

OpenNebula/Reservoir Training, January 27-28

Brussels, Belgium

## Session 2

# Administration and Basic Usage – Part I

Daniel Molina & Javier Fontán  
dmolina/jfontan@opennebula.org

**OpenNebula.org**



Copyright 2002-2011 © OpenNebula Project Leads (OpenNebula.org). All Rights Reserved.  
Creative Commons Attribution Share Alike (CC-BY-SA)

# Installing OpenNebula 2.0

- Grab the source code from /automount/share/reservoir/opennebula/2.0.1/opennebula-2.0.1.tar.gz and compile it!

```
fe~/SRC$ tar xzvf opennebula-2.0.1.tar.gz
fe~/SRC$ cd opennebula-2.0.1
fe~/SRC$ scons
```

- If there are problem with PKG\_CONFIG\_PATH:

```
fe~/SRC$ export PKG_CONFIG_PATH=/usr/lib/pkgconfig
```

- Install the software in /srv/cloud/one (ONE\_LOCATION)

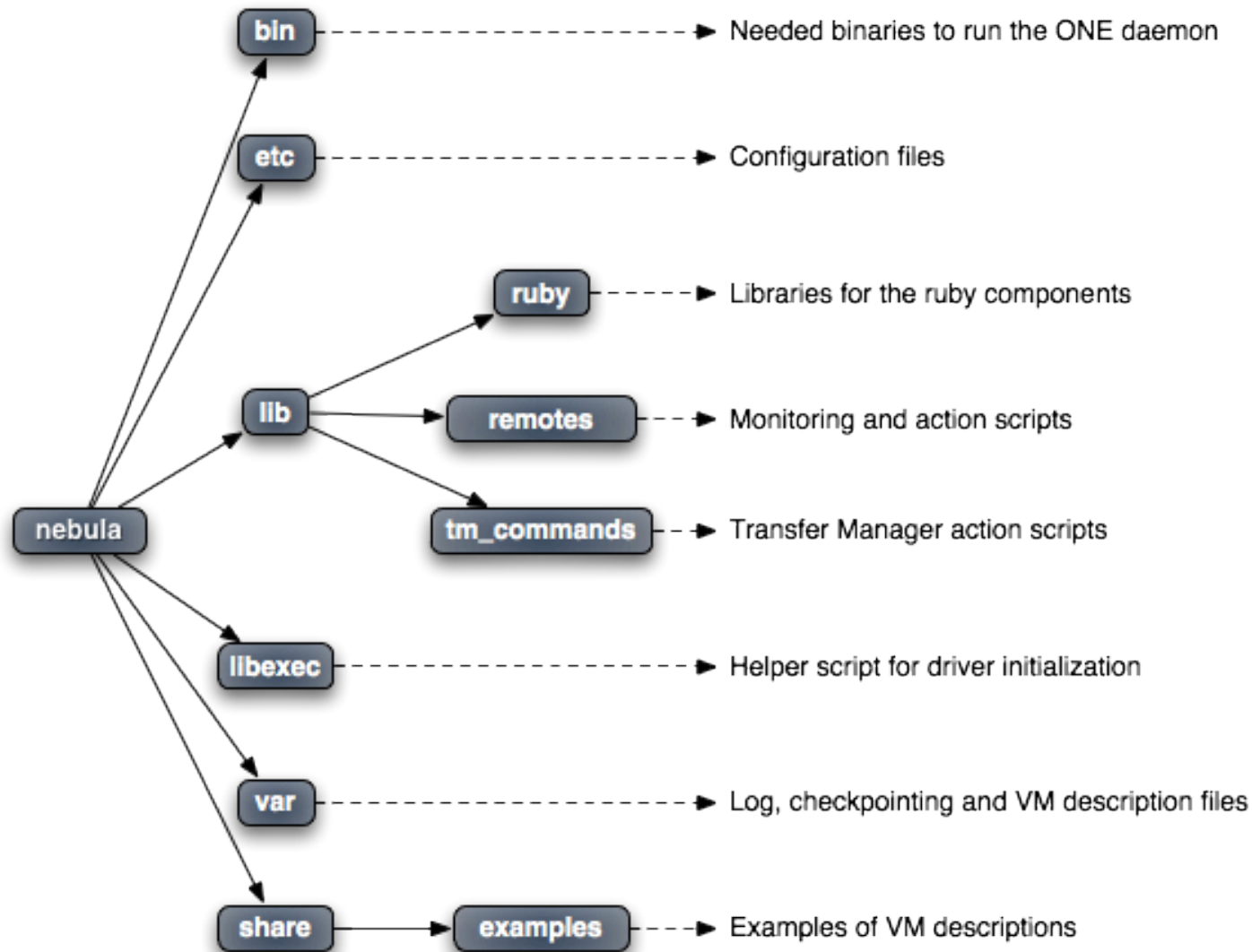
```
fe$ export ONE_LOCATION=/srv/cloud/one/
fe$ ./install.sh -d $ONE_LOCATION
```

