OVA appliance to your vCenter infrastructure. It is based on CentOS 7 and has the VMware tools enabled.

The appliance requirements are kept to a strict minimum so it can be executed in any vCenter installation. However, before deploying it, please read the *system requirements*.

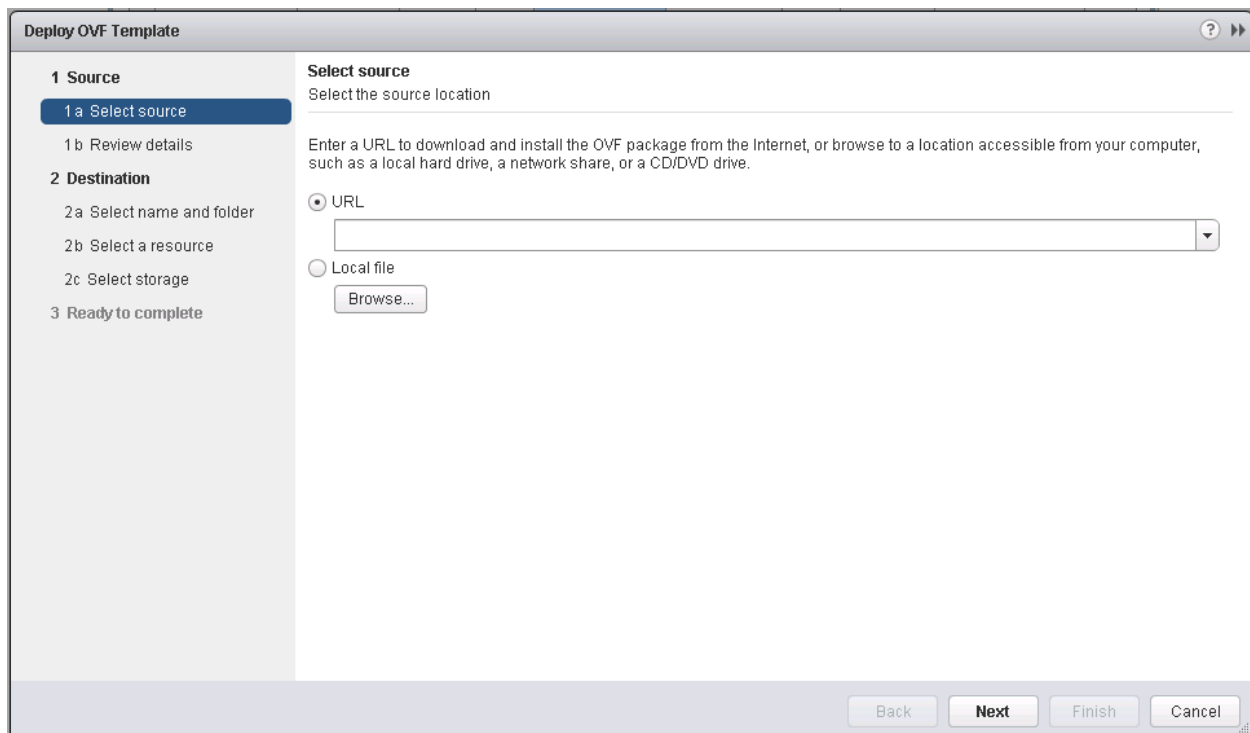
Follow the next steps to deploy a fully functional vOneCloud:

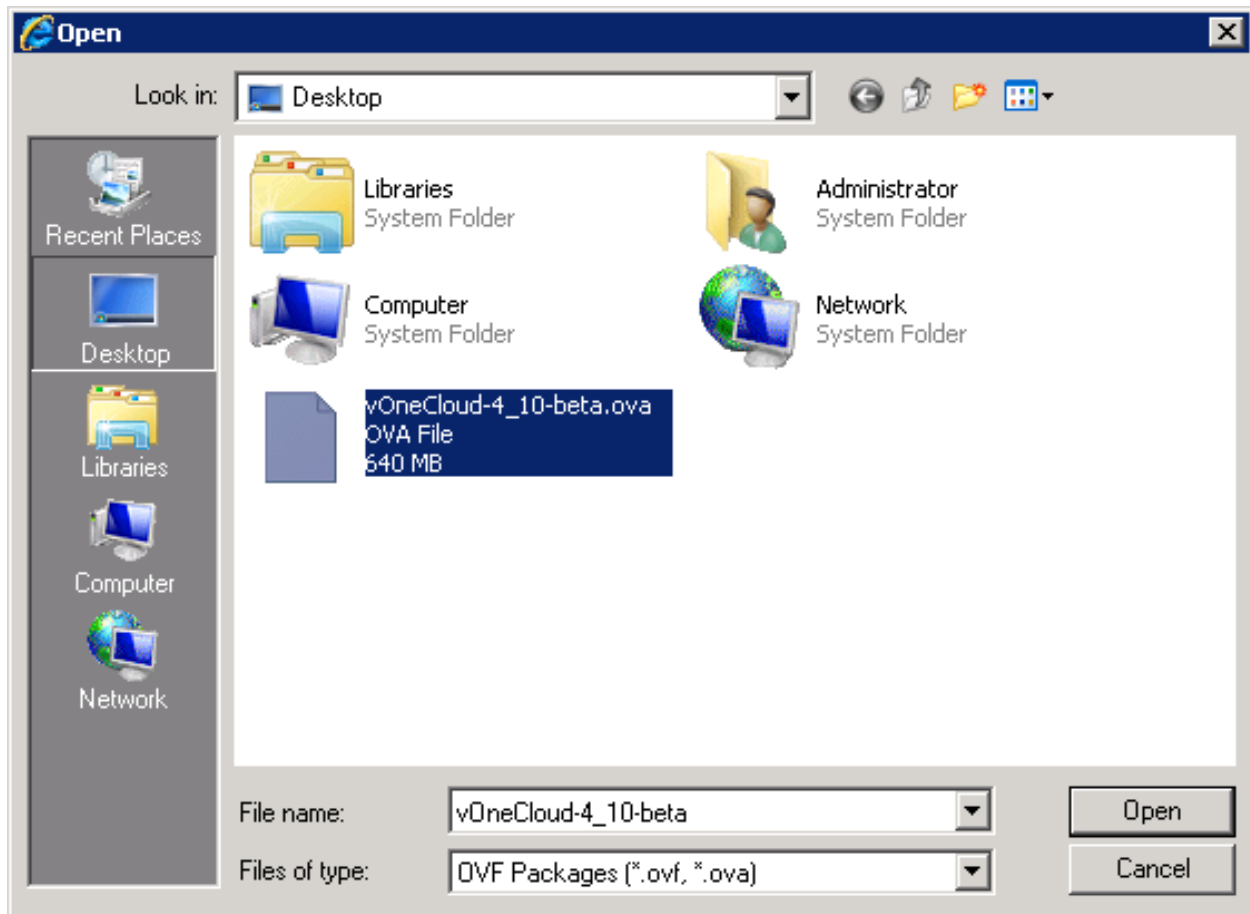
2.2.1 Step 1. Deploying the OVA

Login to your vCenter installation and select the appropriate datacenter and cluster, where you want to deploy the appliance. Select the Deploy OVF Template.

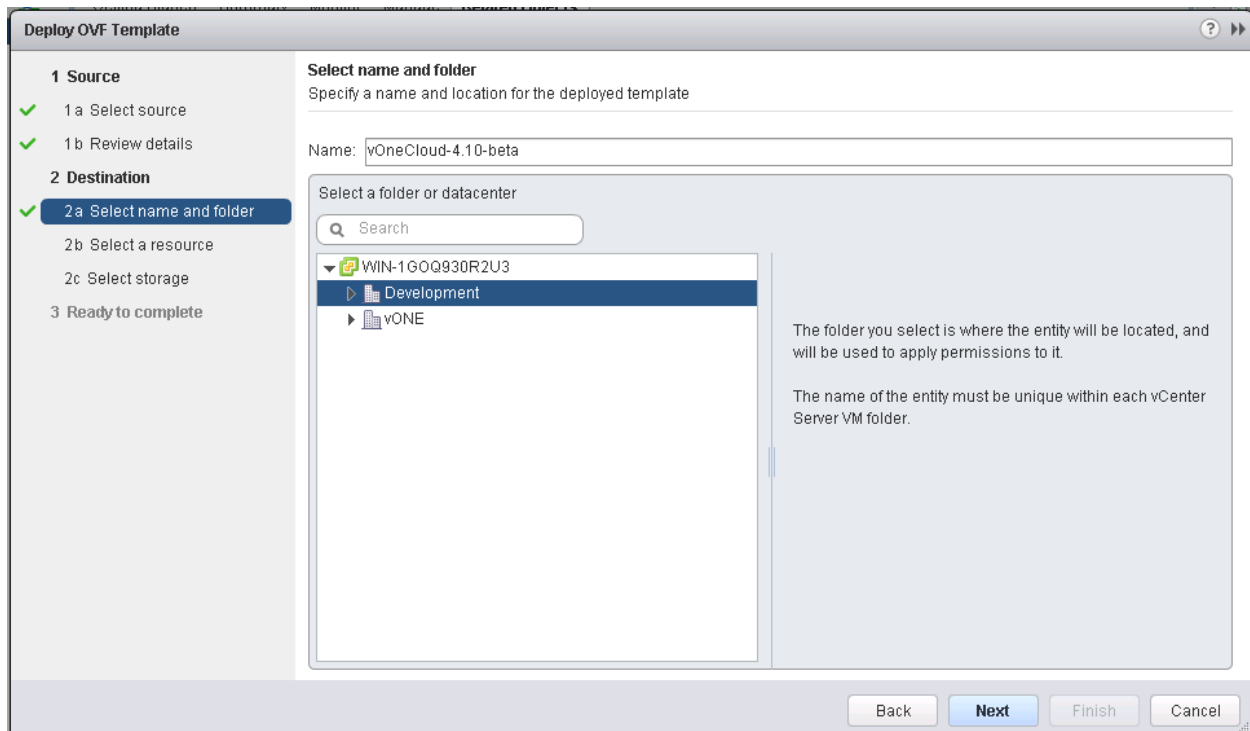


You have the option now to input the URL of the appliance (you can find it at the top of this page), or if you have previously downloaded it, you can simply browse to the download path as such:

