

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

Installing & Configuring OpenNebula

Constantino Vázquez Blanco

dsa-research.org | OpenNebula.org

Distributed Systems Architecture Research Group

Universidad Complutense de Madrid



- Installing OpenNebula 3.4
- Configuring OpenNebula 3.4
- Managing Hosts
- Managing Datastores
- Managing Clusters
- Managing Users
- Logging and Debugging Information

Installing OpenNebula 3.4

Compiling the Software

- Grab the source code and compile it (oneadmin)

```
~/SRC$ tar xzvf one-3.4.tar.gz
~/SRC$ cd opennebula-3.4/
~/SRC$ scons
```

- Install the software in /srv/cloud/one (ONE_LOCATION)

```
$ export ONE_LOCATION=/srv/cloud/one/
$ ./install.sh -d $ONE_LOCATION
```

Check install.sh -h for other options

- Check and explore the installation tree

```
~$ ls -F
bin/      etc/      include/  lib/      share/    var/
```

Upgrading from previous versions

onedb utility

- From 2.2+ onwards there is a process to upgrade
- onedb command updates the database

```
~$ onedb upgrade -v --sqlite /var/lib/one/one.db
```

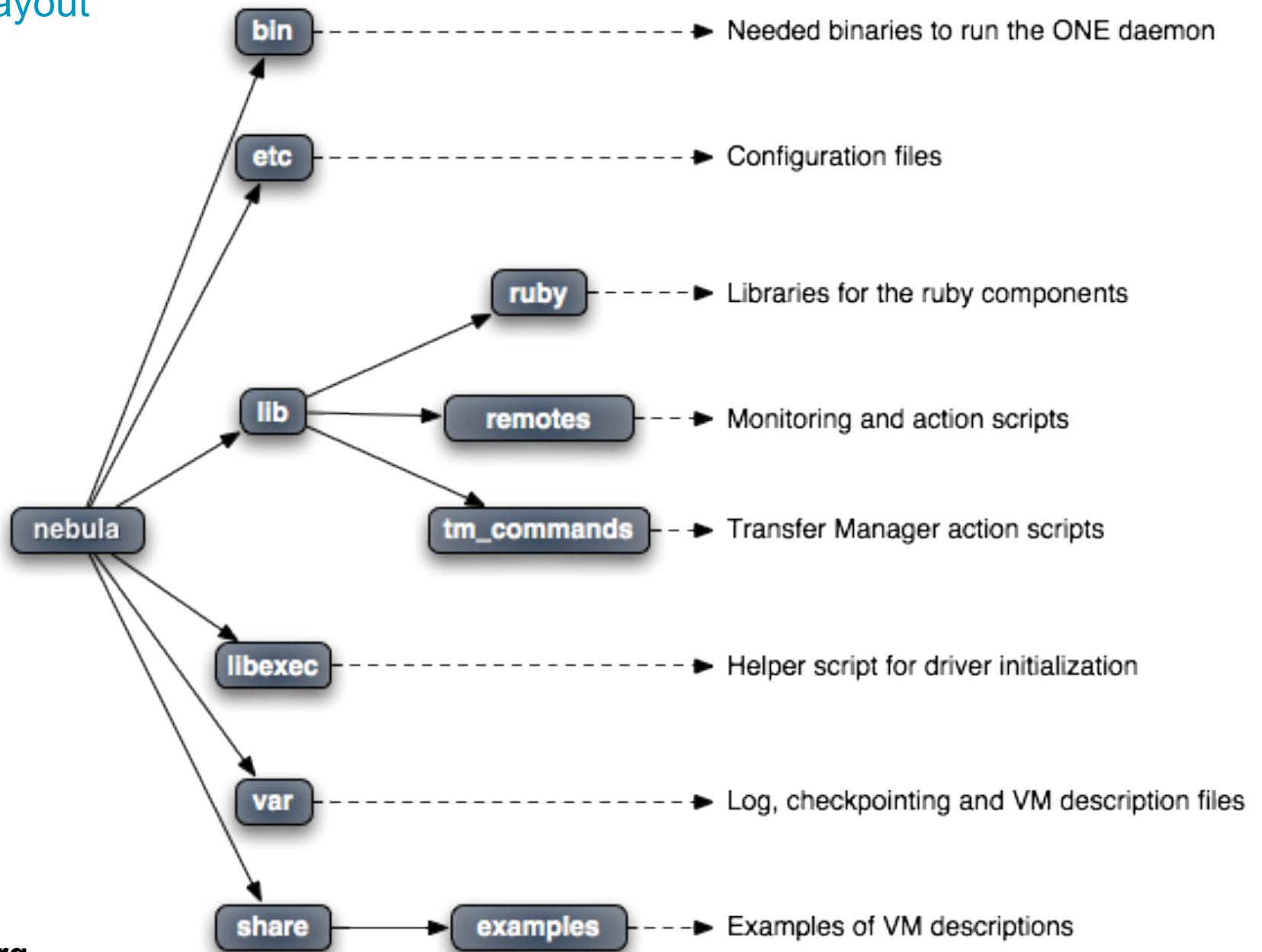
<http://opennebula.org/documentation:rel3.4:upgrade>