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

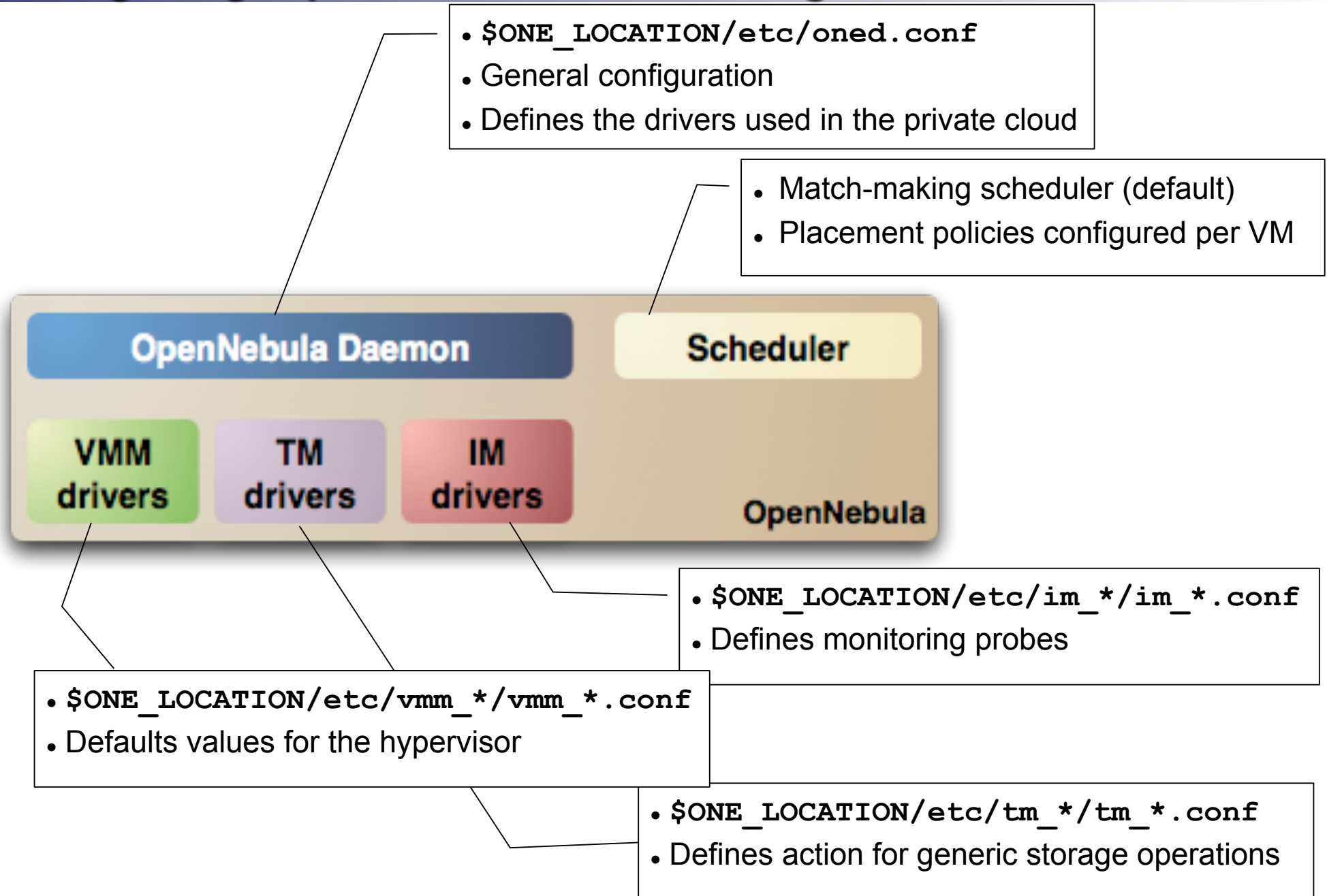
- Check and explore the installation tree

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

# Installing OpenNebula 2.0



# Configuring OpenNebula: The configuration interface



# 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.
  - SCRIPTS\_REMOTE\_DIR: Remote path to store the monitoring and VM management script.
  - PORT : Port where oned will listen for xml-rpc calls
  - DB: Configuration attributes for the database backend
  - VNC\_BASE\_PORT: VNC ports are set to VNC\_BASE\_PORT + VMID
  - DEBUG\_LEVEL

```
HOST_MONITORING_INTERVAL = 60
VM_POLLING_INTERVAL      = 60

#VM_DIR      = /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

- Physical Networks configuration
  - NETWORK\_SIZE: default size for the virtual networks
  - MAC\_PREFIX: Default prefix to be used in the auto-generated MAC addresses