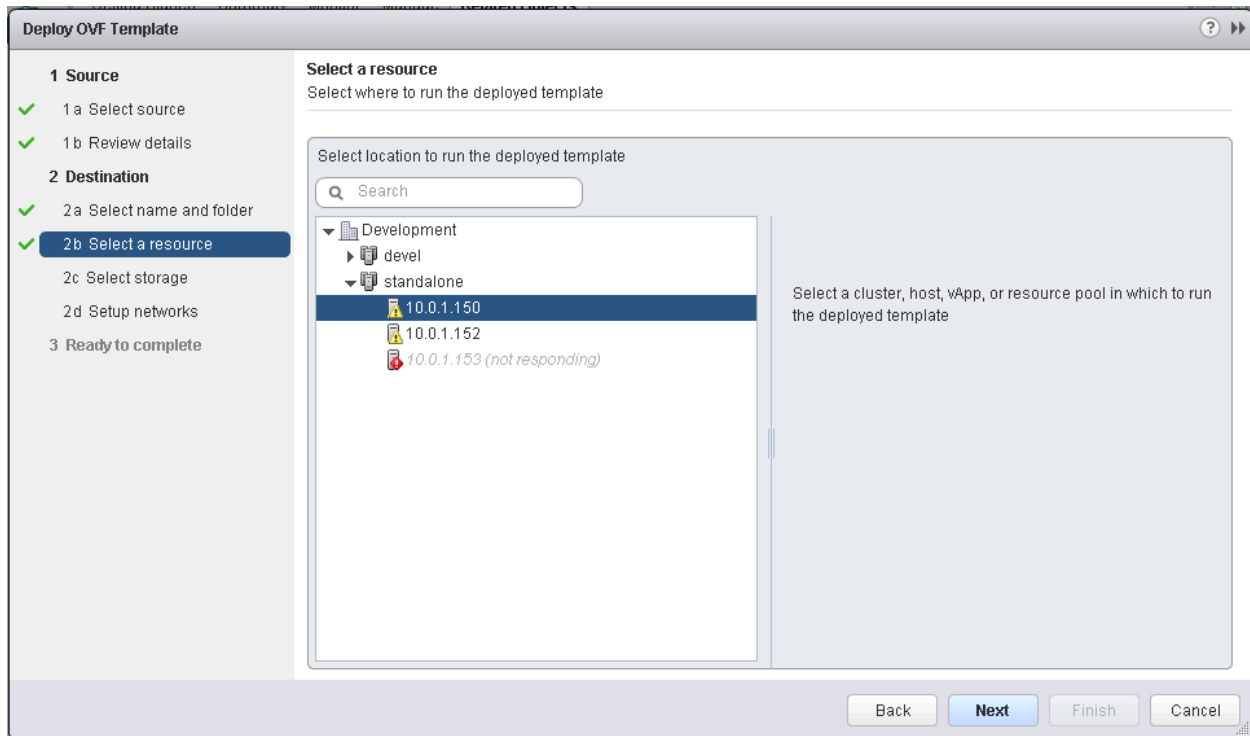




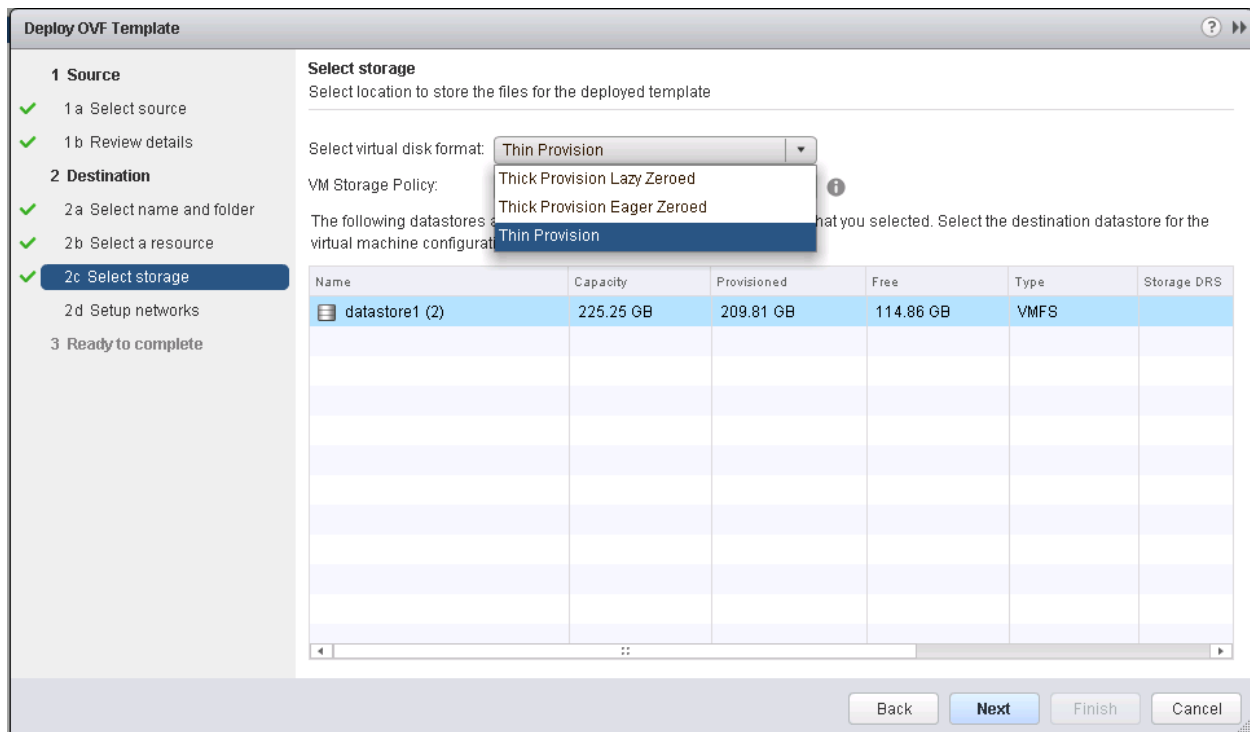
Select the name and folder:



Select a resource to run the appliance:



Select the datastore:



Select the Network. You will need to choose a network that has access to the vCenter hosts.

Deploy OVF Template

1 Source

- ✓ 1 a Select source
- ✓ 1 b Review details

2 Destination

- ✓ 2 a Select name and folder
- ✓ 2 b Select a resource
- ✓ 2 c Select storage
- ✓ **2 d Setup networks**
- ✓ 3 Ready to complete

Setup networks
Configure the networks the deployed template should use

Source	Destination	Configuration
internal	internal	✓
	VM Network	
	internal	

IP protocol: IPv4 IP allocation: Static - Manual ⓘ

Source: internal - Description
The internal network

Destination: internal - Protocol settings
No configuration needed for this network

Back Next Finish Cancel

Review the settings selection and click finish. Wait for the Virtual Machine to appear in the cluster.

Deploy OVF Template

1 Source

- ✓ 1 a Select source
- ✓ 1 b Review details

2 Destination

- ✓ 2 a Select name and folder
- ✓ 2 b Select a resource
- ✓ 2 c Select storage
- ✓ 2 d Setup networks
- ✓ **3 Ready to complete**

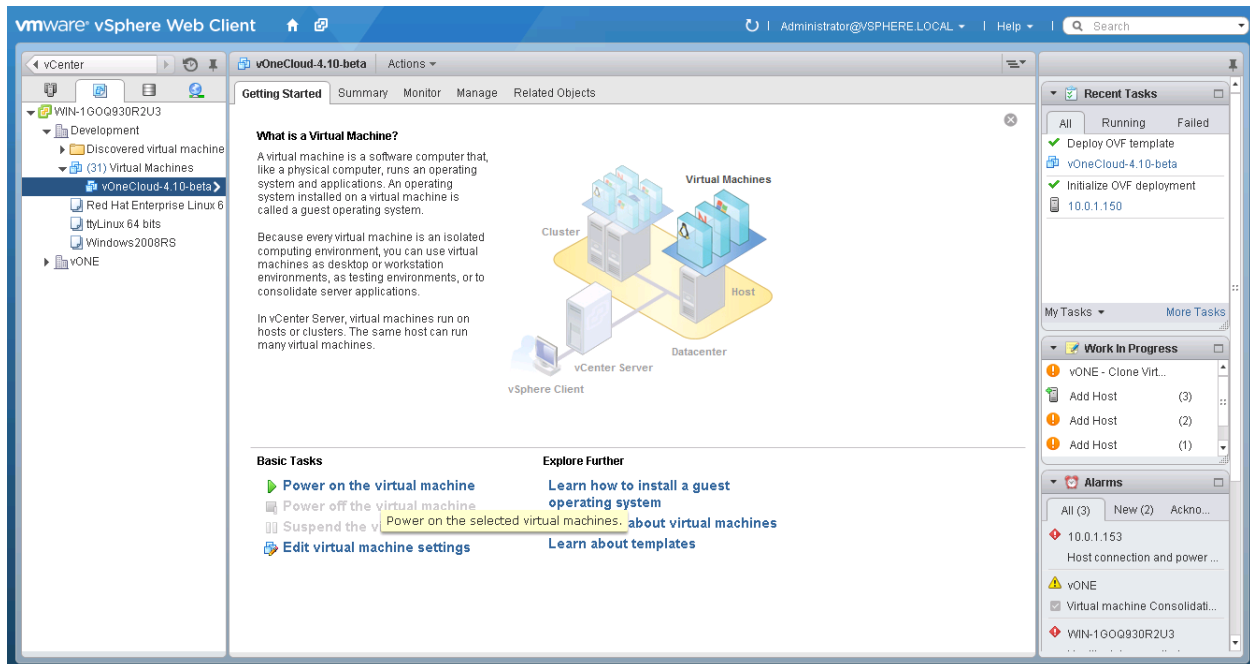
Ready to complete
Review your settings selections before finishing the wizard.

