

OpenStack Icehouse installation on virtual box's virtual machines using vasabilab's OpenStack installation scripts

Kasidit Chanchio

Vasabi Laboratory

Department of Computer Science,

Faculty of Science and Technology,

Thammasat University

<http://vasabilab.cs.tu.ac.th>

Version 0.2

Aug 29, 2014

1. setting up virtual box hosts and networks

First, you have to create four virtual machines using virtual box and create virtual box (vbox)'s network on them as described in Figure 1.

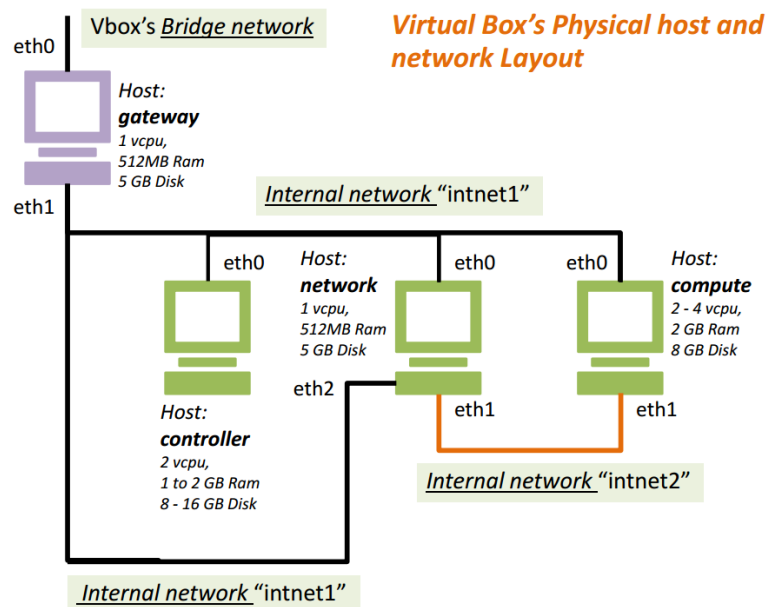


Figure 1. Virtual box's physical host and network layout for OpenStack installation

These machines are:

1. Host name: **gateway**, CPU: 1 vcpu, Ram: 512MB to 1 GB, Disk: 5 GB

Network:

- 1st NIC: eth0 is set to use "bridge" network connection. The NIC promiscuous mode is set to "Allow All". For the sake of simplicity, set promiscuous mode of NICs on every machine to "Allow All" if possible.
- 2nd NIC: The eth1 is set to use vbox's "Internal network" with "intnet1" tag value.

This gateway machine simulates the gateway computer (or a router) of a network that your OpenStack installation will reside. It will also run pptp VPN server. Note that this gateway machine is **not** a part of OpenStack installation. We merely use it to simulate a network in an organization before OpenStack installation.

2. Host name: **controller**, CPU: 2 vcpus, Ram: 1 to 2 GB, Disk: 8 – 16 GB

Network: the eth0 is configured to connect to the internal "intnet1" network.

This machine is the OpenStack controller node.

- Host name: **network**, CPU: 1 vcpu, Ram: 512MB, Disk: 5 GB

Network:

- 1st NIC: eth0: connect to the internal "intnet1" network
- 2nd NIC: eth1: connect to a new internal network, namely "intnet2"
- 3rd NIC: eth2: connect to the internal "intnet1" network

From Figure 1, notice that the "intnet1" network is the only network that connect to the internet via the **gateway** machine. This machine will run as the network node of OpenStack.

- Host name: **compute**, CPU: 2 – 4 vcpus, Ram: 2 – 4 GB, Disk: 10 GB – 32 GB

Network:

- 1st NIC: eth0: connect to the internal "intnet1" network
- 2nd NIC: eth1: connect to a new internal network, namely "intnet2"

Before continue, **MAKE SURE THAT THE DATEs and TIMEs of every machine are correct.**

2. Recommendations for reliable and efficient installation

Openstack installation involved multiple computers and extensive data transfer across network. Network glitches and shortages can cause installation corruptions. Some of them is undetectable during installation time and may cause failures when OpenStack operates. Also, high communication latency or low network bandwidth can cause the installation to take a long time. Therefore, we highly recommend the followings before embarking on your installation.

