Session 2
Administration and Basic Usage – Part I

Constantino Vázquez
tinova@fdi.ucm.es

Installing OpenNebula 1.4

- Grab the source code and compile it!

```
~/SRC$ scp gw:one-1.4.0.tar.gz .
~/SRC$ tar xzvf one-1.4.0.tar.gz
~/SRC$ cd one-1.4/
~/SRC$ scons
```

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

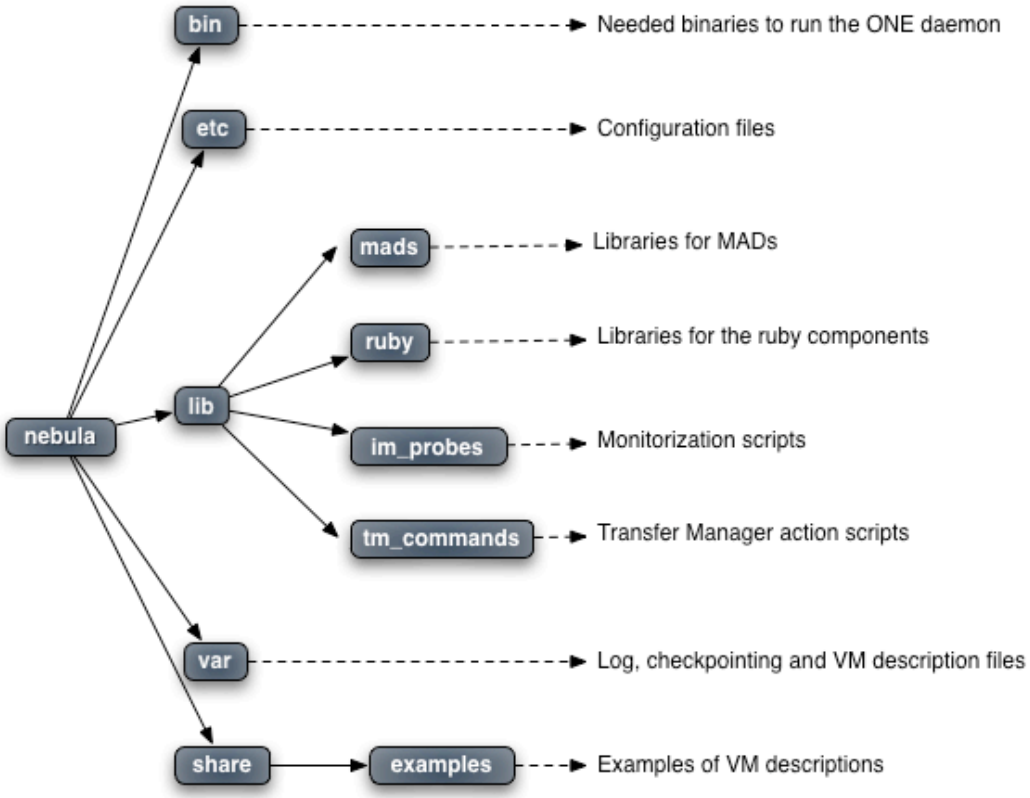
```
$ export ONE_LOCATION=/srv/cloud/one/
$ ./install.sh -d $ONE_LOCATION

Check install.sh -h for other options
```

- Check and explore the installation tree

```
~$ ls -F
bin/  etc/  examples.desktop  include/  lib/  share/  SRC/  var/
```

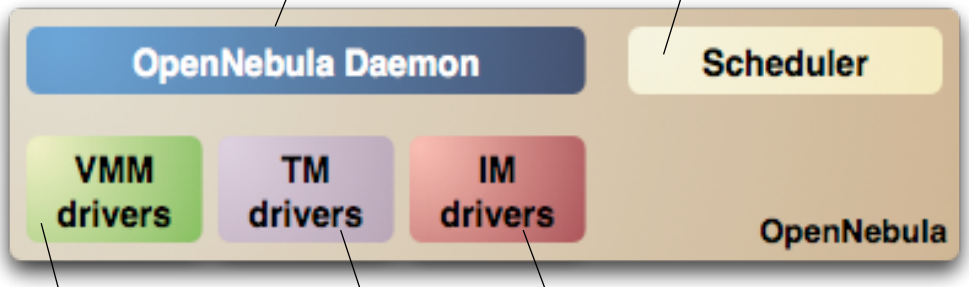
Installing OpenNebula 1.4



Configuring OpenNebula: The 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/im_*/im_*.conf`
- Defines monitoring probes

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

- `$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
  - VM_DIR: Path to the VM directory for all the cluster nodes.
  - Network parameters, MAC_PREFIX, NETWORK_SIZE
  - PORT : Port where oned will listen for xml-rpc calls
  - DEBUG_LEVEL

```
HOST_MONITORING_INTERVAL = 60
VM_POLLING_INTERVAL      = 60

#VM_DIR      = /srv/cloud/one/var

MAC_PREFIX   = "00:01"
NETWORK_SIZE = 254

PORT         =  2633
DEBUG_LEVEL  = 3
```


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: a probe configuration file

```
IM_MAD = [
      name       = "im_xen",
      executable = "one_im_ssh",
      arguments  = "im_xen/im_xen.conf" ]
```

- 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

- Virtualization Drivers, to interface the hypervisors
  - name: identifies the driver
  - executable: absolute or relative to $ONE_LOCATION/lib/mads
  - arguments: (not needed for the distribution drivers)
  - default: default values for the hypervisor
  - type: format of the VM description file to be passed to the driver: xen, kvm or xml

```
VM_MAD = [
    name       = "vmm_xen",
    executable = "one_vmm_xen",
    default    = "vmm_xen/vmm_xen.conf",
    type       = ”xen" ]
```