OVF file	C:\Users\Administrator\Desktop\wOneCloud-4_10-beta.ova
Download size	640.3 MB
Size on disk	1.3 GB
Name	vOneCloud-4.10-beta
Datastore	datastore1 (2)
Target	10.0.1.150
Folder	Development
Disk storage	Thin Provision
Network mapping	internal to internal
IP allocation	Static - Manual, IPv4

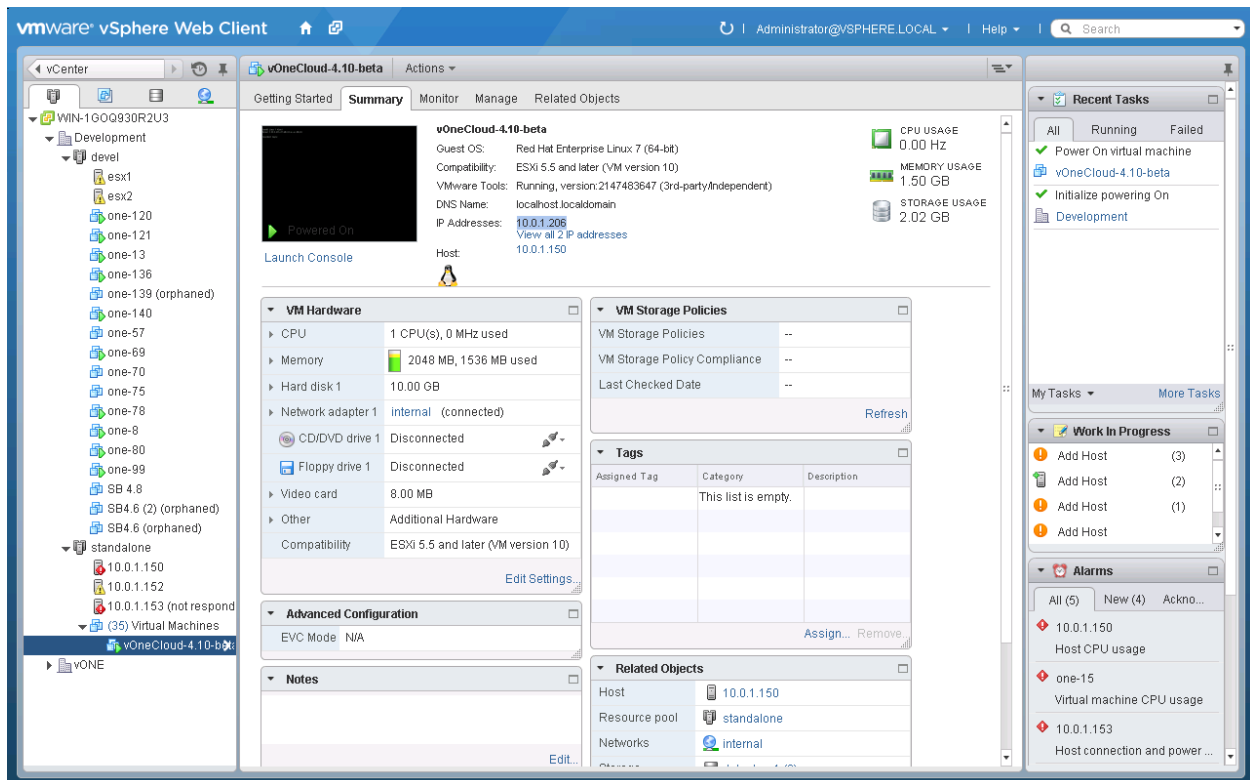
☐ Power on after deployment

Back Next Finish Cancel

Now you can power on the Virtual Machine:



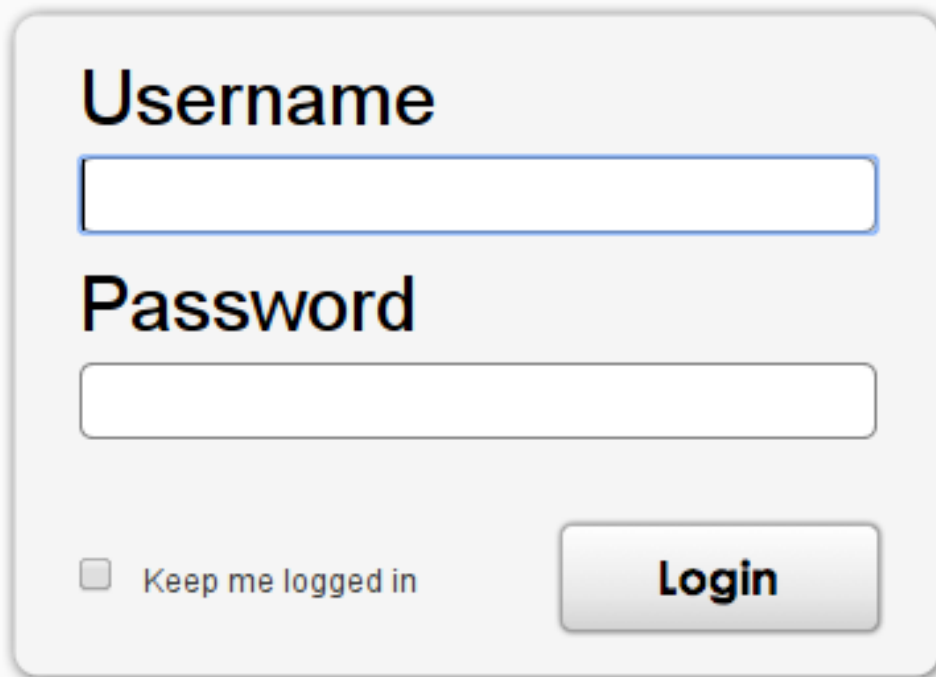
Allow for a few minutes for the Appliance start and report the IP.



2.2.2 Step 2. Connecting to the Sunstone Interface

Now that you have the IP of the appliance you can open the Sunstone Web Interface: `http://<appliance-ip>:9869`.

OpenNebula Sunstone

A login form for OpenNebula Sunstone. It features a light gray background with a rounded rectangle containing the login fields. The fields are labeled 'Username' and 'Password' in a large, bold, black font. Below the 'Password' field is a checkbox labeled 'Keep me logged in'. To the right of the checkbox is a 'Login' button with a gradient background and a shadow effect.

Username

Password

☐ Keep me logged in

Login

To login type in these credentials:

- **Username:** vOneCloud
- **Password:** opennebula

Now you will be able to use the *out-of-the-box features* of vOneCloud!

2.2.3 Step 3. Advanced Usage - Login to the Appliance [Optional]

All the functionality you need to run your vOneCloud can be accessed via Sunstone. However, in order to enable some of the *advanced features* of vOneCloud some extra configurations steps are needed that must be performed in the command line of vOneCloud, and can be carried out opening a console in your vCenter client and log in with these credentials:

- **Username:** root

- **Password:** opennebula

Note: SSH access to the root account has been disabled.

2.3 Import Existing vCenter

Importing a vCenter infrastructure into vOneCloud can be carried out easily through the Sunstone Web UI. Follow the next steps to import an existing vCenter as well as any already defined VM Template.

