INTERNSHIP REPORT

RTDS

(February 2022 - June 2022)

Internship report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Harsh Mishra (181006)

Under the supervision of

Prof. Dr Sumedha Arora

Prateek karna (**Product & Development** , RTDS).

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

ACKNOWLEDGEMENT

Firstly, We would like to express our heartiest thanks and gratefulness to Almighty God for His divine blessing in making us possible to complete the project work successfully. We are grateful and wish our profound indebtedness to Dr.Sumedha Arora for her profound Knowledge, Professor of the Department of CSE & IT, Jaypee University of Information Technology, Waknaghat.

I am highly indebted to Prateek karna (Product & development Manager) for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude to Ace cloud hosting for their kind cooperation and encouragement which helped me in the completion of this project. Deep Knowledge & keen interest of my supervisor in the field of in Devops, Research & Development, Managing Product Line and Product Architecture" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project. We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Regards,

Harsh Mishra 181006
IT | JUIT

Table of content:

Certificate

Candidate’s Declaration

I hereby declare that the work presented in this training report for company “ **RTDS** ”in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information technology and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2022 to May 2022 under the supervision of Prateek karna (Product & development Team Lead) .
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Student Name: Harsh Mishra
Roll no.: 181006
This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Prof. Dr Sumedha Arora
Designation: Assistant Professor (SG)
Department name: Computer Science and Engineering
Dated: 27th May 2022
Supervisor Name: Prateek karna
Designation: SDM (Product & development Team Lead)
Department name: Product & development
Dated: 27th May 2022

List of Figures :

CHAPTER 1

INTRODUCTION

OpenStack is a cloud working framework that uses APIs and standard validation techniques to monitor and provision vast pools of processing, storage, and system management resources in the data center. It also provides a dashboard that guides the board and allows clients to place assets through the web interface.

Despite the general framework for management capabilities, additional parts provide various managed executive organizations, boards, and controls to ensure the accessibility of client applications. increase.

OpenStack is divided into management that allows it to be customized and played according to needs. The Openstack map allows you to see the Openstack climate "first" and understand where these controls fit and how they can be integrated.

Objectives

To create testable, structured, clean and maintainable web applications by using industrial best practices.

Motivation

To apply industrial best practices and create a fast, scalable and secure web application.

Purpose

This is a work underway. Its will probably portray the OpenStack people group's vision for the by and large OpenStack task's result as it develops over the long haul. It is likewise optimistic instead of distinct. That is, it characterizes the OpenStack that the local area is endeavoring toward, as opposed to the OpenStack that existed at any one time.

While investigating proposed includes and making points of interaction, project groups can allude to this report to help confirm that their plans fit easily inside a more extensive structure and add to the general scene of utilization organization designs.

Scope

The degree of this chronicle is limited to the cloud helps that an end-client associates with. This analyzes to the principal 'OpenStack' bucket and divides of the 'OpenStack Operations' holder in theOpenstack project map. While OpenStack in like manner has various kinds of genuine exercises (for instance plan gadgets and client libraries), nothing at all can be understood about our vision for them from this report.

The Pillars of Cloud

There are as various specific understandings of the saying "cloud" as there are modifying engineers. Regardless, we can all agree that the cloud has meaning. Dispersed registering maintains more effective resource use by decreasing trade costs related with giving and deprovisioning structure to move toward nothing, and it can do as such because it differentiates abstractly from earlier figuring models (counting virtualization). Two explicitly stick out.

OpenStack-specific Considerations

Most prohibitive fogs are worked by programming arranged by and for a singular affiliation. OpenStack is one of a kind - there are various OpenStack fogs, both public and private, each worked by a substitute relationship with different targets and making different decisions. These fogs could have covering sets of clients. This prompts a couple of necessities that are planned for OpenStack, and may not be shared by various fogs.

These thoughts are not commonly fitting to all pieces of the structure, yet we guess that all endeavors in OpenStack ought to change in accordance with them any spot they are important.

Partitioning

A region in an OpenStack cloud is portrayed as an alternate plan of organization endpoints in the Keystone organization record, yet a typical Keystone - allowing a selected client to get to any locale of the cloud starting from a comparative confirmation URL. This importance is compelled by the OpenStack programming, and subsequently will overall be unsurprising across fogs.

Strangely, groupings of resources that are portrayed by hardware or the genuine geology of the server ranch are vigorously impacted by individual cloud executives. For example, many fogs integrate the possibility of 'openness zones' - groupings inside a region that share no typical points of weakness. The OpenStack programming has no genuine method for approving this importance across fogs, and there are different various purposes behind cloud heads to have to assemble resources. OpenStack projects are encouraged to push toward allowing overseers to make conflicting, different evened out groupings of the resources they make due, and to do whatever it takes not to credit genuine ramifications to the groupings.

Design Goals

The going with plan goals address the limits that we should see the OpenStack organizations all things considered provide for applications and clients. It isn't commonplace that every help or part would (or even could) bear on each goal recorded. Rather, any assist that adds to accomplishing something with enjoying one of the targets under is probably going to assist with propelling the mission of the OpenStack project.

Hardware Virtualisation

For any assistance that is commonly given by a particular piece of gear, OpenStack hopes to give a vender free API that gives purchasers programming portrayed control of circulating the resource in a multi-tenant environment. This isn't restricted to virtual servers, yet may (for instance) additionally coordinate such things as cutoff, switches, load balancers, firewalls, HSMs, GPGPUs, FPGAs, ASICs (for example video codecs, etc.

A piece of these equipment classes could have unadulterated programming accomplices that can be utilized behind tantamount API, permitting applications to be moderate even to hazes that don't have express stuff in those cases.

Graphical User Interface

A GUI is many times the most effective way for new clients to move toward a cloud and for clients overall to explore different avenues regarding new areas of it. Introducing choices and work processes graphically manages the cost of disclosure of capacities such that perusing API or CLI documentation can't. A GUI is likewise frequently the most effective way for even experienced clients and cloud administrators to get an expansive outline of the condition of their cloud assets, and to envision connections between them. Consequently, notwithstanding the API and some other UIs, OpenStack ought to incorporate an electronic graphical UI.

Customisable Integration

OpenStack forces no specific sending model or design on applications. Each application has special necessities, and OpenStack obliges them by permitting administrations to be wired together in 'userspace' - through open APIs - as opposed to permanently set up moves made in the background that help just predefined arrangement models.

This permits the application engineer to modify anything utilizing client-side paste, yet shouldn't need it. OpenStack organizations should be satisfactorily integrated that they can be related together by the cloud buyer without requiring any client-side joint effort past the basic wiring.

Security models should permit the two sorts of communication - between OpenStack administrations, and among applications and OpenStack administrations in the two headings. They ought to likewise allow the cloud buyer to appoint just the negligible honors important to permit the application to work as planned, and consider normal disavowal and substitution of qualifications to keep up with however much security as could reasonably be expected in a climate where Internet-confronting machines are probably going to ultimately be compromised.

Built-in Reliability and Durability

In a climate loaded with problematic (in other words, genuine) equipment, making an application dependable is troublesome and, for more modest applications specifically, costly. (Normally parts stumbling into at least three hypervisor hubs are required.)

OpenStack intends to give natives (for instance, dependable conveyance of messages and sturdy stockpiling) that permit designers to assemble solid applications on top of it. The fundamental assets can be divided among applications and occupants with the goal that the expense is amortized across them, instead of requiring every application to pay the full expense.

The presence of these natives permits another administrations to be less difficult and more versatile, for instance by utilizing possible consistency.

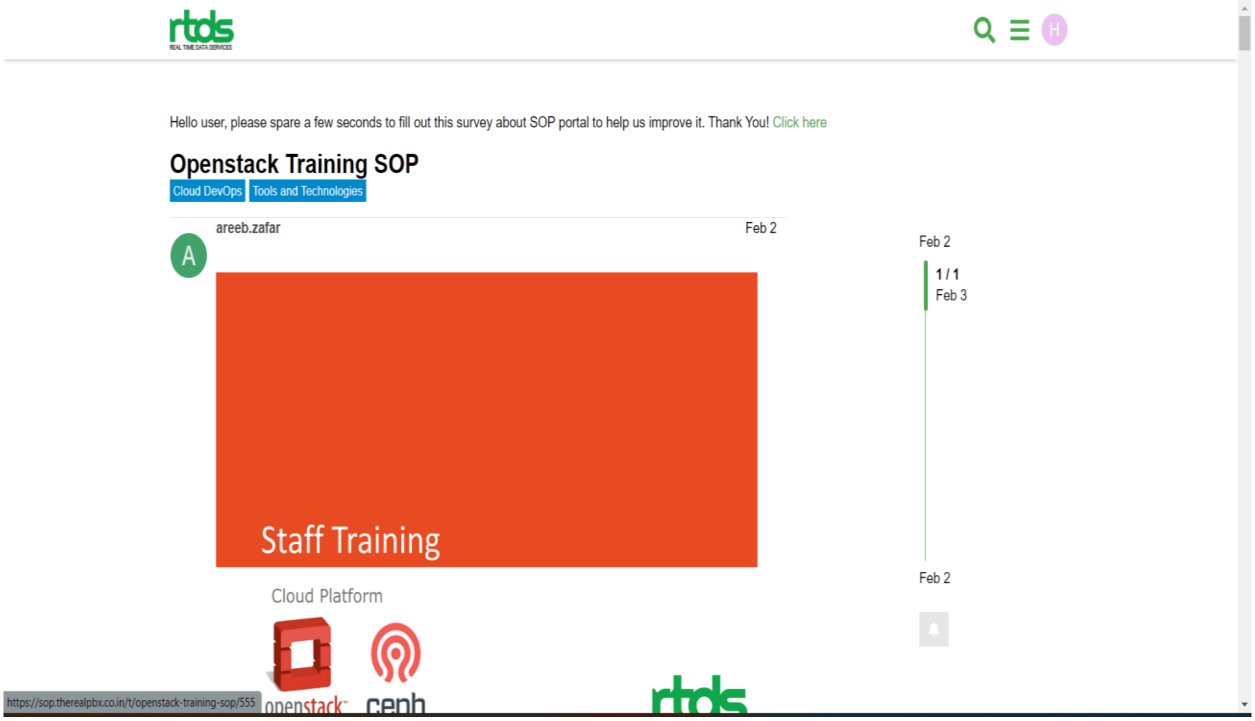
Infinite, Continuous Scaling

OpenStack endeavors to furnish application engineers with interfaces that permit them, on a fundamental level, to scale productively from tiny to exceptionally enormous responsibilities without rearchitecting their applications.

