OpenStack Architecture & Deployment Training
5 Day Course | Lecture & Labs

COURSE OUTLINE
This course consists of five training modules
- Day 1 Openstack Architecture
- Day 2 Virtualization (NOVA)
- Day 3 Networking (Neutron) – This section has an optional one-day expansion
- Day 4 Storage (cinder, swift, ceph glance, ephemeral)
- Day 5 Subsystems, high availability, and DEVOPS

== DAY ONE: OPENSTACK ARCHITECTURE ===================
This section provides an overview of Openstack. In this section we are focused on what OpenStack is, its architecture, and how OpenStack components fit together. The labs are focused on OpenStack operations. This module consists of Chapters 1 which describes OpenStack’s architecture, chapter 2 which describes how to use the browser user interface, and chapter 3 which describes how to use the command line to configure OpenStack. We consider the skills imparted in this module to be essential for everyone working or managing OpenStack.

1. OpenStack Architecture Overview
   - Big Picture
     - OpenStack Big Picture
     - IaaS (Infrastructure as a Service)
     - PaaS (Platform as a Service)
     - SaaS (Software as a Service)
     - OpenStack XaaS
   - OpenStack Architecture
     - Nodes
     - Controller Node
     - Neutron Node (aka Networking Node)
     - Compute
     - Storage Node
     - Storage Node - Ceph
     - Node Roles
2. Horizon
   - OpenStack operations and management from the browser
   - OpenStack Services in Detail
   - Horizon aka the “Dashboard”
   - Project vs User
   - Overview Screen
   - Projects
   - Launching and managing Instances
   - Assigning Floating IP addresses
   - Assigning Key pairs
   - Network Topology

3. Managing OpenStack from the Command Line
   - How to quickly search and select the correct command
   - Some services that are only supported by command line, not GUI
   - How to use the command line to do all the tasks we did previously with a browser
   - Mainly we will focus on
     - Command structure
     - Managing an instance
     - Booting an instance
     - Deleting an instance
     - The base skills that are essential for more complex deployments

Day 1 LABS:

Lab 0: Learning the lab environment
Lab 1: Openstack python-clients
Lab 2: Introduction to OpenStack API Endpoints
Lab 3: Verifying OpenStack Services
Lab 4: Exploring Horizon
Lab 5: Adding a Project (Tenant) and Defining a Quota in Horizon
Lab 6: Adding Users to a Project in Horizon
Lab 7: Launching Instances as a User
Lab 8: Taking Control of a Project as admin
Lab 9: Edit the Horizon Log-In Timeout Setting
Lab 10: Creating a Project and Adding Users at the CLI
Lab 11: User Roles and Permissions
Lab 12: Adding a User at the CLI within the OpenStack-python Environment
Lab 13: Keystone (identity) credentials
Lab 14: CLI OpenStack RC files
Lab 15: Using Linux Environmental Variables
This section is concerned with the virtual machines themselves, or in OpenStack’s vernacular, the “instance”. Chapter 4 studies the compute node’s perspective and talks about virtualization, providing enough detail to clearly understand the work that an OpenStack compute node performs. Chapter 5 covers enough details regarding permissions to that you can successfully launch instances. Chapter 6 describes Nova, the “heart” of openstack, what Nova can do, and how to configure Nova. The labs associated with this module are more advanced than day one, requiring students to launch business-driven instances rather than cookie-cutter instances we launched in day 1. This section is important any people that will use or develop virtual machines.

4. Hypervisors

- Overview
  - The Virtual Machine Stack
  - The Old Way without Virtualization
  - Virtualization
  - A Snowflake Computer on Bare Metal
  - A Virtualized System installed on a Bare Metal Hypervisor
  - The “Top Half” of the Hypervisor
  - The “Bottom Half” of the Hypervisor
  - Why is the Hypervisor shown as an upper and lower portions?

- Images
  - “Imaging” the Operating System
  - Imaging our Machine

- Migration
  - “Migrating” the Virtual Machine

- Compute
  - “Compute”
  - Compute Space Example
  - Assemble the Server
  - Install the Hypervisor
  - The Compute Space
  - Resulting Compute Space with just 2 Intel Xeon E5-2699 v3 CPUs
  - Intel’s Most Powerful Processor (July 20, 2015)
  - Compute Space Example

- Compute Cloud
  - A “360-Core” Cloud
  - Clouds not Cloud
  - Remember we talked about “Migrating”?
  - Migrating
  - Instantiation
  - Oversubscribing 16:1 is the OpenStack Default!
  - Agenda