- Good read to find out changes between versions (Compatibility Guide)

<http://opennebula.org/documentation:rel3.4:compatibility>

Installing OpenNebula 3.4

Installation layout

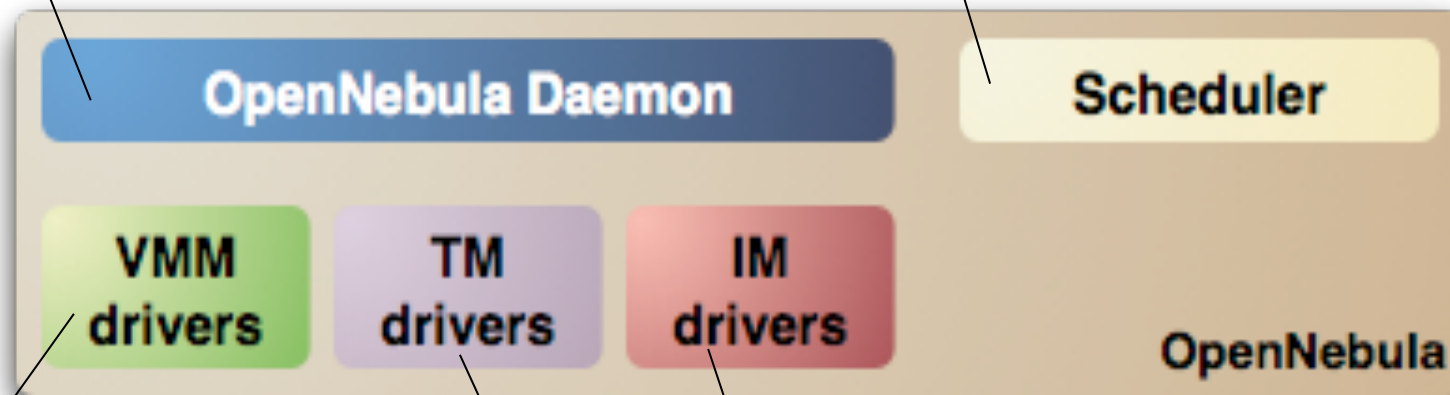


Configuring OpenNebula 3.4

Configuration Interface

- `$ONE_LOCATION/etc/oned.conf`
- General configuration
- Defines the drivers used in the private cloud

- Match-making scheduler (default)
- Placement policies configured per VM



- `$ONE_LOCATION/etc/vmm_*/vmm_*.conf`
- Defaults values for the hypervisor

- `$ONE_LOCATION/etc/im_*/im_*.conf`
- Defines monitoring probes

- `$ONE_LOCATION/etc/tm_*/tm_*.conf`
- Defines action for generic storage operations