To some extent, this implies permitting customers to utilize limit depending on the situation and offer the hidden assets with different applications and occupants, in inclination to dispensing discrete pieces to specific applications and squandering any abundance limit inside the lumps that they don't use.

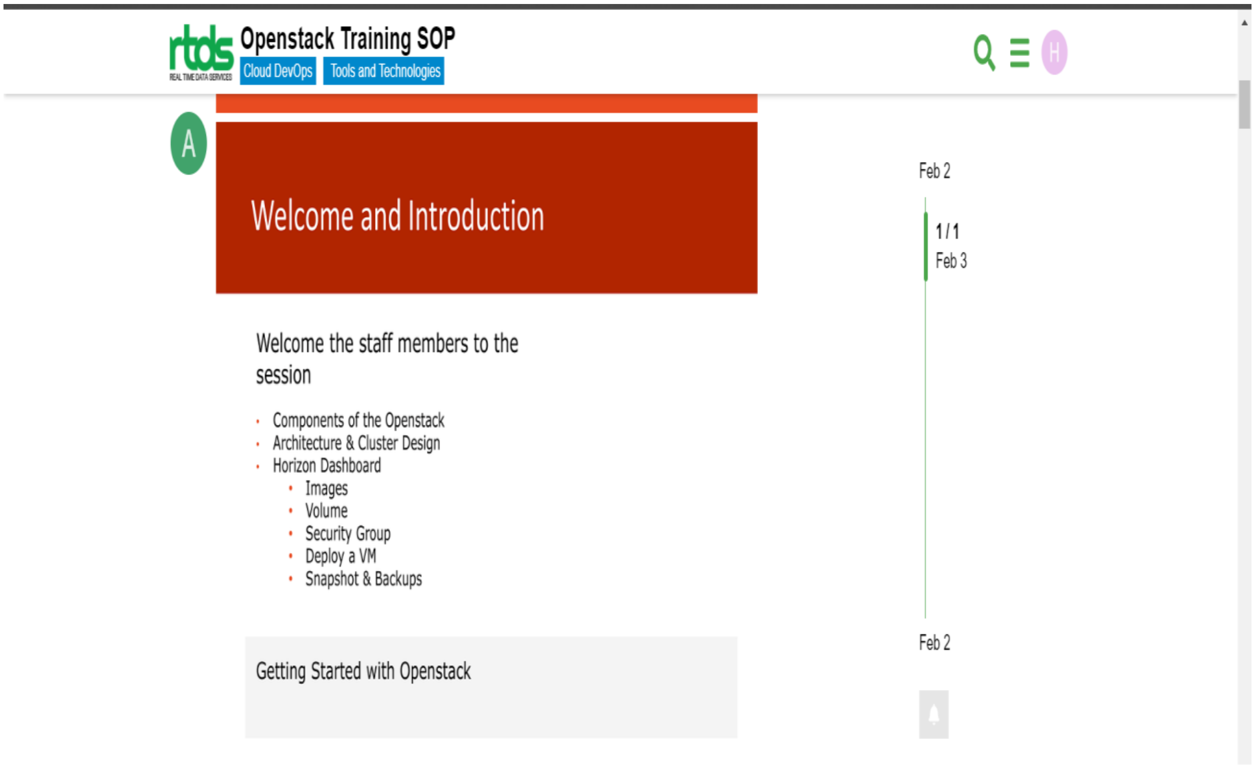
CHAPTER-2

Openstack training SOP



Hello user, please spare a few seconds to fill out this survey about SOP portal to help us improve it. Thank You! Click here

Openstack Training SOP

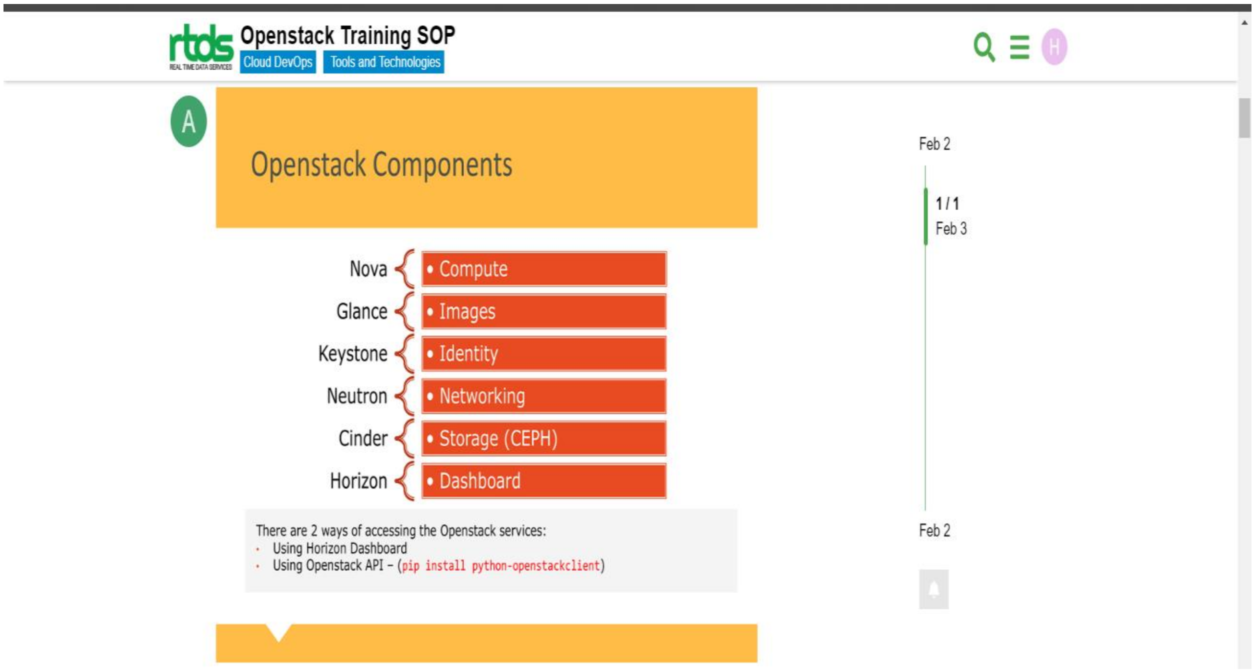


Welcome and Introduction

Welcome the staff members to the session

- Components of the Openstack
- Architecture & Cluster Design
- Horizon Dashboard
  - Images
  - Volume
  - Security Group
  - Deploy a VM
  - Snapshot & Backups

Getting Started with Openstack



Openstack Components

Nova	Compute
Glance	Images
Keystone	Identity
Neutron	Networking
Cinder	Storage (CEPH)
Horizon	Dashboard

There are 2 ways of accessing the Openstack services:

- Using Horizon Dashboard
- Using Openstack API – (`pip install python-openstackclient`)

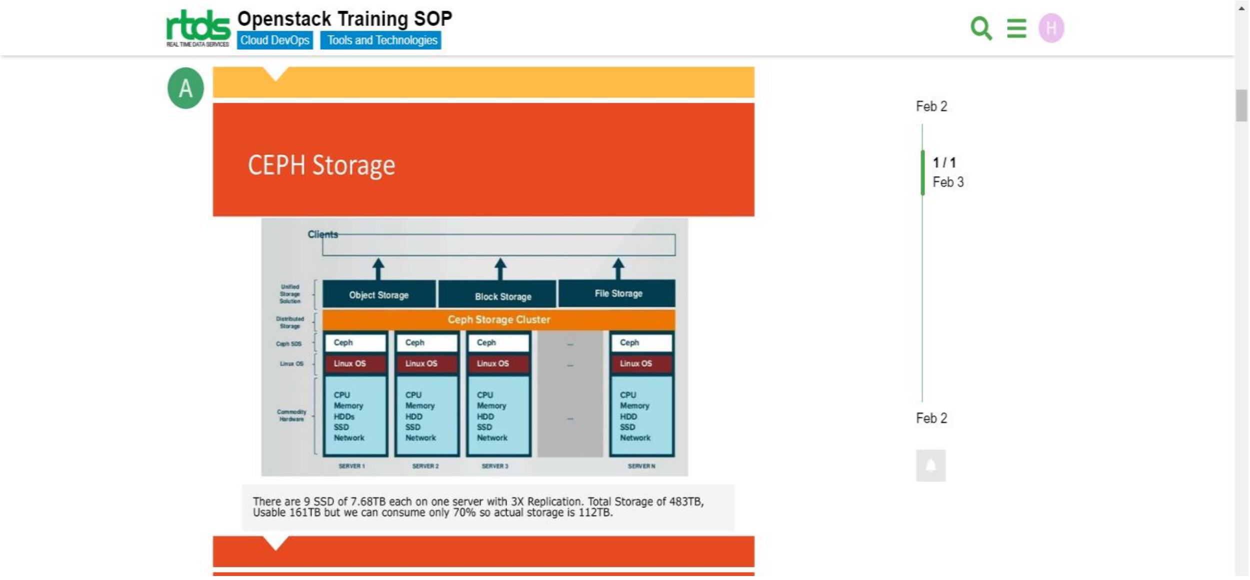
-Nova is the OpenStack project that gives a method for provisioning register examples (otherwise known as virtual servers). Nova upholds making virtual machines, exposed metal servers (using amusing), and has restricted help for framework compartments.

-Look is a picture administration that permits clients to find, recover, and register VM (virtual machine) pictures and compartment pictures, which can involve Swift or Ceph as its genuine stockpiling backend (for example not broadly useful article stockpiling).

- Neutron is an OpenStack venture to give "organizing as a help" between interface gadgets (e.g., vNICs) oversaw by other Openstack administrations (e.g., nova).