1. Create a local ubuntu mirror repository in your LAN. This can be done using the apt-mirror utility. The repository will contain ubuntu version that you will use for OpenStack. From our experiences, we found that the local repository can significantly reduce failures from network shortages and reduce installation time.
2. Unless your wireless signals are very strong, use wired connection.

3. Installing Ubuntu on vbox hosts and set initial network configuration

Figure 2 shows initial OS installation and setups before using vasabi's installer script to install OpenStack.

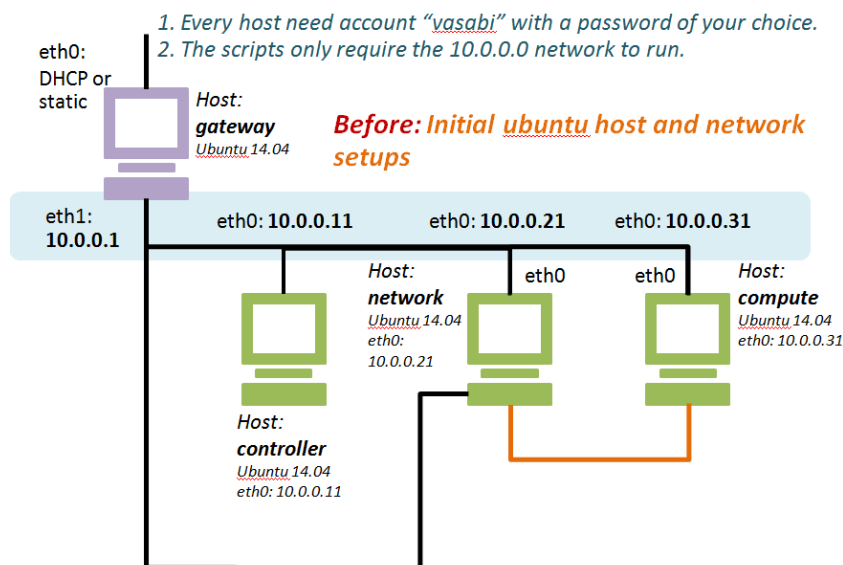


Figure 2 initial OS installation and setups

From the figure, we install ubuntu 14.04 and create a “vasabi” account on every vbox host. The account name “vasabi” is required by the installer but the password can be anything you like. The installer also requires the following **initial network configuration**.

1. On the gateway, set eth0 to connect to the internet and eth1 to connect to the internal network “intnet1” which in turn connect to other hosts.
2. Base on the OpenStack Icehouse installation document at [1], we set the network address for the “intnet1” network to 10.0.0.0/24. You can change this network address to a different value. However, you will have to define the new value as an installation parameter before starting OpenStack installation in the next section.

To set up IP addresses on each host, you have to modify the /etc/network/interfaces file on the gateway, controller, network, and compute machines as follows:

On gateway:

```
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 10.0.0.1
netmask 255.255.255.0
network 10.0.0.0
broadcast 10.0.0.255
dns-nameservers 8.8.8.8
```

save the file and restart network with

```
$ sudo ifdown eth1
$ sudo ifup eth1
```

On controller:

```
auto eth0
iface eth0 inet static
address 10.0.0.11
netmask 255.255.255.0
network 10.0.0.0
gateway 10.0.0.1
```

save the file and restart network with

```
$ sudo ifdown eth0
$ sudo ifup eth0
```

On network:

```
auto eth0
iface eth0 inet static
address 10.0.0.21
netmask 255.255.255.0
network 10.0.0.0
gateway 10.0.0.1
```

save the file and restart network with

```
$ sudo ifdown eth0
$ sudo ifup eth0
```

On compute:

```
auto eth0
iface eth0 inet static
address 10.0.0.31
netmask 255.255.255.0
network 10.0.0.0
gateway 10.0.0.1
```

save the file and restart network with

```
$ sudo ifdown eth0
$ sudo ifup eth0
```

Use ping to check connectivity among these hosts.

