BRAC
UNIVERSITY

Inspiring Excellence

CSE484(Cloud Computing)

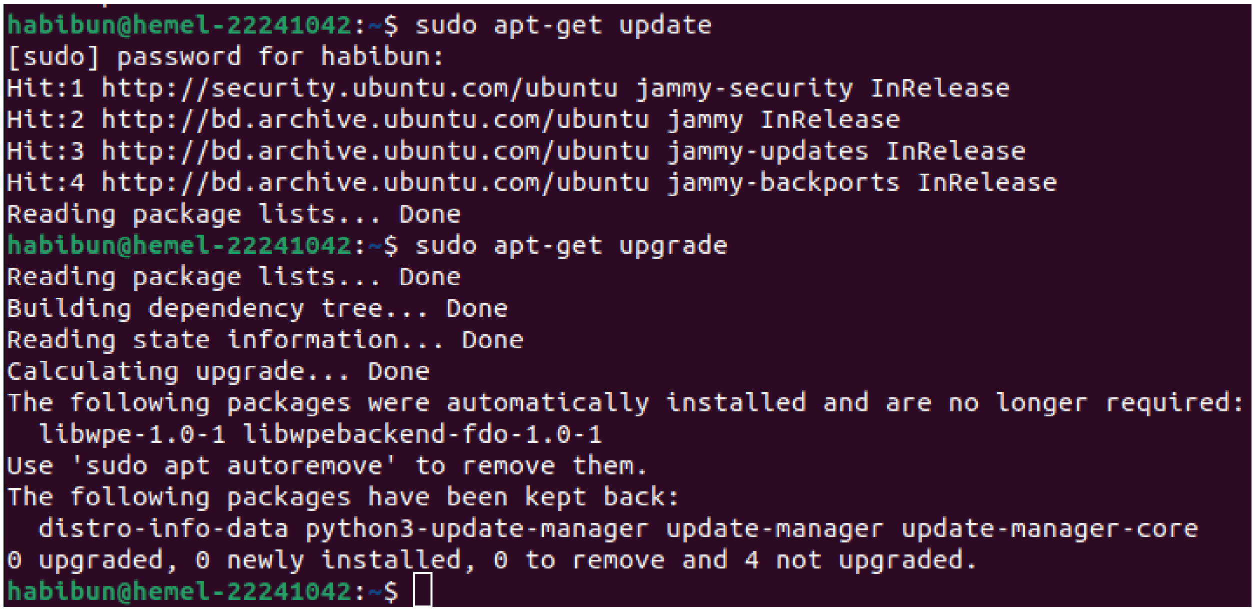
Assignment 5 : Time to create Own Storage

Name: Habibun Nabi Hemel
Id: 22241042
Semester: Sping24
Submitted to: Jannatun Noor

Install OpenStack Swift SAIO in any Linux platform

We start the installation by running the commands on an Ubuntu system before moving on to install and update the system's dependencies.

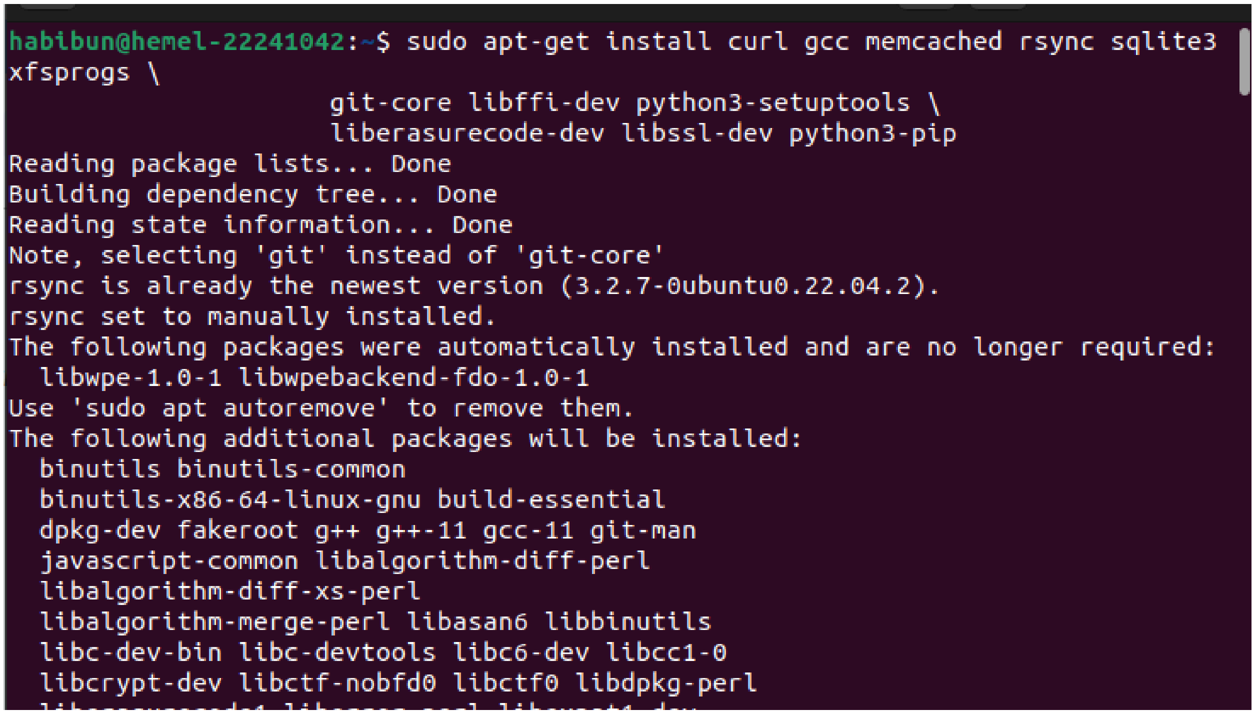
```
sudo apt-get update
sudo apt-get upgrade
```



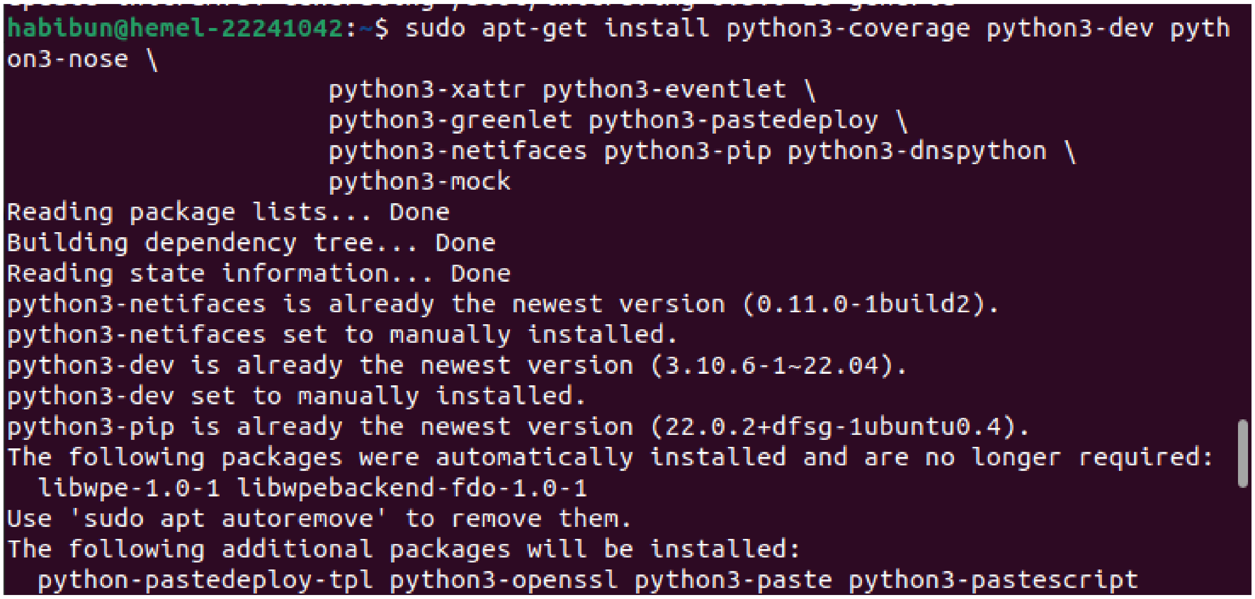
```
habibun@hemel-22241042:~$ sudo apt-get update
[sudo] password for habibun:
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://bd.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://bd.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://bd.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
habibun@hemel-22241042:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  distro-info-data python3-update-manager update-manager update-manager-core
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
habibun@hemel-22241042:~$ 
```