-Soot is open source programming intended to make and deal with a help that gives tireless information stockpiling to distributed computing applications. Soot is the code name for the OpenStack Block Storage project

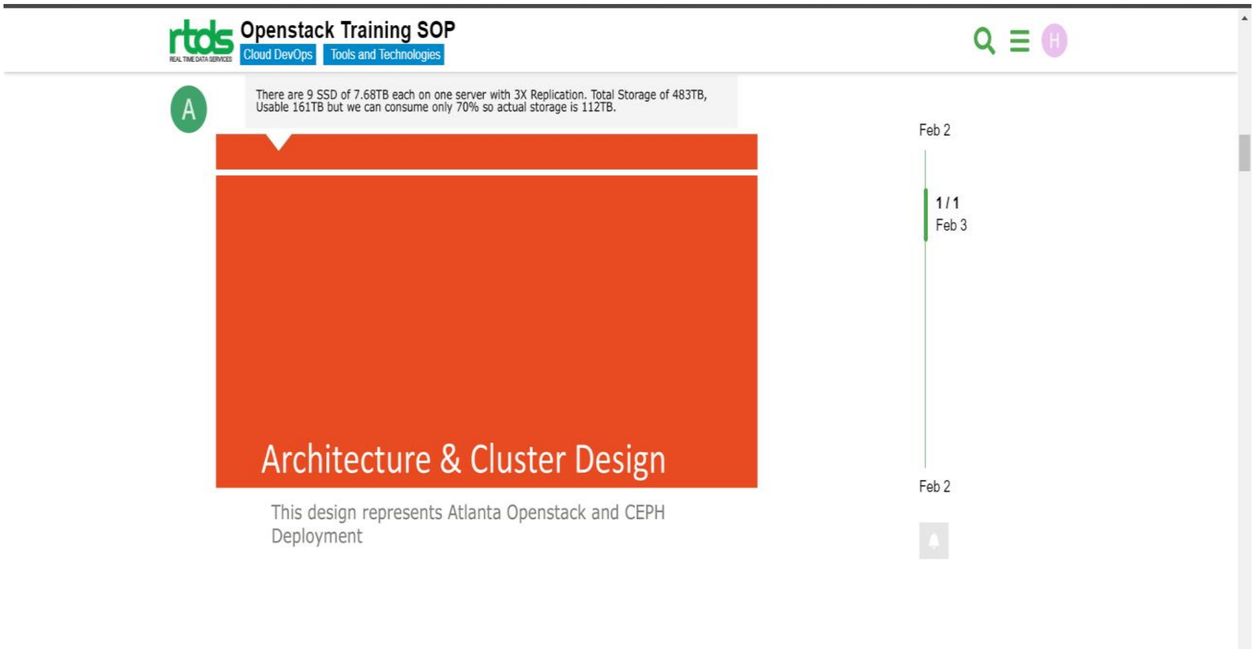
- Horizon is the sanctioned execution of Openstack's Dashboard, which gives an online UI to OpenStack administrations including Nova, Swift, Keystone, and so forth. Kindly see Introducing Horizon for an exhaustive glance at what Horizon is and what the points of the venture are.



CEPH Storage

There are 9 SSD of 7.68TB each on one server with 3X Replication. Total Storage of 483TB, Usable 161TB but we can consume only 70% so actual storage is 112TB.

- Ceph capacity is a product characterized capacity arrangement that conveys information across bunches of capacity assets. It is a shortcoming open minded and scale-out capacity framework, where numerous Ceph stockpiling hubs (servers) participate to introduce a solitary stockpiling framework that can hold numerous petabytes (1PB = 1,000 TB = a million GB) of information.



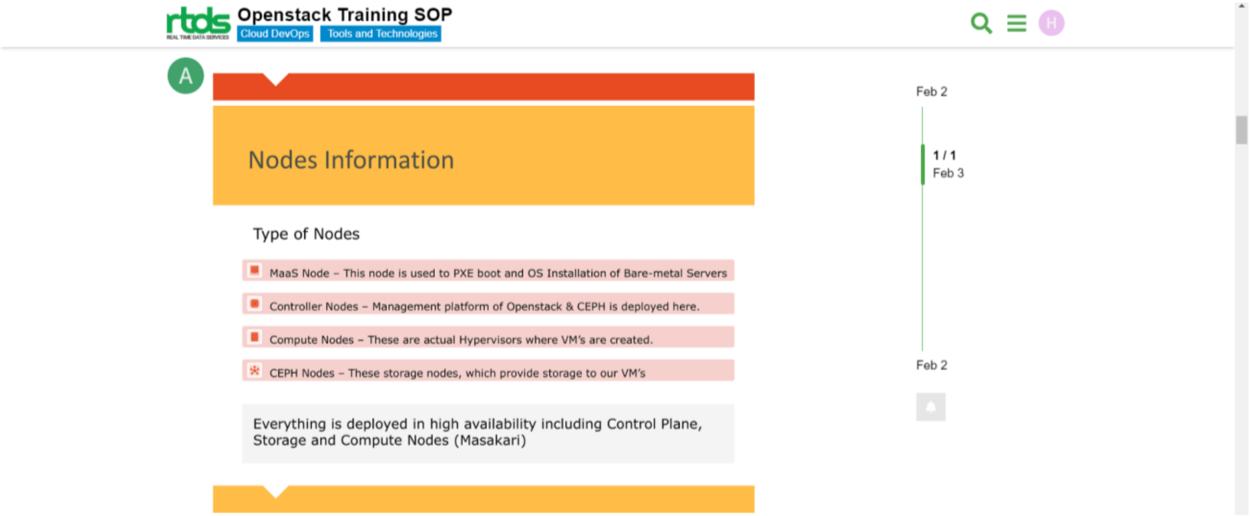
Openstack Training SOP

There are 9 SSD of 7.68TB each on one server with 3X Replication. Total Storage of 483TB, Usable 161TB but we can consume only 70% so actual storage is 112TB.

Architecture & Cluster Design

This design represents Atlanta Openstack and CEPH Deployment

- The Architecture Design Guide gives data on arranging and planning an OpenStack cloud. It makes sense of center ideas, cloud engineering plan prerequisites, and the plan models of key parts and administrations in an OpenStack cloud. The aide additionally depicts five normal cloud use cases.



Openstack Training SOP

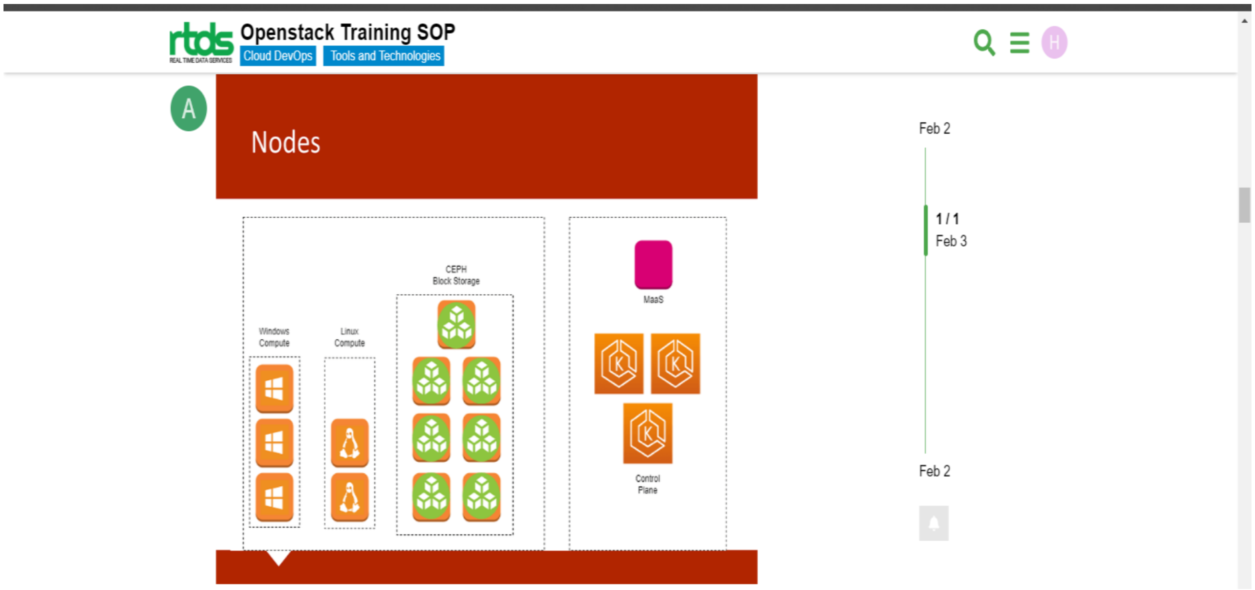
Nodes Information

Type of Nodes

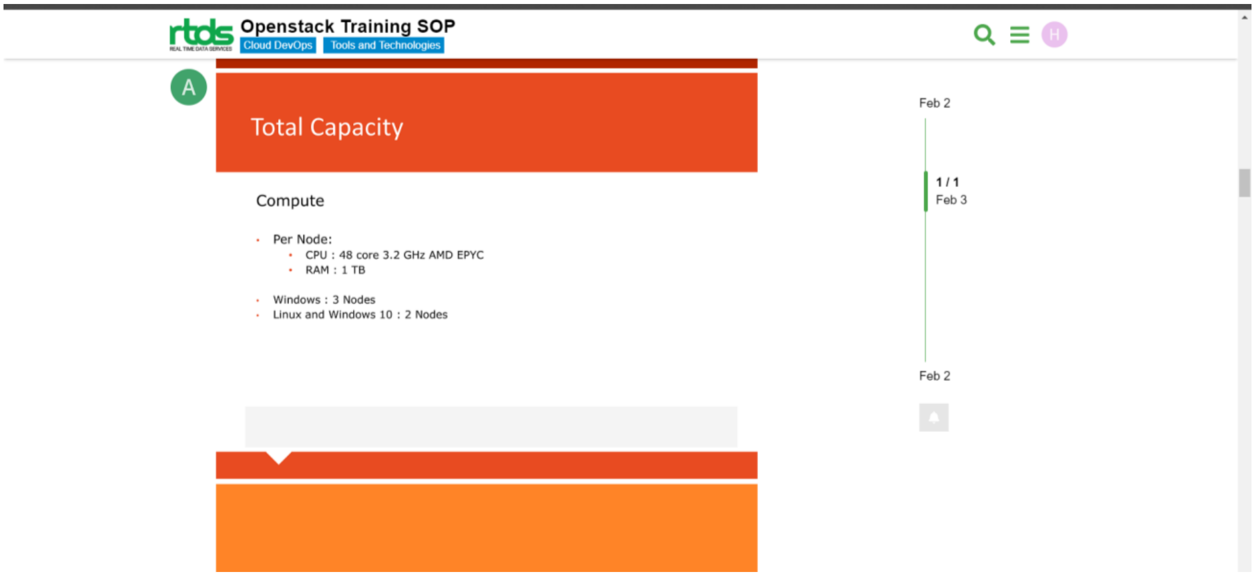
- MaaS Node – This node is used to PXE boot and OS Installation of Bare-metal Servers
- Controller Nodes – Management platform of Openstack & CEPH is deployed here.
- Compute Nodes – These are actual Hypervisors where VM's are created.
- CEPH Nodes – These storage nodes, which provide storage to our VM's

Everything is deployed in high availability including Control Plane, Storage and Compute Nodes (Masakari)

The Architecture Design Guide gives data on arranging and planning an OpenStack cloud. It makes sense of center ideas, cloud engineering plan prerequisites, and the plan models of key parts and administrations in an OpenStack cloud. The aide additionally depicts five normal cloud use cases.