Configuring OpenNebula

The oned.conf file

- **General configuration attributes**

- Monitoring intervals:
 - HOST_MONITORING_INTERVAL
 - VM_POLLING_INTERVAL
- Global Paths
 - DATASTORE_LOCATION: Unique mount point for datastores **in the** cluster nodes.
 - SCRIPTS_REMOTE_DIR: to store driver actions in the cluster nodes
- PORT : Port where oned will listen for xml-rpc calls
- DEBUG_LEVEL
- DB, configuration for the DB backend driver:
 - Sqlite
 - MySQL
- VNC_BASE_PORT, for VNC port generation (BASE + ID)

Configuring OpenNebula

The oned.conf file

```
#MANAGER_TIMER=30

HOST_MONITORING_INTERVAL = 60
VM_POLLING_INTERVAL      = 60

#DATSTORE_LOCATION=/srv/cloud/one/var

SCRIPTS_REMOTE_DIR=/var/tmp/one

PORT=2633

DB = [ backend = "sqlite" ]

VNC_BASE_PORT = 5900

DEBUG_LEVEL=3
```


Configuring OpenNebula

The oned.conf file

- **Networking**

- MAC_PREFIX, for MAC address generation
- NETWORK_SIZE, default value

- **Image Repository**

- IMAGE_REPOSITORY_PATH, to store the images
- DEFAULT_IMAGE_TYPE: OS, CDROM, DATABLOCK
- DEFAULT_IMAGE_PREFIX: hd, sd, xvd, vd

```
MAC_PREFIX      = "00:02"  
NETWORK_SIZE   = 254  
  
IMAGE_REPOSITORY_PATH = /srv/cloud/images  
DEFAULT_IMAGE_TYPE    = "OS"  
DEFAULT_DEVICE_PREFIX = "hd"
```

Configuring OpenNebula

The oned.conf file

- **Information Drivers**, to monitor cluster nodes
 - name: identifies the driver
 - executable: absolute or relative to \$ONE_LOCATION/lib/mads
 - arguments:
 - hypervisor probe set (remotes dir)
 - Number of retries (-r)
 - Concurrency (-t number of threads)

```
IM_MAD = [  
    name          = "im_kvm",  
    executable    = "one_im_ssh",  
    arguments     = "-r 0 -t 15 kvm" ]
```

Configuring OpenNebula

The oned.conf file

- **Transfer Drivers**, to interface with the storage
 - name: identifies the driver
 - executable: path to driver executable
 - arguments: storage commands configuration file

```
TM_MAD = [  
    name      = "tm_nfs",  
    executable = "one_tm",  
    arguments  = "tm_nfs/tm_nfs.conf" ]
```

Configuring OpenNebula

The oned.conf file

- **Datastore drivers**, to add images to OpenNebula
 - executable: path to driver executable
 - arguments: type of allowed datastores

```
DATASTORE_MAD = [  
    executable = "one_datastore",  
    arguments  = "-t 15 -d fs,vmware,iscsi"  
]
```

Configuring OpenNebula

The oned.conf file

- **Virtualization Drivers**, to interface the hypervisors
 - name: identifies the driver
 - executable: absolute or relative to \$ONE_LOCATION/lib/mads
 - arguments: same as Information Drivers
 - default: default values for the hypervisor
 - type: format of the VM description used by the driver: xen, kvm or xml

```
VM_MAD = [  
    name           = "vmm_kvm",  
    executable     = "one_vmm_ssh",  
    arguments      = "-t 15 -r 0 kvm",  
    default        = "vmm_ssh/vmm_ssh_kvm.conf",  
    type           = "kvm" ]
```

Configuring OpenNebula

The oned.conf file

- **Other sections**, to interface the hypervisors
 - Hooks
 - Auth Manager
 - Different auth mechanisms: x509, ldap, plain

