



Introduction to Virtualization & OpenStack

Provided By: **S. Saeed Hosseini**



[SayidHosseini](#)



[SayidHosseini](#)

Index

- Virtualization
 - Types of Virtualization
 - Operating System-level Virtualization
 - Operating System-level Virtualization softwares
 - VirtualBox and Ubuntu
- Introduction to OpenStack
 - OpenStack Architecture
 - OpenStack Keystone
 - OpenStack Glance
 - OpenStack Neutron
 - OpenStack Nova
 - OpenStack Cinder
 - OpenStack Horizon

Index

- Introduction to DevStack
 - DevStack Installation
 - Installation and Debugging Notes

Virtualization

- In computing, **virtualization** refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.
- Virtualization began in the 1960s, as a method of logically dividing the system resources provided by mainframe computers between different applications. Since then, the meaning of the term has broadened.

Types of Virtualization

- Execution Virtualization
 - Hardware Level
 - Operating System Level
 - Programming Language Level
- Network Virtualization
- Storage Virtualization
- Desktop Virtualization
- ...

Hardware-assisted Virtualization

- This term refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation
- This technique was originally introduced in the IBM System/370
- At present, examples of hardware-assisted virtualization are the extensions to the x86-64 bit architecture introduced with Intel-VT and AMD-V

Operating System-level Virtualization

- It offers the opportunity to create different and separated execution environments for applications that are managed concurrently
- Differently from hardware virtualization, there is no virtual machine manager or hypervisor, and the virtualization is done within a single operating system, where the OS kernel allows for multiple isolated user space instances

Operating System-level Virtualization

- The kernel is also responsible for sharing the system resources among instances and for limiting the impact of instances on each other
- A user space instance in general contains a proper view of the file system, which is completely isolated, and separate IP addresses, software configurations, and access to devices

OS-level Virtualization Softwares

- VMWare Workstation Pro/Player
- Microsoft Hyper-V
- Parallels Desktop
- VirtualBox
- KVM
- QEMU
- Xen
- ...

VirtualBox



- VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.
- Presently, VirtualBox runs on Windows, Linux, Macintosh

Ubuntu on VirtualBox



- It's the time to create a virtual machine on your OS
- We are going to install Ubuntu 18.04.3 LTS on VirtualBox with the following minimum requirements:
 - 4 Virtual CPU Cores
 - 6 GB of RAM
 - 50 GB of Storage
 - 2 Networks (NAT + Host-Only)
- Make sure to take a snapshot after a clean install.

Enable SSH for easier communication with your VM!

Introduction to OpenStack



- OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.
- A dashboard is also available, giving administrators control while empowering their users to provision resources through a web interface.

Introduction to OpenStack



- Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.

OPENSTACK

OPENSTACK-USER



OpenStackClient
Python SDK

OPENSTACK-ADJACENTENABLERS



CONTAINER SERVICES

Kuryr



NFV

Tacker



WEB FRONTEND

Horizon



API PROXIES

EC2API



WORKLOAD PROVISIONING

Magnum Trove
Sahara



APPLICATION LIFECYCLE

Murano Freezer
Solum Masakari



ORCHESTRATION

Heat Mistral Aodh
Senlin Zaqr Blazar



COMPUTE

VIRTUAL MACHINES

Nova

CONTAINERS

Zun

FUNCTIONS

Qinling



NETWORKING

SDN

Neutron

LOAD BALANCING

Octavia

DNS

Designate



HARDWARE LIFECYCLE

BARE METAL

Ironic

ACCELERATORS

Cyborg



STORAGE

OBJECT

Swift

BLOCK

Cinder

FILE

Manila



SHARED SERVICES

Keystone

Placement

Glance

Barbican

Searchlight

Karbor

OPENSTACK-LIFECYCLEMANAGEMENT



DEPLOYMENT / LIFECYCLE TOOLS

Kolla-Ansible OpenStack-Charms TripleO Bifrost
OpenStack-Helm OpenStack-Ansible OpenStack-Chef



PACKAGING RECIPES FOR...

RPM Puppet
OCI containers

OPENSTACK-OPERATIONS



MONITORING TOOLS

Ceilometer
Monasca Panko



OPTIMIZATION / POLICY TOOLS

Watcher Vitrage
Congress Rally



BILLING / BUSINESS LOGIC

CloudKitty



MULTI-REGION TOOLS

Tricircle

Bold represents **Core Functionality**

Version 2019.06.02

OpenStack Keystone



- Keystone is an OpenStack service that provides API client authentication, service discovery, and distributed multi-tenant authorization by implementing OpenStack's Identity API
- Keystone is organized as a group of internal services exposed on one or many endpoints. Many of these services are used in a combined fashion by the frontend. For example, an authenticate call will validate user/project credentials with the Identity service and, upon success, create and return a token with the Token service

OpenStack Glance



- The Image service (glance) project provides a service where users can upload and discover data assets that are meant to be used with other services. This currently includes images and metadata definitions
- Glance image services include discovering, registering, and retrieving virtual machine (VM) images. Glance has a RESTful API that allows querying of VM image metadata as well as retrieval of the actual image

OpenStack Neutron



- Neutron is an OpenStack project to provide “network connectivity as a service” between interface devices (e.g., vNICs) managed by other OpenStack services (e.g., nova). It implements the OpenStack Networking API.

OpenStack Nova



- Nova is the OpenStack project that provides a way to provision compute instances (aka virtual servers). Nova supports creating virtual machines, baremetal servers (through the use of ironic), and has limited support for system containers. Nova runs as a set of daemons on top of existing Linux servers to provide that service.