- Virtualization Types
  - Virtualization Requirements
  - The VMM and the VM
  - x86 modes: Privilege Levels
  - Evolution of Virtualization
  - Full Virtualization - Advantages
5. Keystone
   - Overview
     - Keystone
     - Alternatives to Tokens
     - Tokens - what they are and why you need them
     - OpenStack Shell Command Permissions
     - Keystone Authentication Message Flow
     - How to use curl with the Keystone Identity API
     - Keystone Access – Token and Service Catalog

6. Nova
   - nova-api
   - nova-scheduler
   - Nova Architecture
     - Nova Cert
     - Nova Compute
     - Nova Hypervisor Support
     - Nova Console
     - Nova Conductor
     - Nova Compute Interfaces
     - Nova Components
     - Nova Review
   - Management
     - Nova-api CLI
     - Nova Networking

Day 2 LABS:

Lab 16: APIs, RESTful messages, and Keystone (Identity) authentication
Lab 17: Creating new public and private flavors at the CLI
Lab 18: Launching Instances from the CLI
Lab 19: OpenStack Virtual Machine Metadata
Lab 20: Creating Security Groups in Horizon and CLI
Lab 21: SSH Keypairs
Lab 22: Floating IP Addresses
Lab 23: Logging
== DAY THREE: NETWORKING ==========================

There are many aspects to understanding OpenStack Clouds, Layer 2 networking, Layer 3 networking, L2 and L3 plugins, DHCP, qrouter, OVS or Linux bridge, Provider and Tenant networks. Chapter 7 starts this section with an explanation of software defined networking as well as network function virtualization. Enough information is provided here to understand what it is. If you want a deep-dive into this technology is available in our SDN class and can be inserted here on request. Chapter 8 is a practical description of the provider network and who to interface with it. Chapter 9 dives into neutron, imparting enough skills to deploy most OpenStack deployment types. If you are looking for a deep dive into Neutron, that can be included here on request. The (+++) indicator on the labs indicate optional and difficult labs for those students looking for a deep dive into Neutron.

7. SDN and NFV
   - Stack
     - Network Functions Virtualization (NFV)
   - Overview
     - Virtualize the Middleboxes
     - Goal: All routers share the same picture
     - Goal: "Network Omniscience"
     - Goal: "Networking Becomes Software Defined"
     - The Current State of Networking
     - SDN Déjà vu?
     - If H.248 behaved like SDN (OpenFlow)
     - Defining Software “Abstraction”
     - Abstracting the Network
     - SDN in a Nutshell
     - Isolating Networks
     - Yet Another Abstraction: Network Slicing
     - SDN Architecture
     - Merchant Silicon Example
     - Requirement: Non blocking at 10 Gbps
   - OpenFlow
     - OpenFlow Example
     - Forwarding an Audio Packet
     - Destination MAC Address
     - Type Field means “An IP header is next”
     - IP Source Address Validity Check
     - IP Destination Address
     - Deep Packet Inspection (DPI)
     - The “Really Bad” Stuff is Always BELOW the Line
     - Fabrics are “Really” Fast

8. Data Center Fabric
   - Overview
     - Classic Hierarchical Network Design
     - TRILL
     - Path #1
     - Path #2
     - Path #3
Path #4
- Interfacing NFV with the Fabric
  - NFV Integration
    - vSwitch
    - Namespace and the “veth”
    - Heat Makes Clouds Rise
    - vSwitch vs Hardware Switch
    - Virtual Router
  - Interfacing with the provider network
    - Configuration
    - Verification
    - Overview of the options

9. OpenStack Networking
   - Neutron
     - Why Traditional Networking is Inefficient
     - Network Virtualization
     - Agents and Plugins
     - Agents
     - Neutron Configuration
     - Open vSwitch Rules of Thumb
   - Architecture
     - Nova Networking – Flat Mode
     - Nova Networking – Flat Mode + DHCP
     - Nova Networking – Multi-Host Mode
     - Compute Node Network OVS Integration
     - Neutron Networking
     - East West Traffic in Neutron Networking without Distributed Virtual Router
     - Neutron Networking Distributed Virtual Router
     - East West Traffic with Distributed Virtual Router
     - North South SNAT Traffic with Distributed Virtual Router
     - North South Floating IP Traffic with Distributed Virtual Router
     - Neutron Big Picture with Open vSwitch
   - OpenvSwitch
     - Traditional VM Ethernet Processing
     - Intel VMDq (Virtual Machine Device Queues)
     - Intel SR-IOV (Single Root IO Virtualization)
   - NFV
     - Namespaces
     - Linux Bridge
     - SDN and NFV Vendors
     - ML2 Layer
     - L3 layer