4. Obtain OpenStackInstaller scripts

You can download the OpenStackInstaller scripts from <https://github.com/kasidit/OpenStackInstaller> and save them on the **gateway** host using the following commands.

```
$ cd $HOME
$ wget https://github.com/kasidit/OpenStackInstaller/archive/master.zip
$ unzip master.zip
$ cd OpenStackInstaller-master
```

Alternatively, you can get a tar file directly from vasabilab web site and extract it on the gateway using the following commands.

```
$ cd $HOME
$ wget http://vasabilab.cs.tu.ac.th/projects/OpenStackInstaller.tar
$ tar xvf OpenStackInstaller.tar
$ cd OpenStackInstaller
```

3. Define installation parameters

Before proceeding with OpenStack installation, you need to create an install-paramrc.sh file to define a number of installation parameters, especially the networking configuration of your OpenStack deployment.

Based on an example implementation in the OpenStack Icehouse installation webpage [1], there are three kinds of network that users must define. They are the management network, the data tunnel network, and the external network as depicted in Figure 4 below.

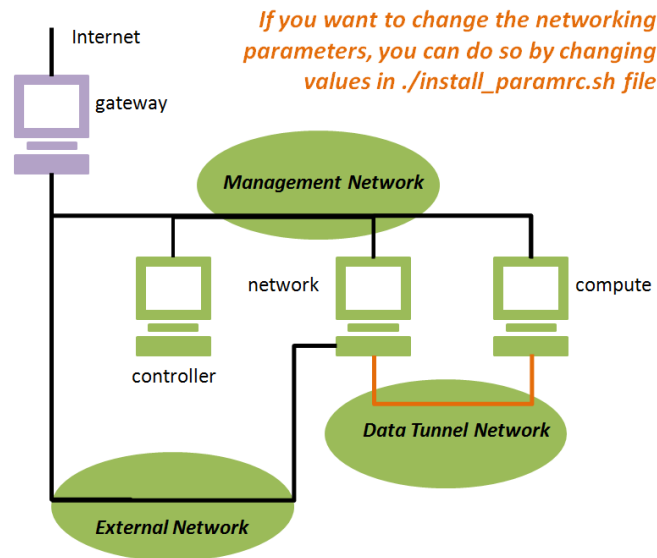


Figure 3. Three networks in an OpenStack implementation

The management network connects the controller, network, and compute host of OpenStack, for which we assign a subnet 10.0.0.0/24 to it as mentioned earlier. The data tunnel network handles data communication to and from virtual machine instances in OpenStack, while the external network forward data between the data tunnel network and the network of the outside world. According to the example in [1], the parameter values of all three networks are described Figure 4.

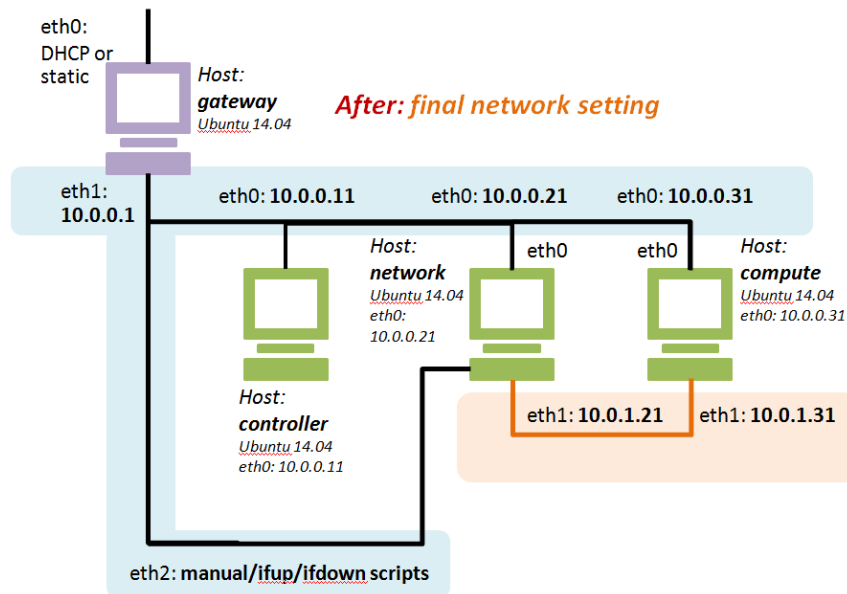


Figure 4. Example parameter values of the three OpenStack network.

3.1 Creating the install-paramrc.sh file

We have developed a web page (using javascript) to help you customize installation parameters and create the install-paramrc.html file for your installation. It is included in the installer package under the “helpers” directory. You can also access it at the URL <http://vasabilab.cs.tu.ac.th/projects/openstack-install-env.html> where you will see the web UI as shown in Figure 5.

