



Inspiring Excellence

**CSE-491 Cloud Computing
Assignment- 4**

**Time to Create Own
Storage**

Submitted By:

Mehedi Hasan

Id: 17301046

Submitted To,

Jannatun Noor Mukta

1. Installing Dependencies

Sudo apt-get update

Sudo apt-get install curl gcc memcached rsync sqlite3 xfsprogs \
git-core libffi-dev python-setuptools \
librasurecode-dev libssl-dev

```
mehedi@hasan-17301046: ~  
mehedi@hasan-17301046:~$ sudo apt-get install curl gcc memcached rsync sqlite3 xfsprogs \  
> git-core libffi-dev python-setuptools \  
> librasurecode-dev libssl-dev  
[sudo] password for mehedi:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
gcc is already the newest version (4:5.3.1-1ubuntu1).  
rsync is already the newest version (3.1.1-3ubuntu1.3).  
  
update-initramfs: deferring update (trigger activated)  
Setting up librasurecode1:i386 (1.1.0-3) ...  
Setting up librasurecode-dev (1.1.0-3) ...  
Setting up libffi-dev:i386 (3.2.1-4) ...  
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...  
Processing triggers for ureadahead (0.100.0-19.1) ...  
Processing triggers for systemd (229-4ubuntu21.31) ...  
Processing triggers for initramfs-tools (0.122ubuntu8.17) ...  
update-initramfs: Generating /boot/initrd.img-4.15.0-142-generic
```

After That, Install More Python dependencies

Sudo apt-get install python-coverage python-dev python-nose \
python-xattr python-eventlet \
python-greenlet python-pastedeploy \
python-netifaces python-pip python-dnspython \
python-mock

```
mehedi@hasan-17301046: ~  
mehedi@hasan-17301046:~$ sudo apt-get install python-coverage python-dev python-nose \  
> python-xattr python-eventlet \  
> python-greenlet python-pastedeploy \  
> python-netifaces python-pip python-dnspython \  
> python-mock  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:
```

```
Setting up python-pip-whl (8.1.1-2ubuntu0.6) ...
Setting up python-pip (8.1.1-2ubuntu0.6) ...
Setting up python-scgi (1.13-1.1build1) ...
Setting up python-wheel (0.29.0-1) ...
Setting up python-xattr (0.7.8-3) ...
Setting up libjs-jquery-isonscreen (1.2.0-1) ...
Setting up python-pbr (1.8.0-4ubuntu1) ...
update-alternatives: using /usr/bin/python2-pbr to provide /usr/bin/pbr (pbr) in
auto mode
Setting up python-mock (1.3.0-2.1ubuntu1) ...
```

After Installing dependencies now need to clone python-swift from the Git

git clone <https://github.com/openstack/python-swiftclient.git>

```
mehedi@hasan-17301046: ~
mehedi@hasan-17301046:~$ sudo git clone https://github.com/openstack/python-swif
tclient.git
Cloning into 'python-swiftclient'...
remote: Enumerating objects: 5619, done.
remote: Counting objects: 100% (252/252), done.
remote: Compressing objects: 100% (169/169), done.
remote: Total 5619 (delta 142), reused 174 (delta 82), pack-reused 5367
Receiving objects: 100% (5619/5619), 3.25 MiB | 2.27 MiB/s, done.
Resolving deltas: 100% (3805/3805), done.
Checking connectivity... done.
```

Installing -r requirements

Sudo pip install -r requirements.txt

```
mehedi@hasan-17301046:~/python-swiftclient$ sudo pip install -r requirements.txt

The directory '/home/mehedi/.cache/pip/http' or its parent directory is not owne
d by the current user and the cache has been disabled. Please check the permissi
ons and owner of that directory. If executing pip with sudo, you may want sudo's
-H flag.
The directory '/home/mehedi/.cache/pip' or its parent directory is not owned by
the current user and caching wheels has been disabled. check the permissions and
owner of that directory. If executing pip with sudo, you may want sudo's -H fla
```

Installing Python

Sudo python setup.py install

```
mehedi@hasan-17301046:~/python-swiftclient$ sudo python setup.py install
running install
running build
running build_py
creating build
creating build/lib.linux-i686-2.7
creating build/lib.linux-i686-2.7/swiftclient
copying swiftclient/version.py -> build/lib.linux-i686-2.7/swiftclient
copying swiftclient/utils.py -> build/lib.linux-i686-2.7/swiftclient
copying swiftclient/__init__.py -> build/lib.linux-i686-2.7/swiftclient
copying swiftclient/command_helpers.py -> build/lib.linux-i686-2.7/swiftclient
copying swiftclient/authv1.py -> build/lib.linux-i686-2.7/swiftclient
copying swiftclient/service.py -> build/lib.linux-i686-2.7/swiftclient
```

Now we Need to Cloning Swift from Github as we will install swift

First, i make a folder name pythonSwift

Mkdir pythonSwift

then cloning the git

git clone <https://github.com/openstack/swift.git>

```
mehedi@hasan-17301046:~$ mkdir pythonSwift
mehedi@hasan-17301046:~$ cd pythonSwift
mehedi@hasan-17301046:~/pythonSwift$ git clone https://github.com/openstack/swift.git
Cloning into 'swift'...
remote: Enumerating objects: 92221, done.
remote: Counting objects: 100% (2996/2996), done.
remote: Compressing objects: 100% (929/929), done.
remote: Total 92221 (delta 2294), reused 2717 (delta 2066), pack-reused 89225
Receiving objects: 100% (92221/92221), 62.96 MiB | 9.44 MiB/s, done.
Resolving deltas: 100% (71070/71070), done.
Checking connectivity... done.
```

Need to Check the Git Status

git status

```
mehedi@hasan-17301046:~/pythonSwift$ ls
swift
mehedi@hasan-17301046:~/pythonSwift$ cd swift
mehedi@hasan-17301046:~/pythonSwift/swift$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

As its in Master Branch so need to change it to Stable/train branch

git checkout stable/train

```
mehedi@hasan-17301046:~/pythonSwift/swift$ git checkout stable/train
Branch stable/train set up to track remote branch stable/train from origin.
Switched to a new branch 'stable/train'
mehedi@hasan-17301046:~/pythonSwift/swift$ git status
On branch stable/train
Your branch is up-to-date with 'origin/stable/train'.
nothing to commit, working directory clean
```

Now Installing Python Again in this repo

python setup.py install

```
mehedi@hasan-17301046:~/pythonSwift/swift$ sudo python setup.py install
running install
running build
running build_py
creating build
creating build/lib.linux-i686-2.7
creating build/lib.linux-i686-2.7/swift
creating build/lib.linux-i686-2.7/swift/obj
copying swift/obj/replicator.py -> build/lib.linux-i686-2.7/swift/obj
copying swift/obj/__init__.py -> build/lib.linux-i686-2.7/swift/obj
copying swift/obj/auditor.py -> build/lib.linux-i686-2.7/swift/obj
```

```

changing mode of /usr/local/bin/swift-account-auditor to 755
changing mode of /usr/local/bin/swift-object-relinker to 755
changing mode of /usr/local/bin/swift-container-sharder to 755
changing mode of /usr/local/bin/swift-account-reaper to 755
changing mode of /usr/local/bin/swift-object-expirer to 755
changing mode of /usr/local/bin/swift-account-audit to 755
changing mode of /usr/local/bin/swift-init to 755
Installing swift-container-deleter script to /usr/local/bin
Installing swift-manage-shard-ranges script to /usr/local/bin

```

If we change our Directory to etc we can see lots of conf files. So need to copy those conf files.