Configuring OpenNebula

The oneadmin account

- **Accounts** in OpenNebula
 - **oneadmin**, has enough privileges to perform any operation on any object. It is created the first time OpenNebula is started using the ONE_AUTH data
 - Regular **user accounts** must be created by oneadmin and they can only manage their own objects.
- Configuring **the oneadmin account**
 - Environment variables: ONE_AUTH, ONE_LOCATION and ONE_XMLRPC

```
$ tail .bashrc
export ONE_LOCATION=/srv/cloud/one
export ONE_AUTH=$HOME/.one/one_auth
export PATH=$PATH:$ONE_LOCATION/bin
```

Configuring OpenNebula

The oneadmin account


- Create the password file

```
$ mkdir .one
$ cd .one
$ cat one_auth
oneadmin:onecloud
```

- **Start OpenNebula** using the init scripts

```
$ source .bashrc
$ echo $ONE_AUTH
/srv/cloud/one/.one/one_auth

$ one start
$ less $ONE_LOCATION/var/oned.log
Thu May 05 18:03:11 2011 [ONE][I]: Init OpenNebula Log system
...
```

 Be sure to configure the oneadmin account (specially, create the ONE_AUTH file) before starting OpenNebula for the first time.

Configuring the Private Cloud

Managing hosts

- **Hosts** are cluster worker-nodes defined with
 - *Hostname* of the node or IP
 - *Information Driver* to be used to monitor the host
 - *Storage Driver* to clone, delete, move or copy images into the host
 - *Virtualization Driver* to boot, stop, resume VMs in the host
- Hosts are managed with the **onehost** utility
 - Create & delete hosts
 - List the hosts
 - Show detailed information from a host
 - Enable/Disable a host

Configuring the Private Cloud

Managing hosts

```
$ onehost add pcaulaX -i im_kvm -v vmm_kvm -n 802.1q
$ onehost add pcaulaY -i im_kvm -v vmm_kvm -n 802.1q
$ onehost list
```

| ID | NAME | CLUSTER | RVM | TCPU | FCPU | ACPU | TMEM | FMEM | STAT |
|----|---------|---------|-----|------|------|------|------|------|------|
| 0 | pcaulaX | default | 0 | 0 | 0 | 100 | 0K | 0K | on |
| 1 | pcaulaY | default | 0 | 0 | 0 | 100 | 0K | 0K | on |

```
...
$ cat $ONE_LOCATION/var/oned.log
Mon May  2 18:06:35 2011 [InM][I]: Monitoring host pcaula7 (0)
Mon May  2 18:06:35 2011 [InM][I]: Monitoring host pcaula10 (1)
Mon May  2 18:06:38 2011 [InM][D]: Host 0 successfully monitored.
Mon May  2 18:06:39 2011 [InM][D]: Host 1 successfully monitored.
...
$ onehost list
```

| ID | NAME | CLUSTER | RVM | TCPU | FCPU | ACPU | TMEM | FMEM | STAT |
|----|----------|---------|-----|------|------|------|------|------|------|
| 0 | pcaula7 | default | 0 | 200 | 199 | 200 | 1.9G | 1.5G | on |
| 1 | pcaula10 | default | 0 | 200 | 200 | 200 | 1.9G | 1.5G | on |

Configuring the Private Cloud

Managing clusters

- Clusters are pools of hosts that share datastores and virtual networks. Clusters are used for load balancing, high availability, and high performance computing.
- By default, all hosts belong to the *default* **cluster**. Clusters are managed using the **onecluster** command
 - Create & delete clusters
 - List the available clusters
 - Add & remove hosts from the clusters
 - Add & remove Datastores from the clusters
 - Add & remove Virtual Networks from the clusters

Configuring the Private Cloud

Managing clusters

- Several Hosts configured to use (for example)
 - iSCSI datastore drivers
 - Open vSwitch networks,
 - ...
- They can be grouped the same **cluster**.
- The scheduler will know that VMs using these resources can be deployed in any of the hosts of the **cluster**.