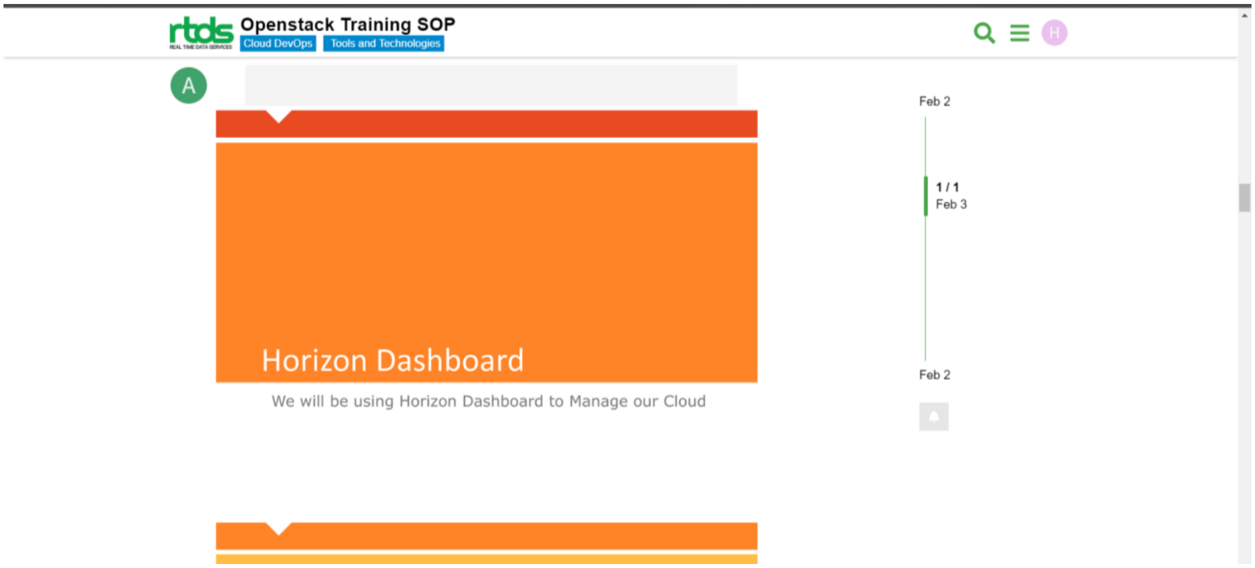
A regulator hub is a framework running Oracle Linux, and is where the majority of the OpenStack administrations are introduced. The term regulator hub is utilized to examine hubs that don't run virtual machine examples. The regulator hubs might have all the non-figure administrations or just some of them. A regulator hub may likewise incorporate the Oracle OpenStack for Oracle Linux tool compartment, which is utilized to play out the sending of OpenStack administrations to different hubs.



Total Capacity

Compute

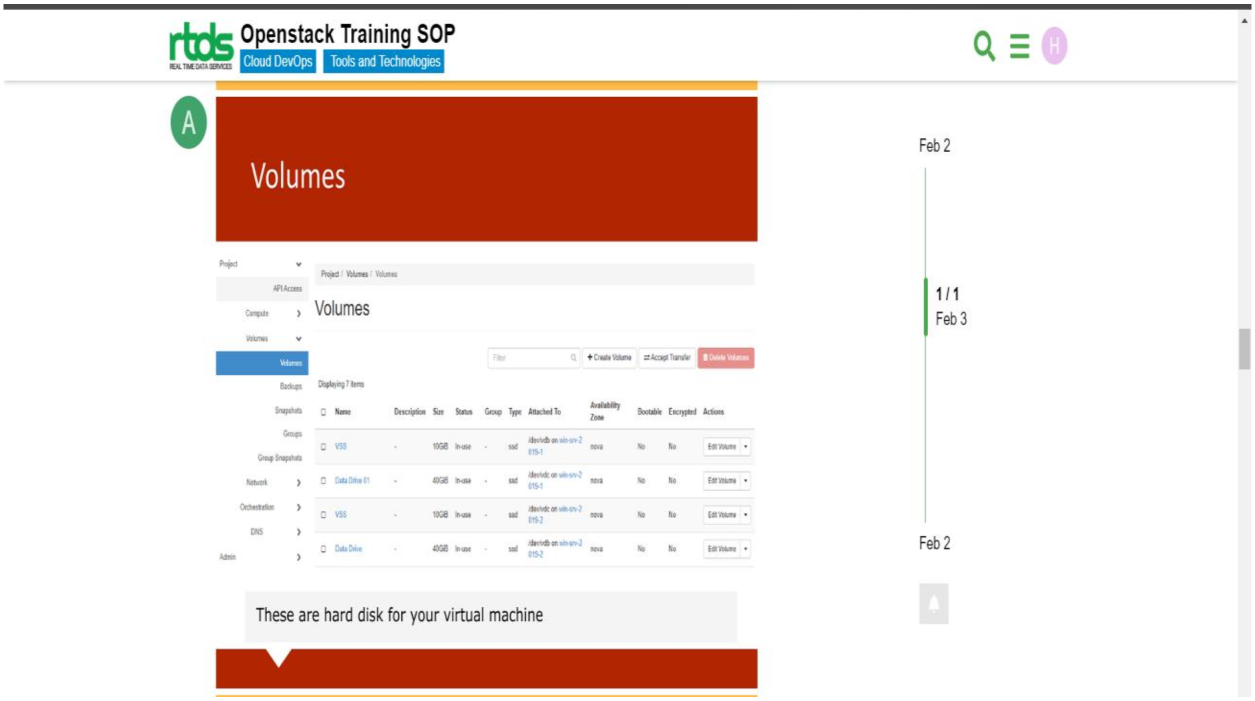
- Per Node:
  - CPU : 48 core 3.2 GHz AMD EPYC
  - RAM : 1 TB
- Windows : 3 Nodes
- Linux and Windows 10 : 2 Nodes



Horizon Dashboard

We will be using Horizon Dashboard to Manage our Cloud

Horizon is the standard execution of OpenStack's Dashboard, which gives an online UI to OpenStack administrations including Nova, Swift, Keystone, and so on.



Volumes

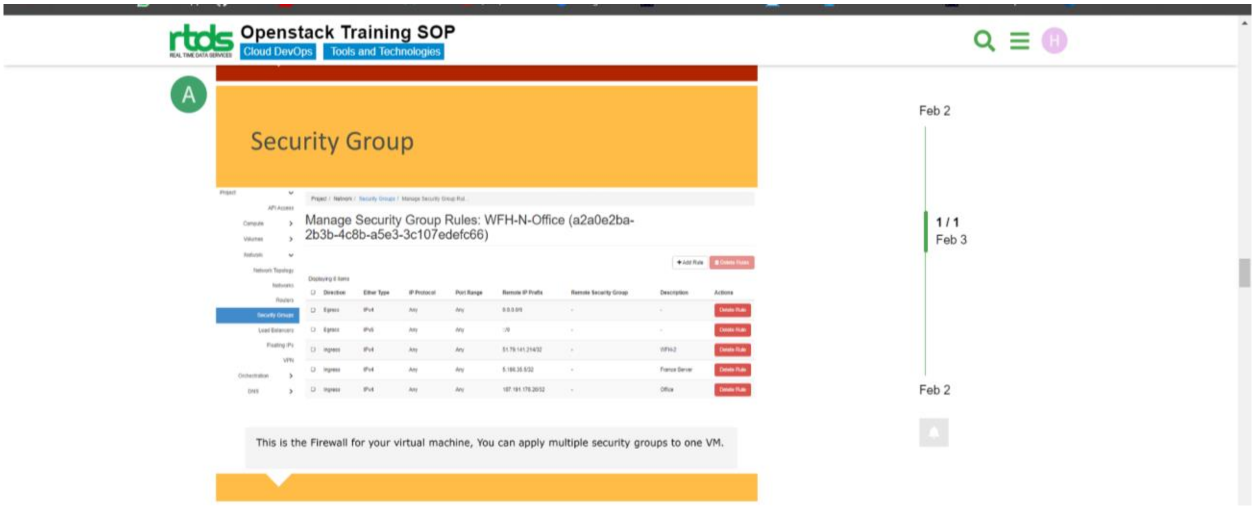
Volumes

Displaying 7 items

□	Name	Description	Size	Status	Group	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions
□	VSS	-	10GiB	In-use	-	ssd	/dev/vdb on win-srv-2019-1	nova	No	No	Edit Volume ▾
□	Data Drive 01	-	40GiB	In-use	-	ssd	/dev/vdc on win-srv-2019-1	nova	No	No	Edit Volume ▾
□	VSS	-	10GiB	In-use	-	ssd	/dev/vdc on win-srv-2019-2	nova	No	No	Edit Volume ▾
□	Data Drive	-	40GiB	In-use	-	ssd	/dev/vdb on win-srv-2019-2	nova	No	No	Edit Volume ▾

These are hard disk for your virtual machine

- Utilize the openstack client orders to make and oversee volumes. This model makes a my-new volume in view of a picture. List pictures, and note the ID of your desired picture to use for your volume:



Security Group

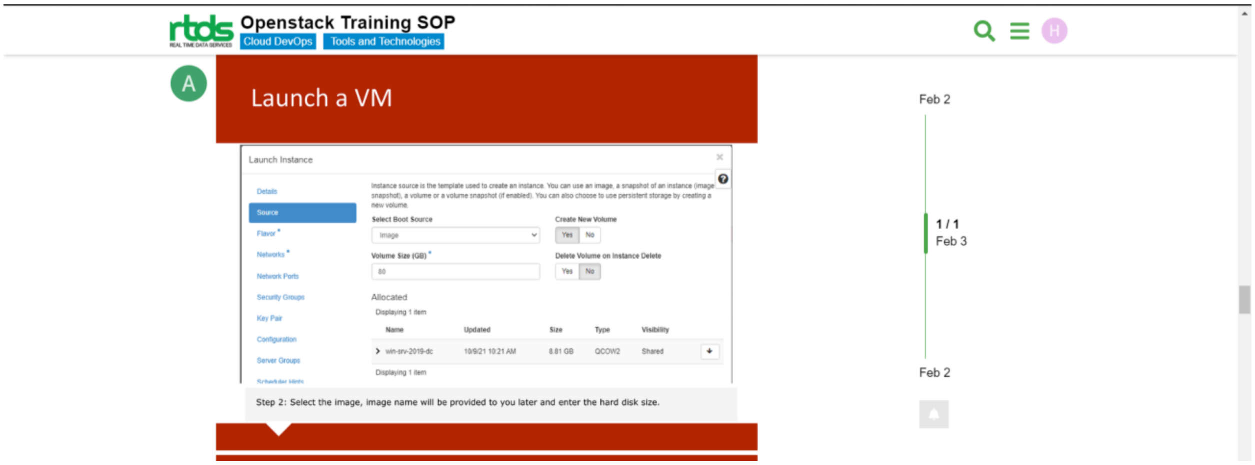
Manage Security Group Rules: WFH-N-Office (a2a0e2ba-2b3b-4c8b-a5e3-3c107edefc66)

Displaying 5 items

□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
□	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
□	Egress	IPv6	Any	Any	::/0	-	-	Delete Rule
□	Ingress	IPv4	Any	Any	51.79.141.214/32	-	WFH-2	Delete Rule
□	Ingress	IPv4	Any	Any	5.196.35.5/32	-	France Server	Delete Rule
□	Ingress	IPv4	Any	Any	107.191.176.20/32	-	Office	Delete Rule

This is the Firewall for your virtual machine, You can apply multiple security groups to one VM.