OpenStack Cinder



- Cinder is a Block Storage service for OpenStack. It's designed to present storage resources to end users that can be consumed by the OpenStack Compute Project (Nova). This is done through the use of either a reference implementation (LVM) or plugin drivers for other storage.
- The short description of Cinder is that it virtualizes the management of block storage devices and provides end users with a self service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device.

OpenStack Horizon



- Horizon is the canonical implementation of OpenStack's Dashboard, which provides a web based user interface to OpenStack services including Nova, Swift, Keystone, etc
- Horizon ships with three central dashboards, a "User Dashboard", a "System Dashboard", and a "Settings" dashboard. Between these three they cover the core OpenStack applications and deliver on Core Support

OpenStack Releases

Series	Status	Initial Release Date	Next Phase	EOL Date
Ussuri	Future	2020-05-13 <i>estimated (schedule)</i>	Maintained <i>estimated 2020-05-13</i>	
Train	Development	2019-10-16 <i>estimated (schedule)</i>	Maintained <i>estimated 2019-10-16</i>	
Stein	Maintained	2019-04-10	Extended Maintenance <i>estimated 2020-10-10</i>	
Rocky	Maintained	2018-08-30	Extended Maintenance <i>estimated 2020-02-24</i>	
Queens	Maintained	2018-02-28	Extended Maintenance <i>estimated 2019-10-25</i>	
Pike	Extended Maintenance	2017-08-30	Unmaintained <i>estimated TBD</i>	
Ocata	Extended Maintenance	2017-02-22	Unmaintained <i>estimated TBD</i>	
Newton	End Of Life	2016-10-06		2017-10-25
Mitaka	End Of Life	2016-04-07		2017-04-10
Liberty	End Of Life	2015-10-15		2016-11-17
Kilo	End Of Life	2015-04-30		2016-05-02
Juno	End Of Life	2014-10-16		2015-12-07
Icehouse	End Of Life	2014-04-17		2015-07-02
Havana	End Of Life	2013-10-17		2014-09-30
Grizzly	End Of Life	2013-04-04		2014-03-29
Folsom	End Of Life	2012-09-27		2013-11-19
Essex	End Of Life	2012-04-05		2013-05-06
Diablo	End Of Life	2011-09-22		2013-05-06
Cactus	End Of Life	2011-04-15		
Bexar	End Of Life	2011-02-03		
Austin	End Of Life	2010-10-21		

DevStack



- DevStack is a series of extensible scripts used to quickly bring up a complete OpenStack environment based on the latest versions of everything from git master. It is used interactively as a development environment and as the basis for much of the OpenStack project's functional testing
- Check out the "stein" documentation from here:
<https://docs.openstack.org/devstack/stein/>

DevStack



- Following the instruction from the DevStack documentation, first you'll need to create a non-root user with sudo privileges and change to that user:

```
$ sudo useradd -s /bin/bash -d /opt/stack -m stack  
$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack  
$ sudo su - stack
```

- Now, since opendev.org sanctioned Iran, you may use the GitHub repository instead:

```
$ git clone https://github.com/openstack/devstack.git -b stable/stein  
$ cd devstack
```

DevStack



- Create a local.conf file with the following contents:

```
[[local|localrc]]
HOST_IP=w.x.y.z

ADMIN_PASSWORD=secret1
DATABASE_PASSWORD=secret2
RABBIT_PASSWORD=secret3
SERVICE_PASSWORD=secret4

LOGFILE=./stack.sh.log
VERBOSE=True
ENABLE_DEBUG_LOG_LEVEL=True
ENABLE_VERBOSE_LOG_LEVEL=True

GIT_BASE=${GIT_BASE:-https://github.com}
```

DevStack



- ❑ Check the configuration file documentation from [here](#).
- ❑ The “**HOST_IP**” parameter defines the IP that horizon will be available on. Assign the IP from the **Host-Only** network card.
- ❑ The passwords can be the same or different.
- ❑ The logs of the installation will be available in “**stack.sh.log**” file in the devstack installation directory. Check that out if something goes wrong!

DevStack



- ❑ As stated before opendev.org sanctioned Iran, so the last line states that every repository should be downloaded from GitHub. However, some repository may want to download from opendev.org. So to get around that you may use **openconnect** on you ubuntu; or an alternative that you are familiar with.
- ❑ After preparing the file, enter **"./stack.sh"** to start the process. Make sure you are connected to the internet.
- ❑ If installation finished successfully, make sure to take a snapshot.

DevStack



- ☐ If something went wrong, you always have the option to revert to your clean install snapshot.
- ☐ If you want to stop all of the devstack services, enter **`"/unstack.sh"`**.
- ☐ If you want to stop and uninstall devstack and all of its dependencies, enter **`"/clean.sh"`**.

Good Luck with the installation!

Resources

- Cloud-Computing course slides of Ferdowsi University of Mashhad
- <https://www.virtualbox.org/>
- <https://www.openstack.org/software/>
- <https://docs.openstack.org/keystone/latest/>
- <https://docs.openstack.org/keystone/stein/getting-started/architecture.html>
- <https://docs.openstack.org/glance/stein/>
- <https://docs.openstack.org/neutron/stein/>
- <https://docs.openstack.org/nova/stein/>
- <https://wiki.openstack.org/wiki/Cinder>
- <https://wiki.openstack.org/wiki/Horizon>
- <https://releases.openstack.org/>
- <https://docs.openstack.org/devstack/stein/>
- <https://docs.openstack.org/devstack/stein/configuration.html>
- <https://stackoverflow.com/questions/20390267/installing-openstack-errors>
- <https://ask.openstack.org/en/question/58887/how-to-complete-uninstall-devstack/>