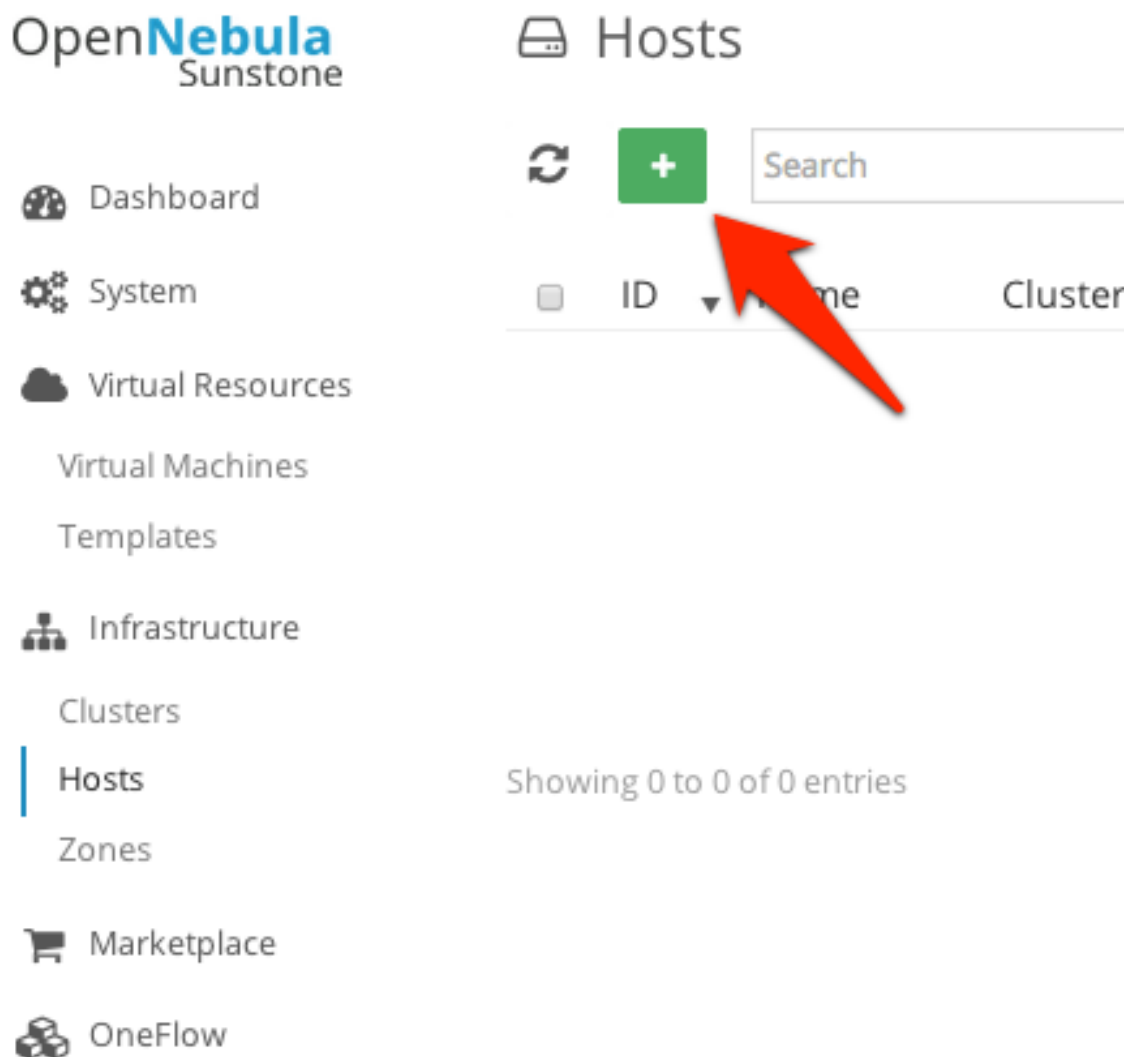
You will need the IP or hostname of the vCenter server, as well as an administrator credentials to successfully import resources from vCenter.

2.3.1 Step 1. Sunstone login

Log in into Sunstone as **vonecloud**, as explained in *the previous section*.

2.3.2 Step 2. Acquire vCenter Resources

In Sunstone, proceed to the `Infrastructure --> Hosts` tab and click on the “+” green icon.



In the dialog that pops up, select vCenter as Type in the dropdown. You now need to fill in the data according to the following table:

Hostname	vCenter hostname or IP address
User	Username of a vCenter user with administrator rights
Password	Password for the above user

Create Host

Type: vCenter Cluster: Default (none)

Hostname: vCenter

User: Administrator@vsphere.local Hostname: 10.1.1.23

Password:

Please select the vCenter Clusters to be imported to OpenNebula. Each vCenter Cluster will be included as a new OpenNebula Host

Development Clusters

☐ standalone

☒ devel

Host created successfully ID:24

Templates

☐ Hadoop Experiment 42162c75-4bd7-fb6f-0cb4-aa94474146d1

☒ HPC Env 4216d5af-7c51-914c-33af-1747667c1019

Template created successfully ID:13

vONE Clusters

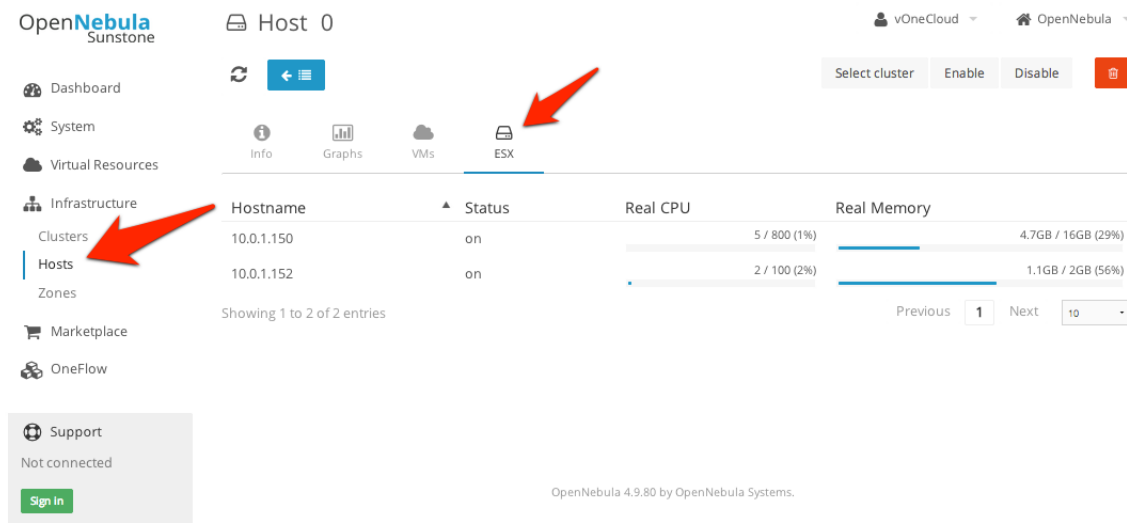
No clusters found in this DataCenter

After the vCenter cluster is selected in Step 2, a list of vCenter VM templates will be presented to be imported into vOneCloud. Select all the templates you want to import, and vOneCloud will generate vOneCloud VM template resources representing the vCenter VM templates.

2.3.3 Step 3. Check resources

Now it's time to check that the vCenter import has been succesful. In Infrastructure --> Hosts check vCenter has been imported, and if all the ESX hosts are available:

Note: Take into account that one vCenter cluster (with all its ESX hosts) will be represented as one vOneCloud host.



2.3.4 Step 4. Instantiate a VM Template

Everything is ready! Now vOneCloud is prepared to manage Virtual Machines. In Sunstone, go to Virtual Resources --> Templates, select one of the templates imported in **Step 2** and click on Instantiate. Now you will be able to control the lifecycle of the VM.

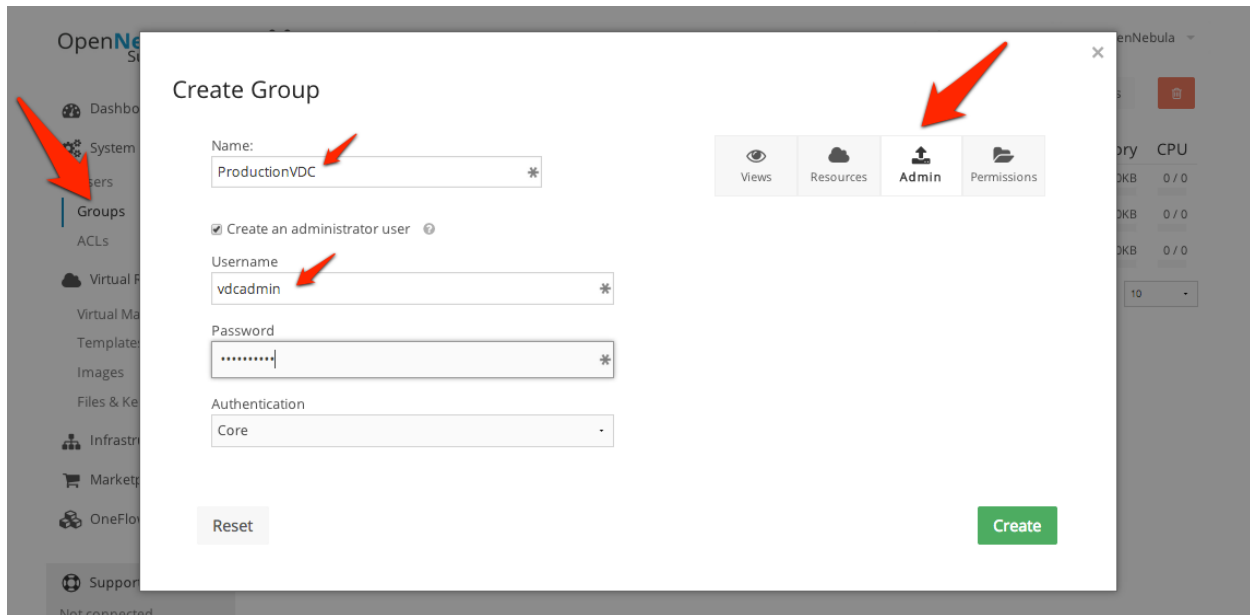
More information on available operations over VMs [here](#).

2.4 Create a Virtual Datacenter

The provisioning model by default in vOneCloud is based on three different *roles* using three different web *interfaces*. *vOneCloud* user comes preconfigured and is the **cloud administrator**, in full control of all the physical and virtual resources and using the vCenter view.

