

# Deploying OpenStack DevStack on VMware Fusion



RichB, 2016-29-12

For developing and testing the F5 OpenStack agent and driver, I created a [DevStack](#) virtual machine on my MacBook Pro using VMware Fusion Pro for OSX version 7.1.3. DevStack is a collection of scripts that allows users to rapidly deploy an OpenStack environment. I run this DevStack virtual machine along with a BIG-IP Virtual Edition (VE) as a self-contained environment for quick development and test. If you would like to install this along with a BIG-IP Virtual Edition, I recommend reading [Deploying F5 BIG-IP Virtual Edition on VMware Fusion](#) by Chase Abbott. I will build upon his article and use the same networks so that the reader can integrate the BIG-IP into an OpenStack environment. In order to install both a DevStack VM and F5 BIG-IP Virtual Edition on the same host, I would recommend using a host with 16GB of RAM.

## DevStack Install

1. [Create VMware networks](#)
2. [Install Ubuntu 14.04](#)
3. [Install DevStack](#)

## Create Custom API and Provider Networks

If you have followed the [instructions](#) on how to install and configure additional networking, you can use **vmnet2** and **vmnet4** as OpenStack public and API networks. If you do not wish to install VE and only need to set up VMware networks, then do the following:

1. Start VMware Fusion Pro, and select the menu **VMware Fusion > Preferences**
2. Click the **Network** icon
3. Click the lock icon to authenticate and create additional networks
4. Click the **+** icon to create an additional network, for my example I will use **vmnet2** and **vmnet4**.
5. Select **vmnet2** and configure the following (provider) network:
  - Select the option, "**Allow virtual machines on this network to connect to external networks (using NAT).**"
  - Select the option, "**Connect the host Mac to this network.**"
  - Select "**Provide addresses on this network via DHCP.**"
  - In the **Subnet IP** field, enter **10.128.1.0**
  - In the **Subnet Mask** field, enter **255.255.255.0**
6. Select **vmnet4** and configure the following (OpenStack API) network:
  - Select the option, "**Allow virtual machines on this network to connect to external networks (using NAT).**"
  - Select the option, "**Connect the host Mac to this network.**"
  - Select "**Provide addresses on this network via DHCP.**"
  - In the **Subnet IP** field, enter **10.128.1.0**
  - In the **Subnet Mask** field, enter **255.255.255.0**

## Create Ubuntu 14.04 Instance

Download an ISO image [Ubuntu Server 14.04.5 Trusty](#) for the DevStack VM. There are more recent versions of Ubuntu available; however, these may not be supported by the OpenStack Mitaka release.

The VM that I create has 2 processor cores and 4GB RAM of -- it is not a very beefy setup, and more resources could be added, but this is good enough for a small development environment. If you want to add a BIG-IP VE on the same development host, you should stay within these constraints.

## Provision Processors and Memory

1. Start VMware Fusion Pro, and select the menu **File > New**, and click Continue.
2. Chose to install from the **ubuntu-14.04.5-server-amd64.iso**
3. Select **Customize Settings**
4. Chose a name for the virtual machine and click **Save**
5. Select **Processors & Memory**
  - Select 2 processor cores from the **Processors** drop down
  - Change the **Memory** amount to **4096**
  - Click on **Advanced options** and select **Enable hypervisor applications in this virtual machine**

## Connect Network Adapters

1. Click Network Adapter, and click **vmnet2**
2. Click **Show All**, and click **Add Device**
3. Click Network Adapter, and **Add...**
4. Click **vmnet4**, and click **Show All**
5. Close the Settings window and start the install.

## Install the Ubuntu 14.04.5 Server VM

Chose all the defaults during install until the network configuration.

1. Select **eth0** as the primary network interface. This will be the management interface.
2. Enter whatever you would like for hostname
3. Add a new user and add a password.
4. For disk partitioning:
  - Select the **Guided – use entire disk and set up LVM** method
  - Select **Yes**, when prompted, **Write the changes to disks and configure LVM?**
  - Accept the default when prompted, **Amount of volume group to use for guided partitioning**. Click **Continue**
  - Select **Yes**, when prompted; **Write the changes to disks?**
5. Choose **Continue** when prompted for a proxy.
6. When prompted; “**How do you want to manage upgrades on this system?**” Select: **No automatic updates**
7. Select **OpenSSH server** in the **Software Selection** screen. Wait for software to install.
8. Select **Yes**, when prompted; **Install the GRUB boot loader to the master boot record?**
9. To finish the install select **Continue**

## Configure the Virtual DevStack VM

1. After the DevStack VM reboots, you’ll be presented with a login screen.
2. Log in to the Ubuntu server with the username and password you created during the install
3. Update and install packages

```
$ sudo apt-get update
$ sudo apt-get upgrade -y
$ sudo apt-get install git -y
```

4. Reboot the host.

## Configure the VM networking

It is a good idea to create static IP addresses on the guest interfaces for the OpenStack service API endpoints. Edit the network configuration, /etc/network/interfaces. I modified mine based on the VM’s network adapter configuration.