- OpenStack security bunches are IP channels allocated to virtual ports. A security bunch contains a named rundown of rules, which characterizes the traffic that is allowed to pass. It adjusts a default deny strategy, so traffic that don't match any of the principles is dropped.



Launch a VM

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Create New Volume: [✓ Yes] No

Volume Size (GB) *: 80

Delete Volume on Instance Delete: Yes [✓ No]

Allocated

Displaying 1 item

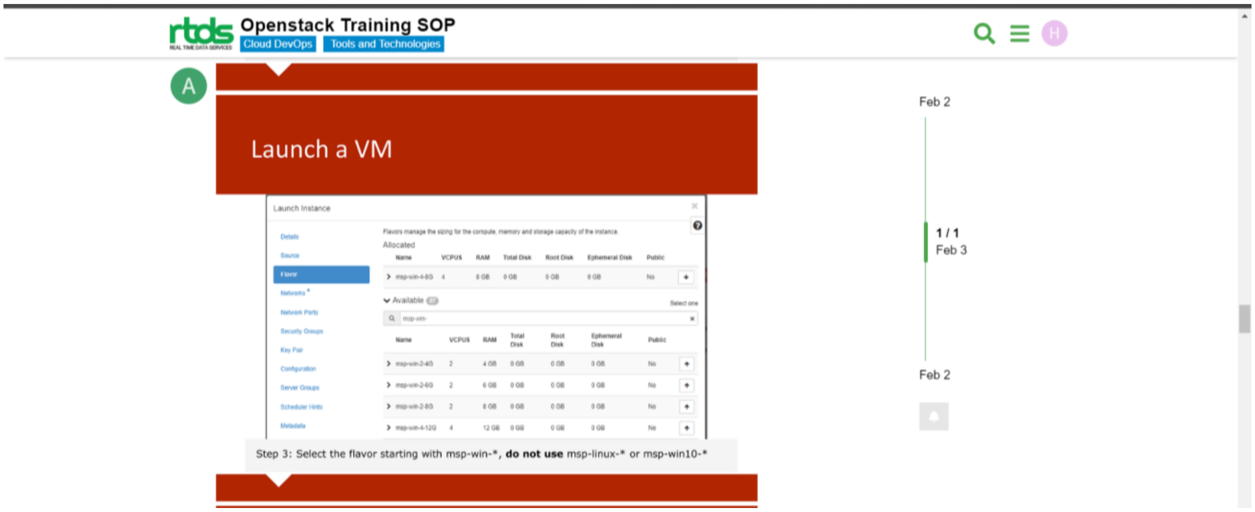
Name	Updated	Size	Type	Visibility
win-srv-2019-dc	10/9/21 10:21 AM	8.81 GB	QCOW2	Shared

Displaying 1 item

Step 2: Select the image, image name will be provided to you later and enter the hard disk size.

- One of the "Four Opens" which outline the reason for an OpenStack project is Open

Source. OpenStack projects don't deliver "open center" programming, they rather produce simply open source programming. Furthermore, the product is delivered with a local area and patron acknowledged permit.



Launch a VM

Launch Instance

Flavors manage the sizing for the compute, memory and storage capacity of the instance.

Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
msp-win-4-8G	4	8 GB	0 GB	0 GB	0 GB	No

Available

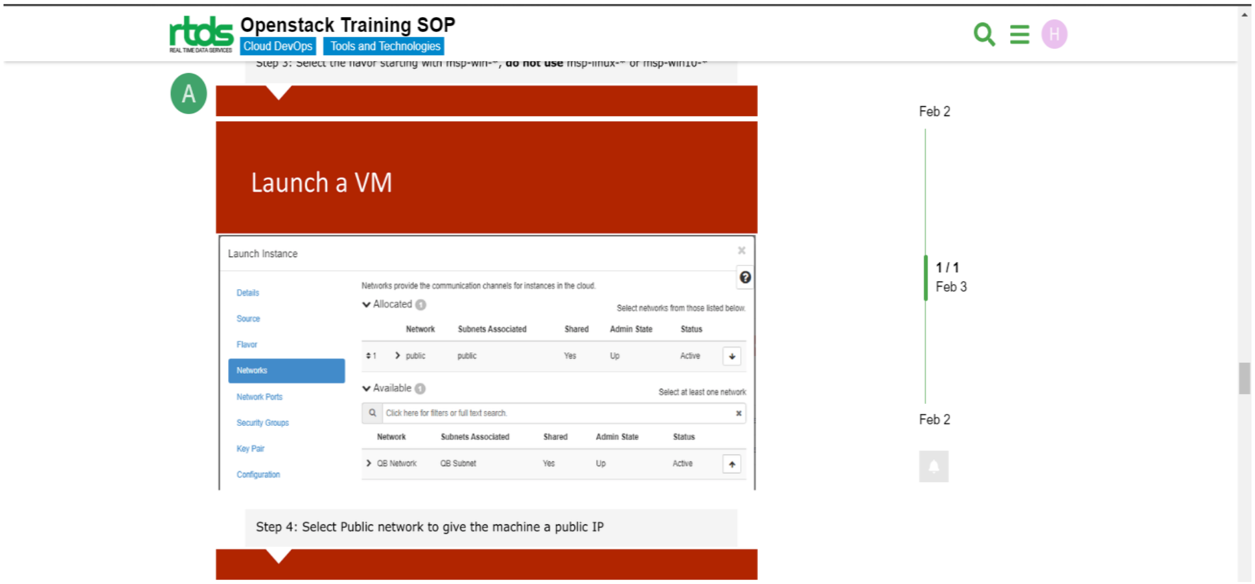
Select one

msp-win-

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
msp-win-2-4G	2	4 GB	0 GB	0 GB	0 GB	No
msp-win-2-6G	2	6 GB	0 GB	0 GB	0 GB	No
msp-win-2-8G	2	8 GB	0 GB	0 GB	0 GB	No
msp-win-4-12G	4	12 GB	0 GB	0 GB	0 GB	No

Step 3: Select the flavor starting with msp-win-*, **do not use** msp-linux-* or msp-win10-*

In OpenStack, flavors characterize the register, memory, and capacity limit of nova figuring occasions. To lay it out plainly, a flavor is an accessible equipment setup for a server. It characterizes the size of a virtual server that can be sent off.



Openstack Training SOP

Step 3: Select the flavor starting with msp-win-*, **do not use** msp-linux-* or msp-win10-*

Launch a VM

Launch Instance

Networks provide the communication channels for instances in the cloud.

Allocated 1

Select networks from those listed below.

	Network	Subnets Associated	Shared	Admin State	Status	
1	public	public	Yes	Up	Active	↓

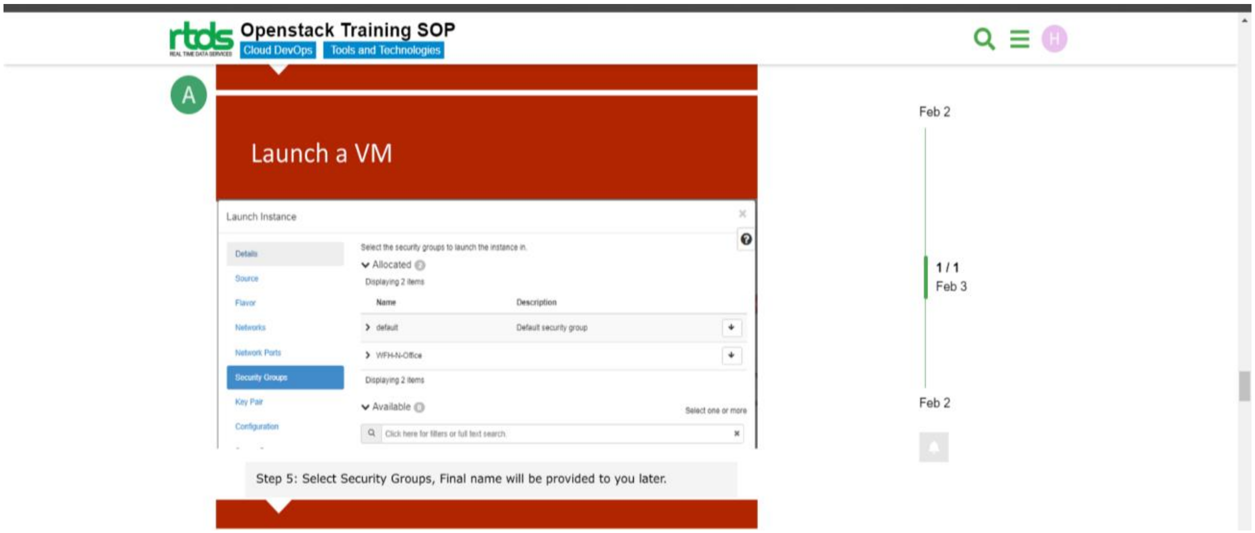
Available 1

Select at least one network

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status	
QB Network	QB Subnet	Yes	Up	Active	↑

Step 4: Select Public network to give the machine a public IP



Launch a VM

Launch Instance

Select the security groups to launch the instance in.