```

mehedi@hasan-17301046:~/pythonSwift/swift$ cd etc
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ ls
account-server.conf-sample      memcache.conf-sample
container-reconciler.conf-sample  mime.types-sample
container-server.conf-sample     object-expirer.conf-sample
container-sync-realms.conf-sample object-server.conf-sample
dispersion.conf-sample          proxy-server.conf-sample
drive-audit.conf-sample         rsyncd.conf-sample
internal-client.conf-sample      swift.conf-sample
keymaster.conf-sample           swift-rsyslog.conf-sample
mehedi@hasan-17301046:~/pythonSwift/swift/etc$

```

First we need to make a folder in home/etc/swift

So, mkdir -p /etc/swift

Need to change the directory and Copy all the etc conf file to /etc/swift folder

cd swift/etc

Sudo cp account-server.conf-sample /etc/swift/account-server.conf

Sudo cp container-server.conf-sample /etc/swift/container-server.conf

Sudo cp object-server.conf-sample /etc/swift/object-server.conf

Sudo cp proxy-server.conf-sample /etc/swift/proxy-server.conf

Sudo cp drive-audit.conf-sample /etc/swift/drive-audit.conf

Sudo cp swift.conf-sample /etc/swift/swift.conf

```

mehedi@hasan-17301046:~/pythonSwift/swift/etc$ sudo cp account-server.conf-sample /etc/swift/account-server.conf
[sudo] password for mehedi:
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ sudo cp container-server.conf-sample /etc/swift/container-server.conf
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ sudo cp object-server.conf-sample /etc/swift/object-server.conf
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ sudo cp proxy-server.conf-sample /etc/swift/proxy-server.conf
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ sudo cp drive-audit.conf-sample /etc/swift/drive-audit.conf
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ sudo cp swift.conf-sample /etc/swift/swift.conf

```


After Copying all the files we can check our swift is installed

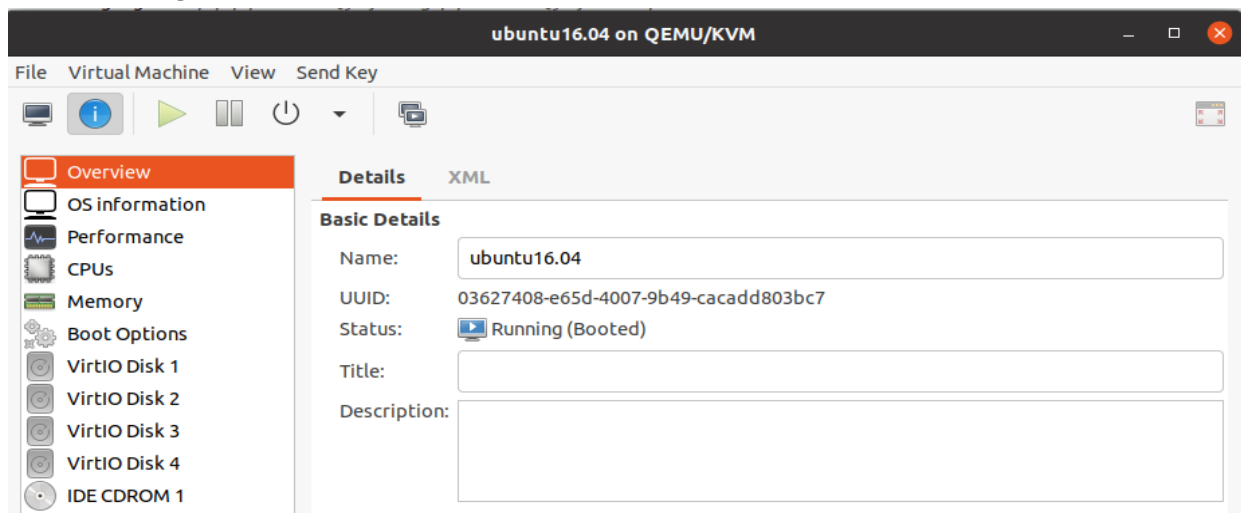
swift-init -h

```
mehedi@hasan-17301046:~/pythonSwift/swift/etc$ swift-init -h
Usage: swift-init <server>[.<config>] [<server>[.<config>] ...] <command> [options]

where:
  <server> is the name of a swift service e.g. proxy-server.
           The '-server' part of the name may be omitted.
           'all', 'main' and 'rest' are reserved words that represent a
           group of services.
           all: Expands to all swift daemons.
           main: Expands to main swift daemons.
                 (proxy, container, account, object)
           rest: Expands to all remaining background daemons (beyond
                 "main").
                 (updater, replicator, auditor, etc)
  <config> is an explicit configuration filename without the
           .conf extension. If <config> is specified then <server> should
           refer to a directory containing the configuration file, e.g.:
```

2. Adding Storage

After Installing Swift now we need to add 3 storage to our Swift. So firstly i create 3 virtual storage from KVM



Df

```
mehedi@hasan-17301046: ~
mehedi@hasan-17301046:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            2042068         0   2042068  0% /dev
tmpfs           413708      6300   407408   2% /run
/dev/vda1       20509308 5210012 14234440 27% /
tmpfs           2068528      252   2068276  1% /dev/shm
tmpfs           5120         4      5116   1% /run/lock
tmpfs           2068528         0   2068528  0% /sys/fs/cgroup
tmpfs           413708        64   413644  1% /run/user/1000
mehedi@hasan-17301046:~$
```

Check the driver in our sys Block

ls /sys/block

```
mehedi@hasan-17301046:~$ ls /sys/block
loop0 loop1 loop2 loop3 loop4 loop5 loop6 loop7 sr0 vda vdb vdc vdd
mehedi@hasan-17301046:~$
```

Now need to add those (VDB, VDC, VDD) Drive to our Swift not vda as its the main OS.

sudo mkfs.xfs -f -L drv1 /dev/vdb

sudo mkfs.xfs -f -L drv2 /dev/vdc

sudo mkfs.xfs -f -L drv3 /dev/vdd

```
mehedi@hasan-17301046:~$ sudo mkfs.xfs -f -L drv1 /dev/vdb
[sudo] password for mehedi:
meta-data=/dev/vdb            isize=512    agcount=4, agsize=393216 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=0
data      =                   bsize=4096    blocks=1572864, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2          bsize=4096    ascii-ci=0 ftype=1
log       =internal log      bsize=4096    blocks=2560, version=2
      =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096    blocks=0, rtextents=0
```

```
mehedi@hasan-17301046:~$ sudo mkfs.xfs -f -L drv2 /dev/vdc
meta-data=/dev/vdc            isize=512    agcount=4, agsize=393216 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=0
data      =                   bsize=4096    blocks=1572864, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2          bsize=4096    ascii-ci=0 ftype=1
log       =internal log      bsize=4096    blocks=2560, version=2
      =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096    blocks=0, rtextents=0
```

```
mehedi@hasan-17301046:~$ sudo mkfs.xfs -f -L drv3 /dev/vdd
meta-data=/dev/vdd            isize=512    agcount=4, agsize=393216 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=0
data      =                   bsize=4096    blocks=1572864, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2          bsize=4096    ascii-ci=0 ftype=1
log       =internal log      bsize=4096    blocks=2560, version=2
      =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096    blocks=0, rtextents=0
```

Now Need to Create directories in /srv/node locations

Sudo mkdir -p /srv/node/drv1

Sudo mkdir -p /srv/node/drv2

Sudo mkdir -p /srv/node/drv3