# This file describes the network interfaces available on your system

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface

auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static address
    10.128.1.128
    netmask 255.255.255.0
    gateway 10.128.1.2

auto eth1
iface eth1 inet static
    address 10.128.20.150
    netmask 255.255.255.0
    network 10.128.20.0
```

You must also update resolver configuration, /etc/resolvconf/resolv.conf.d/base

```
nameserver 8.8.8.8
search localdomain
```

Once you have finished, you can either restart networking and resolvconf or just reboot the guest.

## Install DevStack

These instructions are taken from the DevStack website:

<http://docs.openstack.org/developer/devstack>

### Add Stack User

```
$ sudo adduser stack
$ sudo echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

### Download DevStack

Login as the stack user or as the user you created when installing Ubuntu and clone the devstack repository. To work on Mitaka, checkout the stable mitaka branch.

```
$ git clone https://git.openstack.org/openstack-dev/devstack
$ cd devstack
$ git checkout -b stable/mitaka origin/stable/mitaka
```

### Create a local.conf

The local.conf is used as configuration input to the DevStack deployment scripts. It is very configurable, but not well documented. Most of the default values in the configuration file are acceptable; however, if you want to explore the options available you will very likely find yourself delving into the deployment scripts. What I am providing is a sparse configuration to get **Keystone, Glance, Nova, Neutron and Neutron-LBaaS** running. The devstack repository contains a sample local.conf, devstack/samples/local.conf.

1. Copy the sample configuration into the devstack directory.

```
$ cp devstack/samples/local.conf devstack
```

2. Modify the HOST\_IP and HOST\_IP\_IFACE to reference the API endpoint. In the case of this example we use **eth1**

```
HOST_IP=10.128.20.150
```

```
HOST_IP_IFACE=eth1
```

3. Append a neutron configuration:

```
# Neutron
# -----
disable_service n-net
enable_service neutron
enable_service q-svc
enable_service q-agt
enable_service q-dhcp
enable_service q-l3
enable_service q-meta
enable_service q-lbaasv2

OVS_ENABLE_TUNNELING=True

NEUTRON_CREATE_INITIAL_NETWORKS=False

Q_USE_SECGROUP=True
PUBLIC_INTERFACE=eth0
PUBLIC_BRIDGE=br-ex

Q_USE_PROVIDERNET_FOR_PUBLIC=True

OVS_PHYSICAL_BRIDGE=$PUBLIC_BRIDGE
PHYSICAL_NETWORK=physnet
PROVIDER_NETWORK_TYPE=flat
```

1. Disable **swift**, **tempest** and **horizon** and fix the noVNC version to 0.6.0

```
# Swift
# -----
disable_service s-proxy
disable_service s-object
disable_service s-container
disable_service s-account

# Disable services
disable_service horizon
disable_service tempest

# Fix for console issues on noVNC
NOVNC_BRANCH=v0.6.0
# Make noVNC available on eth0
NOVNC_PROXY_URL="http://10.128.1.128:6080/vnc_auto.html"
```

The swift project can be enabled, but I find it superfluous to manage object storage on a laptop virtual machine. The current version of Horizon exposes a bug in the **openstack-sdk** package that breaks the installation. It can be added back in on a subsequent build, but a change to the Mitaka requirements must be made first. I will add details below. The noVNC branch must be fixed at 0.6.0 or you will experience problems with console access to VM's.

## Deploy Stack

Now run the stack script. This will take about 15 minutes the first time to install, but once the service repositories are cloned will be much shorter on subsequent runs.

```
$ cd devstack
$ ./stack.sh
```

Congratulations! You should now have a running Mitaka stack.

```
=====
DevStack Components Timed
=====
run_process - 46 secs
test_with_retry - 3 secs
apt-get-update - 6 secs
pip_install - 238 secs
restart_apache_server - 6 secs
wait_for_service - 8 secs
git_timed - 134 secs
apt-get - 101 secs
```

## Teardown and cleanup stack

Once you are finished with the deployment you should teardown the services and cleanup the stack.

```
$ cd devstack
$ ./unstack.sh
$ ./clean.sh
```

## Install Horizon

In order to get Horizon to work on the current stable mitaka branch, you will need to teardown your current stack and restack. Once the stack is cleaned up, edit the following file, */opt/stack/requirements/upper-constraints.txt* by changing the upper constraint on openstacksdk. You can uncomment the **disable\_service horizon** line in *local.conf* and then restack.

Change the line:

```
openstacksdk===0.8.1
```

To:

```
openstacksdk===0.9.6
```

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

---

F5 Networks, Inc.  
Corporate Headquarters  
info@f5.com

F5 Networks  
Asia-Pacific  
apacinfo@f5.com

F5 Networks Ltd.  
Europe/Middle-East/Africa  
emeainfo@f5.com

F5 Networks  
Japan K.K.  
f5j-info@f5.com

---

©2018 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113