From the figure, the webpage is a form that let user specify configuration parameters consisting of:

- A simulated intranet of an organization that has the gateway machine as a NAT server.** You must specify the domain name of the host in the intranet. You also have to define the CIDR network address of it, which is set to be the same as the management network for vasabilab’s installer.
 The installer will also use the gateway host as a *pptp VPN server* so that users can access OpenStack dashboard via vpn. Thus, you have to specify the range of intranet IP addresses the VPN server can give to its clients.
 Finally, you have to specify the IP address or URL of the ubuntu repository and security repository you plan to use for the installation. The default values are set to *jp.archive*, *ubuntu.com* and *security.ubuntu.com*, respectively.
- The OpenStack management network.** Following the example in [1], the default configuration parameter values of this network are shown in the green box of every host box in Figure 5. You can change these values as you wish; however, you must redefine the initial network configuration described in Section 3 above accordingly.
- The Data Tunnel Network.** The default parameter values of the data tunnel network that connect the network host and compute host are defined in the golden boxes.
- The external network.** The parameter values defined in the light blue box of the network host indicate the address of a network that connects the data tunnel network to the outside world. Following [1], the external network is set to use the management network by default.

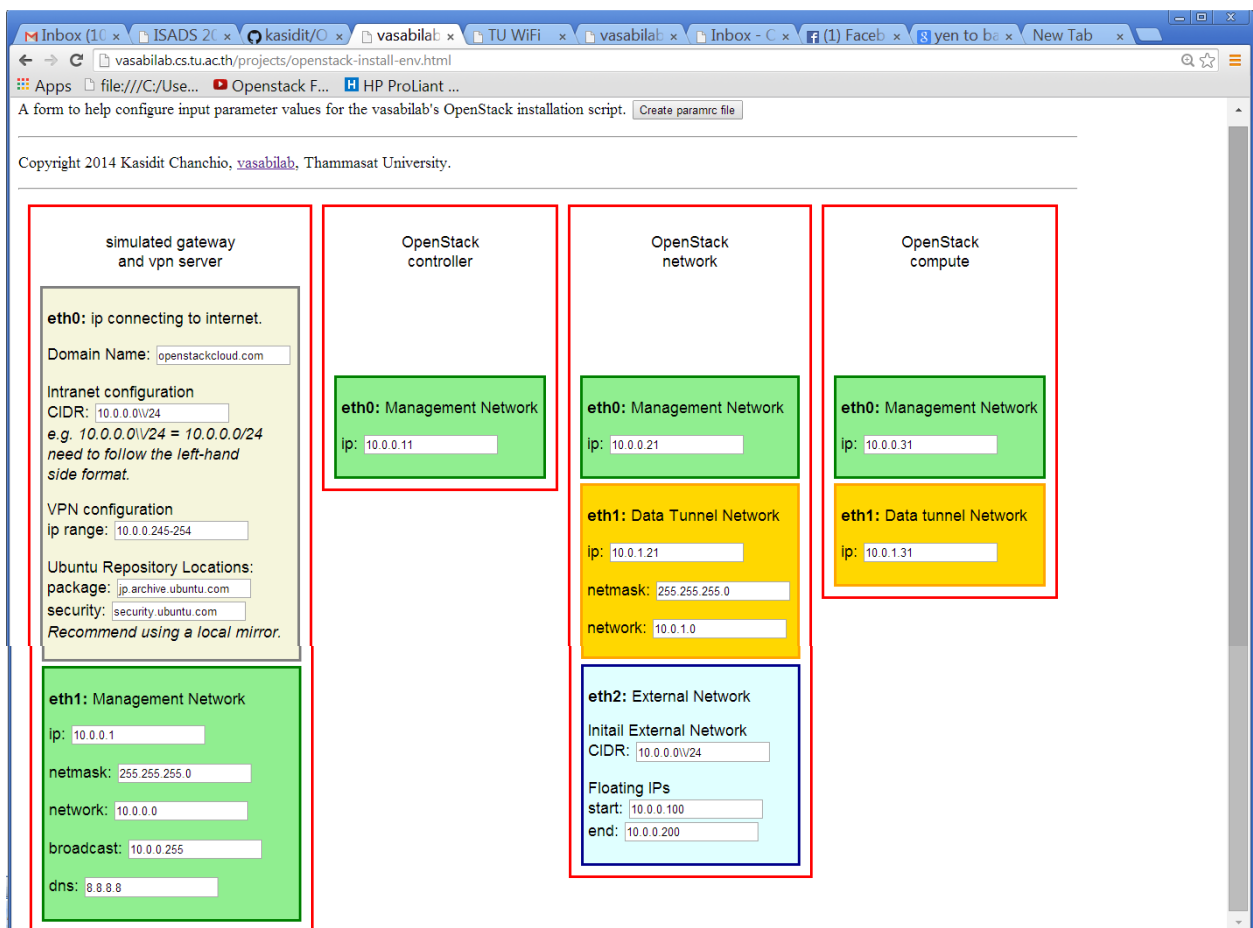


Figure 5. The installation parameter generator webpage for vasabilab’s OpenStack installer scripts

Supposed we change the ubuntu repository to use the local repository to , say 10.100.20.202, the textbox “package” and “security” on the “simulated gateway...” must both be changed to the new

value. Also, since I am going to install OpenStack at my university, I will change the Domain Name box to “tu.ac.th”.

To generate the install-paramrc.sh file, you must click the “Create paramrc file” button. The “Download” link will appear underneath the button. By clicking the link, the “install-paramrc.sh” file is generated and downloaded to your local computer. The contents of the file are illustrated in Figure 6.

