OSDC 2012

24th April, Nürnberg

Building Clouds with OpenNebula 3.4

Constantino Vázquez Blanco

dsa-research.org | OpenNebula.org

Distributed Systems Architecture Research Group
Universidad Complutense de Madrid

Building Clouds with OpenNebula 3.4
Preparing the Cloud for OpenNebula

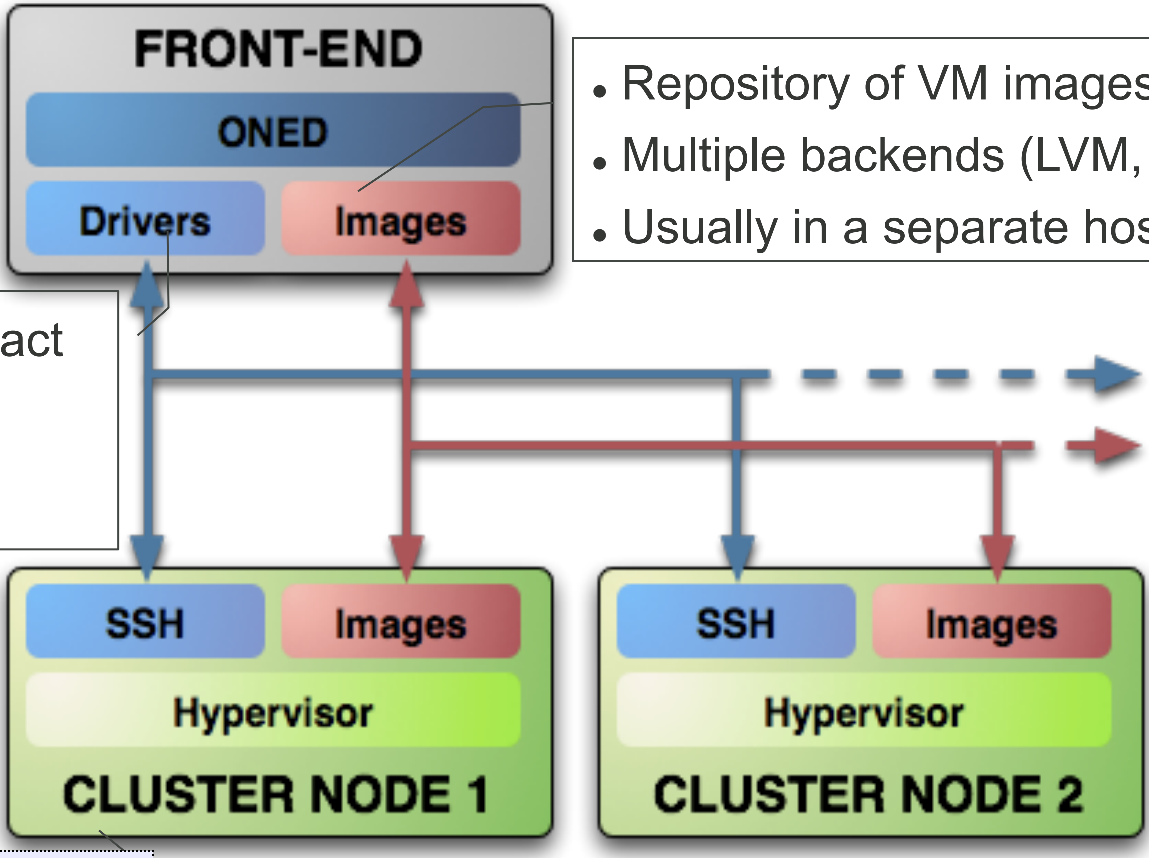
Constantino Vázquez Blanco
dsa-research.org | OpenNebula.org
Distributed Systems Architecture Research Group
Universidad Complutense de Madrid

- Overview of OpenNebula Components
- Runtime & Compilation Requirements
- Users & File-System
- Storage for the Private Cloud
- Networking for the Private Cloud
- Hypervisor Configuration
- Checklist

Component Overview

Preparing the cloud for OpenNebula

- Executes the OpenNebula Services
- *Usually* acts as a classical cluster front-end



- Repository of VM images
- Multiple backends (LVM, iSCSI..)
- Usually in a separate host

- Modular components to interact with the cluster services
- *Types:* storage, monitoring, virtualization and network