Then we need to download all the dependencies for the installation. Then command for download the dependencies:

```
sudo apt-get install curl gcc memcached rsync sqlite3 xfsprogs \
                 git-core libffi-dev python3-setuptools \
                 liberasurecode-dev libssl-dev python3-pip

sudo apt-get install python3-coverage python3-dev python3-nose \
                 python3-xattr python3-eventlet \
                 python3-greenlet python3-pastedeploy \
                 python3-netifaces python3-pip python3-dnspython \
                 python3-mock
```



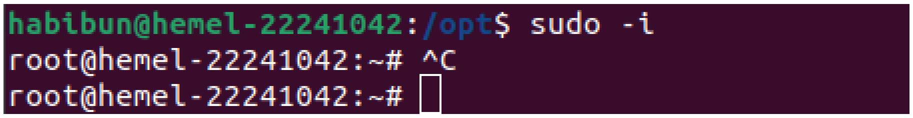
```
habibun@hemel-22241042:~$ sudo apt-get install curl gcc memcached rsync sqlite3
xfsprogs \
                     git-core libffi-dev python3-setuptools \
                     liberasurecode-dev libssl-dev python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'git' instead of 'git-core'
rsync is already the newest version (3.2.7-0ubuntu0.22.04.2).
rsync set to manually installed.
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  binutils binutils-common
  binutils-x86-64-linux-gnu build-essential
  dpkg-dev fakeroot g++ g++-11 gcc-11 git-man
  javascript-common libalgorithm-diff-perl
  libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils
  libc-dev-bin libc-devtools libc6-dev libcc1-0
  libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl
```



```
habibun@hemel-22241042:~$ sudo apt-get install python3-coverage python3-dev pyth
on3-nose \
                     python3-xattr python3-eventlet \
                     python3-greenlet python3-pastedeploy \
                     python3-netifaces python3-pip python3-dnspython \
                     python3-mock
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-netifaces is already the newest version (0.11.0-1build2).
python3-netifaces set to manually installed.
python3-dev is already the newest version (3.10.6-1~22.04).
python3-dev set to manually installed.
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.4).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python-pastedeploy-tpl python3-openssl python3-paste python3-pastescript
```

Then we use the command to give the root access now
root@hemel-22241042 will be occur and get all the permissions to do any change

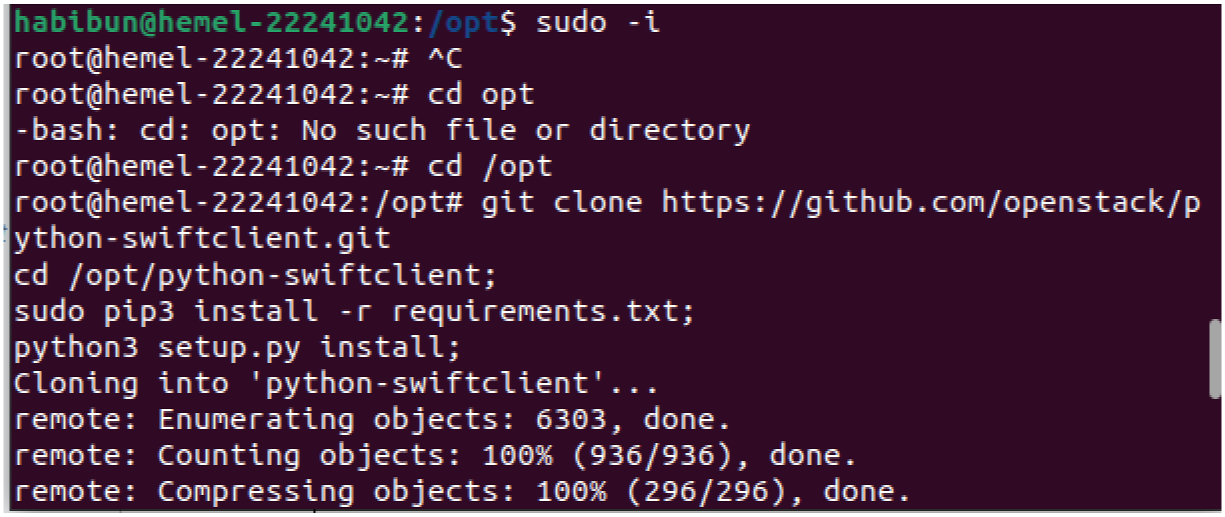
```
sudo -i
```



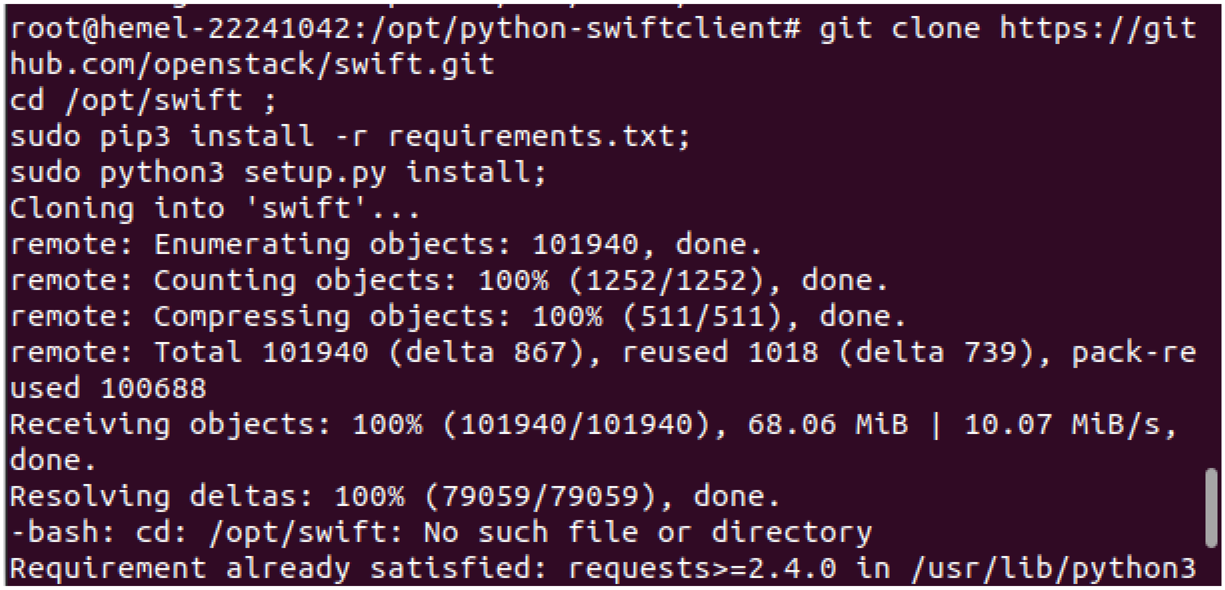
```
habibun@hemel-22241042:/opt$ sudo -i
root@hemel-22241042:~# ^C
root@hemel-22241042:~# 
```