```
# This shell script file contains parameter definitions
# for the vasabilab's OpenStack Installer. It is generated by the compenv.js
# program in the OpenStackInstaller/helpers directory of OpenStackInstaller.tar.
#
# Change the values of mysql root password, admin user password
# and demo user passwords below as you like.
#
export OPS_MYSQL_PASS=yourpassword
export DEMO_PASS=yourDemoUserPassword
export ADMIN_PASS=yourAdminUserPassword
export VPN_USERNAME=yourVpnUsername
export VPN_PASSWORD=yourVpnPassword
export HYPERVISOR=qemu
export INIT_IMAGE_LOCATION=http://cdn.download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86_64-disk.img
#export INIT_IMAGE_LOCATION=http://localrepo/images/cirros-0.3.2-x86_64-disk.img
export INIT_IMAGE_NAME=cirros-0.3.2-x86_64-disk
#
# ---- gateway host ----
#
export DOMAINNAME=openstackcloud.com
export NAT_INTRANET_CIDR=10.0.0.0/24
export VPN_IP_RANGE=10.0.0.245-254
export LOCAL_REPO=jp.archive.ubuntu.com
export LOCAL_SECURITY_REPO=security.ubuntu.com
export GATEWAY_IP=10.0.0.1
#
# ---- controller host ----
#
export CONTROLLER_IP=10.0.0.11
export MANAGEMENT_NETWORK_NETMASK=255.255.255.0
export MANAGEMENT_NETWORK=10.0.0.0
export MANAGEMENT_BROADCAST_ADDRESS=10.0.0.255
export DNS_IP=8.8.8.8
#
# ---- network host ----
#
export NETWORK_IP=10.0.0.21
export DATA_TUNNEL_NETWORK_NODE_IP=10.0.1.21
export DATA_TUNNEL_NETWORK_NETMASK=255.255.255.0
export DATA_TUNNEL_NETWORK_ADDRESS=10.0.1.0
export EXTERNAL_CIDR=10.0.0.0/24
export START_FLOATING_IP=10.0.0.100
export END_FLOATING_IP=10.0.0.200
#
# ---- compute host ----
#
export COMPUTE_IP=10.0.0.31
export DATA_TUNNEL_COMPUTE_NODE_IP=10.0.1.31
```

Figure 6. The contents of the generated “install-paramrc.sh” file

3.2 Defining parameter values in the install-paramrc.sh file

In Figure 6, there are a few parameter values that must be defined before running the OpenStack installer. These parameter and their default values are:

```
export OPS_MYSQL_PASS=yourpassword
export DEMO_PASS=yourDemoUserPassword
export ADMIN_PASS=yourAdminUserPassword
export VPN_USERNAME=yourVpnUsername
export VPN_PASSWORD=yourVpnPassword
export HYPERVISOR=qemu
export INIT_IMAGE_LOCATION=http://cdn.download.cirros-
cloud.net/0.3.2/cirros-0.3.2-x86_64-disk.img
#export INIT_IMAGE_LOCATION=http://localrepo/images/cirros-0.3.2-
x86_64-disk.img
export INIT_IMAGE_NAME=cirros-0.3.2-x86_64-disk
```