- Hooks, custom programs that are executed on specific events, e.g. VM creation.
- Hands on... Check and adjust the values of oned.conf for your cloud

Configuring OpenNebula: Accounts

- 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
```

  - Create the password file

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

Configuring OpenNebula: Start & Stop

- Use the `one` script

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

$one start
oned and scheduler started

$ more $ONE_LOCATION/var/oned.log
Thu Jan 14 18:03:11 2010 [ONE][I]: Init OpenNebula Log system
Thu Jan 14 18:03:11 2010 [ONE][I]: Log Level: 3 [0=ERROR,1=WARNING,
2=INFO,3=DEBUG]
Thu Jan 14 18:03:11 2010 [ONE][I]: ------------------------------------
Thu Jan 14 18:03:11 2010 [ONE][I]:      OpenNebula Configuration File
Thu Jan 14 18:03:11 2010 [ONE][I]: ------------------------------------
```

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

Configuring OpenNebula: Hosts

- Cluster nodes are defined with
  - *Hostname* of the cluster 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
- Cluster nodes are managed with the onehost utility
  - Create & delete hosts
  - List the hosts in the cluster
  - Show detailed information from a host
  - Enable/Disable a host

Configuring OpenNebula: Hosts

- Hands on... configure the hosts of your private cloud

```
$ onehost create host01 im_xen vmm_xen tm_nfs
$ onehost create host02 im_xen vmm_xen tm_nfs

$ onehost list
  ID NAME                RVM   TCPU   FCPU   ACPU    TMEM    FMEM STAT
   0 host01                0      0      0    100       0       0   on
   1 host02                0      0      0    100       0       0   on

$ tail -f $ONE_LOCATION/var/oned.log
Thu Jan 14 18:07:39 2010 [InM][I]: Monitoring host host01(0)
Thu Jan 14 18:07:39 2010 [InM][I]: Monitoring host host02 (1)
Thu Jan 14 18:07:43 2010 [InM][D]: Host 0 successfully monitored.
Thu Jan 14 18:07:44 2010 [InM][D]: Host 1 successfully monitored.

$ onehost list
  ID NAME                RV   TCPU   FCPU   ACPU    TMEM    FMEM STAT
   0 host01                0    200    184    184 2017004 1848172   on
   1 host02                0    200    200    200 2017004 1857172   on

$ onehost show 0
```

- Hands on... Explore and test the onehost command in your cloud

Configuring OpenNebula: 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
- Hands on... create new users in your private cloud and configure the "*user*" UNIX account

```
$ 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 OpenNebula: Log Files

- 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)

Using the Private Cloud: Virtual Networks

- A Virtual Network in OpenNebula
  - Defines a separated MAC/IP address space to be used by VMs
  - Each virtual network is associated with a physical network through a bridge
  - Virtual Networks can be isolated (at layer 2 level) with ebtables and hooks
- Virtual Network definition
  - **Name,** of the network
  - **Type**
    - **Fixed**, a set of IP/MAC leases
    - **Ranged,** defines a network range
  - **Bridge**, name of the physical bridge in the physical host where the VM should connect its network interface.
- Virtual Networks are managed with the `onevnet` utility


Networks created by oneadmin are *public, i.e.* can be used by VMs of any other user

Using the Private Cloud: Virtual Networks

- Hands on...

```
$ vi real.net
NAME = "One-TD"
TYPE = RANGED
BRIDGE = xenbr0
NETWORK_SIZE    = 125
NETWORK_ADDRESS = 192.168.$CN.128

$ vi fake.net
NAME = "One-TD-Invisible"
TYPE = FIXED
BRIDGE = xenbr0
LEASES = [IP=192.168.($CN+100).5]
LEASES = [IP=192.168.($CN+100).10]
LEASES = [IP=192.168.($CN+100).15]
LEASES = [IP=192.168.($CN+100).20]
LEASES = [IP=192.168.($CN+100).25]

$ onevnet -v create real.net
$ onevnet -v create fake.net
```

- Hands on... explore the use of onevnet list, show, delete

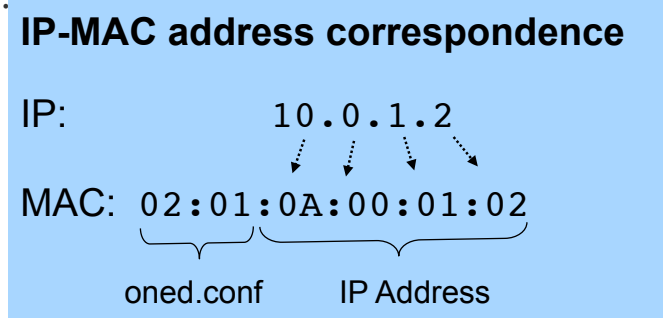
Using the Private Cloud: Virtual Networks

- Using a Virtual Network with your VMs
  - Define NICs attached to a given virtual network. The VM will get a NIC with a free MAC in the network and attached to the corresponding bridge

```
#A VM with two interfaces each one in a different vlan
NIC=[NETWORK="One-TD"]
NIC=[NETWORK="One-TD-Invisible"]

#Ask for a specific IP/MAC of the Red vlan
NIC=[NETWORK="One-TD", IP=192.168.$CN.140]
```

  - Prepare the VM to use the IP. Sample scripts to set the IP based on the MAC are provided for several Linux distributions.



Administration and Basic Usage – Part I
Creative Commons Attribution Share Alike (CC-BY-SA)