```
mehedi@hasan-17301046: ~  
mehedi@hasan-17301046:~$ mkdir -p /srv/node/drv1  
mkdir: cannot create directory '/srv/node': Permission denied  
mehedi@hasan-17301046:~$ sudo mkdir -p /srv/node/drv1  
mehedi@hasan-17301046:~$ sudo mkdir -p /srv/node/drv2  
mehedi@hasan-17301046:~$ sudo mkdir -p /srv/node/drv3  
mehedi@hasan-17301046:~$
```

Now need to Add the User to Swift

Useradd swift

chown -R swift:swift /srv/node

```
mehedi@hasan-17301046: ~  
mehedi@hasan-17301046:~$ useradd swift  
useradd: Permission denied.  
useradd: cannot lock /etc/passwd; try again later.  
mehedi@hasan-17301046:~$ sudo useradd swift  
mehedi@hasan-17301046:~$ sudo chown -R swift:swift /srv/node
```

Now need to Mount Those dishes to swift

sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv1 /srv/node/drv1

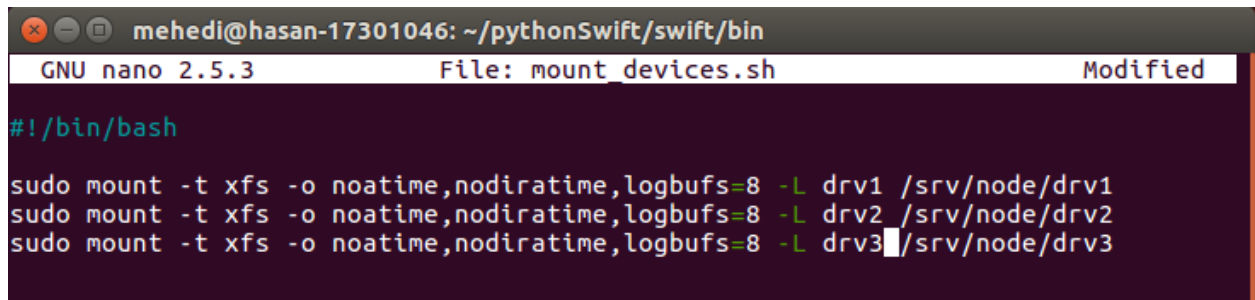
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv2 /srv/node/drv2

sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv3 /srv/node/drv3

```
mehedi@hasan-17301046:~$ sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv1 /srv/node/drv1  
mehedi@hasan-17301046:~$ sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv2 /srv/node/drv2  
mehedi@hasan-17301046:~$ sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv3 /srv/node/drv3  
mehedi@hasan-17301046:~$
```



```
cd swift/bin
nano mount_devices.sh
cat mount_devices.sh
#!/bin/bash
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv1 /srv/node/drv1
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv2 /srv/node/drv2
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv3 /srv/node/drv3
```

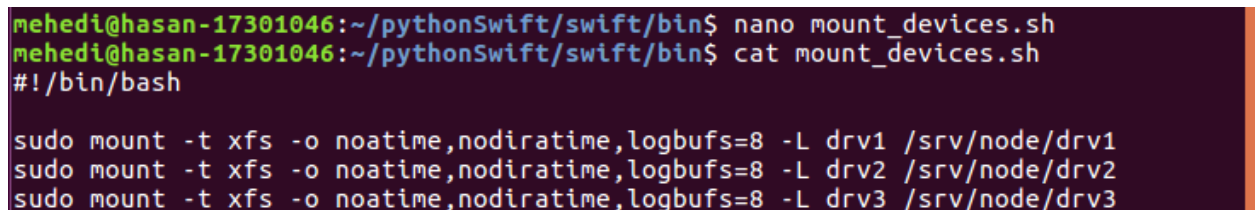


The screenshot shows a terminal window with the title bar 'mehedi@hasan-17301046: ~/pythonSwift/swift/bin'. The window contains the GNU nano 2.5.3 editor editing the file 'mount_devices.sh'. The file content is as follows:

```
#!/bin/bash

sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv1 /srv/node/drv1
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv2 /srv/node/drv2
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv3 /srv/node/drv3
```

```
cat mount_devices.sh
```

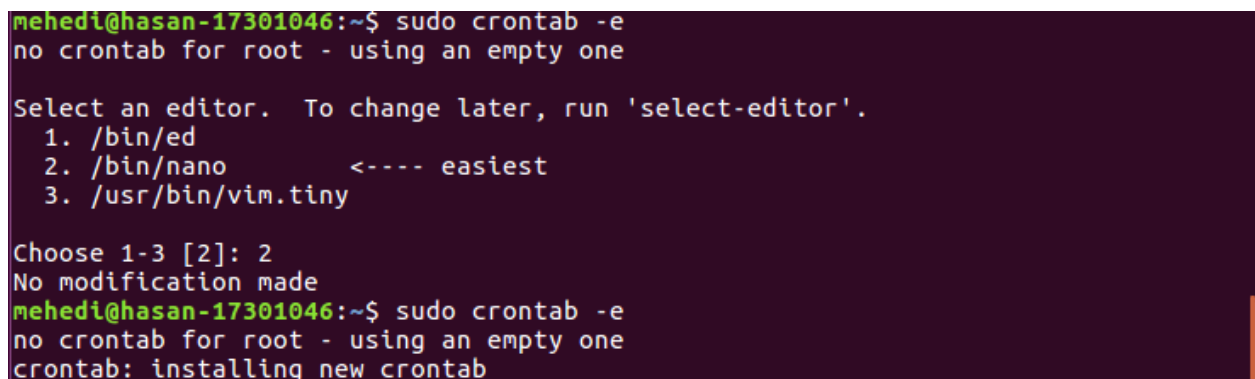


The screenshot shows a terminal window with the following commands and output:

```
mehedi@hasan-17301046:~/pythonSwift/swift/bin$ nano mount_devices.sh
mehedi@hasan-17301046:~/pythonSwift/swift/bin$ cat mount_devices.sh
#!/bin/bash

sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv1 /srv/node/drv1
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv2 /srv/node/drv2
sudo mount -t xfs -o noatime,nodiratime,logbufs=8 -L drv3 /srv/node/drv3
```

Adding crontab



The screenshot shows a terminal window with the following commands and output:

```
mehedi@hasan-17301046:~$ sudo crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/ed
 2. /bin/nano <---- easiest
 3. /usr/bin/vim.tiny

Choose 1-3 [2]: 2
No modification made
mehedi@hasan-17301046:~$ sudo crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
```

Edit the crontab and apply reboot. It will help us to mount devices after reboot

```
mehedi@hasan-17301046: ~  
GNU nano 2.5.3      File: /tmp/crontab.YXgkzX/crontab      Modified  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
@reboot sh swift/bin/mount_devices.sh
```

Now we can check our drive is mounted or not

Mount

Here we can see our 3 storage were Successfully mounted

```
/dev/vdb on /media/mehedi/driv1 type xfs (rw,nosuid,nodev,relatime,attr2,inode64,  
noquota,uhelper=udisks2)  
/dev/vdc on /media/mehedi/driv2 type xfs (rw,nosuid,nodev,relatime,attr2,inode64,  
noquota,uhelper=udisks2)  
/dev/vdd on /media/mehedi/driv3 type xfs (rw,nosuid,nodev,relatime,attr2,inode64,  
noquota,uhelper=udisks2)
```

3. Ring Builders

To add ring first we need to go to our /etc/swift Directory

cd /etc/swift

Now, here we need to add ring builder to our account, container and object

swift-ring-builder account.builder create 3 3 1

swift-ring-builder container.builder create 3 3 1

swift-ring-builder object.builder create 3 3 1