Then after directing to the opt directory we use the following command to clone the swift client and swift :

```
cd /opt

git clone https://github.com/openstack/python-swiftclient.git
cd /opt/python-swiftclient;
sudo pip3 install -r requirements.txt;
python3 setup.py install;



git clone https://github.com/openstack/swift.git
cd /opt/swift ;
sudo pip3 install -r requirements.txt;
sudo python3 setup.py install;
```



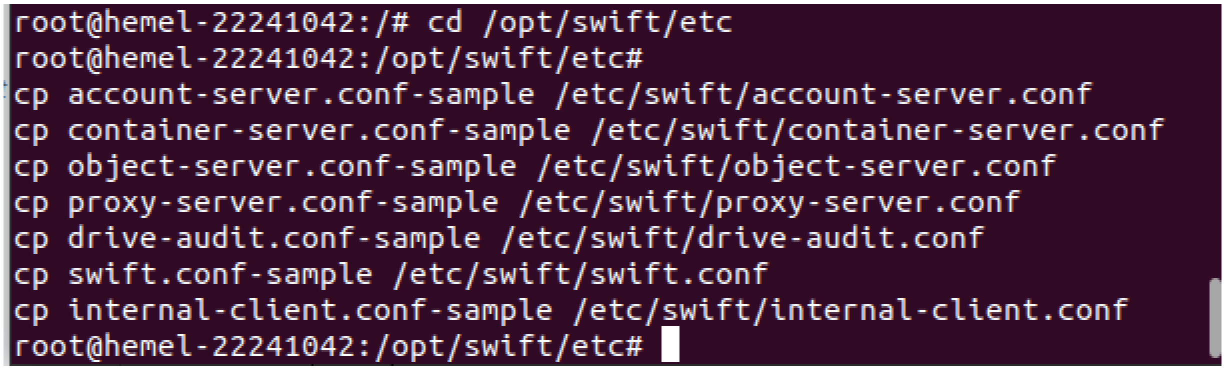
```
habibun@hemel-22241042:/opt$ sudo -i
root@hemel-22241042:~# ^C
root@hemel-22241042:~# cd opt
-bash: cd: opt: No such file or directory
root@hemel-22241042:~# cd /opt
root@hemel-22241042:/opt# git clone https://github.com/openstack/p
ython-swiftclient.git
cd /opt/python-swiftclient;
sudo pip3 install -r requirements.txt;
python3 setup.py install;
Cloning into 'python-swiftclient'...
remote: Enumerating objects: 6303, done.
remote: Counting objects: 100% (936/936), done.
remote: Compressing objects: 100% (296/296), done.
```



```
root@hemel-22241042:/opt/python-swiftclient# git clone https://git
hub.com/openstack/swift.git
cd /opt/swift ;
sudo pip3 install -r requirements.txt;
sudo python3 setup.py install;
Cloning into 'swift'...
remote: Enumerating objects: 101940, done.
remote: Counting objects: 100% (1252/1252), done.
remote: Compressing objects: 100% (511/511), done.
remote: Total 101940 (delta 867), reused 1018 (delta 739), pack-re
used 100688
Receiving objects: 100% (101940/101940), 68.06 MiB | 10.07 MiB/s,
done.
Resolving deltas: 100% (79059/79059), done.
-bash: cd: /opt/swift: No such file or directory
Requirement already satisfied: requests>=2.4.0 in /usr/lib/python3
```

Firstly i create a directory Now we are going to make the directory for our swift.Now I will copy the files from /opt/swift/etc/ to /etc/swift/

```
cd ..

mkdir -p /etc/swift

cd ..


cd /opt/swift/etc


cp account-server.conf-sample /etc/swift/account-server.conf
cp container-server.conf-sample /etc/swift/container-server.conf
cp object-server.conf-sample /etc/swift/object-server.conf
cp proxy-server.conf-sample /etc/swift/proxy-server.conf
cp drive-audit.conf-sample /etc/swift/drive-audit.conf
cp swift.conf-sample /etc/swift/swift.conf
cp internal-client.conf-sample /etc/swift/internal-client.conf
```



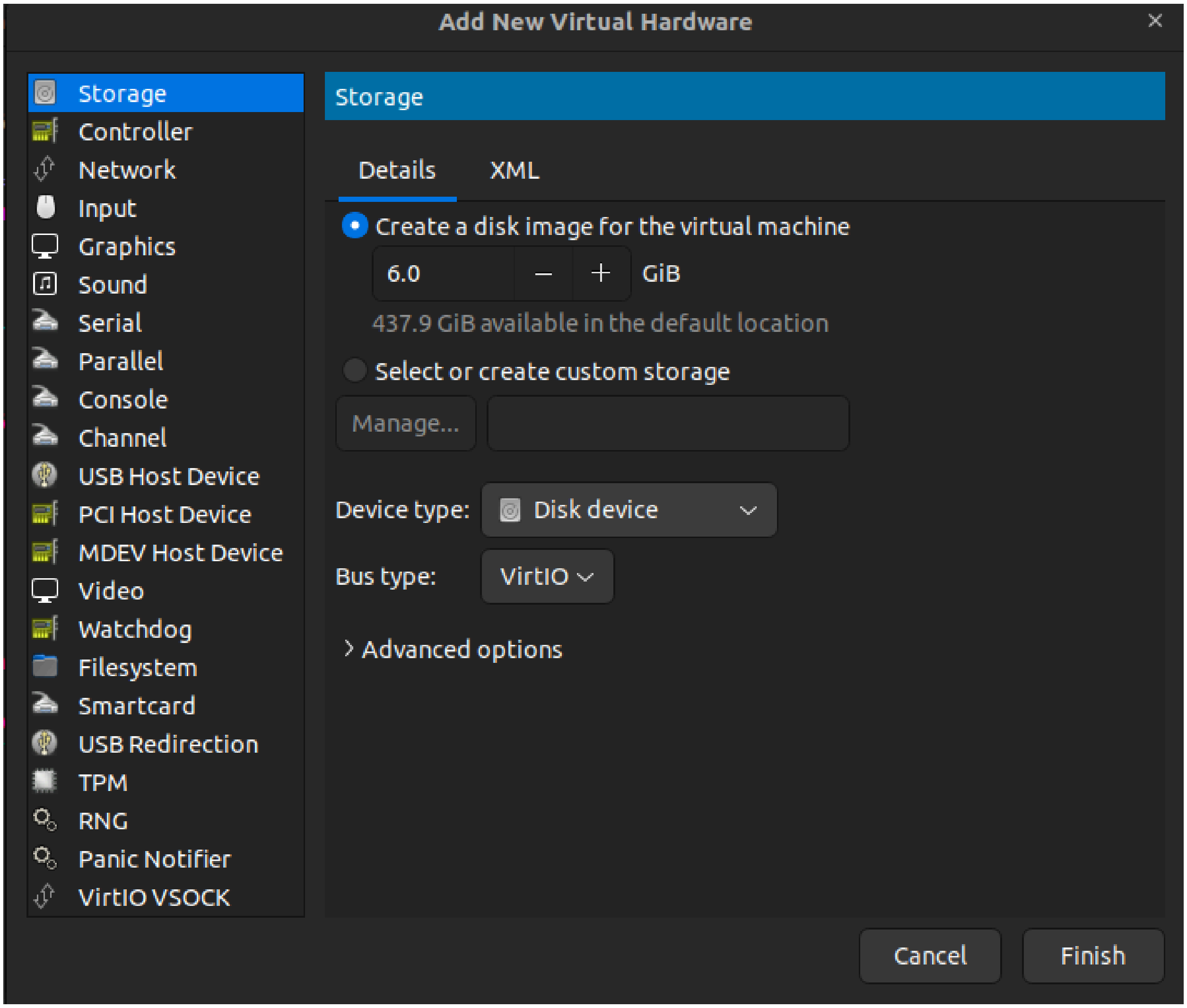
```
root@hemel-22241042:/# cd /opt/swift/etc
root@hemel-22241042:/opt/swift/etc#
cp account-server.conf-sample /etc/swift/account-server.conf
cp container-server.conf-sample /etc/swift/container-server.conf
cp object-server.conf-sample /etc/swift/object-server.conf
cp proxy-server.conf-sample /etc/swift/proxy-server.conf
cp drive-audit.conf-sample /etc/swift/drive-audit.conf
cp swift.conf-sample /etc/swift/swift.conf
cp internal-client.conf-sample /etc/swift/internal-client.conf
root@hemel-22241042:/opt/swift/etc#
```