❗ The same host can be can be a the front-end and a node

- Provides physical resources to VMs
- *Must have* a hypervisor installed

Runtime Requirements

Cluster Front-end

- Choose your installation mode
  - system wide (/usr, /etc...)
  - *self-contained* (under $ONE_LOCATION)
- Install software dependencies (runtime)

```
# apt-get install ruby
# apt-get install sqlite3
# apt-get install libxmlrpc-c3
```

install_gems.sh

Users & File-System Layout

Cluster Front-end

- The Users of the private cloud:
  - **oneadmin**: Account to run the daemons, manage the system and do all the low-level operations (e.g. start VMs, move images...).
  - **Users**: create and manage their own VMs and networks. *Need to be defined in OpenNebula*
- Installation layout – Self-contained
  - We will use the **/srv/cloud** directory to place the OpenNebula software
  - **/srv/cloud/one** will hold the OpenNebula installation
  - **/srv/cloud/one/var/datastores** will contain all the datastores

Users & File-System Layout

Cluster Front-end

- Installation layout

```
# tree /srv
/srv/
`-- cloud
    |-- one
        |-- var
           |-- datastores
                |-- 0
                |-- 1
```

The oneadmin account must be created system wide (i.e. front-end and all the nodes) you can use NIS, or a local account with the same ID's in all the hosts. Users do not need a UNIX account in the nodes, nor in the front-end.

Storage for the Private Cloud

Component Overview

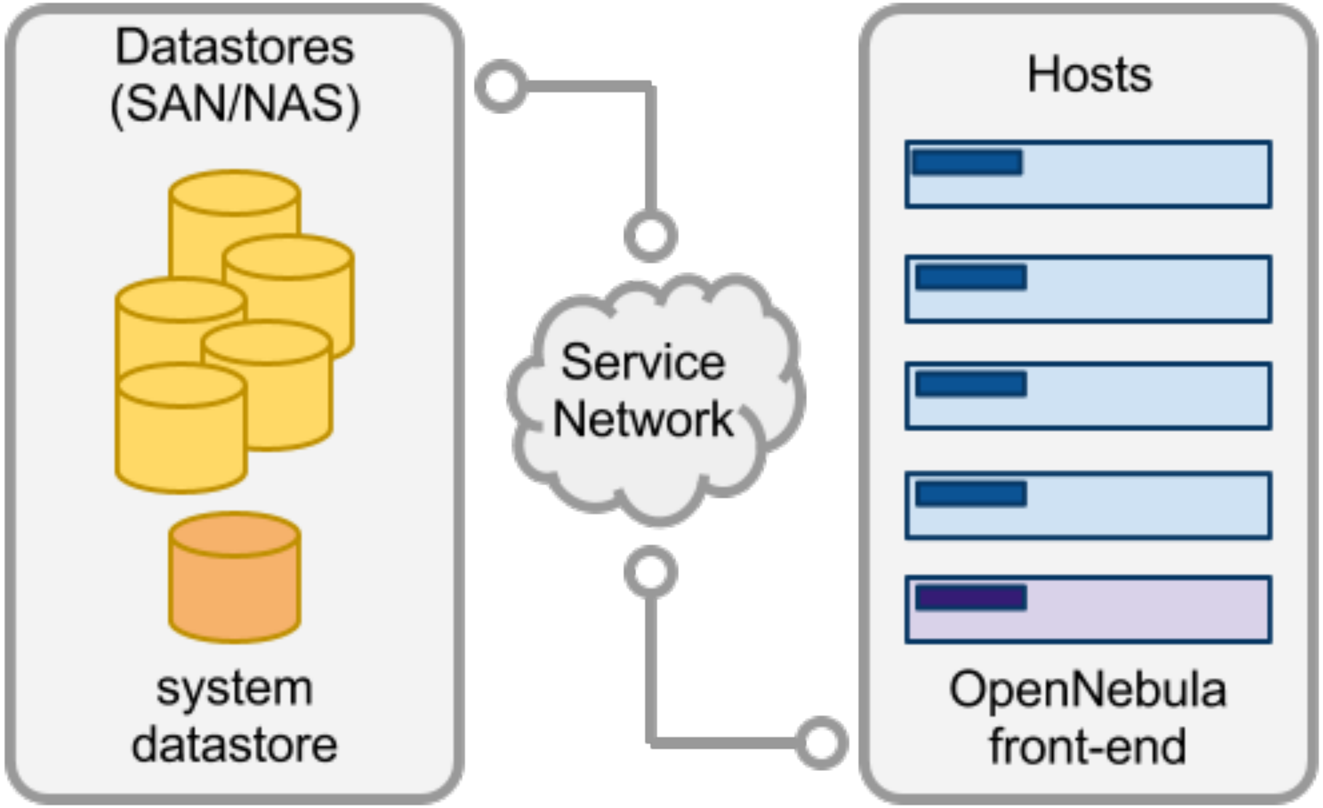
- **Datastore:** Any storage medium for the VM images (usually a high performing SAN). In this course a fs-based repo.
- **Cluster Storage**
  - OpenNebula supports multiple back-ends (e.g. LVM for fast cloning)
  - From 3.4 onwards – use multiple back-ends simultaneously
- **VM Directory:** The home of the VM in the cluster node
  - Stores checkpoints, description files and VM disks
  - Actual operations over the VM directory depends on the storage medium
  - Should be shared for live-migrations
  - You can go on without a shared FS and use the SSH back-end
  - Defaults to `$ONE_LOCATION/var/$VM_ID`