```
mehedi@hasan-17301046: /etc/swift  
mehedi@hasan-17301046:~$ cd /etc/swift  
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder account.builder create  
3 3 1  
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder container.builder crea  
te 3 3 1  
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder object.builder create  
3 3 1  
mehedi@hasan-17301046:/etc/swift$
```

Now it's time to Adding Devices to our Builder files

```
swift-ring-builder account.builder add r1z1-127.0.0.1:6002/drv1 100
swift-ring-builder container.builder add r1z1-127.0.0.1:6001/drv1 100
swift-ring-builder object.builder add r1z1-127.0.0.1:6000/drv1 100
```

```
swift-ring-builder account.builder add r1z1-127.0.0.1:6002/drv2 100
swift-ring-builder container.builder add r1z1-127.0.0.1:6001/drv2 100
swift-ring-builder object.builder add r1z1-127.0.0.1:6000/drv2 100
```

```
swift-ring-builder account.builder add r1z1-127.0.0.1:6002/drv3 100
swift-ring-builder container.builder add r1z1-127.0.0.1:6001/drv3 100
swift-ring-builder object.builder add r1z1-127.0.0.1:6000/drv3 100
```

```
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder account.builder add r1
z1-127.0.0.1:6002/drv1 100
[sudo] password for mehedi:
Device d0r1z1-127.0.0.1:6002R127.0.0.1:6002/drv1_" with 100.0 weight got id 0
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder container.builder add
r1z1-127.0.0.1:6001/drv1 100
Device d0r1z1-127.0.0.1:6001R127.0.0.1:6001/drv1_" with 100.0 weight got id 0
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder object.builder add r1z
1-127.0.0.1:6000/drv1 100
Device d0r1z1-127.0.0.1:6000R127.0.0.1:6000/drv1_" with 100.0 weight got id 0
mehedi@hasan-17301046:/etc/swift$
```

```
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder account.builder add r1
z1-127.0.0.1:6002/drv2 100
Device d1r1z1-127.0.0.1:6002R127.0.0.1:6002/drv2_" with 100.0 weight got id 1
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder container.builder add
r1z1-127.0.0.1:6001/drv2 100
Device d1r1z1-127.0.0.1:6001R127.0.0.1:6001/drv2_" with 100.0 weight got id 1
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder object.builder add r1z
1-127.0.0.1:6000/drv2 100
Device d1r1z1-127.0.0.1:6000R127.0.0.1:6000/drv2_" with 100.0 weight got id 1
```

```
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder account.builder add r1
z1-127.0.0.1:6002/drv3 100
Device d2r1z1-127.0.0.1:6002R127.0.0.1:6002/drv3_" with 100.0 weight got id 2
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder container.builder add
r1z1-127.0.0.1:6001/drv3 100
Device d2r1z1-127.0.0.1:6001R127.0.0.1:6001/drv3_" with 100.0 weight got id 2
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder object.builder add r1z
1-127.0.0.1:6000/drv3 100
Device d2r1z1-127.0.0.1:6000R127.0.0.1:6000/drv3_" with 100.0 weight got id 2
```

Now we need to balancing our Builder files

swift-ring-builder account.builder rebalance
swift-ring-builder container.builder rebalance
swift-ring-builder object.builder rebalance

```
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder account.builder rebalance
No partitions could be reassigned.
There is no need to do so at this time
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder container.builder rebalance
Reassigned 24 (300.00%) partitions. Balance is now 0.00. Dispersion is now 0.00
mehedi@hasan-17301046:/etc/swift$ sudo swift-ring-builder object.builder rebalance
Reassigned 24 (300.00%) partitions. Balance is now 0.00. Dispersion is now 0.00
mehedi@hasan-17301046:/etc/swift$
```

After Balancing if we want, we can check our builder files

Ls

```
mehedi@hasan-17301046:/etc/swift$ ls
account.builder      container.builder    object.builder      swift.conf
account.ring.gz      container.ring.gz    object.ring.gz
account-server.conf  container-server.conf object-server.conf
backups              drive-audit.conf    proxy-server.conf
```

We need to configure our Swift logging by adding local files to our rsyslog

cd /etc
cd rsyslog.d

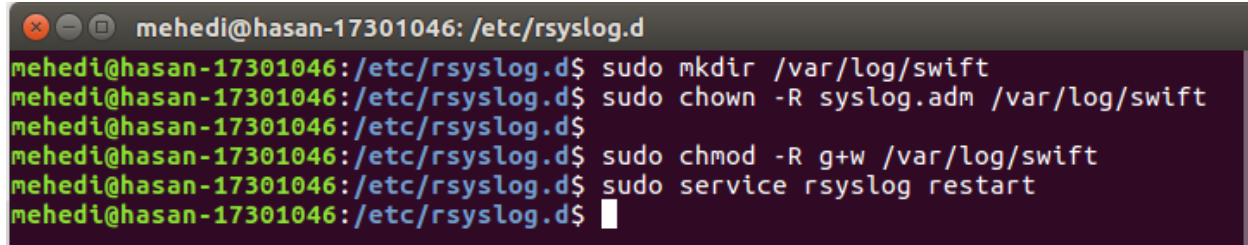
```
mehedi@hasan-17301046: /etc/rsyslog.d
mehedi@hasan-17301046:/etc$ cd rsyslog.d
mehedi@hasan-17301046:/etc/rsyslog.d$ sudo nano vim 0-swift.conf
mehedi@hasan-17301046:/etc/rsyslog.d$ sudo nano cat 0-swift.conf
```

nano 0-swift.conf

local0.* /var/log/swift/all.log

```
mehedi@hasan-17301046: /etc/rsyslog.d
GNU nano 2.5.3      File: vim
local0.* /var/log/swift/all.log
```

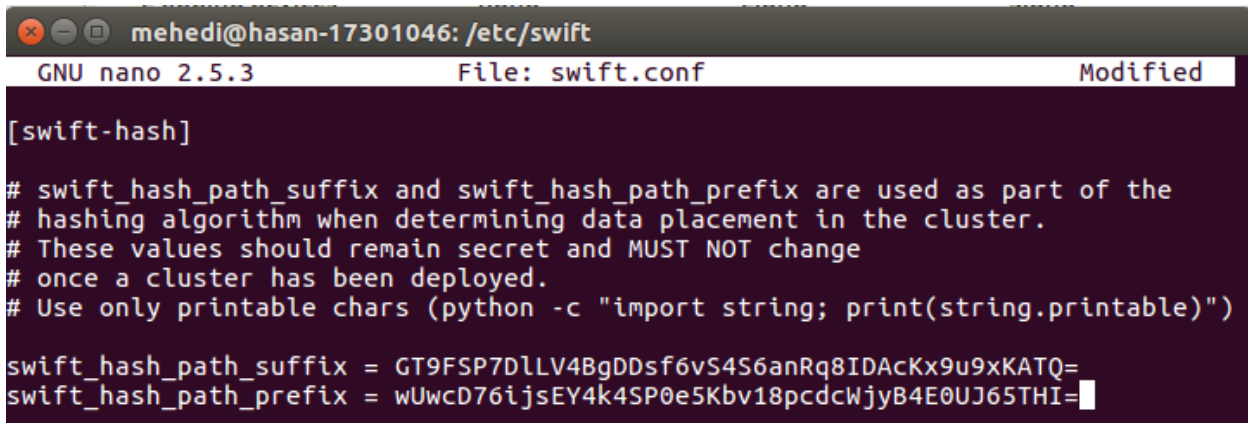
```
cd /etc/rsyslog.d
mkdir /var/log/swift
chown -R syslog.adm /var/log/swift
chmod -R g+w /var/log/swift
service rsyslog restart
```



```
mehedi@hasan-17301046: /etc/rsyslog.d
mehedi@hasan-17301046:/etc/rsyslog.d$ sudo mkdir /var/log/swift
mehedi@hasan-17301046:/etc/rsyslog.d$ sudo chown -R syslog.adm /var/log/swift
mehedi@hasan-17301046:/etc/rsyslog.d$
mehedi@hasan-17301046:/etc/rsyslog.d$ sudo chmod -R g+w /var/log/swift
mehedi@hasan-17301046:/etc/rsyslog.d$ sudo service rsyslog restart
mehedi@hasan-17301046:/etc/rsyslog.d$
```

Now I need to add a Hash path to our swift.conf file, it will help to form many kinds of denial-of-service (DOS) attacks. If someone knows the hash path suffix and prefix, he could determine the actual partition where objects would be stored.