Mounting virtual disks/the Drives and creating ring.

maintained
- Use three virtual hard disks for setup purposes.
-Partition should be 8. i.e. in ring builder change the value of to 3. Use replication value to 3 also.

You will need to add three 6GB(as mentioned in the project task) virtual drives if you are working in a KVM. Go to Show virtual hardware details first, then pick Add hardware and finally select Storage. You can decide how much storage you need in a new window that opens. Click 'Finish' to add the desired amount after you've entered it.

I have done using the GUI and additionally added 3 harddisk with 6gb(as mentioned in question) each



Add New Virtual Hardware

Storage

● Create a disk image for the virtual machine

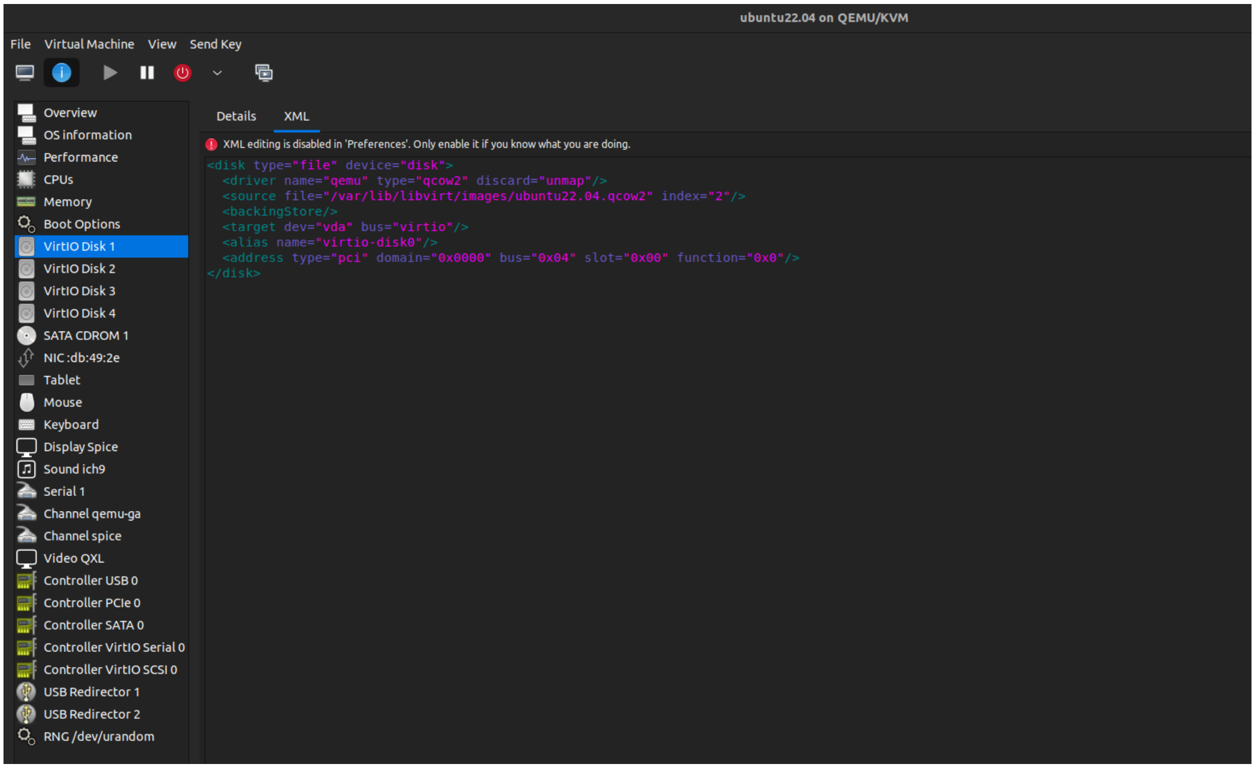
6.0 GiB

437.9 GiB available in the default location

○ Select or create custom storage

Device type: Disk device

Bus type: VirtIO

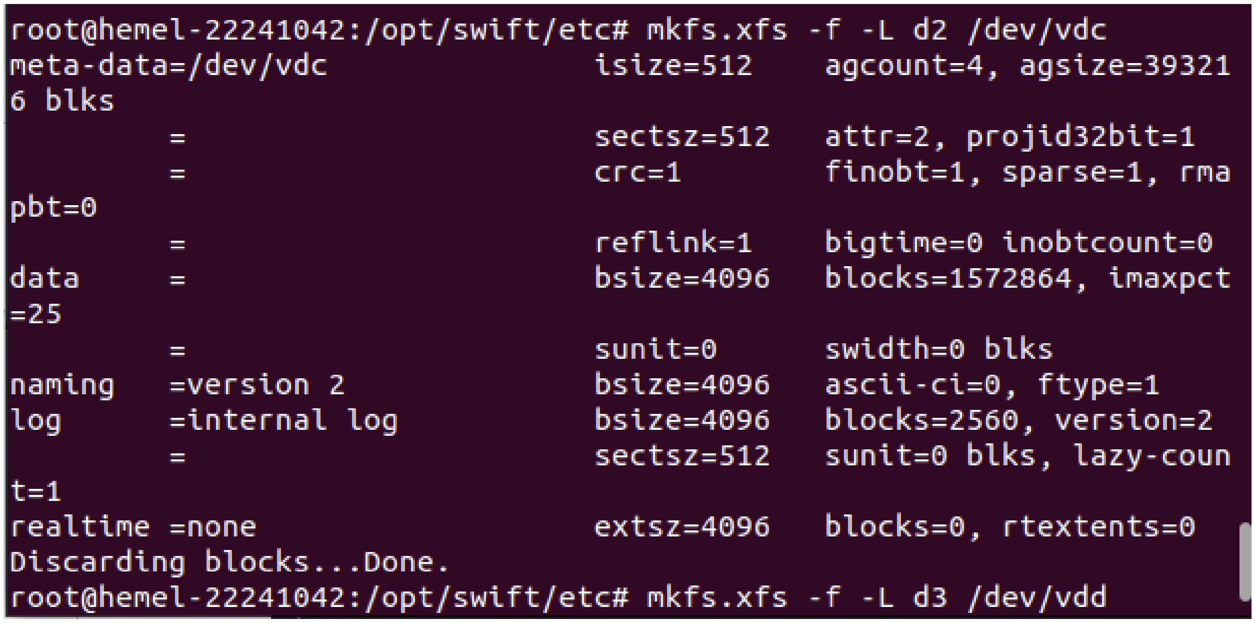


XML editing is disabled in 'Preferences'. Only enable it if you know what you are doing.

```
<disk type="file" device="disk">
  <driver name="qemu" type="qcow2" discard="unmap"/>
  <source file="/var/lib/libvirt/images/ubuntu22.04.qcow2" index="2"/>
  <backingStore/>
  <target dev="vda" bus="virtio"/>
  <alias name="virtio-disk0"/>
  <address type="pci" domain="0x0000" bus="0x04" slot="0x00" function="0x0"/>
</disk>
```