Allocated

Displaying 2 items

Name	Description
default	Default security group
WFH-N-Office	

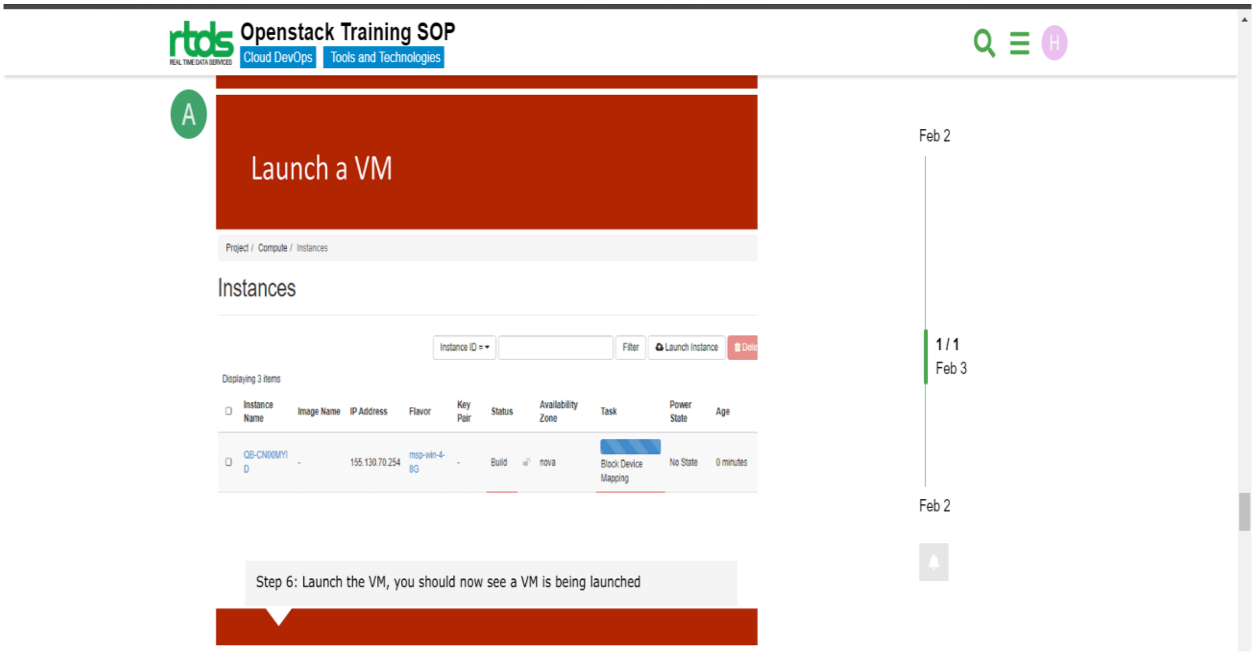
Displaying 2 items

Available

Select one or more

Click here for filters or full text search.

Step 5: Select Security Groups, Final name will be provided to you later.



Launch a VM

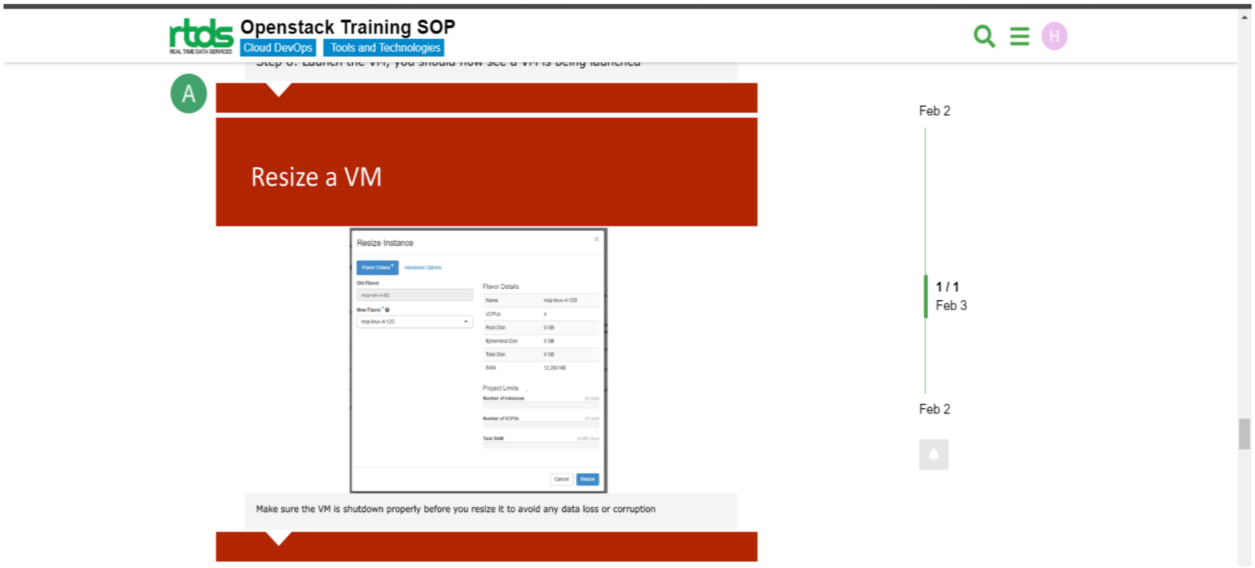
Project / Compute / Instances

Instances

Displaying 3 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age
□	QB-CN00MYID	-	155.130.70.254	msp-win-4-8G	-	Build	nova	Block Device Mapping	No State	0 minutes

Step 6: Launch the VM, you should now see a VM is being launched



Openstack Training SOP

Cloud DevOps Tools and Technologies

Step 6: Launch the VM, you should now see a VM is being launched

Resize a VM

Resize Instance

Flavor Choice * Advanced Options

Old Flavor: msp-win-4-8G

New Flavor *: msp-linux-4-12G

Flavor Details

Name	msp-linux-4-12G
VCPUs	4
Root Disk	0 GB
Ephemeral Disk	0 GB
Total Disk	0 GB
RAM	12,288 MB

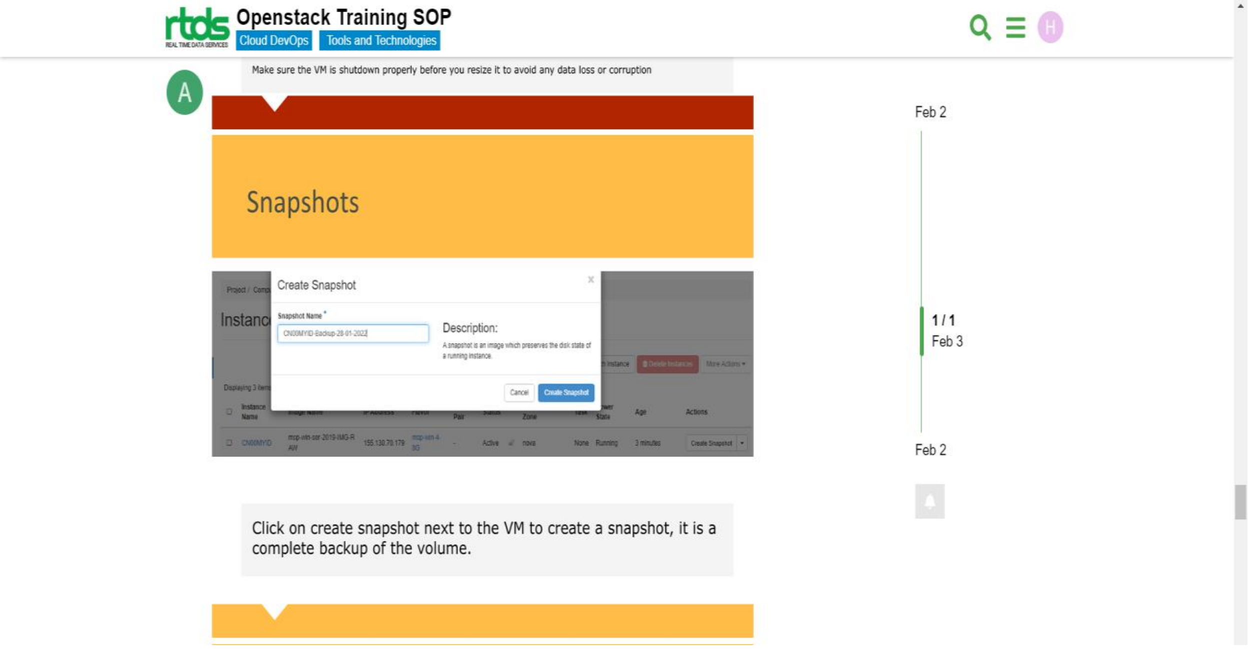
Project Limits

Number of Instances — of Used

Number of VCPUs — of Used

Total RAM — of MB Used

Make sure the VM is shutdown properly before you resize it to avoid any data loss or corruption



Openstack Training SOP

Cloud DevOps Tools and Technologies

Make sure the VM is shutdown properly before you resize it to avoid any data loss or corruption

Snapshots

Create Snapshot

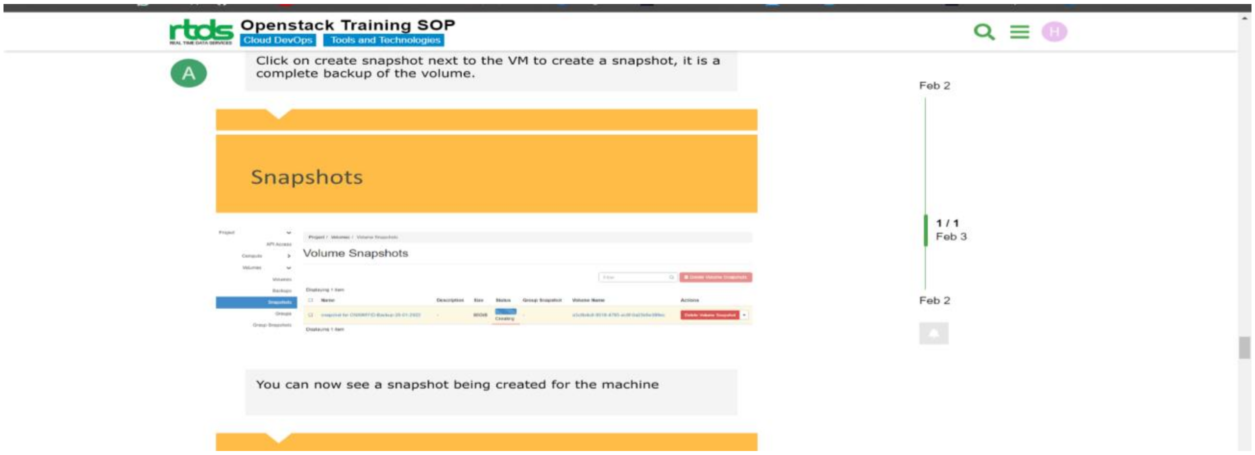
Snapshot Name *: CN00MYID-Backup-28-01-2022

Description:

A snapshot is an image which preserves the disk state of a running instance.

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status		Availability Zone	Task	Power State	Age	Actions
CN00MYID	msp-win-ser-2019-IMG-R AW	155.130.70.179	msp-win-4-8G	-	Active		nova	None	Running	3 minutes	Create Snapshot

Click on create snapshot next to the VM to create a snapshot, it is a complete backup of the volume.