Storage for the Private Cloud

Example, a shared FS architecture

Dimensioning the Storage... Example: A 64 core cluster will typically run around 80VMs, each VM will require an average of 10GB of disk space. So you will need ~800GB for /srv/cloud/one, you will also want to store 10-15 master images so ~200GB for /srv/cloud/one/var/datastores. 1TB /srv/cloud will be enough for this example setup.

- One typical setup is to use NFS to share the VM directories
- The Datastores are contained in /srv/cloud/one/var/datastores



Storage for the Private Cloud

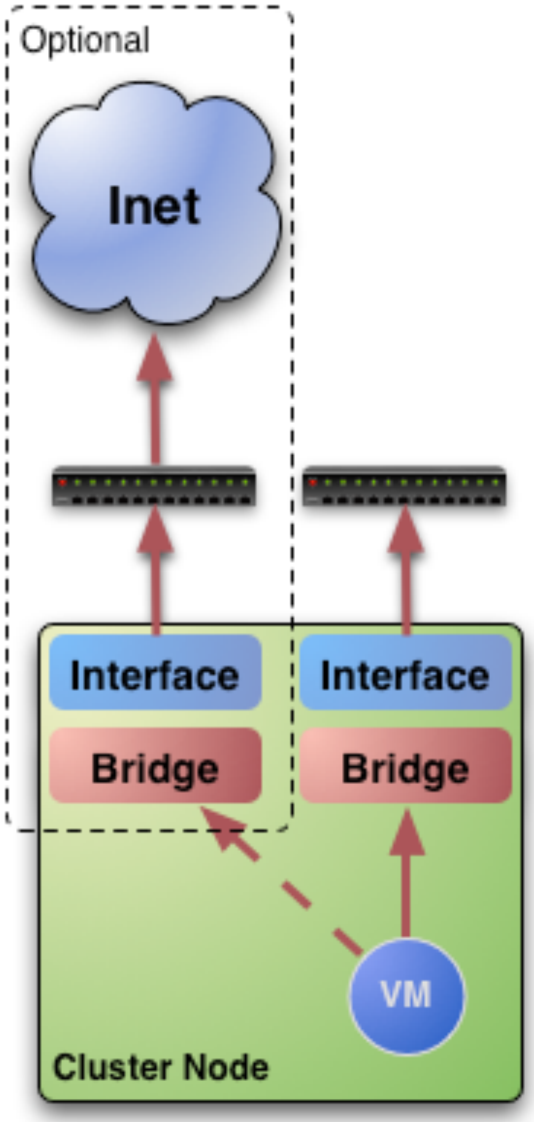
Configuring NFS backend

```
# apt-get install nfs-kernel-server

Export /srv/cloud to your nodes
  - only need /srv/cloud/one/var
  - we also export $HOME of oneadmin for easy SSH key configuration
  - No need to export /srv/cloud/images

# vim /etc/exports
/srv/cloud 193.144.33.YY(rw,async,no_subtree_check,no_root_squash)

# service nfs-kernel-server restart
# service ufw stop
# iptables -F
```

Networking for the Private Cloud

Component Overview

- OpenNebula management operations uses a ssh connections, it does not require a performing NIC
- **Image traffic,** may require the movement of heavy files (VM images, checkpoints). Dedicated storage links may be a good idea
- **VM demands,** consider the typical requirements of your VMs. Several NICs to support the VM traffic may be a good idea
- OpenNebula relies on bridge networking for the VMs