```
NETWORK_SIZE = 254  
MAC_PREFIX   = "02:00"
```

- Image Repository Configuration
  - IMAGE\_REPOSITORY\_PATH: by default \$ONE\_LOCATION/var/images
  - DEFAULT\_IMAGE\_TYPE: Can be: OS, CDROM, DATABLOCK
  - DEFAULT\_DEVICE\_PREFIX: hd, sd, xvd, vd

```
#IMAGE_REPOSITORY_PATH = /srv/cloud/var/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: a probe configuration file

```
IM_MAD = [  
    name          = "im_xen",  
    executable    = "one_im_ssh",  
    arguments     = "xen" ]
```

- 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",  
  arguments  = "xen",  
  default    = "vmm_ssh/vmm_ssh_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, or public ones.
- Configuring the `oneadmin` account
  - Environment variables: `ONE_AUTH`, `ONE_LOCATION` and `ONE_XMLRPC`

```
fe$ 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

```
fe$ mkdir .one
fe$ cd .one
fe$ vi one_auth
oneadmin:onecloud
```

# Configuring OpenNebula: Start & Stop

- Use the `one` script

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

fe$ one start
oned and scheduler started

fe$ 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 & Clusters

- Hosts are 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
- By default, all hosts belong to the *default* logical cluster. Clusters are managed using the **onecluster** command
  - Create & delete clusters
  - List the available clusters
  - Add & remove hosts from the clusters
- Hosts are managed with the **onehost** utility
  - Create & delete hosts
  - List the hosts
  - Show detailed information from a host

# Configuring OpenNebula: Hosts

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

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

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

```
fe$ onehost list
```

ID	NAME	CLUSTER	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
0	host01	default	0	0	0	100	0	0	on
1	host02	default	0	0	0	100	0	0	on

```
fe$ 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.
```

```
fe$ onehost list
```

ID	NAME	CLUSTER	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
0	host01	default	0	200	199	200	3.6G	2G	on
1	host02	default	0	200	200	200	3.6G	2G	on

```
fe$ onehost show 0
```

# Configuring OpenNebula: Clusters

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

```
fe$ oneclasser list
  ID      NAME
   0     default

fe$ oneclasser create testing
fe$ oneclasser addhost host02 testing

fe$ onehost list
  ID NAME      CLUSTER  RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT
   0 host01     default   0    200   184   184   3.6G  2G    on
   1 host02     testing   0    200   200   200   3.6G  2G    on

fe$ oneclasser delete testing

fe$ onehost list
  ID NAME      CLUSTER  RVM  TCPU  FCPU  ACPU  TMEM  FMEM  STAT
   0 host01     default   0    200   184   184   3.6G  2G    on
   1 host02     default   0    200   200   200   3.6G  2G    on
```

- Hands on... Explore and test the **onehost** and **onecluster** commands 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, list and delete users
  - Change users' passwords
- Hands on... create new users in your private cloud and configure the “*user*” UNIX account

```
fe$ oneuser create helen mypass
```

```
User "Helen" should put helen:mypass in $ONE_AUTH or ~/.one/one_auth
```

```
fe$ oneuser list
```

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

```
fe$ 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
  - **Public**: whether or not this Virtual Network can be used by VMs of any other user
- Virtual Networks are managed with the `onevnet` utility



# Using the Private Cloud: Virtual Networks

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

```
fe$ vi public.net
NAME      = "Public"
TYPE      = FIXED
PUBLIC    = YES
BRIDGE    = xenbr0
LEASES    = [ IP=172.16.1.60+$CN  ]

fe$ vi onetd.net
NAME      = "One-TD"
TYPE      = RANGED
PUBLIC    = NO
BRIDGE    = xenbr0
NETWORK_SIZE = 125
NETWORK_ADDRESS = 172.16.10+$CN.0

fe$ onevnet -v create public.net
fe$ onevnet -v create onetd.net
```

# 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="Public"]
```

```
NIC=[NETWORK="One-TD"]
```

```
#Ask for a specific IP/MAC of the One-TD vlan
```

```
NIC=[NETWORK="Public", IP=172.16.1.60+$CN ]
```

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

## IP-MAC address correspondence

IP: 10.0.1.2

MAC: 02:01:0A:00:01:02

oned.conf      IP Address