Click on create snapshot next to the VM to create a snapshot, it is a complete backup of the volume.

Snapshots

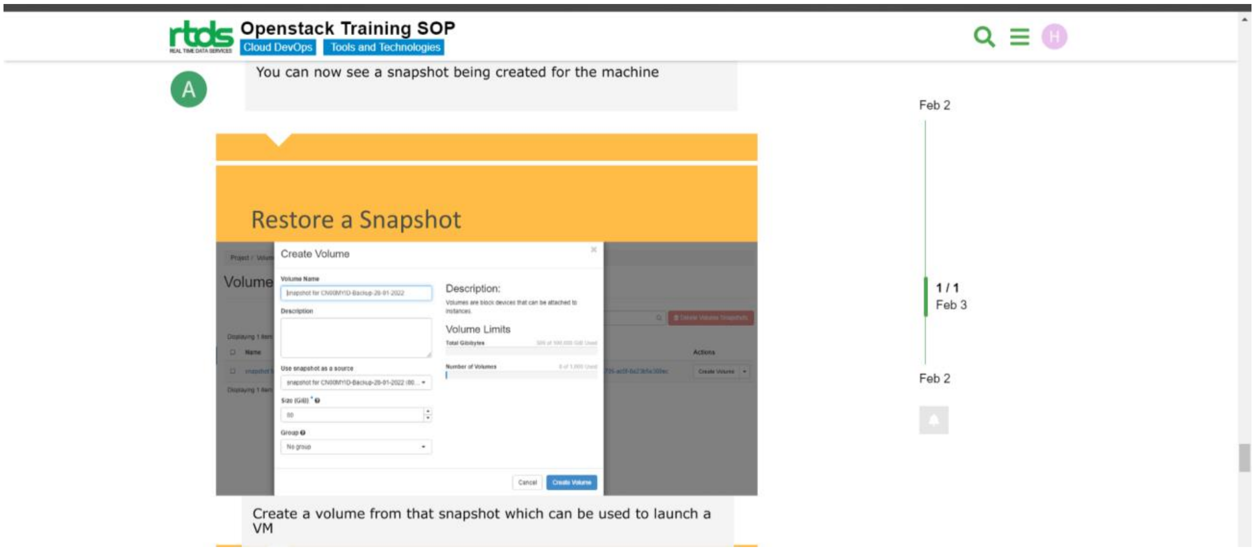
Volume Snapshots

Displaying 1 item

□	Name	Description	Size	Status	Group Snapshot	Volume Name	Actions
□	snapshot for CN00MYID-Backup-28-01-2022	-	80GiB	Creating	-	[illegible]	Delete Volume Snapshot

Displaying 1 item

You can now see a snapshot being created for the machine



You can now see a snapshot being created for the machine

Restore a Snapshot

Create Volume

Volume Name: snapshot for CN00MYID-Backup-28-01-2022

Description:

Use snapshot as a source: snapshot for CN00MYID-Backup-28-01-2022 (80...

Size (GiB) *: 80

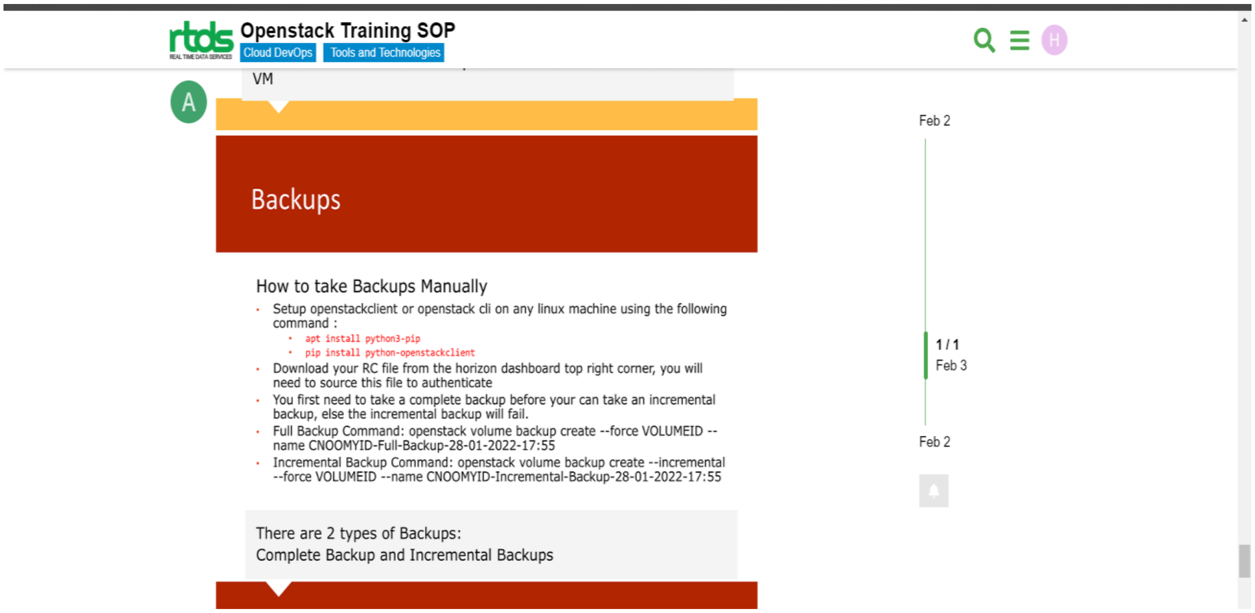
Group: No group

Description:

Volumes are block devices that can be attached to instances.

Volume Limits

Create a volume from that snapshot which can be used to launch a VM

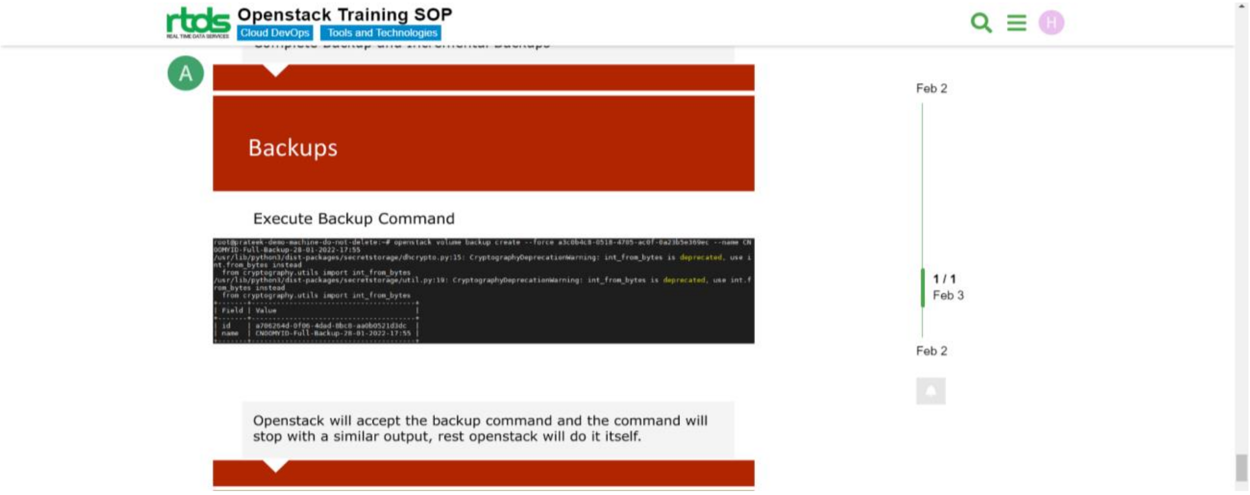


Backups

How to take Backups Manually

- Setup openstackclient or openstack cli on any linux machine using the following command :
  - `apt install python3-pip`
  - `pip install python-openstackclient`
- Download your RC file from the horizon dashboard top right corner, you will need to source this file to authenticate
- You first need to take a complete backup before your can take an incremental backup, else the incremental backup will fail.
- Full Backup Command: openstack volume backup create --force VOLUMEID --name CNOOMYID-Full-Backup-28-01-2022-17:55
- Incremental Backup Command: openstack volume backup create --incremental --force VOLUMEID --name CNOOMYID-Incremental-Backup-28-01-2022-17:55