Now that the Disks are formatted in this XFS file system,

```
mkfs.xfs -f -L d1 /dev/vdb
mkfs.xfs -f -L d2 /dev/vdc
mkfs.xfs -f -L d3 /dev/vdd
```



```
root@hemel-22241042:/opt/swift/etc# mkfs.xfs -f -L d2 /dev/vdc
meta-data=/dev/vdc               isize=512    agcount=4, agsize=39321
6 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rma
pbt=0
         =                       reflink=1    bigtime=0 inobtcount=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-coun
t=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
root@hemel-22241042:/opt/swift/etc# mkfs.xfs -f -L d3 /dev/vdd
```

Now we are going to create three nodes for the files systems here :

```
mkdir -p /srv/node/d1
mkdir -p /srv/node/d2
mkdir -p /srv/node/d3
```

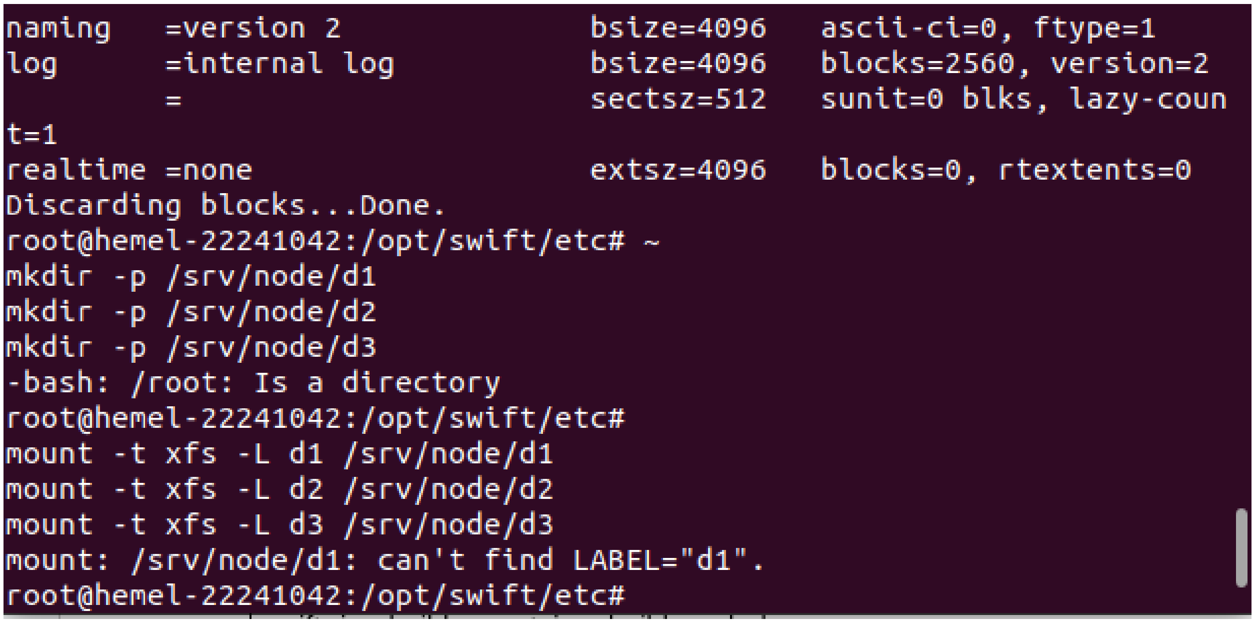
Our VM is now mounting the disks:

```
mount -t xfs -L d1 /srv/node/d1
mount -t xfs -L d2 /srv/node/d2
mount -t xfs -L d3 /srv/node/d3
```

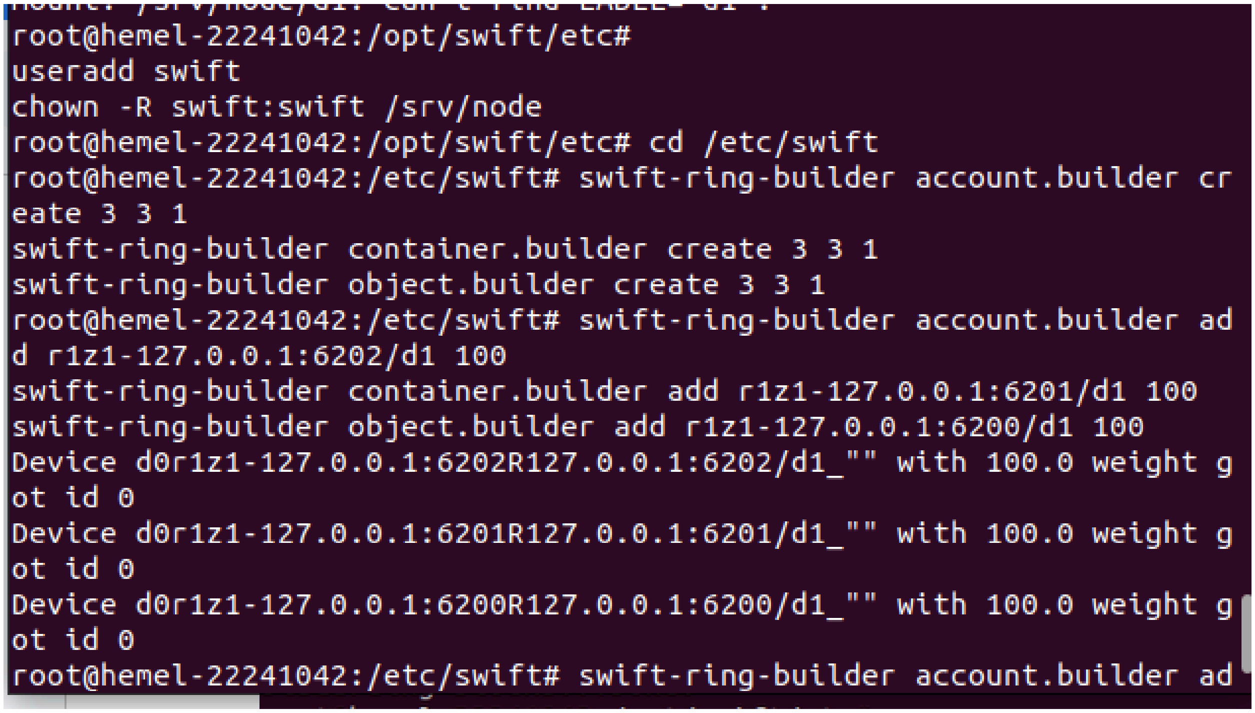
Now lets create a user swift and give it permissions to the nodes. Then we will again go to /etc/swift/ .Now we will build the rings.Now I will add the devices to the ring.swift-ring-builder account

```
useradd swift
chown -R swift:swift /srv/node

cd /etc/swift

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

swift-ring-builder account.builder add r1z1-127.0.0.1:6202/d1 100
swift-ring-builder container.builder add r1z1-127.0.0.1:6201/d1 100
swift-ring-builder object.builder add r1z1-127.0.0.1:6200/d1 100

swift-ring-builder account.builder add r1z2-127.0.0.1:6202/d2 100
swift-ring-builder container.builder add r1z2-127.0.0.1:6201/d2 100
swift-ring-builder object.builder add r1z2-127.0.0.1:6200/d2 100

swift-ring-builder account.builder add r1z3-127.0.0.1:6202/d3 100
swift-ring-builder container.builder add r1z3-127.0.0.1:6201/d3 100
swift-ring-builder object.builder add r1z3-127.0.0.1:6200/d3 100

swift-ring-builder account.builder rebalance
```