```
cd /etc/swift
nano swift.conf
swift_hash_path_suffix = GT9FSP7DILV4BgDDsf6vS4S6anRq8IDAcKx9u9xKATQ=
swift_hash_path_prefix = wUwcD76ijsEY4k4SP0e5Kbv18pcdcWjyB4E0UJ65THI=
```



```
mehedi@hasan-17301046: /etc/swift
GNU nano 2.5.3      File: swift.conf      Modified

[swift-hash]

# swift_hash_path_suffix and swift_hash_path_prefix are used as part of the
# hashing algorithm when determining data placement in the cluster.
# These values should remain secret and MUST NOT change
# once a cluster has been deployed.
# Use only printable chars (python -c "import string; print(string.printable)")

swift_hash_path_suffix = GT9FSP7DILV4BgDDsf6vS4S6anRq8IDAcKx9u9xKATQ=
swift_hash_path_prefix = wUwcD76ijsEY4k4SP0e5Kbv18pcdcWjyB4E0UJ65THI=
```

Sudo cat swift.conf

We can see our swift-hash-Key, Anyone can create hash key simply by using
head -c 32 /dev/random | base64

```
mehedi@hasan-17301046:/etc/swift$ nano vim swift.conf
mehedi@hasan-17301046:/etc/swift$ sudo cat swift.conf
[swift-hash]

# swift_hash_path_suffix and swift_hash_path_prefix are used as part of the
# hashing algorithm when determining data placement in the cluster.
# These values should remain secret and MUST NOT change
# once a cluster has been deployed.
# Use only printable chars (python -c "import string; print(string.printable)")

swift_hash_path_suffix = GT9FSP7DlLV4BgDDsf6vS4S6anRq8IDAcKx9u9xKATQ=
swift_hash_path_prefix = wUwcD76ijsEY4k4SP0e5Kbv18pcdcWjyB4E0UJ65THI=
```

So now we can need to setting up tempAuth Authentication and Authorization with swift

service memcached start
ps aux | grep memcached

```
mehedi@hasan-17301046:/etc/swift$ sudo service memcached start
mehedi@hasan-17301046:/etc/swift$ sudo ps aux | grep memcached
memcache  702  0.0  0.0 424768 3000 ?        Ssl  20:31   0:00 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1 -P /var/run/memcached/memcached.pid
mehedi    1787  0.0  0.0 14764 1016 pts/0    S+   20:36   0:00 grep --color=auto memcached
```

After That need to add user in the proxy-server.conf file and also the blidport

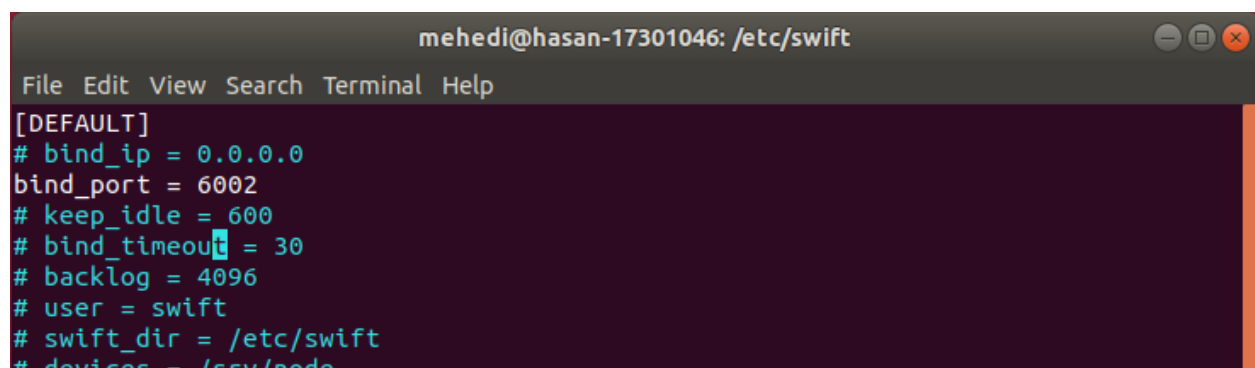
cd /etc/swift
nano proxy-server.conf
user_myaccount_me = secretpassword .admin .reseller_admin <storage
URL:8080>

```
# To what the requester would need to use to reach this host.
# Here are example entries, required for running the tests:
user_admin_admin = admin .admin .reseller_admin
user_test_tester = testing .admin
user_test_tester2 = testing2 .admin
user_test_tester3 = testing3
user_test2_tester2 = testing2 .admin
user_test5_tester5 = testing5 service
user_myaccount_me = secretpassword .admin .reseller_admin <storage
URL:8080>
# To enable Keystone authentication you need to have the auth token
```


Set the Account management and Auto Create to True

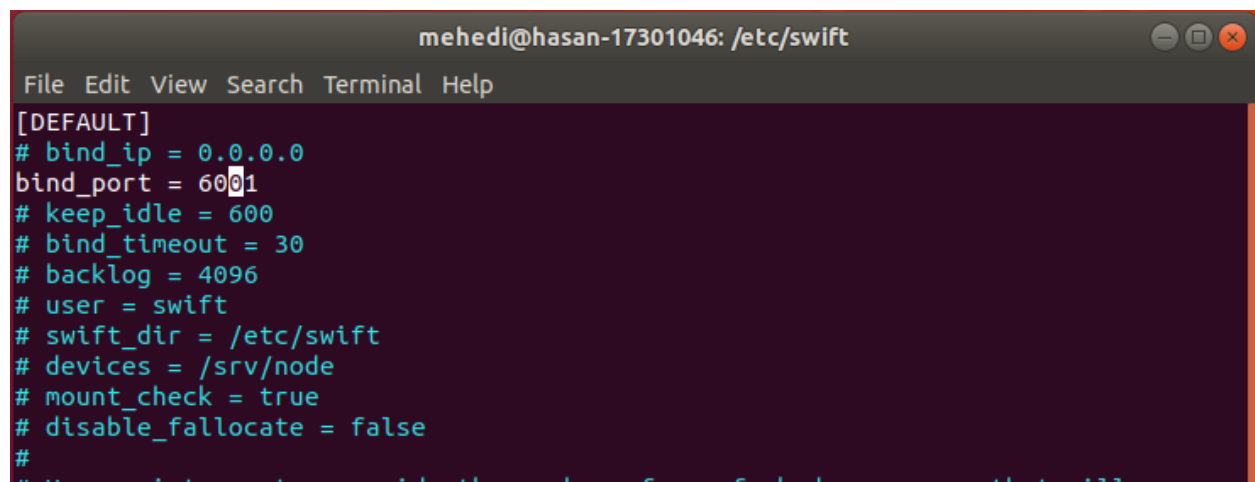
```
# If set to 'true' any authorized user may create and delete accounts; if
# 'false' no one, even authorized, can.
allow_account_management = true
#
# If set to 'true' authorized accounts that do not yet exist within the Swift
# cluster will be automatically created.
account_autocreate = true
#
```

In the Same way need to edit account-server, Container-server, Object-server .conf file to add port address 6002 , 6001 , 6000



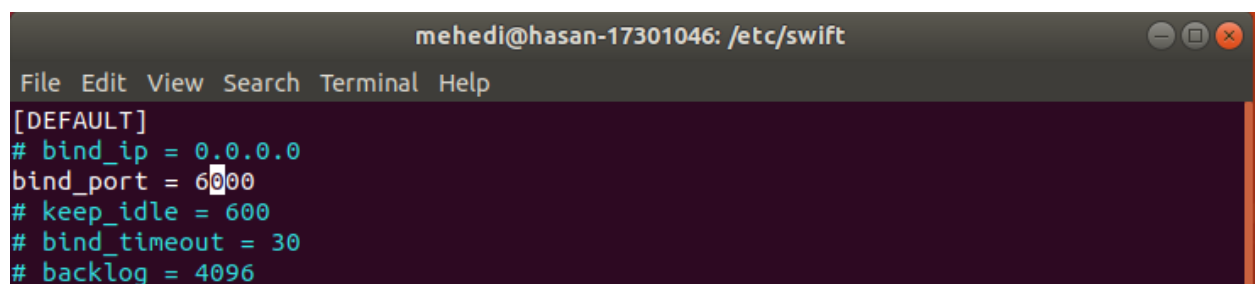
A terminal window titled 'mehedi@hasan-17301046: /etc/swift' with a menu bar (File, Edit, View, Search, Terminal, Help). The content shows the configuration for account-server.conf. The 'bind_port' is set to 6002. The cursor is positioned at the end of the line '# bind_timeout = 30'.

