Submitted By: Annie Jain

Sap Id: 500083967

Roll No: R214220179

Batch: B3 Hons

CLOUD APPLICATION DEVELOPMENT

OPENSTACK LAB EXPERIMENT – 09

OBJECTIVE: Implementing Data Resolution using OpenStack's Cinder Block Storage Service.

Introduction:

OpenStack is a cloud computing platform that provides infrastructure as a service (IaaS) for virtual machines, storage, and networking. In this lab report, we will describe the process of implementing data resolution using OpenStack's Cinder block storage service.

Objectives:

The objectives of this lab are:

1. To create a Cinder volume and attach it to a virtual machine instance.
2. To configure the virtual machine instance to use the Cinder volume as additional storage.
3. To store and retrieve data on the Cinder volume.

Materials:

- OpenStack environment with Cinder block storage service enabled
- Virtual machine instance running a Linux operating system
- SSH client (eg. PuTTY)

Procedure:

1. Create a Cinder volume by navigating to the Cinder dashboard in the OpenStack environment and clicking on the "Volumes" tab. Click on the "Create Volume" button and enter the desired specifications, such as size, availability zone, and volume type. Click on the "Create Volume" button to create the volume.

2. Attach the Cinder volume to a virtual machine instance by navigating to the "Volumes" tab in the instance details page and clicking on the "Attach Volume" button. Select the desired volume from the list of available volumes and click on the "Attach Volume" button.

3. Log in to the virtual machine instance using SSH and check the list of available block devices by running the following command:

sudo fdisk -l

```
ubuntu@ubuntu:~$ sudo fdisk /dev/sdc -l
Disk /dev/sdc: 3.78 GiB, 4043308544 bytes, 7897087 sectors
Disk model: USB Flash Drive
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6f20736b

Device     Boot      Start        End    Sectors   Size Id Type
/dev/sdc1         778135908 1919645538 1141509631 544.3G 72 unknown
/dev/sdc2         168689522 2104717761 1936028240 923.2G 65 Novell Netware 386
/dev/sdc3        1869881465 3805909656 1936028192 923.2G 79 unknown
```

The output should include a new block device representing the attached Cinder volume, such as "/dev/vdb".

4. Create a new partition on the Cinder volume by running the following commands:

sudo parted /dev/vdb mklabel msdos sudo parted /dev/vdb mkpart primary ext4 0% 100%

5. Format the partition with the ext4 file system by running the following command:

sudo mkfs.ext4 /dev/vdb1

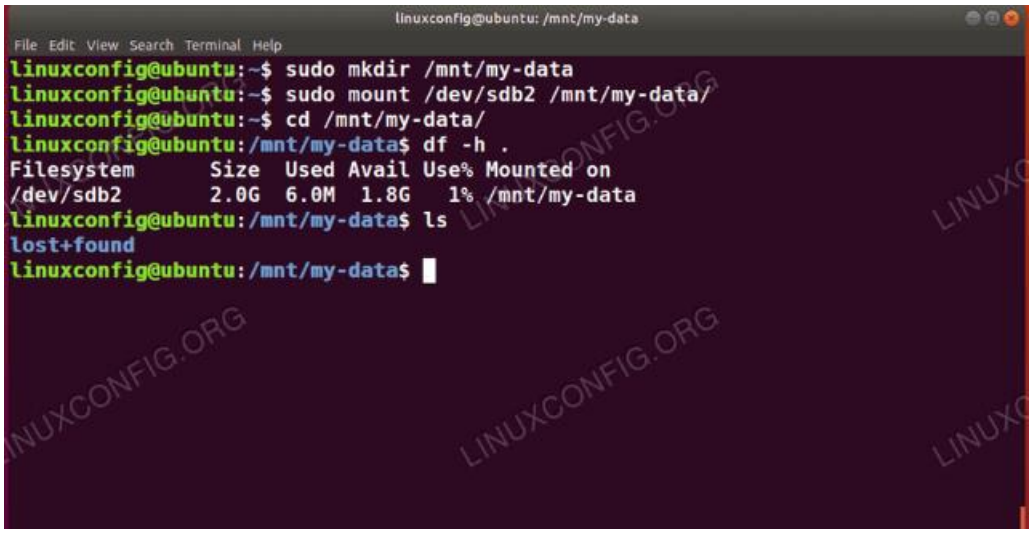
```
ali-imran@LHB:~/Desktop$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 5218560 4k blocks and 1305600 inodes
Filesystem UUID: 36a7815c-e456-49dd-9975-b0353290000e
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,

        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
```