Configuring the Private Cloud

Managing datastores

- A **datastore** is any storage medium used to store disk images for VMs
- Typically, a **datastore** will be backed by SAN/NAS servers.
- previous versions of OpenNebula refer to this concept as Image Repository.
- An OpenNebula installation can have multiple **datastores** of several types to store disk images.
- OpenNebula also uses a special **datastore**, the **system datastore**, to hold images of running VMs.

Configuring the Private Cloud

Managing datastores

OpenNebula is shipped with 4 different datastore types:

- **System**, to hold images for running VMs, depending on the storage technology used these temporal images can be complete copies of the original image, qcow deltas or simple filesystem links.
- **File-system**, to store disk images in a file form. The files are stored in a directory mounted from a SAN/NAS server.
- **iSCSI/LVM**, to store disk images in a block device form. Images are presented to the hosts as iSCSI targets.
- **VMware**, a datastore specialized for the VMware hypervisor that handles the vmdk format.

Configuring the Private Cloud

Managing datastores

- The Disk images registered in a datastore are transferred to the hosts by the transfer manager (TM) drivers
- The transfer mechanism is defined for each datastore. A single host (configured) can simultaneously access multiple datastores that uses different transfer drivers
- OpenNebula includes 4 different ways to distribute datastore images to the hosts:
 - **shared** : exported in a shared filesystem to the hosts.
 - **ssh** : datastore images are copied using the ssh protocol
 - **iscsi**, hosts access datastore targets with dynamic sessions
 - **vmware**, image copies are done using vmdfk filesystem tools
 - **qcow**, a driver specialized to handle qemu-qcow format and take advantage of its snapshotting capabilities

Configuring the Private Cloud

Managing datastores

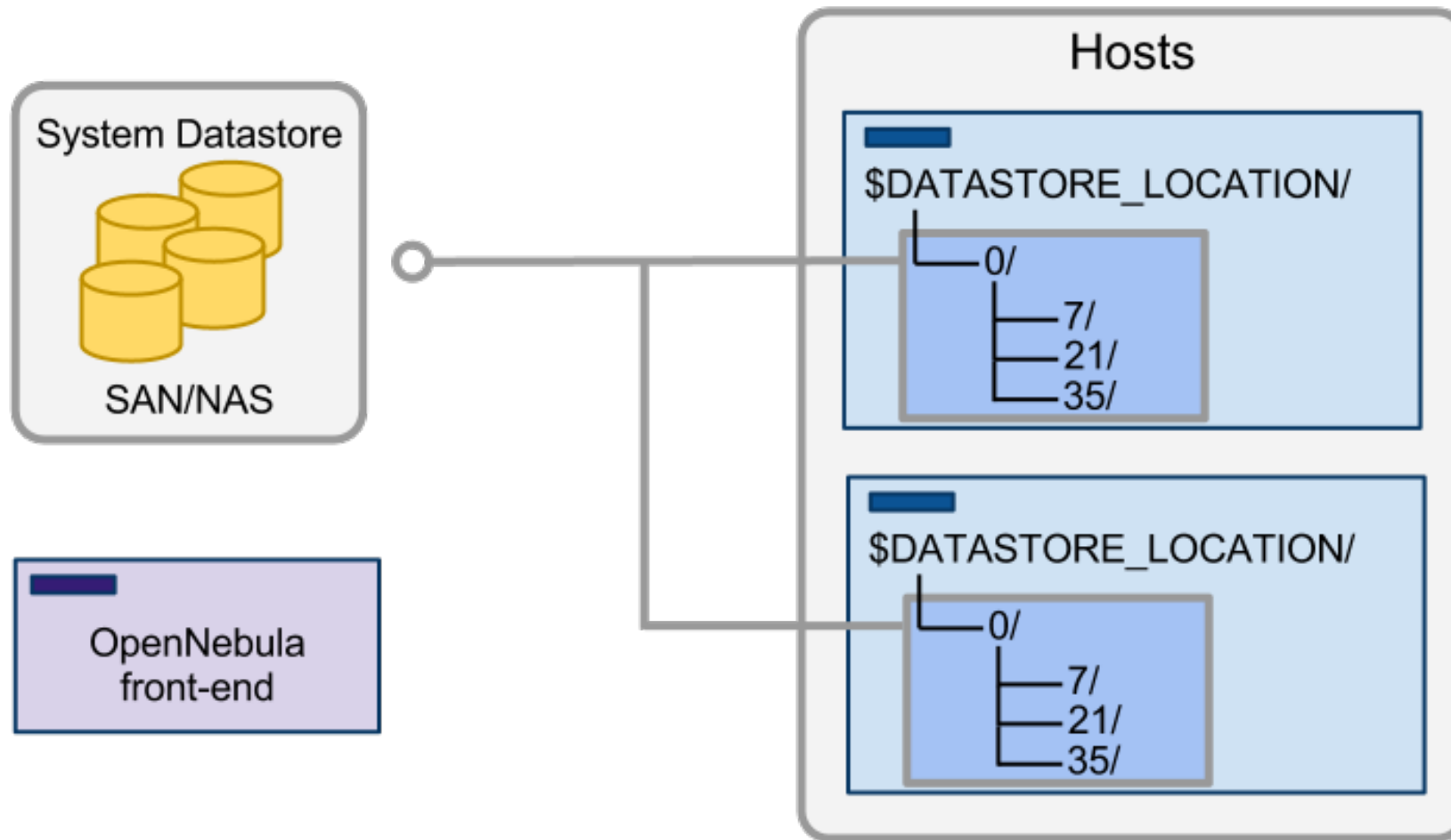
- Valid combinations of datastores and transfer drivers