Day 3 LABS: (+) Number of plusses indicates optional labs for detailed study

Lab 24: The Provider Network - BUI
Lab 25: The Provider Network - CLI
Lab 26: Router deployment - BUI
Lab 27: Router deployment - CLI
Lab 28: DHCP Agent BUI and CLI
Lab 29: The tenant network – BUI
Lab 30: The tenant network: - CLI
Lab 31: L3 agent configuration
Lab 32: L2 agent configuration
Lab 33: routing between tenant networks (+)
Lab 34: Linux Bridge Agent (+++)
Lab 35: OVS agent (+++)
Lab 36: Neutron Internals (++++) These are the skills needed to troubleshoot

**IMPORTANT**: the following adds ONE DAY to this course! If you need this, you probably do not need the storage (day 4) and much of the subsystems sections (day 5), so we can teach that material at a higher level and cover the following in detail.

- Analyzing network namespace
- Analyzing qrouter
- Analyzing veths
- Analyzing floating IP addresses
- Analyzing fixed IP addresses
- Analyzing the L2 network
- IP Tables
- OVS protocol analysis
- BRCTL
- ip link
- ethtool
There are three basic types of cloud storage, ephemeral, object, block and so interesting combination. Chapter 10 covers ephemeral storage, Chapter 11 defines object storage and then describes Swift, which an object storage system similar to Amazon S3. Chapter 12 covers Ceph, which is growing in popularity because it can combine both Block and Object Storage, Chapter 13 covers block storage, and chapter 14 covers the storage of virtual machine images. While it could make sense to place chapter 14 with instances, since that is what image is for, images need storage, and we find it is easily learned under the storage module. OpenStack clouds cannot function without storage, so this section is essential, but if more networking details are needed, this section can be covered at a high level and more detail regarding networking may be included.

10. Ephemeral
   - Definition
   - When to use Ephemeral storage
   - Where ephemeral storage physically exists
   - Ephemeral types
     - Disk
     - Ephemeral

11. Swift
   - Overview
     - Swift
     - Object Storage Examples
     - Defining an Object
     - HTTP RESTful API
   - Working with Swift
     - Analyzing the Swift URL
     - Data Types
     - Comparing Storage Types: Object, File, Block
     - Consistency vs Availability Example
     - Saving a Swift Object
     - Immutable Cluster Sizing
     - Calculating part_power
     - The Swift Proxy
     - Multiple Swift Proxies and Load Balancing
     - Swift Enhanced Consistent Hashing Ring
     - Durability with MD5 Metadata
     - Durability with Replication
     - Swift Background Auditor
     - Background Replicator
     - Enhanced Consistent Hashing Ring
     - Comparing Swift and Ceph
     - Swift Regions
     - How Ceph and Swift fit in OpenStack

12. Ceph
   - Overview
     - Storage Node - Ceph
     - Ceph Advantages
- RADOS
- Ceph Architecture
- Building Ceph Storage Nodes

- OSD
  - OSD (Object Storage Daemons)
  - Building Ceph Cluster for Geographic Disparity
  - Configuration to Prepare and Mount the Storage Devices to be Used with Ceph
  - Prepare and Mount the Storage Devices to be Used with Ceph

- CRUSH
  - Introduction to the CRUSH Map
  - The Ceph Monitors
  - CRUSH Advantages over Other Forms of Data Mapping
  - CRUSH Rules

- Replication
  - Data Storage and OSD Replication

- Durability
  - Handling OSD Failures or other Topography Changes
  - How to Recover from a Ceph SSH Configuration Error
  - Ceph - Troubleshooting the Logs
  - Crush Map from Ceph Lab
  - Storage Node Log: cat /var/log/ceph/cat ceph-osd.0.log
  - Ceph Log /var/log/ceph/ceph.audit.log
  - Watch Cluster Activity in Real-time with ceph -w
  - Erasure Coding Defined

- Deployment
  - Performance Rules of thumb
  - Ceph Deployment

13. Cinder
  - Overview
    - Cinder
    - Block – File - Object
    - The Linux Posix Filesystem Hierarchy
    - Cinder Components
    - Cinder Volume
    - Cinder Snapshot
    - Cinder Backend
    - Cinder Driver
    - Cinder Volume Type
    - Cinder Processes
    - iSCSI
    - NFS
    - Volume Attach Workflow