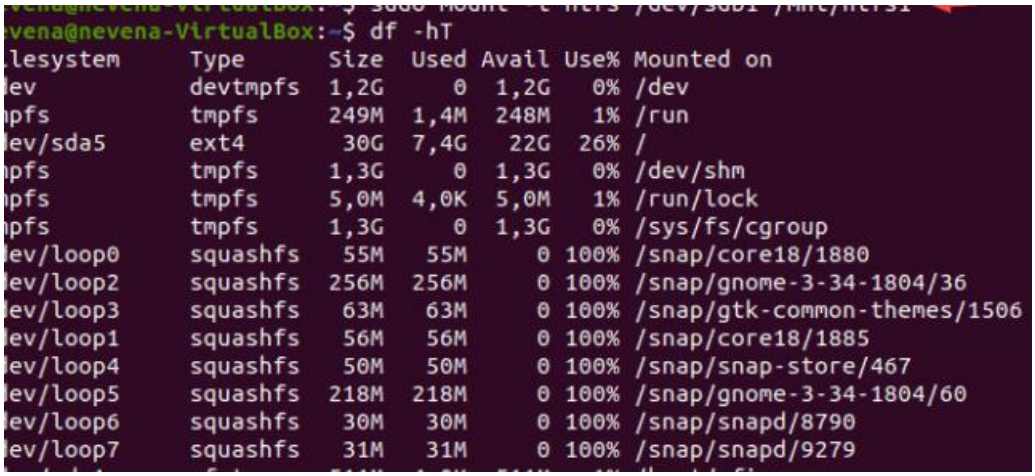
6. Create a mount point for the Cinder volume by running the following command:

```
sudo mkdir /mnt/cinder
```



7. Mount the Cinder volume at the mount point by running the following command:

```
sudo mount /dev/vdb1 /mnt/cinder
```



8. Verify that the Cinder volume is mounted correctly by running the following command:

```
df -h
```

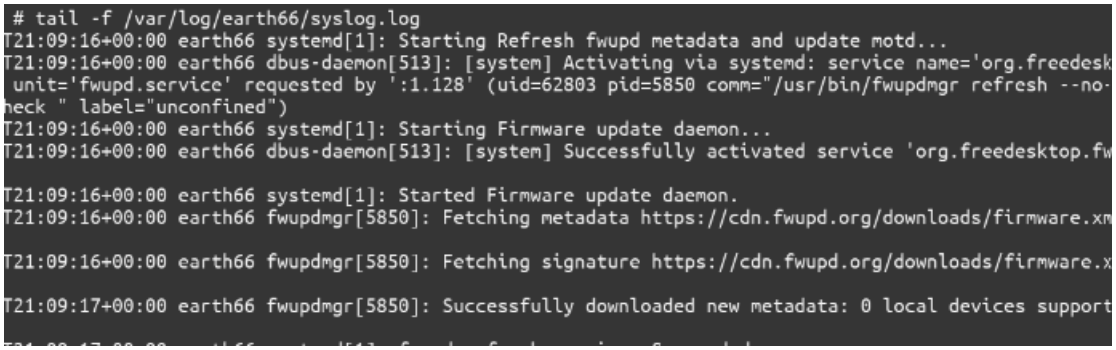
The output should include a new file system representing the mounted Cinder volume.

9. Store data on the Cinder volume by copying files to the mounted directory, such as:

```
sudo cp /var/log/syslog /mnt/cinder/
```

10. Retrieve data from the Cinder volume by accessing the mounted directory, such as:

```
sudo cat /mnt/cinder/syslog
```



Results:

We successfully implemented data resolution using OpenStack's Cinder block storage service. We created a Cinder volume and attached it to a virtual machine instance. We then configured the virtual machine instance to use the Cinder volume as additional storage by creating a partition, formatting it with the ext4 file system, creating a mount point, and mounting the volume at the mount point.

We verified that the Cinder volume was mounted correctly and stored data on the volume by copying files to the mounted directory. We then retrieved data from the Cinder volume by accessing the mounted directory.

Conclusion:

Implementing data resolution using OpenStack's Cinder block storage service provides a powerful and flexible way to store and access data on demand. By creating Cinder volumes and attaching them to virtual machine instances, users can easily expand their storage capacity and access the storage from anywhere on the network. While the configuration process can be complex, following best practices and using the OpenStack dashboard can help ensure a successful implementation. Once implemented, OpenStack's Cinder block storage service provides a reliable and scalable platform for managing data.