Runtime Requirements

Cluster Worker-nodes

- Install **software dependencies**
  - We need SSH daemon running in the cluster nodes
  - Runtime dependencies:

```
# apt-get install ruby
```

- **Users**
  - Create the oneadmin account (**use same UID and GID**)

```
# groupadd -g 3000 cloud
# useradd -d /srv/cloud/one -g cloud -u 3000 -s /bin/bash oneadmin
```

  - Set language environment to english

```
# cat /etc/default/locale
LANG="en_US.UTF-8‟
```

Runtime Requirements

Configuring SSH access

- Enable **password-less SSH** access to cluster (oneadmin)

```
Do not protect the private key with a password
$ ssh-keygen
Generating public/private rsa key pair.
...
Enter passphrase (empty for no passphrase):
Enter same passphrase again:

$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
Tell ssh client not to ask to add hosts to known_hosts (optional)
$ cat /srv/cloud/one/.ssh/config
Host *
    StrictHostKeyChecking no

$ ssh 193.144.33.yy
```

❗ You may need to exchange keys with the nodes (not here as we share /srv/cloud/one the oneadmin home, and so the ~/.ssh directory)

Hypervisor Configuration

Example, configuring KVM in the Worker-nodes

- Installing the **Hypervisor**
  - OpenNebula supports KVM, Xen and Vmware (*even simultaneously*). This course applies to KVM and Xen
  - Refer to the hypervisor documentation for additional (and better information) on setting up them.
- **Setting up KVM and libvirt** (Ubuntu 10.04)
  - Install the packages (should be already installed)

```
#apt-get install qemu-common qemu-kvm libvirt-bin
```

Hypervisor Configuration

Example, configuring KVM in the Worker-nodes

- Add oneadmin to the libvirt group

```
# usermod -G kvm,libvirtd oneadmin
```

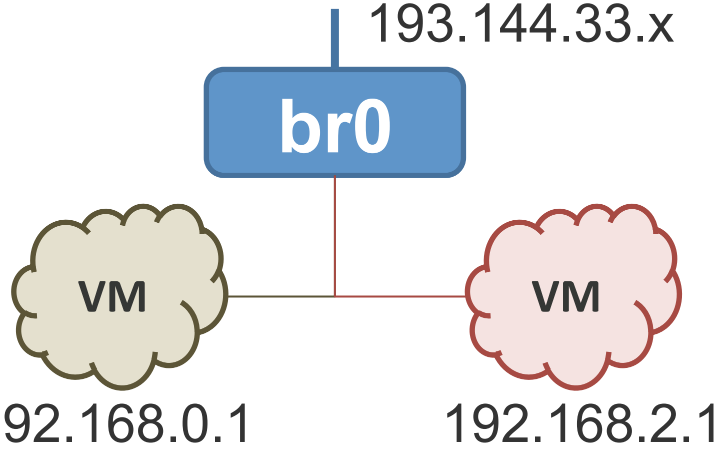
- Test the installation for the oneadmin account

```
$ virsh -c qemu:///system list

 Id Name                    State
----------------------------------
```