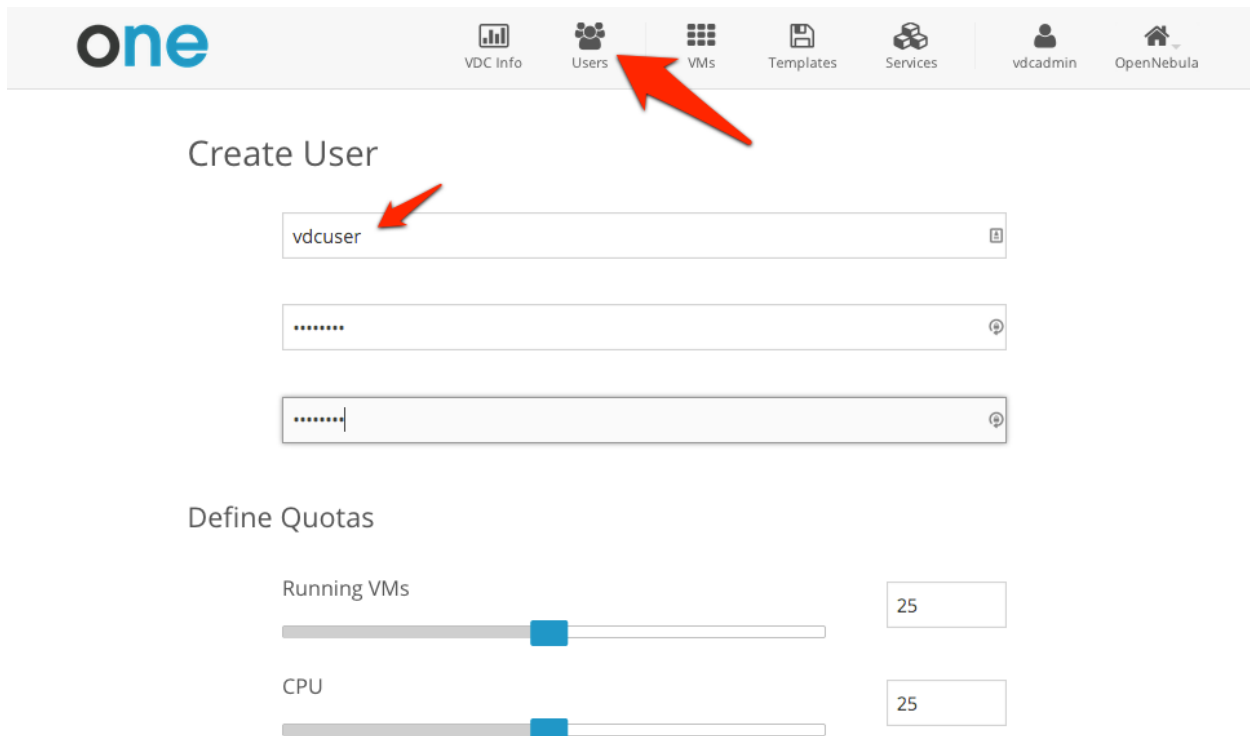
The whole cloud can be divided in isolated partitions, called Virtual Datacenters, or VDCs. VDC are defined as group of users with access to a set of physical hosts and their associated resources in a transparent way. A **VDC Admin** manages her partition of the cloud, including user management, but only within her VDC, not for the whole cloud like the **cloud administrator**.

Let's create a VDC named *ProductionVDC* with an administrator called **vdadmin**:



In the *Resources* tab you can choose which physical resources are assigned to the VDC. By default it will use all the available resources.

Now login again using this newly created **vdcadmin**. The VDC Admin view will kick in. Try it out creating the first *vdcuser* and assign them quotas on resource usage:



As *vOneCloud* user, in the vCenter View, you can assign a VM Template to this VDC:

OpenNebula Sunstone

Template 0

Update Instantiate Clone

Info Template

Information		Permissions:	Use	Manage	Admin
ID	0	Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Name	ttyLinux 64 bits	Group	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Register time	16:30:37 22/10/2014	Other	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ownership

Owner	vOneCloud	<input checked="" type="checkbox"/>
Group	<ul style="list-style-type: none"> 0: oneadmin 1: users 101: ProductionVDC 	<input checked="" type="checkbox"/>

OpenNebula 4.9.80 by OpenNebula Systems.

If you log with *vduser*, the view will change to the Cloud View, where **vduser** can start consuming VMs based on the VM Template shared by the **cloud administrator** and allowed by the **vdadmin**:

one

VMs Templates Services vdcuser OpenNebula

Virtual Machines

+ Add Search Filter Refresh

There are no Virtual Machines

Previous Next 6

OpenNebula 4.9.80 by OpenNebula Systems.

Read more about [VDC managing](#).

2.5 vOneCloud Interfaces

vOneCloud offers a rich set of interfaces to interact with your cloud infrastructure, tailored for specific needs of cloud administrators and cloud users alike.

2.5.1 Web Interface (Sunstone)

vOneCloud web interface, called Sunstone, offers three main views:

- **Sunstone vCenter view:** Aimed at cloud administrators, this view is tailored to present all the available options to manage the physical and virtual aspects of your vCenter infrastructure.
- **Sunstone VDC Admin View:** Aimed at Virtual Datacenter administrators, this interface is designed to manage all the virtual resources of the VDC, including the creation of new users.
- **Sunstone Cloud View:** Aimed at end users, this interface eases virtual resource provisioning and hides all the complexity of the cloud that is going on behind the scenes.

2.5.2 Command Line Interface (CLI)

If you are a SysAdmin, you will probably appreciate vOneCloud's **CLI**, which uses the same design philosophy behind *nix commands (one command for each task).

Moreover, vOneCloud ships with a powerful tool (`onevcenter`) to import vCenter clusters and VM Templates. The tool is self-explanatory, just set the credentials and IP to access the vCenter host and follow on screen instructions.

2.5.3 Cloud Interfaces

Your EC2 ready applications can run on top of vOneCloud, since it implements the **EC2 Query API**.

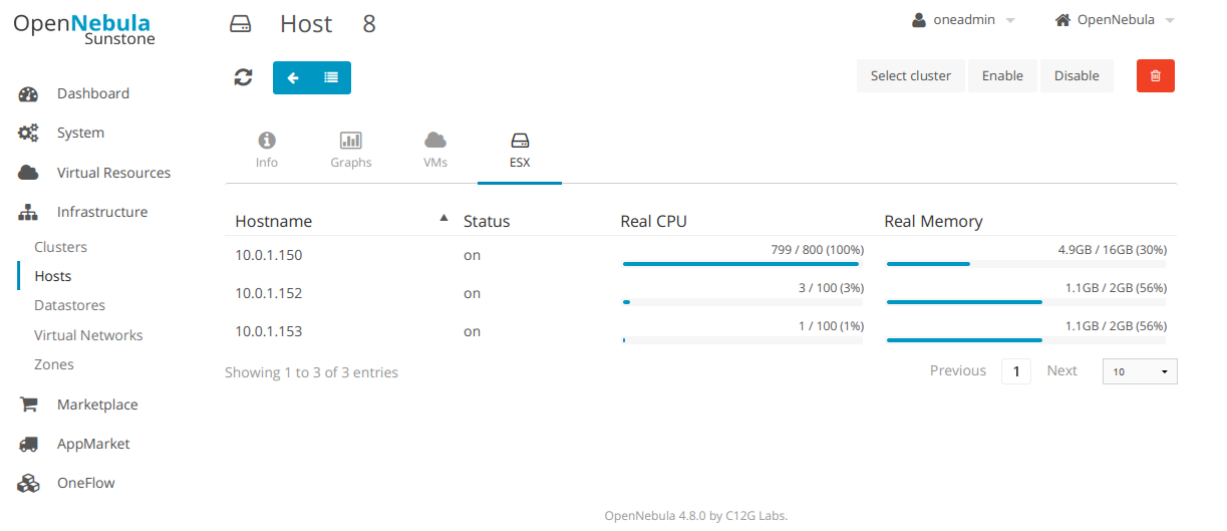
2.5.4 Application Programming Interfaces (API)

If you are a DevOp, you are probably used to build scripts to automate tasks for you. vOneCloud offers a rich set of APIs to build scripts to perform these tasks in different programming languages:

- **xmlrpc API** Talk directly to the OpenNebula core
- **Ruby OpenNebula Cloud API (OCA)** Build tasks in Ruby
- **Java OpenNebula Cloud API (OCA)** Build tasks in Java
- **OneFlow API** Build tasks to manage Multi-VM services