| Datastore | Transfer Manager Drivers | | | | |
|-------------|--------------------------|-----|-------|------|--------|
| | shared | ssh | iscsi | qcow | vmware |
| System | ✓ | ✓ | | | |
| File-System | ✓ | ✓ | | ✓ | |
| iSCSI | | | ✓ | | |
| VMware | ✓ | ✓ | | | ✓ |

Configuring the Private Cloud

Managing datastores

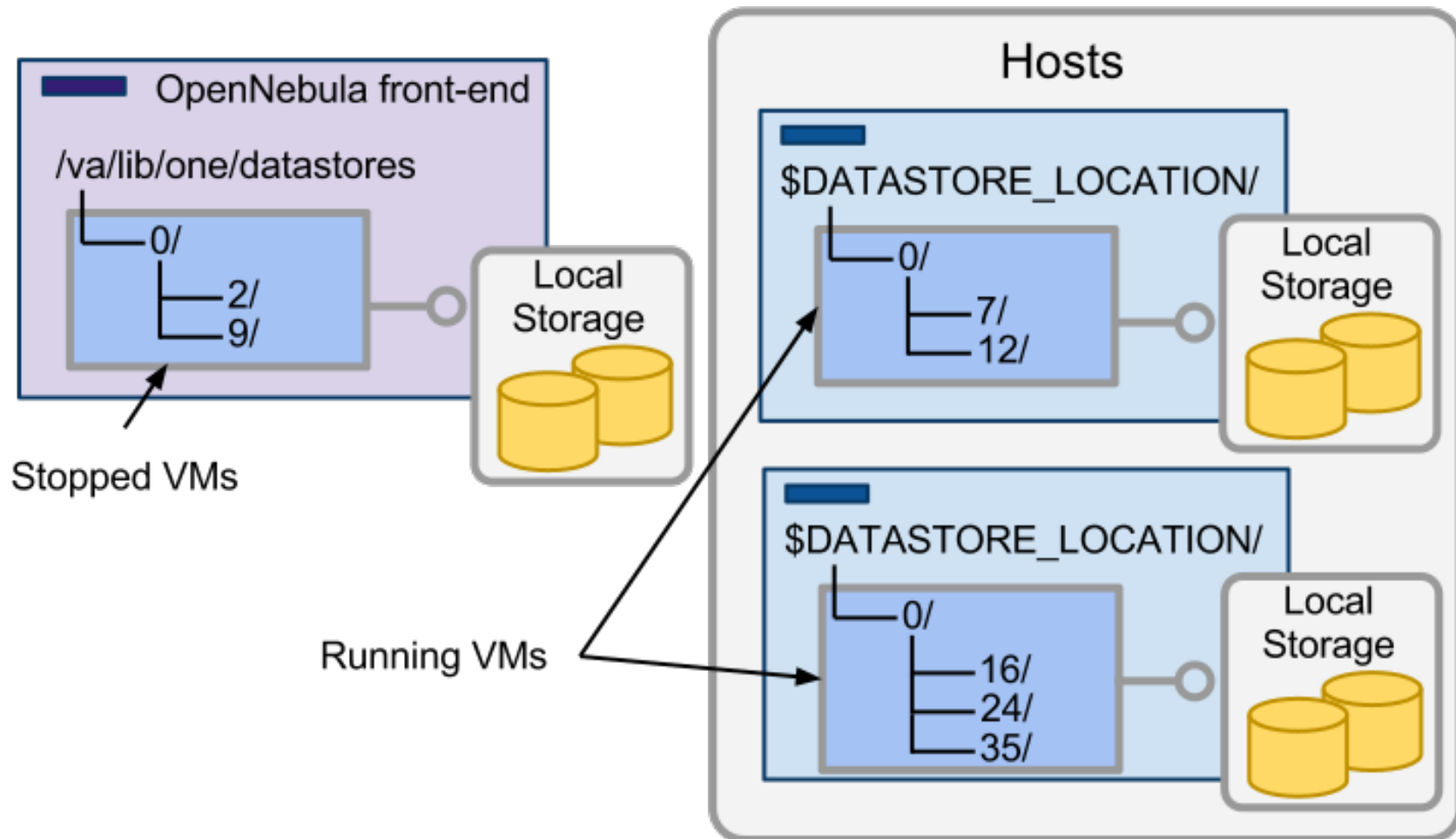
- **shared system datastore**



Configuring the Private Cloud

Managing datastores

- **ssh system datastore**



Configuring the Private Cloud

Managing Users

- **Users** are defined within OpenNebula by:
 - *ID* unique identifier for the user
 - *Name* of the user, used for authentication
 - *Password* used for authentication
- Users are managed with the **oneuser** utility
 - Create & delete users
 - List the users in the cluster

Configuring the Private Cloud

Managing Users

- List current users of your Cloud
- Create a new user

```
$ oneuser create helen mypass
```

```
User "Helen" should put helen:mypass in $ONE_AUTH
```

```
$ oneuser list
```

| UID | NAME | PASSWORD | ENABLE |
|-----|----------|--|--------|