14. Glance
  - Architecture
    - Glance
    - Basic Architecture
    - Glance Command Line
    - Container Format
    - Disk Format
Day 4 LABS:

Lab 37: Ephemeral disk storage
Lab 38: Ephemeral storage
Lab 39: The storage server
Lab 40: Cinder volumes
Lab 41: Moving cinder volumes between instances
Lab 42: Glance Image storage

== DAY FIVE: SUBSYSTEMS =======

There are many other systems required to create a functioning OpenStack system, many of which may be installed and largely forgotten, until there are problems. This section covers the critical subsystems. Chapter 15 covers telemetry and alarming, Chapter 16 covers messaging, which glues the vast Openstack cloud together. Chapter 17 covers services that may be added to OpenStack which are highly specialized. Chapters 18 and 19 are DEVOPS, and normally are very short in duration. These chapters make it clear how to turn a single failed software release every nine months into 15 successful releases every day. Section 20 covers high availability techniques, and the last chapter covers some security basics.

15. Ceilometer
   ▪ Overview
     ○ Ceilometer - Telemetry
     ○ Metering Primer
     ○ Metering
     ○ Metering: Network
   ▪ Architecture
     ○ Notification
     ○ Polling - Agent
     ○ Data Collector
     ○ Ceilometer Architecture
     ○ OpenStack Ceilometer Reference

16. AMQP
   ▪ Overview
     ○ How AMQP Fits with Similar Messaging Protocols
     ○ AMQP in a Nutshell
   ▪ Message Delivery Options
     ○ Direct Exchange
     ○ Fanout
     ○ Topic Exchange
   ▪ Subscribe and Publish
Subscribe and Publish
Subscribe and Publish Message Flow
Embedding RPC in RabbitMQ Messages
Nova Messaging Service AMQP example
MQ Alternatives
RabbitMQ and zeroMQ

17. Other Services
- NTP
- Trove
  - Trove Database Manager
  - Trove - Database Service
  - Trove Architecture
- OSLO
  - OSLO - OpenStack Common Libraries
- MySQL
  - MariaDB - These Databases Define Your Cloud!
- GlusterFS

18. Git Essentials
- Setting up the repository
  - Why OpenStack users should know Git and GitHub
  - Git repo-to-repo collaboration
  - Git usage: git init --bare <directory>
  - Git usage: git clone <repo> <directory>
  - Git usage: git config --global
- Saving changes
  - Git usage: git add <file || directory>
  - Git usage: git commit -m "<message>"
  - Git saves files not deltas
  - Git usage: git status
  - Git usage: git log --oneline
  - Git usage: git checkout <commit || master>
- Collaboration
  - Git usage: git remote <add || rm> <name> <url>
  - Git usage: git branch <branch>
  - Git usage: git pull --rebase <remote>
  - Git usage: git push <remote> <branch>
  - GitHub is git + social

19. Cloud Automation
- Overview
  - You've got VMs! Now what …
  - Configuration Management
  - Automation…
  - Puppet
  - Chef
  - Ansible
  - Ansible Hosts
  - Ansible Playbook - YAML (Yet Another Markup Language)
  - Ansible Playbook - Running
20. HA

- OpenStack Vulnerability
  - Vulnerability Assessment: #1 Databases (STATEFUL)
  - Vulnerability Assessment: #2 Networking
  - Vulnerability Assessment: #3 Stateful Services
  - Vulnerability Assessment: #4 The Stateless Services (Everything Else)

- Planning
  - HA Design Follows RPO and RTO
  - Stateful vs Stateless
  - Failover, Failback, and Switchover
  - Active/Passive vs Active/Active
  - HA Options by Vendor

- Methods
  - Keepalived and HAProxy (keep alive’ dee)
  - VRRP Virtual Router Redundancy Protocol - RFC 3768
  - Native Cluster
  - Pacemaker
  - Totem
  - Resource Agents
  - Corosync
  - DRBD
  - Galera – Write Set Replication (WSREP)
  - Galera – Deadlock
  - MySQL Cluster Replication: Multi-master and Circular Replication
  - RabbitMQ HA
  - MySQL HA using Pacemaker, Corosync, and DRBD
  - HA for MySQL

- Planning
  - Hypervisor Evacuation
  - Keystone Identity Manager

21. Cloud Security

- Overview
  - Keystone Authentication
  - Securing APIs
  - Security groups: iptables and Linux bridges

Day 5 LABS:
Lab 43: GITHUB – Save all your work this week
Lab 44: Booting new instances using cloud-init
Lab 45: Cloud automation using Ansible
Lab 46: Heat