```
mehedi@hasan-17301046: /etc/swift
[DEFAULT]
# bind_ip = 0.0.0.0
bind_port = 6002
# keep_idle = 600
# bind_timeout = 30
# backlog = 4096
# user = swift
# swift_dir = /etc/swift
# devices = /srv/node
```



A terminal window titled 'mehedi@hasan-17301046: /etc/swift' with a menu bar (File, Edit, View, Search, Terminal, Help). The content shows the configuration for container-server.conf. The 'bind_port' is set to 6001. The cursor is positioned at the end of the line '# mount_check = true'.

```
mehedi@hasan-17301046: /etc/swift
[DEFAULT]
# bind_ip = 0.0.0.0
bind_port = 6001
# keep_idle = 600
# bind_timeout = 30
# backlog = 4096
# user = swift
# swift_dir = /etc/swift
# devices = /srv/node
# mount_check = true
# disable_fallocate = false
#
# Use an integer to override the number of open file descriptors that will
```



A terminal window titled 'mehedi@hasan-17301046: /etc/swift' with a menu bar (File, Edit, View, Search, Terminal, Help). The content shows the configuration for object-server.conf. The 'bind_port' is set to 6000. The cursor is positioned at the end of the line '# backlog = 4096'.

```
mehedi@hasan-17301046: /etc/swift
[DEFAULT]
# bind_ip = 0.0.0.0
bind_port = 6000
# keep_idle = 600
# bind_timeout = 30
# backlog = 4096
```

After Assigning port we need to start our proxy server

Sudo swift-init proxy start

And if We see something starting proxy..(/etc/swift..) it means our proxy server is successfully connected.

```
mehedi@hasan-17301046: /etc/swift
File Edit View Search Terminal Help
mehedi@hasan-17301046:/etc/swift$ sudo swift-init proxy start
/usr/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography import x509
Starting proxy-server...(/etc/swift/proxy-server.conf)
/usr/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography import x509
```

In the Same way we need to Start our Every Server file. We can do it by just 1 command

Sudo swift-init main start

It Will Start all of our server

```
mehedi@hasan-17301046:/etc/swift$ sudo swift-init main start
/usr/lib/python2.7/dist-packages/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography import x509
proxy-server running (1688 - /etc/swift/proxy-server.conf)
proxy-server already started...
container-server running (2064 - /etc/swift/container-server.conf)
container-server already started...
account-server running (2083 - /etc/swift/account-server.conf)
account-server already started...
object-server running (2084 - /etc/swift/object-server.conf)
object-server already started...
```

Now it's time for the Authentication and Authorization part.

If We run this command we will get a token for our storage.

```
curl -v -H 'X-Auth-User: myaccount:me' -H 'X-Auth-Key: secretpassword'  
http://localhost:8080/auth/v1.0/
```

```
mehedi@hasan-17301046: /etc/swift
File Edit View Search Terminal Help
object-server running (2084 - /etc/swift/object-server.conf)
object-server already started...
mehedi@hasan-17301046:/etc/swift$ sudo curl -v -H 'X-Auth-User: admin:admin' -H
'X-Auth-Key: admin' http://localhost:8080/auth/v1.0/
* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET /auth/v1.0/ HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.58.0
> Accept: */*
> X-Auth-User: admin:admin
> X-Auth-Key: admin
>
< HTTP/1.1 200 OK
< X-Storage-Url: http://localhost:8080/v1/AUTH_admin
< X-Auth-Token-Expires: 86399
< X-Auth-Token: AUTH_tka8ff45e922bc4f4485699274df25ea22
< Content-Type: text/html; charset=UTF-8
< X-Storage-Token: AUTH_tka8ff45e922bc4f4485699274df25ea22
< Content-Length: 0
< X-Trans-Id: tx7b8906b1ac454538854df-006123bb26
< X-Openstack-Request-Id: tx7b8906b1ac454538854df-006123bb26
< Date: Mon, 23 Aug 2021 15:13:42 GMT
```

After that we need to check our Http server. We need to use Storeger Token/

```
curl -v -H 'X-Storage-Token: AUTH_tka8ff45e922bc4f4485699274df25ea22'  
http://127.0.0.1:8080/v1/AUTH\_myaccount/
```

If it returns something 204 no Content it means our Authentication and Authorization is correct. Here i can see i got 204 no Content, means successfully Authentication done.

N.B: This is the most crucial part. If it fails or shows any error like 503 Server unauthorized we need to do the same work from the mount reboot.

```
mehedi@hasan-17301046:~$ curl -v -H 'X-Storage-Token: AUTH_tk725da61c9aa14bb691f97d8aa290803b' http://127.0.0.1:8080/v1/AUTH_myaccount/
* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> GET /v1/AUTH_myaccount/ HTTP/1.1
> Host: 127.0.0.1:8080
> User-Agent: curl/7.58.0
> Accept: */*
> X-Storage-Token: AUTH_tk725da61c9aa14bb691f97d8aa290803b
>
< HTTP/1.1 204 No Content
```

Now We can Check our Account stat

swift -U myaccount:me -K secretpassword -A http://localhost:8080/auth/v1.0 stat

Here we can see Swift Account was created and there is no container there.