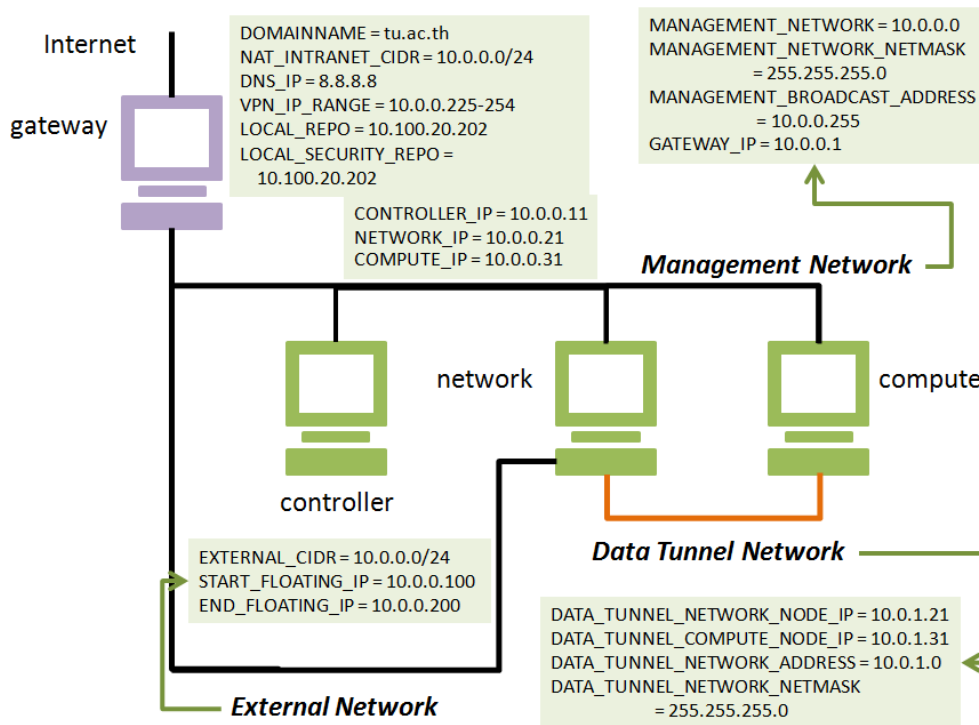


Figure 7. How the input parameter values defined in the “install-paramrc.sh” file before running the OpenStack installer represent the outcome network configurations of OpenStack deployment.

For example, you may redefine their values below:

```
export OPS_MYSQL_PASS=mysqlopenstack
export DEMO_PASS=demopassword
export ADMIN_PASS=adminpassword
export VPN_USERNAME=vasabi
export VPN_PASSWORD=vasabilab
export HYPERVISOR=qemu
export INIT_IMAGE_LOCATION=http://\\//localrepo//images//cirros-0.3.2-x86_64-disk.img
export INIT_IMAGE_NAME=cirros-0.3.2-x86_64-disk
```

You can change the HYPERVISOR value to “kvm” if your compute host supports hardware-assisted virtualization. The INIT_IMAGE_LOCATION is set to a local machine that stores the image of the Cirros OS. Note that the “\\” is required in front of every “/” by our OpenStack installer.

Figure 7 shows how each parameter value is related to the intended configuration of the OpenStack networks illustrated in Figure 3 and 4.

In case you don’t want to use the web UI to create the “install-paramrc.sh” file, you can create the install-paramrc.sh file by copying the install-paramrc.sh.example file in the extracted OpenStackInstaller directory and change the parameter values as you like.

```
$ cp install-paramrc.sh.example install-paramrc.sh
```

The install-paramrc.sh file must be placed in the OpenStackInstaller directory.

Now, you are ready to install OpenStack.

4. Install OpenStack all from one place

There are two methods to use the installer scripts. We describe the first one in this section.