Networking for the Private Cloud

Network Configuration in the Worker-nodes

- **Setting up KVM and libvirt** (Ubuntu 10.04)
  - Networking for this course



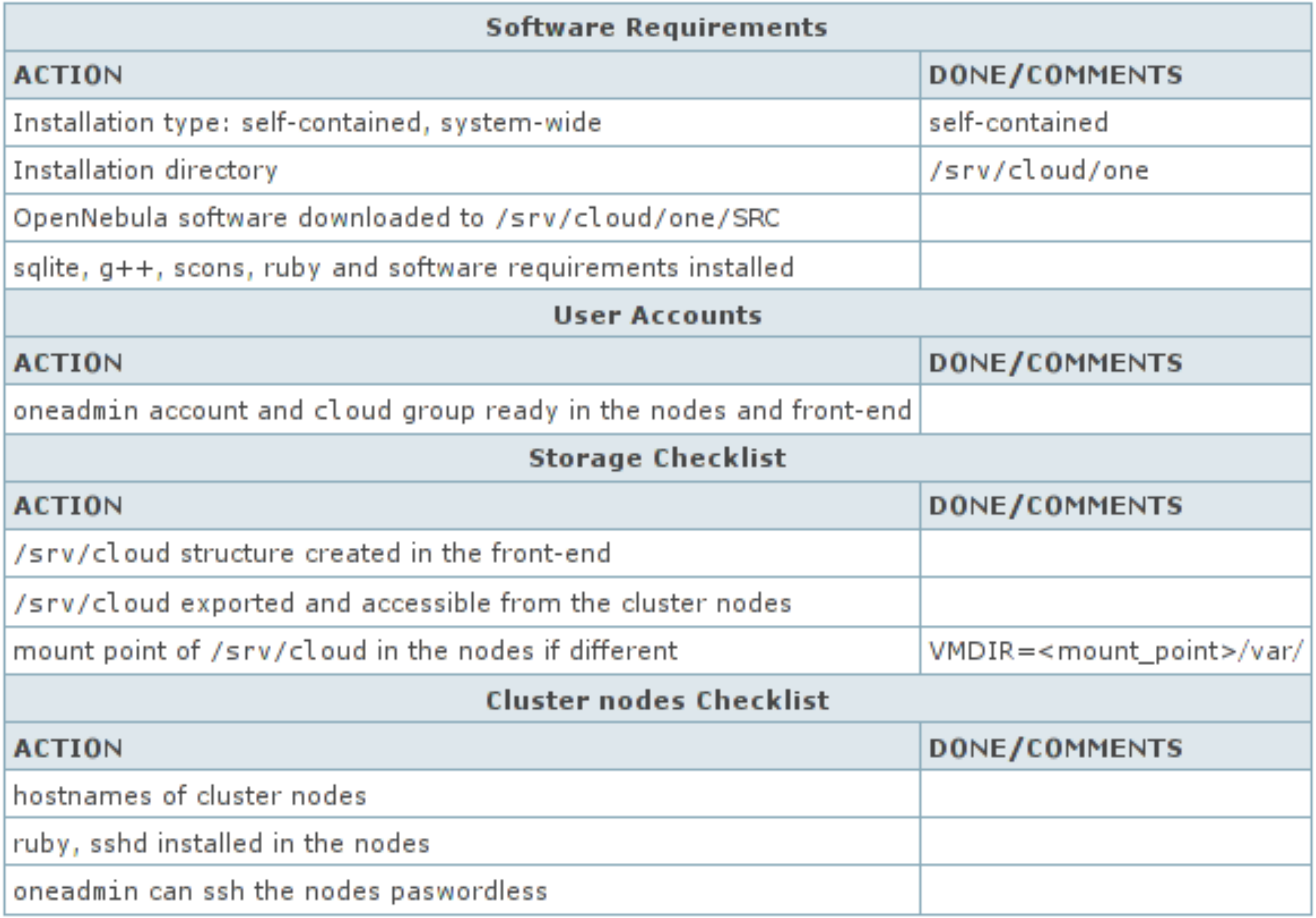
Automated in ONE 3.2+ !!

  - Disable virbr0

```
#rm /etc/libvirt/qemu/networks/autostart/default.xml
#ifconfig virbr0 down
#brctl delbr virbr0
#service libvirt-bin restart
```

Installation Checklist

Preparing the cloud for OpenNebula



Software Requirements	
ACTION	**DONE/COMMENTS**
Installation type: self-contained, system-wide	self-contained
Installation directory	/srv/cloud/one
OpenNebula software downloaded to /srv/cloud/one/SRC	
sqlite, g++, scons, ruby and software requirements installed	
User Accounts	
ACTION	**DONE/COMMENTS**
oneadmin account and cloud group ready in the nodes and front-end	
Storage Checklist	
ACTION	**DONE/COMMENTS**
/srv/cloud structure created in the front-end	
/srv/cloud exported and accessible from the cluster nodes	
mount point of /srv/cloud in the nodes if different	VMDIR=<mount_point>/var/
Cluster nodes Checklist	
ACTION	**DONE/COMMENTS**
hostnames of cluster nodes	
ruby, sshd installed in the nodes	
oneadmin can ssh the nodes paswordless	

Building Clouds with OpenNebula 3.4

Preparing the Cloud for OpenNebula

Constantino Vázquez Blanco
dsa-research.org | OpenNebula.org
Distributed Systems Architecture Research Group
Universidad Complutense de Madrid

OpenNebula.org
The Open Source Toolkit for Cloud Computing

- Overview of OpenNebula Components
- Runtime & Compilation Requirements
- Users & File-System
- Storage for the Private Cloud
- Networking for the Private Cloud
- Hypervisor Configuration
- Checklist