

INSTRUCTIONAL MATERIAL

DEPARTMENT OF INFORMATION TECHNOLOGY

DEVELOPING A PRIVATE CLOUD (OPEN NEBULA) MULTINODE

Dr. P. DARWIN

Associate Professor

Department of Information Technology



TAGORE ENGINEERING COLLEGE

Rathinamangalam, Chennai-600 127

DEVELOPING A PRIVATE CLOUD (Open Nebula) MULTINODE

A Private Cloud has two components. One is the Frontend node and the other acts as the Cluster Node.

1. **Frontend Node:** This node acts the Cloud Manager and has Open Nebula installed on it.
2. **Cluster Node:** The cluster node acts as the Cloud Host running multiple Virtual machines over a Hypervisor. The VMs are created using Virtual Box.

Commands to setup Frontend Node

1. Verify that the Front end can ping the Host. Use wired connectivity such as Ethernet cable or LAN to configure them in same network.
2. Follow these steps to create user account (oneadmin) for managing open nebula.
 - `sudo mkdir -p /srv/cloud`
 - `sudo groupadd -g 1001 cloud`
 - `sudo useradd -u 1001 -g cloud -m oneadmin -d /srv/cloud/one -s /bin/bash`
 - `sudo passwd oneadmin`
 - `sudo chown -R oneadmin:cloud /srv/cloud`

```
su -l oneadmin
```

Note: From admin account you can fire these commands to add one admin in sudo group list so that it can install software without permission problems

```
sudo adduser oneadmin sudo
```

3. Now you have created a user account for one admin. Please note that your main directory of working will be always /srv/cloud/one. Now we will share this directory with NFS clients which are hosts. This is to ensure that they can access the services and files from server which is front end. The steps are as follows:

- `sudo apt-get install nfs-kernel-server` (install the nfs server)
- `sudo gedit /etc/exports` (append to this file the below given information)
 - ❖ `/export 192.168.1.0/24(rw,fsid=0,insecure,no_subtree_check,async)`
 - ❖ Note: 192.168.1.0/24 will be replaced by your network address of LAN or network.
 - ❖ Please refer this guide if facing any issues regarding formats which changes frequently. <https://help.ubuntu.com/community/SettingUpNFSTo>
- `sudo /etc/init.d/nfs-kernel-server restart` (restart the nfs server services)
- Install Open ssh for secure communication between Cloud front end and Host.
- Install open ssh server
 - ❖ `sudo apt-get install openssh-server`
 - ❖ `sudo cp /etc/ssh/sshd_config ~ //`(Make a copy of sshd config file as we are going to edit the original file so in case of problems with your changes you can always revert back)
 - ❖ `sudo gedit /etc/ssh/sshd_config` (edit the configuration file now)
 - ✓ Change the "PermitRootLogin" directive to "no"
 - ✓ Change the "StrictHostKeyChecking" directive "no" //(if you don't find it by this name you will find a similar name in the file below PermitRootLogin.)
 - ❖ `sudo /etc/init.d/ssh restart` (restart the sshd server with the changed configuration) ** `oneadmin@ubuntu:~$ssh-keygen -t rsa` (Create RSA keys and put at remote host so that SSH does not ask for password everytime while doing ssh)
 - ❖ `oneadmin@ubuntu:~$ scp ~/.ssh/id_rsa.pub oneadmin@192.168.111.211:~/.ssh/id_rsa_localbox.pub` (Please note this is a single command and is to place the key at remote host via scp)
 - ❖ Now to complete this process we need to login to remote host:
 1. `user@remotebox: mkdir .ssh` (create a directory .ssh)
 2. `user@remotebox:~cd .ssh`

3. `user@remotebox:~/ssh $ cat id_rsa_localbox.pub >> authorized_keys2`
4. `user@remotebox:~/ssh $ cat id_rsa_localbox.pub >> authorized_keys`
5. `user@remotebox:~/ssh$ rm id_rsa_localbox.pub`

Check after these commands oneadmin will be able to ssh the remote host without password.

Try this

1. `oneadmin@ubuntu: ssh oneadmin@192.168.111.229` a. where oneadmin is name of user account at host and 192.168.111.229 is IP address of the remote host.
2. You will get logged in to remote host, if the above steps went successfully.
3. Now install the dependencies needed for running open nebula
 - `sudo apt-get install libsqlite3-dev libxmlrpc-c3-dev g++ ruby libopenssl-ruby libssl-dev ruby-dev`
 - `sudo apt-get install libxml2-dev libmysqlclient-dev libmysql++-dev libsqlite3-ruby libexpat1-dev`
 - `sudo apt-get install rake rubygems libxml-parser-ruby1.8 libxslt-dev genisoimage scons`
 - Install MySQL for database for opennebula
 - ❖ `sudo apt-get install mysql-server`
 - ❖ `mysql -u root -p`
 - `CREATE USER 'oneadmin'@'localhost' IDENTIFIED BY 'oneadmin';`
 - `CREATE DATABASE opennebula;`
 - `GRANT ALL PRIVILEGES ON opennebula.* TO 'oneadmin' IDENTIFIED BY 'oneadmin';`
 - `Quit;`
 - Note: Please write these commands on terminal sometimes special character copying may cause syntax errors.
4. Now finally install open nebula software
 - `Su -l oneadmin`
 - Download latest Open nebula tarball from <http://downloads.opennebula.org/>
 - `Sudo tar xzf opennebula-3.0.0.tar.gz`
 - `cd opennebula-3.0.0`

- `scons sqlite=no mysql=yes` (build opennebula with my sql support)
- `./install.sh -u oneadmin -g cloud -d /srv/cloud/one`
- Set up the environment variable. Add them to `~/.bash-profile`
- `sudo gedit ~/.bash_profile` (append the text to this file)
 - ❖ `export ONE_LOCATION = /srv/cloud/one`
 - ❖ `export ONE_AUTH=$ONE_LOCATION/.one/one_auth`
 - ❖ `export ONE_XMLRPC=http://localhost:2633/RPC2`

3. `export PATH=$ONE_LOCATION/bin:/usr/local/bin:/var/lib/gems/1.8/bin: $PATH`

(Note: it is a single command)

Source `~/.bash_profile` to initialize those variable

- `mkdir ~/.one`
- `echo "oneadmin:passwd" > ~/.one/one_auth`
 - Configure `~/etc/oned.conf` .Uncomment the following lines
- `Sudo gedit ~/etc/oned.conf`
- `DB = [backend = "mysql",`
- `Server = "local host",`
- `Port =0,`
- `User = "oneadmin",`
- `passwd = "oneadmin",`
- `db_name = "opennebula"]`
 - Run opennebula
- `One start`

If it starts successfully, you are done installing opennebula on front end.

- Note: If you find the error opennebula is not installed in your system try installing from apt-get repository using the command