You have to login to the “vasabi” account of the gateway host and do the following:

```
$ cd $HOME
$ cd OperStackInstaller-master
(or cd OperStackInstaller )
$ ./config-installer.sh
```

The `./config-installer.sh` will assign random number passwords for various OpenStack components and insert them to appropriate OpenStack configuration files. The script is designed to print out operation details on screen. It will also pause and ask you to press the enter key throughout the installation. After the script finishes, you have to enter the following commands. Note that while running these commands, the script will ask you to enter several inputs. You should see our installation demo youtube video at <http://www.youtube.com/...> along with your installation.

```
$ cd OPSInstaller/gateway
$ ./run-verbose.sh 0
$ ./run-verbose.sh 1
$ ./run-verbose.sh 2
```

The above script will generate a public-private key pair for you. You must store the key pair at its default location and choose the passphrase value as you like. You will also be asked to enter the password several times. Once the remote access set up complete, the installer will not asked for password again. Next, you will use the following commands to distribute the OpenStack installer scripts to the controller, network and compute hosts and, finally, launch the `run-install-all-OPENSTACK.sh` script to orchestrate OpenStack installation on them.

```
$ ssh-agent /bin/bash
$ ssh-add
$ ./run-verbose.sh 3
$ ./run-install-all-OPENSTACK.sh
```

While running the last script, the installer will ask you to provide the mysql root password and answer a number of inquiries regarding the removal of test databases and test accounts. You can supply the value of `OPS_MYSQL_PASS` variable defined previously as the mysql root password, and then enter the following responses to the inquiries:

1. Enter mysql root password
2. Enter “n” to not changing the mysql password
3. Enter “Y” when asked if you want to remove anonymous users
4. Enter “n” when asked if you want to “Disallow” root login remotely
5. Enter “Y” when asked if you want to remove test database
6. Enter “Y” when asked if you want to reload privilege tables

For the rest of the installation, you will be asking to press a lot of enter key. We have designed the installer this way to help spot any error report. After completing the installation, you can continue with how to use OpenStack in Section 6.

5. Install OpenStack step-by-step on every host

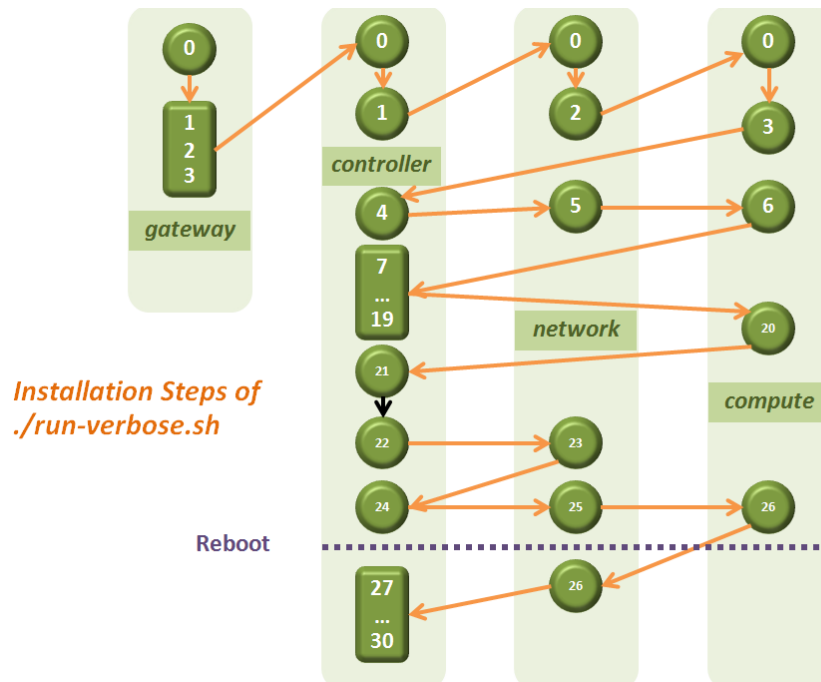


Figure 8. Step-by-step OpenStack installation using the `./run-verbose.sh` script

Alternatively, you can run the installation script step by step using the `./run-verbose.sh` script. Figure 8 shows a flow of steps that you have to enter command

```
$ ./run-verbose.sh N
```

across the four hosts, where N is an integer parameter value given to the script at each step. This N is corresponding to the integer number ranging from 0 to 30 in the figure. Again, you should look at our installation video at

<http://www.youtube.com/...>

while following instructions below.