2.6 vCenter View

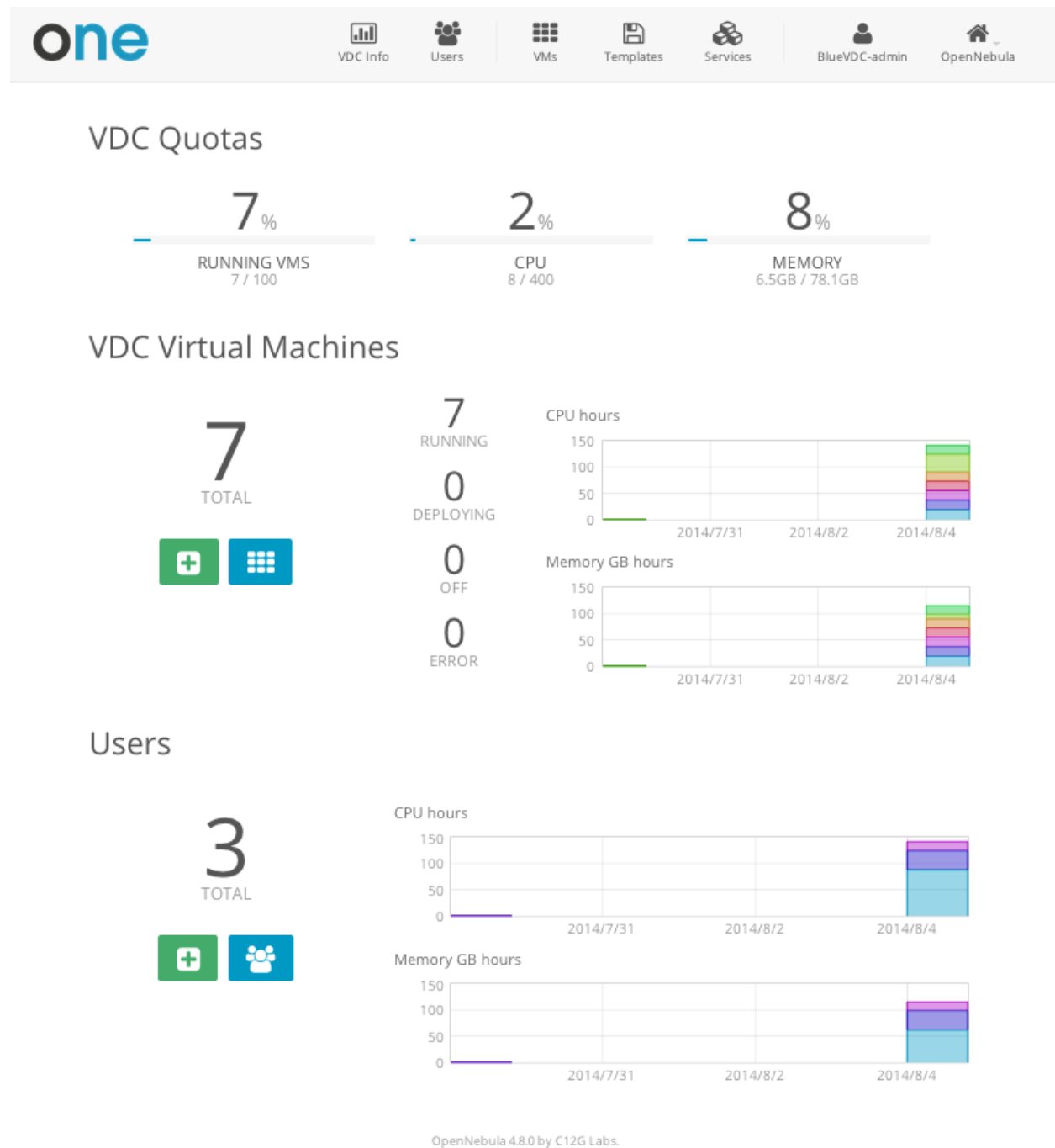
Tailored for common operations with vCenter infrastructures.



More on [vCenter](#) (and other Sunstone) views.

2.7 VDC Admin View

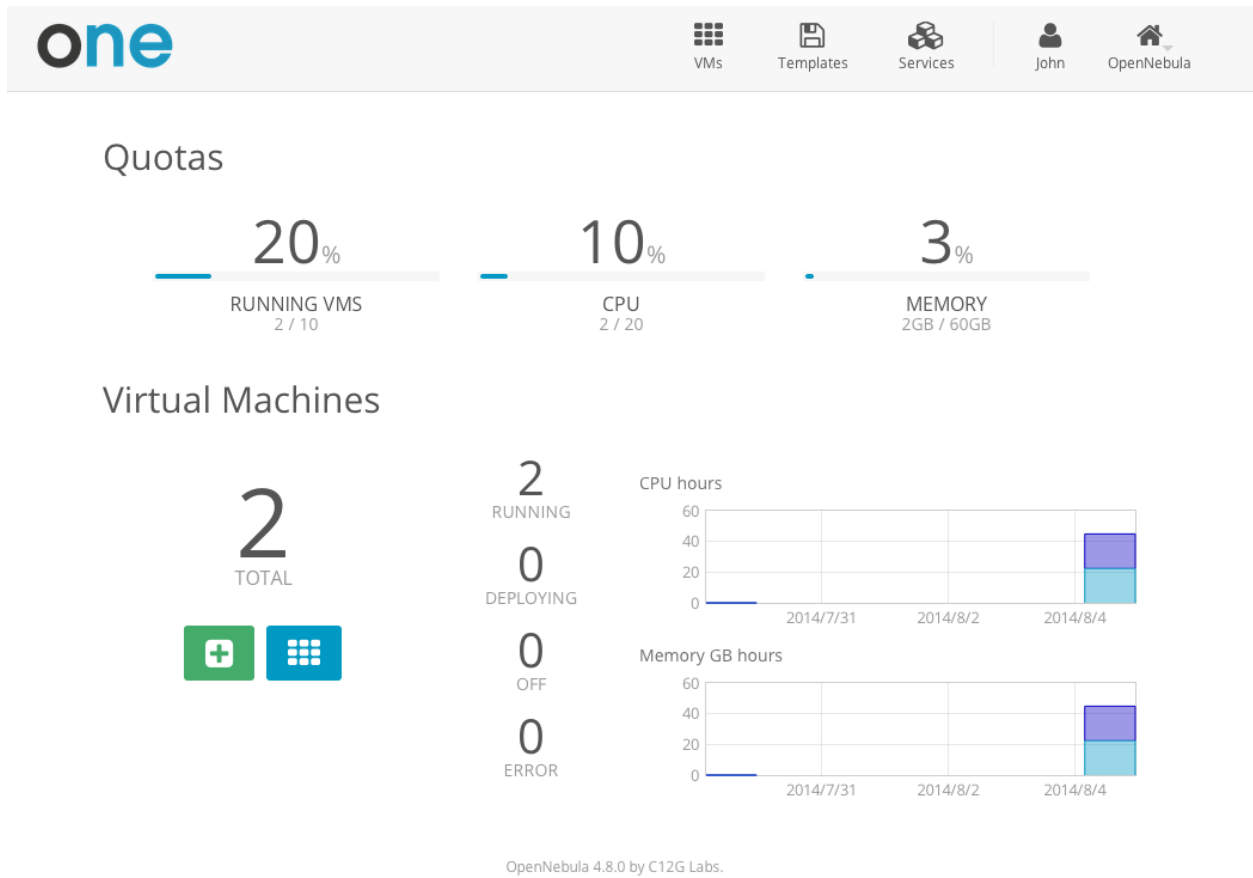
Designed to manage all the virtual resources of the VDC, including the creation of new users.



More on the [VDC Admin View](#).

2.8 Cloud View

Built to ease provisioning to cloud end users.



More on the [Cloud View](#).

SECURITY AND RESOURCE CONSUMPTION CONTROL

3.1 Introduction

vOneCloud ships with several authentication plugins that can be configured to pull user data from existing authentication backends.

vOneCloud also implements a powerful permissions, quotas and ACLs mechanisms to control which users and groups are allowed to use which physical and virtual resources, keeping a record of the consumption of these resources as well as monitoring their state periodically.

Take control of your cloud infrastructure!

3.2 Users, Groups and ACLs

vOneCloud offers a powerful mechanism for managing, grouping and assigning roles to users. Permissions and Access Control List mechanisms ensures the ability to allow or forbid access to any resource controlled by vOneCloud, being physical or virtual.

3.2.1 User & Roles

vOneCloud can manage different types of users, attending to the permissions they have over infrastructure and logical resources.

User Type	Permissions
Cloud Administrators	enough privileges to perform any operation on any object
Infrastructure User	may access most of the functionality to manage resources
VDC Administrators	manage a limited set of resources and users.
VDC Users	access a simplified Sunstone view with limited actions to create new VMs
Public users	access vOneCloud through a public API

Note: VDC is the acronym for Virtual Datacenter

OpenNebula Sunstone Users

oneadmin OpenNebula

Dashboard System Users Groups ACLs Virtual Resources Infrastructure Marketplace OneFlow Support

Showing 1 to 5 of 5 entries

ID	Name	Group	Auth driver	VMs	Memory	CPU
4	Doe	BlueVDC	core	1 / 5	1GB / 10GB	1 / 5
3	John	BlueVDC	core	10 / 10	10GB / 60GB	10 / 20
2	BlueVDC-admin	BlueVDC	core	4 / -	3.5GB / -	5 / -
1	serveradmin	oneadmin	server_cipher	0 / 0	0KB / 0KB	0 / 0
0	oneadmin	oneadmin	core	0 / 0	0KB / 0KB	0 / 0

6 TOTAL

Learn more about user management [here](#).

3.2.2 Group & VDC Management

A **group** in vOneCloud is an authorization boundary for users, but it can also be used to partition the cloud infrastructure and define what resources are available to each group.

A **resource provider** is a set of physical hosts and associated datastores and virtual networks, which is logically grouped into a cluster. When you assign a resource provider to a group, users in that group will be able to use resources of that cluster.

A group and an associated resource provider forms a **Virtual Datacenter (VDC)**. VDCs are a great way to partition your cloud into smaller clouds, with their administrator and users, completely isolated from other VDCs.

OpenNebula Sunstone Groups

oneadmin OpenNebula

Dashboard System Users Groups ACLs Virtual Resources Infrastructure Marketplace OneFlow Support

Showing 1 to 3 of 3 entries

ID	Name	Users	VMs	Memory	CPU
100	BlueVDC	3	15 / 100	14.5GB / 78.1GB	16 / 400
1	users	0	0 / 0	0KB / 0KB	0 / 0
0	oneadmin	2	0 / 0	0KB / 0KB	0 / 0

3 TOTAL

OpenNebula 4.8.0 by C12G Labs.

Read more about [groups](#) and [VDCs](#).

3.2.3 Access Control Lists

vOneCloud implements a very useful ACL mechanism that enables fine-tuning of allowed operations for any user, or group of users. Each operation generates an authorization request that is checked against the registered set of ACL rules. There are predefined ACLs that implements default behaviors (like VDC isolation), but they can be altered by the cloud administrator.

OpenNebula Sunstone

Access Control Lists

oneadmin OpenNebula

Dashboard System Users Groups ACLs Virtual Resources Infrastructure Marketplace OneFlow Support

Search

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

Showing 1 to 10 of 10 entries

« 1 » 10

Writing (or even reading) ACL rules is not trivial, more information about [ACLs here](#).