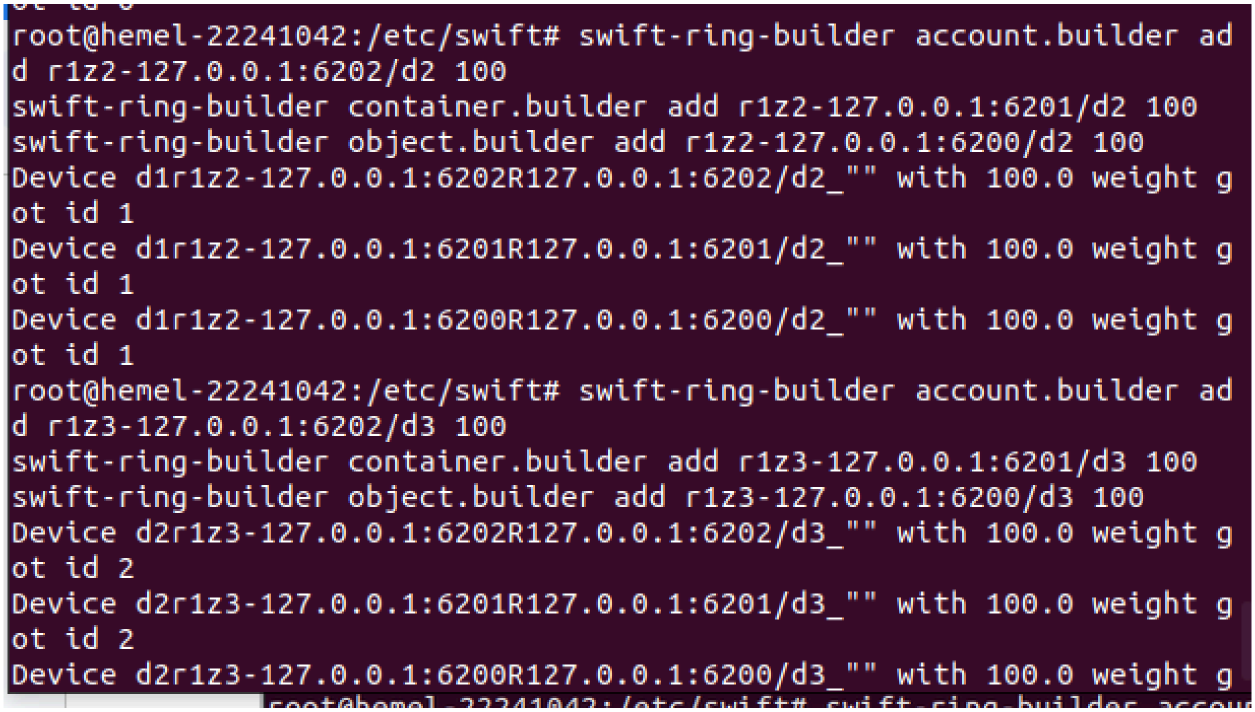
```
swift-ring-builder container.builder rebalance
swift-ring-builder object.builder rebalance
```



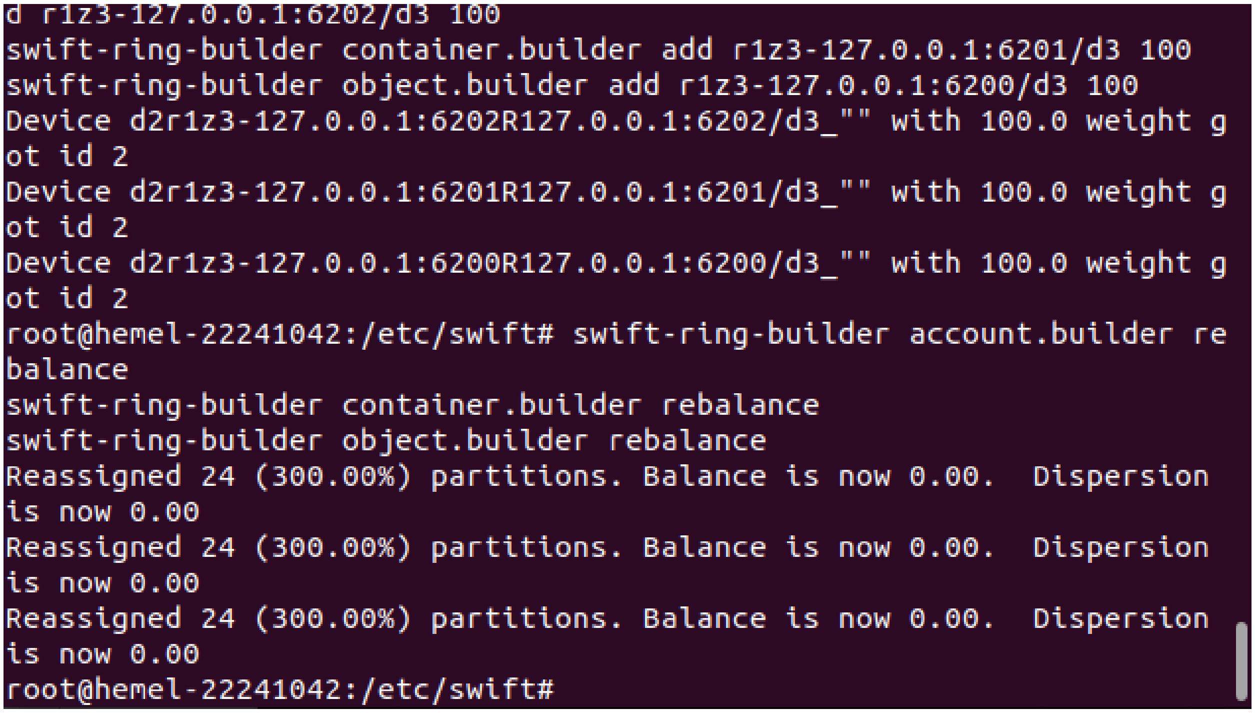
```
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-coun
t=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
root@hemel-22241042:/opt/swift/etc# ~
mkdir -p /srv/node/d1
mkdir -p /srv/node/d2
mkdir -p /srv/node/d3
-bash: /root: Is a directory
root@hemel-22241042:/opt/swift/etc#
mount -t xfs -L d1 /srv/node/d1
mount -t xfs -L d2 /srv/node/d2
mount -t xfs -L d3 /srv/node/d3
mount: /srv/node/d1: can't find LABEL="d1".
root@hemel-22241042:/opt/swift/etc#
```



```
mount: /srv/node/d1: can't find LABEL="d1".
root@hemel-22241042:/opt/swift/etc#
useradd swift
chown -R swift:swift /srv/node
root@hemel-22241042:/opt/swift/etc# cd /etc/swift
root@hemel-22241042:/etc/swift# swift-ring-builder account.builder cr
eate 3 3 1
swift-ring-builder container.builder create 3 3 1
swift-ring-builder object.builder create 3 3 1
root@hemel-22241042:/etc/swift# swift-ring-builder account.builder ad
d r1z1-127.0.0.1:6202/d1 100
swift-ring-builder container.builder add r1z1-127.0.0.1:6201/d1 100
swift-ring-builder object.builder add r1z1-127.0.0.1:6200/d1 100
Device d0r1z1-127.0.0.1:6202R127.0.0.1:6202/d1_"" with 100.0 weight g
ot id 0
Device d0r1z1-127.0.0.1:6201R127.0.0.1:6201/d1_"" with 100.0 weight g
ot id 0
Device d0r1z1-127.0.0.1:6200R127.0.0.1:6200/d1_"" with 100.0 weight g
ot id 0
root@hemel-22241042:/etc/swift# swift-ring-builder account.builder ad
```



```
ot id 0
root@hemel-22241042:/etc/swift# swift-ring-builder account.builder ad
d r1z2-127.0.0.1:6202/d2 100
swift-ring-builder container.builder add r1z2-127.0.0.1:6201/d2 100
swift-ring-builder object.builder add r1z2-127.0.0.1:6200/d2 100
Device d1r1z2-127.0.0.1:6202R127.0.0.1:6202/d2_"" with 100.0 weight g
ot id 1
Device d1r1z2-127.0.0.1:6201R127.0.0.1:6201/d2_"" with 100.0 weight g
ot id 1
Device d1r1z2-127.0.0.1:6200R127.0.0.1:6200/d2_"" with 100.0 weight g
ot id 1
root@hemel-22241042:/etc/swift# swift-ring-builder account.builder ad
d r1z3-127.0.0.1:6202/d3 100
swift-ring-builder container.builder add r1z3-127.0.0.1:6201/d3 100
swift-ring-builder object.builder add r1z3-127.0.0.1:6200/d3 100
Device d2r1z3-127.0.0.1:6202R127.0.0.1:6202/d3_"" with 100.0 weight g
ot id 2
Device d2r1z3-127.0.0.1:6201R127.0.0.1:6201/d3_"" with 100.0 weight g
ot id 2
Device d2r1z3-127.0.0.1:6200R127.0.0.1:6200/d3_"" with 100.0 weight g
root@hemel-22241042:/etc/swift# swift-ring-builder accou
```