On gateway host, login to “vasabi” account and enter the followings.

```
$ cd $HOME
$ cd OperStackInstaller-master
(or cd OperStackInstaller )
$ ./config-installer.sh
$ cd OPSInstaller/gateway
$ ./run-verbose.sh 0
$ ./run-verbose.sh 1
$ ./run-verbose.sh 2
$ ssh-agent /bin/bash
$ ssh-add
$ ./run-verbose.sh 3
```

Then, login to the “vasabi” account on the controller host and enter:

```
$ cd OPSInstaller/controller
$ ./run-verbose.sh 0
$ ./run-verbose.sh 1
```

Next, login to the “vasabi” account on the network host and run:

```
$ cd OPSInstaller/network
$ ./run-verbose.sh 0
$ ./run-verbose.sh 2
```

Login to the “vasabi” account on the compute host and run:

```
$ cd OPSInstaller/compute
$ ./run-verbose.sh 0
$ ./run-verbose.sh 3
```

Next, on the controller host, run:

```
$ cd OPSInstaller/controller
$ ./run-verbose.sh 4
```

On the network host, run:

```
$ cd OPSInstaller/network
$ ./run-verbose.sh 5
```

On the compute host, run:

```
$ cd OPSInstaller/compute
$ ./run-verbose.sh 6
```

Back to the controller host and run:

```
$ cd OPSInstaller/controller
$ ./run-verbose.sh 7
$ ./run-verbose.sh 8
$ ./run-verbose.sh 9
$ ./run-verbose.sh 10
$ ./run-verbose.sh 11
...
$ ./run-verbose.sh 19
```

On the compute host, run:

```
$ cd OPSInstaller/compute
$ ./run-verbose.sh 20
```

Back to the controller host and run:

```
$ cd OPSInstaller/controller
$ ./run-verbose.sh 21
$ ./run-verbose.sh 22
```

On the network host, run:

```
$ cd OPSInstaller/network
$ ./run-verbose.sh 23
```

Back to the controller host and run:

```
$ cd OPSInstaller/controller
$ ./run-verbose.sh 24
```

On the network host, run:

```
$ cd OPSInstaller/network
$ ./run-verbose.sh 25
```

On the compute host, run:

```
$ cd OPSInstaller/compute
$ ./run-verbose.sh 26
```

Next, you have to reboot the controller and wait until it comes back to operation. You will have to do the same for the network and then the compute hosts.

Login to the “vasabi” account on the network host and run:

```
$ cd OPSInstaller/network
$ ./run-verbose.sh 26
```

Login to the controller host and run:

```
$ cd OPSInstaller/controller
$ ./run-verbose.sh 27
$ ./run-verbose.sh 28
$ ./run-verbose.sh 29
$ ./run-verbose.sh 30
```

6. Using OpenStack

After the installation, you should be able to use VPN client connect to the VPN server (the gateway host) from the graphical display console of any computer that can communicate with the IP address on the eth0 of the gateway host. In doing so, first, you have to find out that IP address by login to the gateway and use the following command.

```
$ ifconfig eth0
```

You can use the IP address reported from the above command as the VPN server. You can also display the VPN server configuration using

```
$ cat /etc/pptp.conf
$ cat /etc/ppp/chap-secrets
```

where you shall notice that the default VPN username and password are set to “vasabi” and “vasabilab” (which have been defined previously in Section 3.2), respectively.

On the client computer that you want to use to access OpenStack dashboard, you have to configure the VPN client software to connect to the VPN server by entering the VPN server IP address and supplying the software with the VPN username and password.

After the VPN connection is established, you can open a browser on your client computer and access OpenStack dashboard at URL <http://10.0.0.11/horizon/>.

7. Conclusion

We have created a simple OpenStack installer to run on a number of virtual box’s virtual machines. OpenStack is the software that consists of many Open Source software components. Its installation

involves installing several pieces of software on multiple machines. Therefore, it is important to create an environment that minimizes network error and system error during the installation. Thus, we highly recommend setting up a local repository for mirroring ubuntu packages and storing virtual machine images before the installation. You should also see the installation video for more details. We also plan to extend the installer to include other missing component in the near future.

References

[1]. <http://docs.openstack.org/icehouse/install-guide/install/apt/content/>