3.3 Resource Quotas

vOneCloud quota system tracks user and group usage of system resources, allowing the cloud administrator to set limits on the usage of these resources.

Quota limits can be set for:

- **users**, to individually limit the usage made by a given user.
- **groups**, to limit the overall usage made by all the users in a given group.

Tacking the usage on:

Datastores	Control the amount of storage capacity allocated to each user/group for each datastore
Compute	Limit the overall memory, cpu or VM instances
Network	Limit the number of IPs a user/group can get from a given network
Images	Limit the how many VM instances from a given user/group are using a given image

Quotas can be updated either from the vCenter View:

✕

Update Quota

VM

Datastore

Image

VNet

VMs

1 / 5

∞

CPU

1 / 5

∞

Memory

1024 / 10240

MB

∞

Volatile disks

0 / Default (∞)

MB

∞

Apply changes

Or from the VDC Admin View:

one

VDC Info

Users

VMs

Templates

Services

vOneCloud

OpenNebula

Users

VDCUser

Running VMs

25

CPU

25

Memory (GBs)

25

Update User Quota

Refer to [this guide](#) to find out more.

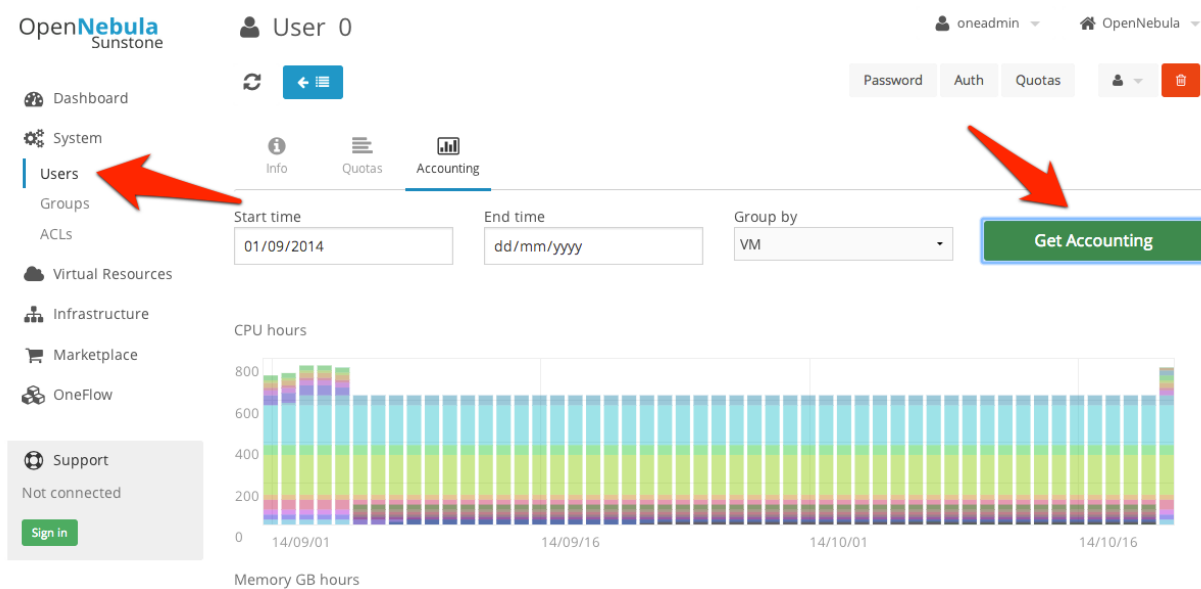
3.4 Accounting & Monitoring

vOneCloud is constantly monitoring the infrastructure resources to keep track of resource consumption. The objective is twofold: being able to have a clear picture of the infrastructure to aid in the resource scheduling, as well as being able to enforce *resource quotas* and give accounting information.

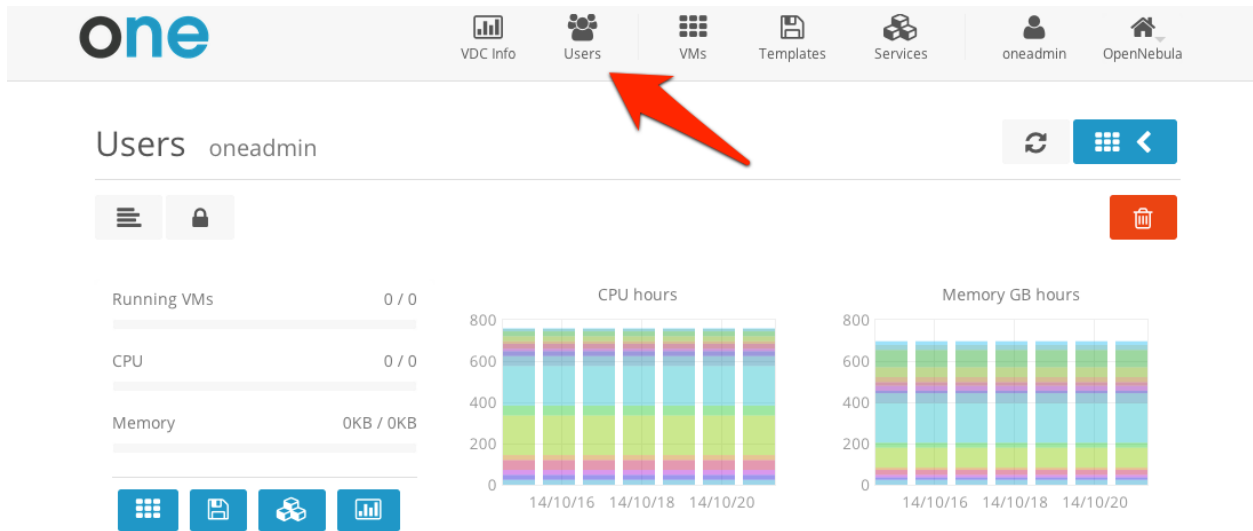
The monitoring subsystem gathers information relative to hosts and virtual machines, such as host and VM status, basic performance indicators and capacity consumption. vOneCloud comes preconfigured to retrieve such information directly from vCenter.

Using the information from the monitoring subsystem, vOneCloud is able to provide accounting information, both in text and graphically. An administrator can see the consumption of a particular user or group in terms of hours of CPU consumed, or total memory used in a given time window. This information is useful to feed a chargeback or billing platform.

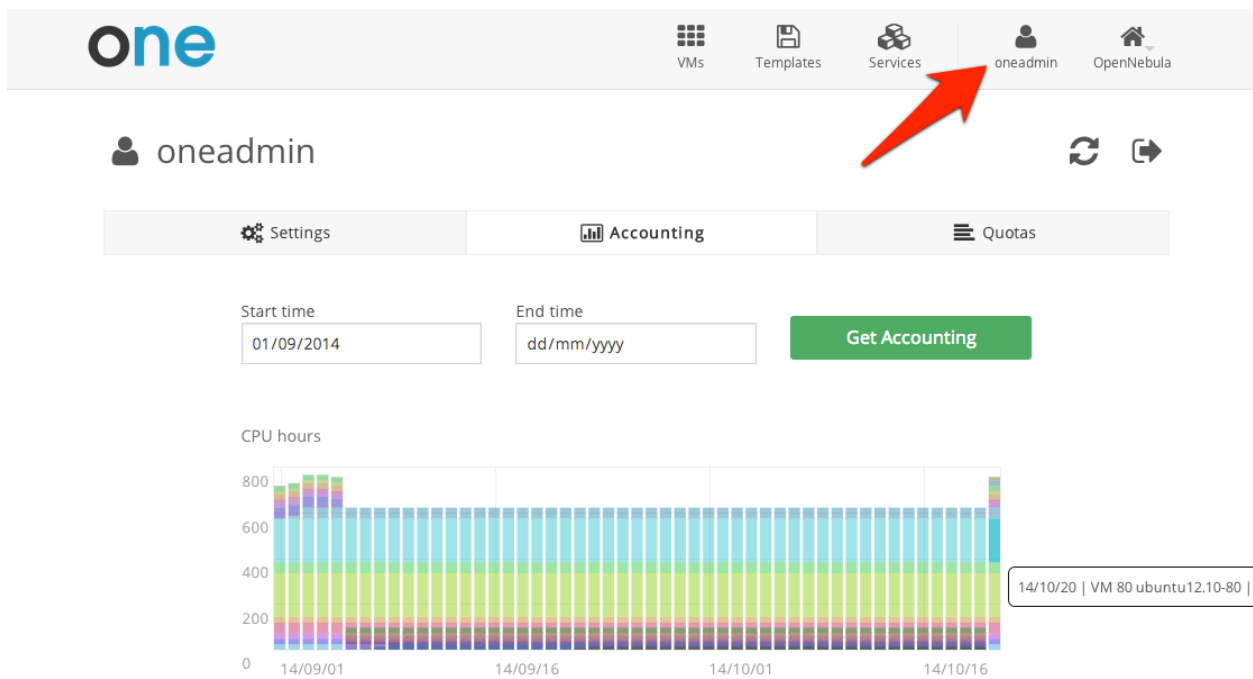
Accounting information is available from the vCenter View:



From the VDC Admin View:



And from the Cloud View:



Learn more on the [monitoring](#) and [accounting](#) subsystems

INFRASTRUCTURE CONFIGURATION

4.1 Introduction

Now that you are familiar with vOneCloud concepts and operations, it is time to extend its functionality by adding new infrastructure components and/or configuring options that do not come enabled by default in vOneCloud, but are present in the software nonetheless.

4.2 Add New vCenters And VM Templates

vOneCloud can manage an unlimited number of vCenters. Each vCenter is going to be represented by an vOneCloud host, which in turn abstracts all the ESX hosts managed by that particular instance of vCenter.