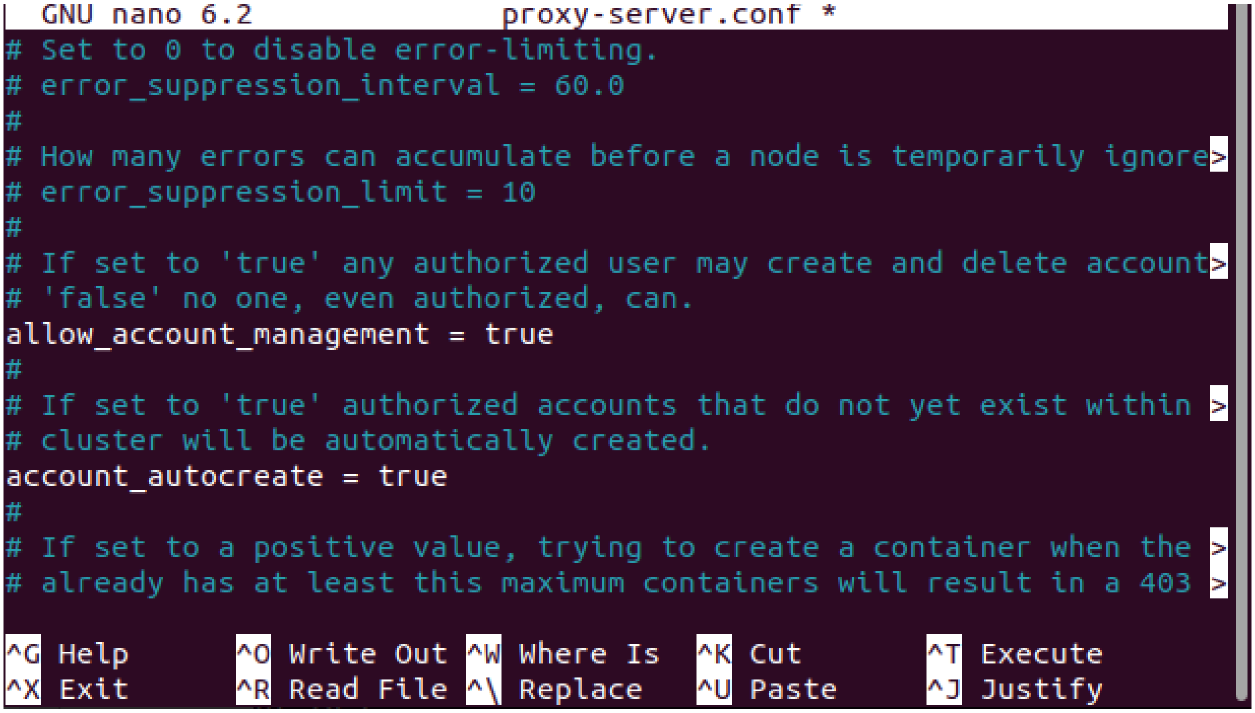


```
d r1z3-127.0.0.1:6202/d3 100
swift-ring-builder container.builder add r1z3-127.0.0.1:6201/d3 100
swift-ring-builder object.builder add r1z3-127.0.0.1:6200/d3 100
Device d2r1z3-127.0.0.1:6202R127.0.0.1:6202/d3_"" with 100.0 weight g
ot id 2
Device d2r1z3-127.0.0.1:6201R127.0.0.1:6201/d3_"" with 100.0 weight g
ot id 2
Device d2r1z3-127.0.0.1:6200R127.0.0.1:6200/d3_"" with 100.0 weight g
ot id 2
root@hemel-22241042:/etc/swift# swift-ring-builder account.builder re
balance
swift-ring-builder container.builder rebalance
swift-ring-builder object.builder rebalance
Reassigned 24 (300.00%) partitions. Balance is now 0.00.  Dispersion
is now 0.00
Reassigned 24 (300.00%) partitions. Balance is now 0.00.  Dispersion
is now 0.00
Reassigned 24 (300.00%) partitions. Balance is now 0.00.  Dispersion
is now 0.00
root@hemel-22241042:/etc/swift#
```

Once every command has been run, use **cd /etc/swift/** to get to the next place. The proxy-server configuration file can then be accessed by pasting the following command.

```
nano proxy-server.conf
```

Set the values of the following two variables to **true**.
allow_account_management = true, account_autocreate = true
After that save it by using command + s (for mac) . After save the use command+x (for mac) to exit the window