```
mehedi@hasan-17301046:/etc/swift$ sudo swift -U myaccount:me -K secretpassword -A http://127.0.0.1:8080/auth/v1.0 stat
Account: AUTH_myaccount
Containers: 0
Objects: 0
Bytes: 0
X-Put-Timestamp: 1629798413.67656
X-Timestamp: 1629798413.67656
X-Trans-Id: txa11745467d464884a6fe3-006124c00d
Content-Type: text/plain; charset=utf-8
X-Openstack-Request-Id: txa11745467d464884a6fe3-006124c00d
```

Now we can upload file in our storage by using this command

swift -A http://127.0.0.1:8080/auth/v1.0/ -U myaccount:me -K secretpassword upload mycontainer text.txt

```
mehedi@hasan-17301046:~$ cat test.txt
cat: test.txt: No such file or directory
mehedi@hasan-17301046:~$ cat text.txt
This is a text file
mehedi@hasan-17301046:~$
```

We Can Create a container in our Swift

curl -v -H 'X-Storage-Token: AUTH_tka8f76378a27f844e79c17e09a96efa5cb' -X PUT http://127.0.0.1:8080/v1/AUTH_myaccount/mycontainer

If it shows something 201 created. Congratulations, successfully swift container is created.

```
mehedi@hasan-17301046:/etc/swift$ curl -v -H 'X-Storage-Token: AUTH_tka8f76378a27f844e79c17e09a96efa5cb' -X PUT http://127.0.0.1:8080/v1/AUTH_myaccount/mycontainer
* Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> PUT /v1/AUTH_myaccount/mycontainer HTTP/1.1
> Host: 127.0.0.1:8080
> User-Agent: curl/7.58.0
> Accept: */*
> X-Storage-Token: AUTH_tka8f76378a27f844e79c17e09a96efa5cb
>
< HTTP/1.1 201 Created
```

Now We can check again if there was any container there or not

swift -U myaccount:me -K secretpassword -A http://localhost:8080/auth/v1.0 stat

See, Previously it was 0 but now it is 1. It means container Created Successfully.

```
mehedi@hasan-17301046:/etc/swift$ swift -U myaccount:me -K secretpassword -A http://localhost:8080/auth/v1.0 stat
Account: AUTH_myaccount
Containers: 1
Objects: 0
Bytes: 0
Containers in policy "policy-0": 1
Objects in policy "policy-0": 0
Bytes in policy "policy-0": 0
X-Openstack-Request-Id: tx15b1b98c37f4401a880ad-006124c1b7
```

4. Uploading And Downloading Files from container.

Again Uploading a file in my new created container

```
mehedi@hasan-17301046:~$ swift -A http://127.0.0.1:8080/auth/v1.0/ -U myaccount:me -K secretpassword upload mycontainer mehedi-test-file
mehedi-test-file/test.txt
mehedi-test-file/~lock.test.txt#
```

We can Download the file from the container. We can see the every detail. It means download is successful

```
mehedi@hasan-17301046:~$ swift -A http://127.0.0.1:8080/auth/v1.0/ -U myaccount:me -K secretpassword download mycontainer
mehedi-test-file/~lock.test.txt# [auth 0.006s, headers 0.018s, total 0.018s, 0.006 MB/s]
mehedi-test-file/test.txt [auth 0.008s, headers 0.025s, total 0.025s, 0.002 MB/s]
mehedi@hasan-17301046:~$
```

5. Replication Process

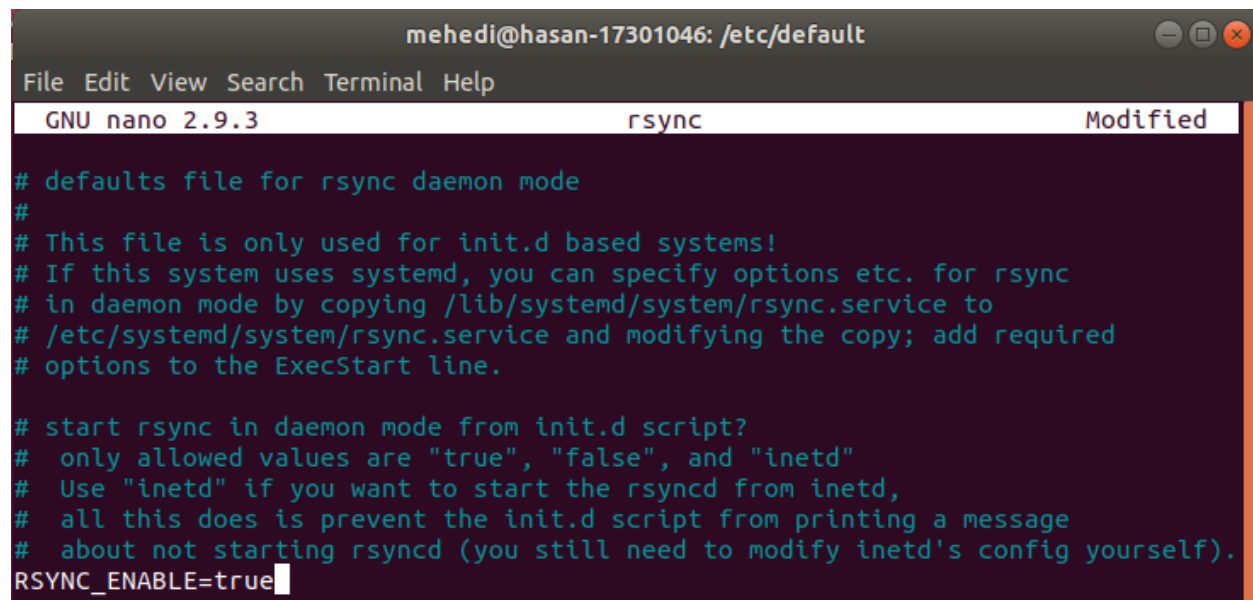
If we go to /srv/node and perform ls -l we can see our 3 old drives.

```
mehedi@hasan-17301046:~$ cd /srv/node
bash: cd: /srv/node: No such file or directory
mehedi@hasan-17301046:~$ cd /srv/node
mehedi@hasan-17301046:/srv/node$ ls
d1 d2 d3
mehedi@hasan-17301046:/srv/node$ ls -l
total 0
drwxr-xr-x 5 swift swift 55 24 16:00 d1
drwxr-xr-x 5 swift swift 55 24 16:00 d2
drwxr-xr-x 5 swift swift 55 24 16:00 d3
mehedi@hasan-17301046:/srv/node$
```

Before Perform replication first we need to Enable rsync

```
mehedi@hasan-17301046:/srv/node$ cd /etc
mehedi@hasan-17301046:/etc$ cd default
mehedi@hasan-17301046:/etc/default$ ls
acpid          console-setup  kerneloops     rsyslog
acpi-support   crda           keyboard        saned
alsa           cron           locale          speech-dispatcher
amd64-microcode  dbus          memcached       ufw
anacron        grub           networkd-dispatcher useradd
appport        im-config     networking
avahi-daemon   intel-microcode nss
bsdmainutils   irqbalance    rsync
mehedi@hasan-17301046:/etc/default$ nano rsync
mehedi@hasan-17301046:/etc/default$ sudo nano rsync
[sudo] password for mehedi:
mehedi@hasan-17301046:/etc/default$
```

In the rsync part we need to perform a task which is `RSYNC_ENABLE = true`
It will help us to perform our rsync of drive.