There are 2 types of Backups:
Complete Backup and Incremental Backups

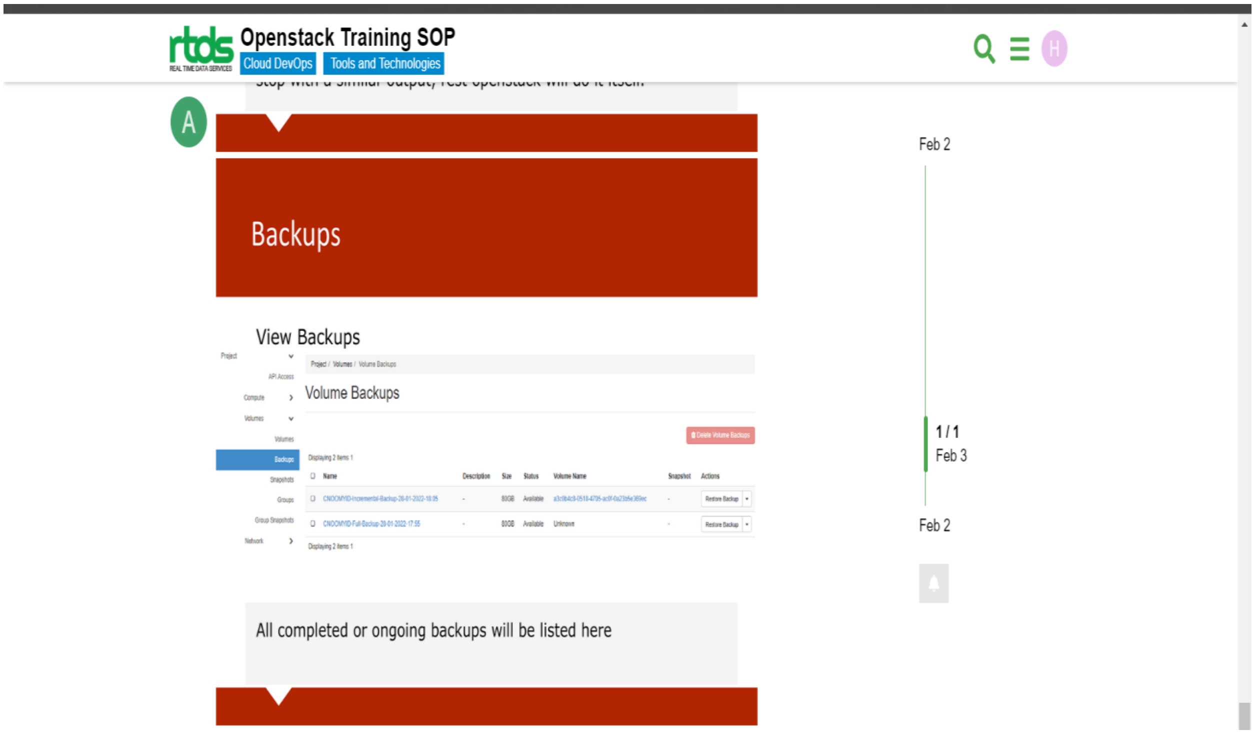


Backups

Execute Backup Command

```
root@prateek-demo-machine-do-not-delete:~# openstack volume backup create --force a3c0b4c8-0518-4705-ac0f-0a23b5e369ec --name CN
OOMYID-Full-Backup-28-01-2022-17:55
/usr/lib/python3/dist-packages/secretstorage/dhcrypto.py:15: CryptographyDeprecationWarning: int_from_bytes is deprecated, use i
nt.from_bytes instead
  from cryptography.utils import int_from_bytes
/usr/lib/python3/dist-packages/secretstorage/util.py:19: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.f
rom_bytes instead
  from cryptography.utils import int_from_bytes
+-------+--------------------------------------+
| Field | Value                                |
+-------+--------------------------------------+
| id    | a706264d-0f06-4dad-8bc8-aa0b0521d3dc |
| name  | CNOOMYID-Full-Backup-28-01-2022-17:55 |
+-------+--------------------------------------+
```

Openstack will accept the backup command and the command will stop with a similar output, rest openstack will do it itself.



Openstack Training SOP

Stop with a similar output, rest openstack will do it itself.

Backups

View Backups

Project / Volumes / Volume Backups

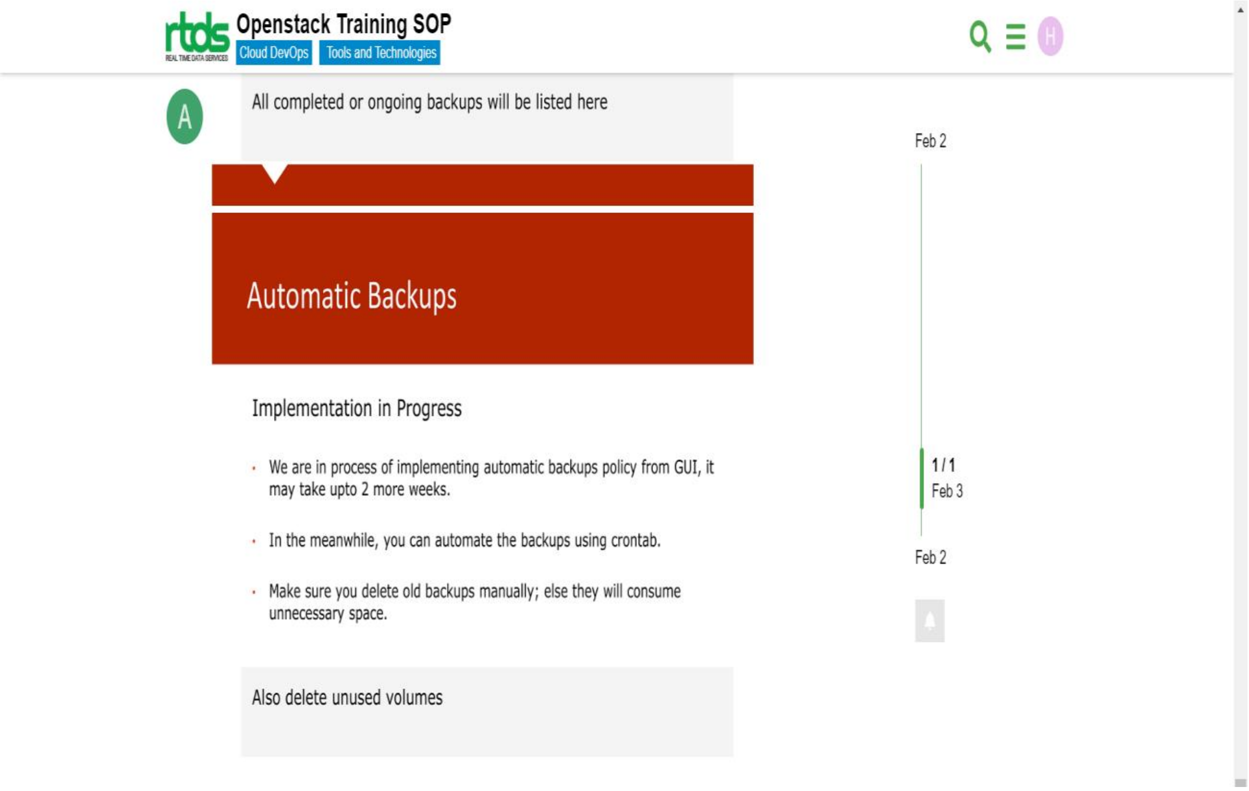
Volume Backups

Displaying 2 items 1

□	Name	Description	Size	Status	Volume Name	Snapshot	Actions
□	CNOOMYID-Incremental-Backup-28-01-2022-18:05	-	80GB	Available	a3c0b4c8-0518-4705-ac0f-0a23b5e369ec	-	Restore Backup ▾
□	CNOOMYID-Full-Backup-28-01-2022-17:55	-	80GB	Available	Unknown	-	Restore Backup ▾

Displaying 2 items 1

All completed or ongoing backups will be listed here



All completed or ongoing backups will be listed here

Automatic Backups

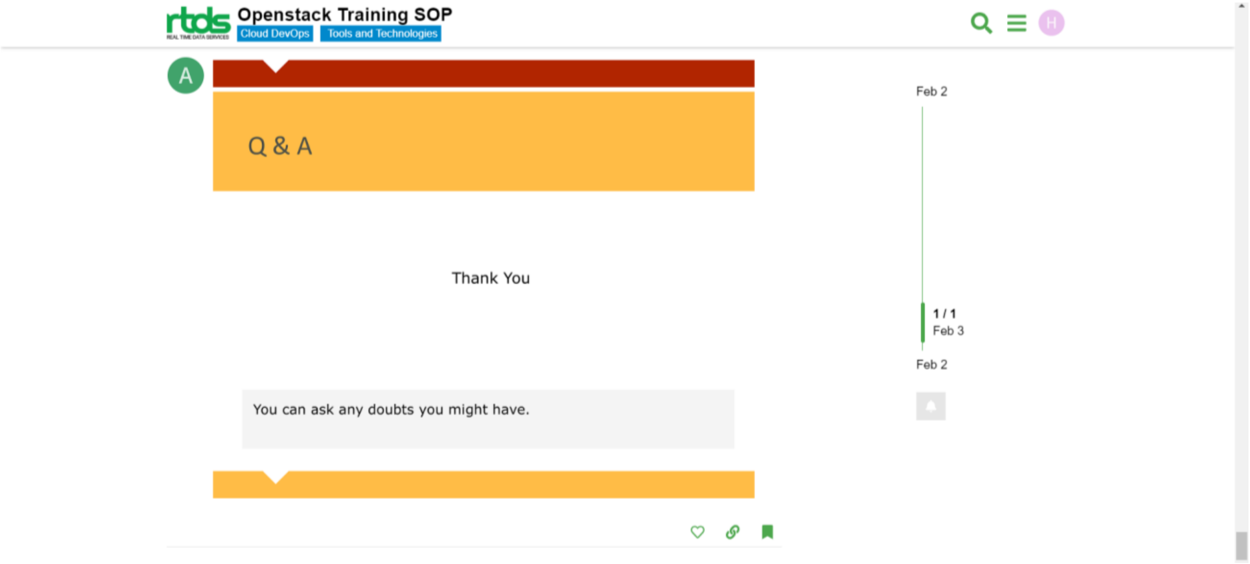
Implementation in Progress

- We are in process of implementing automatic backups policy from GUI, it may take upto 2 more weeks.
- In the meanwhile, you can automate the backups using crontab.
- Make sure you delete old backups manually; else they will consume unnecessary space.

Also delete unused volumes

The figure part of OpenStack, nova, has different choices to make pictures (previews) of an occurrence. One is with the nova picture make order. This makes a moment preview and transfers that to Glance, the picture stockpiling part.

The subsequent choice is nova reinforcement. This is practically equivalent to picture make, with the expansion of revolution of the reinforcement depictions. There are an additional two boundaries, backup_type and revolution. The documentation is a piece inadequate on this, however the programming interface and the source code give more detail.The reinforcements are typical look pictures. In view of the backup_type a set number of pictures is saved. When there are more pictures of that sort then determined in pivot, the most established is erased. On the off chance that we make a reinforcement consistently with backup_type set to everyday and turn set to 7, on the 8'th day, the most established picture will be consequently eliminated (on production of the new reinforcement).



Q & A

Thank You

You can ask any doubts you might have.

OpenStack Services

An OpenStack deployment contains a number of components providing APIs to access infrastructure resources. This page lists the various services that can be deployed to provide such resources to cloud end users.

Compute

Nova - Computer service

Zun- Containers service

Storage

Swift -Object store

Cinder- Block storage

Networking

Neutron- networking

Octavia -Load balancer

DESIGNATE -DNS service

Client tools

Client-side tools and libraries for interacting with OpenStack APIs.

CLIs- Command line interface for all openstack services

SDKs- official-python SDK for openstack APIs

Deployment tools

Openstack Ansible Ansible playbooks to deploy Opentstack

Openstack chef- chef cookbooks to build operate and consume openstack

CHAPTER-3

CONCLUSION

Finally, this internship has been a fantastic and gratifying experience. I can honestly state that my time with RTDS was quite useful to me. Needless to say, the technical aspects of my work aren't ideal and might be improved with more time. As someone with little prior expertise with cloud & devops , I believe the time I spent learning and comprehending was well spent, as it aided in the creation of a completely functional app service. Time management and self-motivation are two of the most important principles I've learnt.

Future Work

As a result, any features and technologies I develop in the future will have a huge impact on small businesses by technologically elevating them and supporting them in easily constructing smart stores through touch. As a consequence, further work will be done in the future to create a better self-checkout experience for consumers, as well as a more accurate and up-to-date point of sale application

Mentor’s Review

He performed admirably during training; he is a quick learner who takes less time to comprehend concepts. At the same time, he has exceptional problem-solving skills, making him an invaluable asset to our organization.

References

https://www.openstack.org/

https://sop.therealpbx.co.in/