The suggested usage is to build vOneCloud templates for each VM Template in each vCenter. The built in scheduler in vOneCloud will decide which vCenter has the VM Template needed to launch the VM.

The mechanism to add a new vCenter is exactly the same as the one used to *import the first one into vOneCloud*. It can be performed graphically from the vCenter View:

The screenshot shows the OpenNebula Sunstone web interface. On the left sidebar, the 'Hosts' menu item under the 'Infrastructure' section is highlighted with a red arrow. In the center, a 'Create Host' dialog box is open. It contains the following fields: 'Type' (set to 'vCenter'), 'Cluster' (set to 'Default (none)'), 'User' (set to 'vCenter_username'), 'Hostname' (set to 'vCenter_IP_or_Hostname'), and 'Password' (masked with dots). A blue button labeled 'Get vCenter Clusters' is located to the right of the password field and is highlighted with a red arrow. A 'Reset' button is at the bottom left of the dialog. The background shows the main dashboard with a 'Status' section on the right.

Everything is clearer with an example, so here we go:

First things first, to avoid misunderstandings, there are two VM templates we will refer to: the vOneCloud VM Templates and the vCenter VM Templates. The formers are created in the vOneCloud web interface (Sunstone), whereas the latters are created directly through the vCenter Web Client.

A cloud administrator builds two vOneCloud templates to represent one vCenter VM Template available in vCenterA and another available in vCenterB. As previous work, the cloud administrator creates [two vCenter VM templates](#), one in each vCenter.

To create a vOneCloud VM template representing a vCloud vM Template, log in into Sunstone as **nonecloud** user as in explained [here](#), proceed to the Virtual Resources -> Templates, and click on the + sign. In the upper right corner there will be two tabs, click on “Advanced” and input something along the lines of:

```
CPU=1
MEMORY=1024

PUBLIC_CLOUD=[
    TYPE="vcenter",
    VM_TEMPLATE="uuidA" ]

REQUIREMENTS = HOSTNAME = "vCenterA"
```

The above template instantiates vCenter VM Template with **uuidA** in vCenterA. Repeat for vCenterB.

If a user instantiates one of these templates, the vOneCloud scheduler [<http://docs.opennebula.org/>](http://docs.opennebula.org/)

Using [the automated process for importing vCenter infrastructures](#), vOneCloud will generate the above template for you at the time of importing vCenterA.

Read more about the [vCenter drivers](#).

4.3 Hybrid Clouds

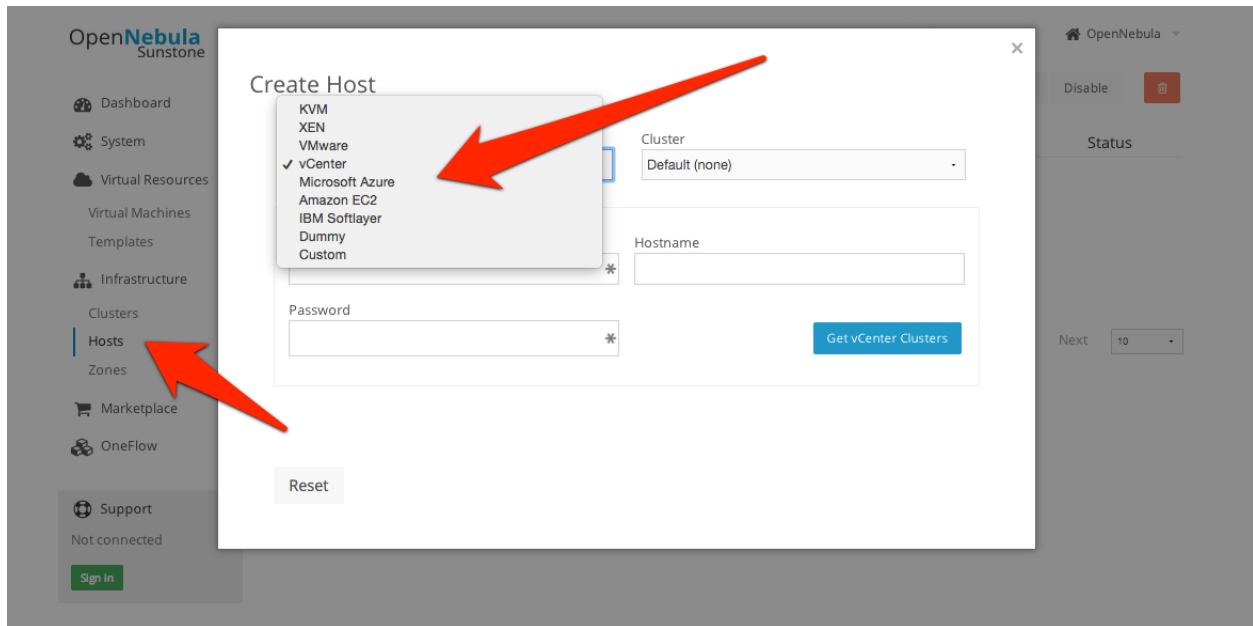
vOneCloud is capable of outsourcing virtual machines to public cloud providers. This is known as cloud bursting, and it is a feature of hybrid clouds where VMs are launched in public clouds if the local infrastructure is saturated.

If you want to extend your private cloud (formed by vOneCloud and vCenter) to create a hybrid cloud, you will need to configure at least one of the supported public clouds: Amazon EC2, IBM SoftLayer and Microsoft Azure. All hybrid drivers are already enabled in vOneCloud, but you need to configure them first with your public cloud credentials.

To configure the drivers you need first to [log into the vOneCloud console](#), and set in the following files (all of them can be found in /etc/one, and edited as root) depending on which public cloud you want to enable(refer to the linked guide for more detailed information):

Amazon EC2	ec2_driver.conf
IBM SoftLayer	sl_driver.conf
Microsoft Azure	az_driver.conf

Afterwards, hybrid hosts can be added from the vCenter View:



The hybrid approach is carried out using hybrid templates, which represents the virtual machines locally and remotely. The idea is to build a vOneCloud hybrid VM template that represents the same VM in vCenter and in the public cloud. Let's see an example of such a template, assuming that vCenter Template uuidA and Amazon EC2 AMI ami-xxxx represents the same virtual machine (for instance, a web server serving a web portal):

```

CPU=1
MEMORY=1024







PUBLIC_CLOUD=[
  TYPE="vcenter",
  VM_TEMPLATE="uuidA" ]

PUBLIC_CLOUD=[
  TYPE="ec2",
  AMI="ami-00bafcb5",
  KEYPAIR="gsg-keypair",
  INSTANCETYPE=m1.small]

```

Once templates are ready, they can be consumed at VM creation time from the Cloud View:

Select a Template

System	VDC	Saved
<input type="text" value="Search"/>		
<div>ttylinux server</div> <div></div> <div>A simple linux server</div>	<div>Windows 7 Desktop</div> <div></div> <div>A windows 7 installation with MS Office suite</div>	<div>RHEL 6.5 - EC2</div> <div></div> <div>RHEL Server 6.5 in Amazon EC2, us-east-1</div>
<div>Windows 8 - Azure</div> <div></div> <div>A Windows VM in Azure cloud</div>	<div>AzureUbuntu14.04</div> <div></div> <div>...</div>	<div>SoftLayerRedHat7</div> <div></div> <div>...</div>
<div>« 1 » 6 ▾</div>		

Learn more about [hybrid support](#).

4.4 Multi VM Applications

vOneCloud enables the management of individual VMs, but also the management of sets of VMs (services) through the OneFlow component.

vOneCloud ships with a running OneFlow, ready to manage services, allowing administrators to define multi-tiered applications using the vCenter View:

OpenNebula Sunstone

Dashboard
System
Virtual Resources
Virtual Machines
Templates
Infrastructure
Clusters
Hosts
Zones
Marketplace
OneFlow
Services
Templates
Support
Not connected
Sign in

Create Service Template

Reset Create

Name *

Description

Network Configuration
Advanced Service Parameters

Roles

Role 0

Role Name

VM template

VMs 1

+ Add another role

End users can consume services from the Cloud View:

one

VMs Templates Services oneadmin OpenNebula

Create Service

Service Name

Select a Template

Search

Hadoop

Frontend 1 VMs
Workers 2 VMs

« 1 » 6

Elasticity of each service can be defined in relation with chosen Key Performance Indicators, either as reported by the hypervisor or by the service itself through the [OneGate component](#).

More information on this component in the [OneFlow guide](#). Also, extended information on how to manage multi-tier applications is available [this guide](#).

4.5 Authentication

By default, vOneCloud authentication uses an internal user/password system with user and group information stored in an internal database.

If you want vOneCloud to have your users pulled from a corporate Active Directory, all the needed components are enabled and just an extra configuration step is needed. As requirements, you will need an Active Directory server with support for simple user/password authentication, as well as a user with read permissions in the Active Directory user's tree.

Log into the vOneCloud console, and then proceed to fill the following information in `/etc/one/auth/ldap_auth.conf`:

You will need to change the following values in the configuration file (`/etc/one/auth/ldap_auth.conf`):

user	Active Directory user with read permissions in the user's tree plus the domain.
password	password of this user
host	hostname or IP of the Domain Controller
base	base DN to search for users.
user_field	Set it to "sAMAccountName"

You can find more information on the integration with Active Directory [in this guide](#).

vOneCloud supports a variety of other authentication methods that can be configured, follow the links to find the configuration steps needed:

LDAP	vOneCloud will connect to an existing LDAP server and retrieve information about a user that is trying to login
X509 Authentication	Strengthen your cloud infrastructure security
SSH Authentication	Users will generate login tokens based on standard ssh rsa keypairs for authentication