```
mehedi@hasan-17301046: /etc/default
File Edit View Search Terminal Help
GNU nano 2.9.3 rsync Modified
# defaults file for rsync daemon mode
#
# This file is only used for init.d based systems!
# If this system uses systemd, you can specify options etc. for rsync
# in daemon mode by copying /lib/systemd/system/rsync.service to
# /etc/systemd/system/rsync.service and modifying the copy; add required
# options to the ExecStart line.
#
# start rsync in daemon mode from init.d script?
# only allowed values are "true", "false", and "inetd"
# Use "inetd" if you want to start the rsyncd from inetd,
# all this does is prevent the init.d script from printing a message
# about not starting rsyncd (you still need to modify inetd's config yourself).
RSYNC_ENABLE=true
```


Now i need to create a rsync.conf file as there was no rsync.conf file
So sudo nano rsync.conf will open a file and we need to put some values in that files.
Such as Account, container, Object

```
mehedi@hasan-17301046: /etc
File Edit View Search Terminal Help
GNU nano 2.9.3 New Buffer Modified

uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
[account]
max connections = 25
path = /srv/node
read only = false
lock file = /var/lock/account.lock
[container]
max connections = 25
id:17301046
path = /srv/node/
read only = false
lock file = /var/lock/container.lock
[object]
max connections = 25
path = /srv/node/
read only = false
```

```
mehedi@hasan-17301046: /etc$ cat rsyncd.conf
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
[account]
max connections = 25
path = /srv/node
read only = false
lock file = /var/lock/account.lock
[container]
max connections = 25
id:17301046
path = /srv/node/
read only = false
lock file = /var/lock/container.lock
[object]
max connections = 25
path = /srv/node/
read only = false
lock file = /var/lock/object.lock
```

After That need to start our replication process

Sudo swift-init account-replication

In the same way need to do it for container and object also.

```
mehedi@hasan-17301046:/etc$ cd /srv/node
mehedi@hasan-17301046:/srv/node$ sudo swift-init account-replicator start
Starting account-replicator...(etc/swift/account-server.conf)
mehedi@hasan-17301046:/srv/node$ sudo swift-init container-replicator start
Starting container-replicator...(etc/swift/container-server.conf)
mehedi@hasan-17301046:/srv/node$ sudo swift-init object-replicator start
Starting object-replicator...(etc/swift/object-server.conf)
```

Now it's time to check the rsync part

If we start rsync and check in localhost we can see 3 replications.

```
mehedi@hasan-17301046:/srv/node$ sudo service rsync start
mehedi@hasan-17301046:/srv/node$ sudo rsync localhost::
account
container
object
mehedi@hasan-17301046:/srv/node$
```

Now we need to start our Replication process for Account, container and Object

```
mehedi@hasan-17301046:/srv/node$ sudo rsync localhost::account
drwxr-xr-x      4,096 2021/08/24 14:31:59 .
drwxr-xr-x      66 2021/08/24 16:28:45 d1
drwxr-xr-x      66 2021/08/24 16:28:45 d2
drwxr-xr-x      66 2021/08/24 16:28:45 d3
mehedi@hasan-17301046:/srv/node$ sudo rsync localhost::container
drwxr-xr-x      4,096 2021/08/24 14:31:59 .
drwxr-xr-x      66 2021/08/24 16:28:45 d1
drwxr-xr-x      66 2021/08/24 16:28:45 d2
drwxr-xr-x      66 2021/08/24 16:28:45 d3
```

We can see 3 replications done.

```
mehedi@hasan-17301046:/srv/node$ find . -name '*.data'
./d2/objects/0/e5f/1641ad30cc50f9b6e11374710664ce5f/1629799237.61711.data
./d2/objects/4/5d9/8e250ddc24cf10b4c1dd5c7d980cc5d9/1629799237.61504.data
./d3/objects/4/5d9/8e250ddc24cf10b4c1dd5c7d980cc5d9/1629799237.61504.data
./d3/objects/0/e5f/1641ad30cc50f9b6e11374710664ce5f/1629799237.61711.data
./d1/objects/4/5d9/8e250ddc24cf10b4c1dd5c7d980cc5d9/1629799237.61504.data
./d1/objects/0/e5f/1641ad30cc50f9b6e11374710664ce5f/1629799237.61711.data
mehedi@hasan-17301046:/srv/node$
```

Now need to check the replication is working perfectly or not , So we deleted drive 1
So, driver 1 is successfully deleted. It means replication works smoothly.

```
mehedi@hasan-17301046:/srv/node$ sudo rm -rf /srv/node/d1/*
mehedi@hasan-17301046:/srv/node$ find . -name '*.data'
./d2/objects/0/e5f/1641ad30cc50f9b6e11374710664ce5f/1629799237.61711.data
./d2/objects/4/5d9/8e250ddc24cf10b4c1dd5c7d980cc5d9/1629799237.61504.data
./d3/objects/4/5d9/8e250ddc24cf10b4c1dd5c7d980cc5d9/1629799237.61504.data
./d3/objects/0/e5f/1641ad30cc50f9b6e11374710664ce5f/1629799237.61711.data
mehedi@hasan-17301046:/srv/node$
```

6. Swifts command

Swift list will give me all available files in my swift account.

```
mehedi@hasan-17301046:/etc/swift$ swift list -U myaccount:me -K secretpassw
ord -A http://127.0.0.1:8080/auth/v1.0 mycontainer
mehedi-test-file/~lock.test.txt#
mehedi-test-file/test.txt
```

Swift post will create a new container in my swift account

```
mehedi@hasan-17301046:/etc/swift$ swift post -U myaccount:me -K secretpassw
ord -A http://127.0.0.1:8080/auth/v1.0 mehediContainer
mehedi@hasan-17301046:/etc/swift$ swift list -U myaccount:me -K secretpassw
ord -A http://127.0.0.1:8080/auth/v1.0
mehediContainer
mycontainer
```

If i want i can delete my container also.

```
mehedi@hasan-17301046:/etc/swift$ swift delete mehediContainer -U myaccount
:me -K secretpassword -A http://127.0.0.1:8080/auth/v1.0
mehediContainer
mehedi@hasan-17301046:/etc/swift$ swift list -U myaccount:me -K secretpassw
ord -A http://127.0.0.1:8080/auth/v1.0
mycontainer
```

7. Using Curl

Curl Get

It will show me all the available containers i have.

```
mehedi@hasan-17301046:/etc/swift$ curl -i -X GET -H "X-Auth-Token:AUTH_tk8f76378a27f844e79c17e09a96efa5cb" http://127.0.0.1:8080/v1.0/AUTH_myaccount/
HTTP/1.1 200 OK
Content-Length: 29
X-Account-Object-Count: 0
X-Account-Storage-Policy-Policy-0-Bytes-Used: 0
X-Account-Storage-Policy-Policy-0-Container-Count: 2
X-Timestamp: 1629798772.04367
X-Account-Storage-Policy-Policy-0-Object-Count: 0
X-Account-Bytes-Used: 0
X-Account-Container-Count: 2
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: tx46e4b462c7cd421fa9990-006124eb32
X-Openstack-Request-Id: tx46e4b462c7cd421fa9990-006124eb32
Date: Tue, 24 Aug 2021 12:50:58 GMT

amiBossContainer
mycontainer
```

In the same way, I can Delete, Put.

Curl Post

```
mehedi@hasan-17301046:~$ curl -i -X POST -H "X-Storage-Token:AUTH_tk8f76378a27f844e79c17e09a96efa5cb" http://127.0.0.1:8080/v1.0/AUTH_myaccount/mycontainer -T newCurlContainer
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txb34291f3cb134e01bf093-006124ec5c
X-Openstack-Request-Id: txb34291f3cb134e01bf093-006124ec5c
Date: Tue, 24 Aug 2021 12:55:56 GMT
```

N.B: Creating your Own storage is very fun and Enjoyable. It will give so many errors. I have also faced a lot. I did this assignment in Ubuntu-16 and Ubuntu-18 Both. My suggestion is that Ubuntu-18 is best for creating your own Storage. As 16 has very old python packages. It will gives lots of errors. Finally, Never Give Up.

The End