| 0 | oneadmin | c24783ba96a35464632a624d9f829136edc0175e | True |
| 2 | helen | 34a91f713808846ade4a71577dc7963631ebae14 | True |

```
$ oneuser delete helen
```

Configuring the Private Cloud

Logging and debugging information

- The operations of the OpenNebula daemon and scheduler are logged in:
 - **oned:** `$ONE_LOCATION/var/oned.log`, Its verbosity is set by `DEBUG_LEVEL` in `$ONE_LOCATION/etc/oned.conf`.
 - **Scheduler (mm_sched):** All the scheduler information is collected into the `$ONE_LOCATION/var/sched.log` file.
- VM logs and files are in `$ONE_LOCATION/var/<VM_ID>`, more in a few slides...
- Drivers can activate `ONE_MAD_DEBUG` in the associated RC file (or in `$ONE_LOCATION/etc/defaultrc`)

Building Clouds with OpenNebula 3.4

Installing & Configuring OpenNebula

Constantino Vázquez Blanco

dsa-research.org | OpenNebula.org

Distributed Systems Architecture Research Group

Universidad Complutense de Madrid



- Installing OpenNebula 3.4
- Configuring OpenNebula 3.4
- Managing Hosts
- Managing Clusters
- Managing Datastores
- Managing Users
- Logging and Debugging Information