**Devstack Installation Guide**

October 2014 V1.1

**OneCloud Consulting Openstack Installation Guide using DevStack**

**ATTENTION**

The information contained in this guide is for training purposes only.This guide contains information and activities that, while beneficial for purposes of training in close, non-production environment, can result in downtime or other severe consequences and therefore are not intended as a reference guide.

No part of this book may be reproduced in any form or by any means (graphic, electronic, mechanical, including photocopying, recording or taping, or storage in an electronic retrieval system) without permission of OneCloud Consulting.

OneCloud Consulting reserves the right to change the contents herein without any prior notice.

[Overview 4](#_Toc397984038)

[Installation Steps 4](#_Toc397984039)

[References](#_Toc397984043) 5

**Overview**

DevStack is a tool initially developed to speed the deployment of OpenStack for development purposes, hence “Dev” Stack. It has been used in certain instances to also allow for a quick example openstack deployment with all of the latest features, as it pulls its components from the “master” branch of each of the services. One incongruity today is that DevStack still uses the nova-network engine by default.

**Steps to follow**

1. Install a Hypervisor

You need a base operating system on which to install DevStack, and in the spirit of the developer, this is often on a laptop. You will need a hypervisor of some nature to do this on your laptop, and if you have one already, you can skip using VirtualBox, otherwise, you can download the VirtualBox hypervisor and management UI from the following link.

<https://www.virtualbox.org/wiki/Downloads>

You may also decide to use a remote virtual platform (or even a remote physical machine) to deploy onto.

1. Install the latest version of Ubuntu as a base OS for the installation Devstack.

Download the latest Ubuntu Trusty image (Ubuntu 14.04.1 LTS) from the Ubuntu repository (see the end of this section for potential alternatives. It is recommended to use the desktop version, to avoid some of the peculiarities of VirtualBox networking, as you would then potentially use the embedded graphical interface for use of tools like Horizon or other web services interfaces (e.g. if you used RESTclient for Firefox to exercise the actual RESTful APIs).

<http://releases.ubuntu.com/trusty/ubuntu-14.04-desktop-amd64.iso>

Once you have the ISO, you will want to create a new VM in VirtualBox, select Linux and Ubuntu, and provide a name like “devstack” for your instances. You’ll want 4GB of RAM (2GB is the minimum you can likely get away with), and you’ll want to define a 32GB thin provisioned disk (I try to use qcow2 when available). Once complete, start the VM, and add the ISO you downloaded to the machine. Once booted, follow the instructions to install Ubuntu onto the local disk.

NOTE: You may also receive these files from the instructor via USB or similar transfer, or your instructor may provide you with an OVA bundle that you can install in most hypervisors to further accelerate the process of bringing up a DevStack environment. (OVA default user: ubuntu, default password: ubuntu)

1. Once Ubuntu is installed in VirtualBox, or your OVA is imported and running, log into the user interface, and start a terminal session.
2. Then run the following steps:

sudo apt-get update

sudo apt-get install git

git clone <https://github.com/openstack-dev/devstack.git>

cd devstack

# TWO NOTES:

# ONE: If you prefer to run Neutron, jump to step 6 before running stack.sh

# TWO: If you don’t, want to run neutron, just follow the instructions from devstack

./stack.sh

1. Once the installation is complete, user will be provided with the IP address, username and password that can be used to access the Horizon dashboard. In the same directory where the stack.sh script was run, there is an openrc file that can be sourced, allowing the use of the also installed command line client tools:

source ./openrc

nova list

1. It is often preferable to actually leverage the OpenStack Neutron project for the network service, and in fact in order to modify the behavior of DevStack, it is often useful to modify other services (documentation, as with many recent OSS projects is “in the script”). We’ll create a new file on the Ubuntu machine and copy in the following data {note copy *in* your VM}:

# Create a new file called local.conf, and copy this content into it, with vi or your favorite editor.

[[local|localrc]]

# Uncomment the next line after you've run devstack once if you don't want to try to download everything again.

#OFFLINE=true

# Set default passwords, further accelerate deployment, maybe make this less generic if you're in an environment where someone may 'play' with your system

ADMIN\_PASSWORD=ubuntu

DATABASE\_PASSWORD=ubuntu

RABBIT\_PASSWORD=ubuntu

SERVICE\_PASSWORD=ubuntu

# This is the token used to set up Keystone – It just needs to be a long random string

SERVICE\_TOKEN=a682f596-76f3-11e3-b3b2-e716f9080d50

disable\_service n-net

enable\_service q-svc

enable\_service q-agt

enable\_service q-dhcp

enable\_service q-l3

enable\_service q-meta

enable\_service q-metering

NOTE: the OFFLINE parameter reference, use this only *after* having run against this local.conf file, so that all of the code needed for Neutron is downloaded. Then setting it will speed up the re-start of DevStack if you want to tweak configuration parameters (not covered in this simple tutorial).

You may find it easier to grab this file from within the VM by pointing your browser (in the VM) to the following gist: <https://gist.github.com/rstarmer/29b54c26b32387b392f2>

Or

<http://git.io/TQmKbQ>

1. Once you’ve created your local.conf file, as defined above, run the un-stack.sh script to clean up your environment, and then run stack.sh again to build it against the new configuration. You may also want to run re-join-stack.sh in order to load the screen environment (where all of the services log files are accessible as separate screen sessions).

./unstack.sh #Ignore any errors, they are not likely to be relevant

./stack.sh #Should run to completion, otherwise run the previous command again

# Stop here or check step 5 again for basic access/usage instructions or run the following

# for access to all of the log files

./rejoin-stack.sh

Screen commands:

<CTRL>-a “ List all screens in a tabular format

<CTRL>-a D Leave the screen session (but do not destroy the screen sessions)

screen –r Rejoin the screen session if you are in the same terminal as where you just left

Otherwise, use re-join-stack.sh

1. As with step 5, you have a running environment, use either Horizon or the CLI tools to manipulate the system.

**References**

http: //www.devstack.org

http://docs.openstack.org/infra/publications/devstack-tutorial/#(1)