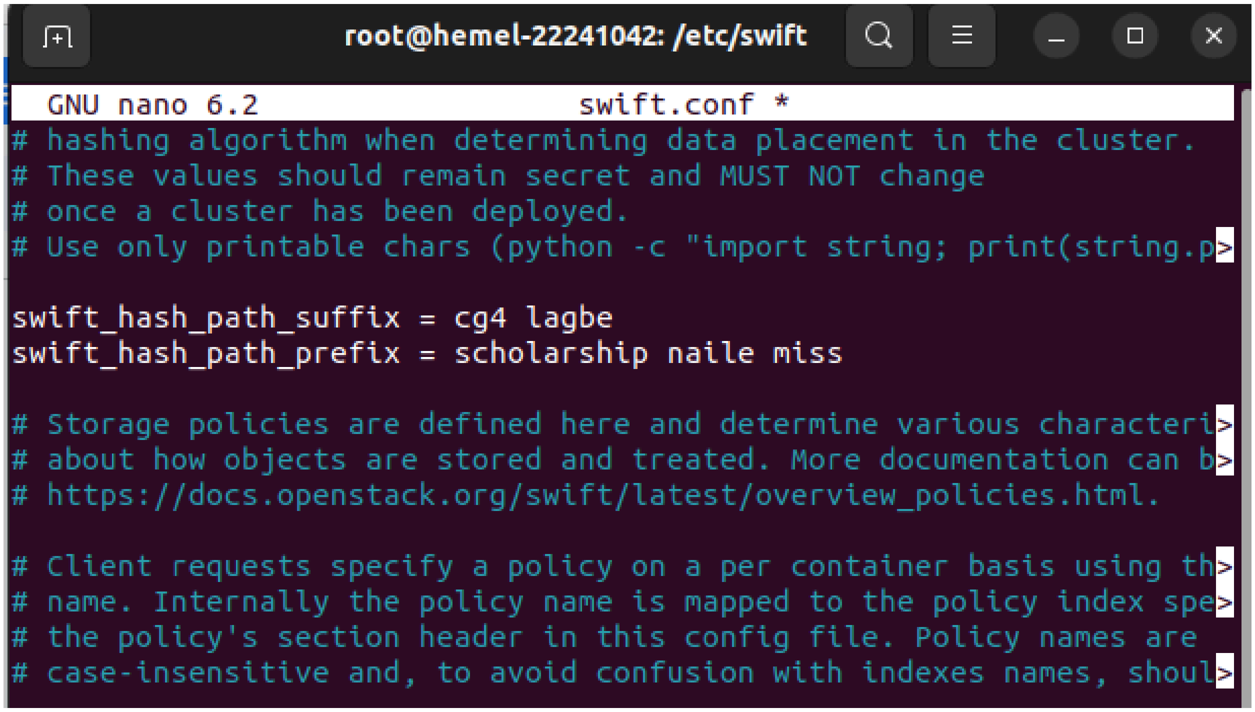
```
  GNU nano 6.2                  proxy-server.conf *
# Set to 0 to disable error-limiting.
# error_suppression_interval = 60.0
#
# How many errors can accumulate before a node is temporarily ignore>
# error_suppression_limit = 10
#
# If set to 'true' any authorized user may create and delete account>
# 'false' no one, even authorized, can.
allow_account_management = true
#
# If set to 'true' authorized accounts that do not yet exist within >
# cluster will be automatically created.
account_autocreate = true
#
# If set to a positive value, trying to create a container when the >
# already has at least this maximum containers will result in a 403 >

^G Help       ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit       ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

After completing the previous step, execute the following command to change the hash values.

```
nano swift.conf
```

From here, change the following options by direct change the value
After that save it by using command + s (for mac) . After save the use command+x (for mac) to exit the window

```
Swift_hash_path_suffix = cg4 lagbe
Swift_hash_path_prefix = scholarship  naile miss
```

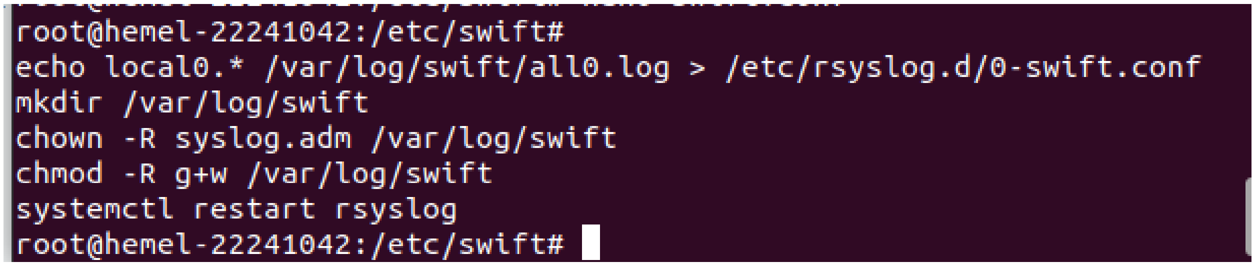


```
  GNU nano 6.2                          swift.conf *
# hashing algorithm when determining data placement in the cluster.
# These values should remain secret and MUST NOT change
# once a cluster has been deployed.
# Use only printable chars (python -c "import string; print(string.p>

swift_hash_path_suffix = cg4 lagbe
swift_hash_path_prefix = scholarship naile miss

# Storage policies are defined here and determine various characteri>
# about how objects are stored and treated. More documentation can b>
# https://docs.openstack.org/swift/latest/overview_policies.html.

# Client requests specify a policy on a per container basis using th>
# name. Internally the policy name is mapped to the policy index spe>
# the policy's section header in this config file. Policy names are
# case-insensitive and, to avoid confusion with indexes names, shoul>
```

SET LOG SYSTEM

To configure rsyslog for Swift logging, copy and paste the following command.

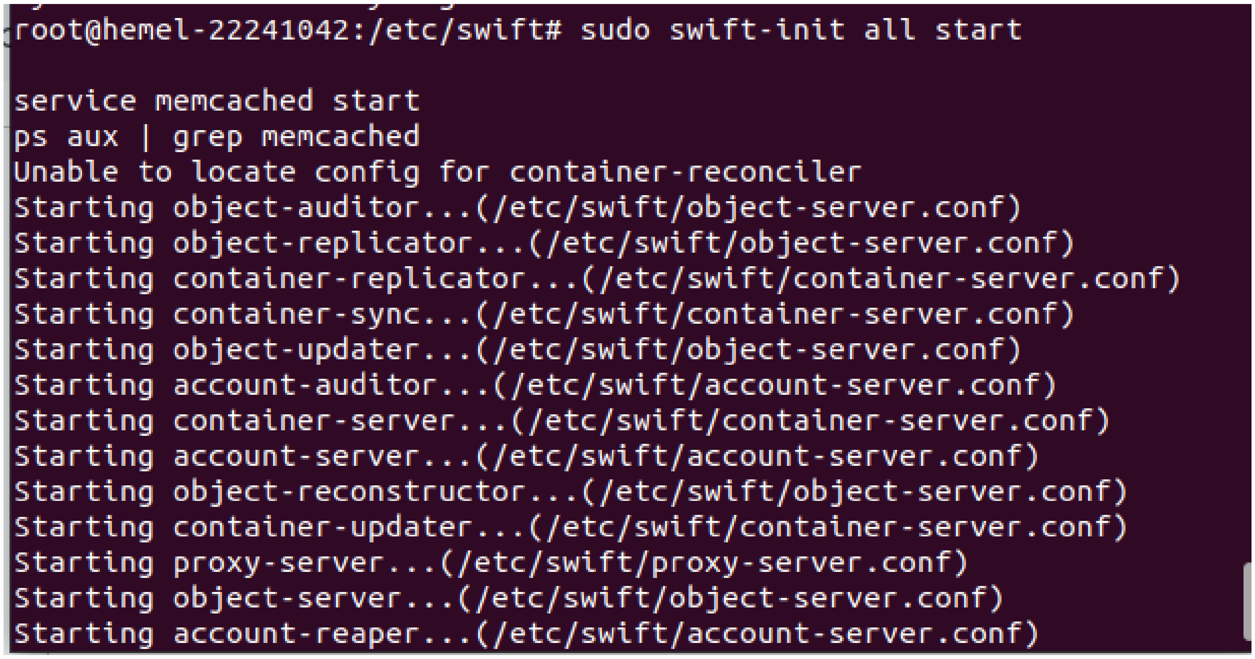
```
echo local0.* /var/log/swift/all0.log > /etc/rsyslog.d/0-swift.conf
mkdir /var/log/swift
chown -R syslog.adm /var/log/swift
chmod -R g+w /var/log/swift
systemctl restart rsyslog
```



```
root@hemel-22241042:/etc/swift#
echo local0.* /var/log/swift/all0.log > /etc/rsyslog.d/0-swift.conf
mkdir /var/log/swift
chown -R syslog.adm /var/log/swift
chmod -R g+w /var/log/swift
systemctl restart rsyslog
root@hemel-22241042:/etc/swift#
```

Launch the following command to start all the Swift-related processes.Starting All The Service and Memcache and other things in a row

```
sudo swift-init all start

service memcached start
ps aux | grep memcached
```



```
root@hemel-22241042:/etc/swift# sudo swift-init all start

service memcached start
ps aux | grep memcached
Unable to locate config for container-reconciler
Starting object-auditor...(/etc/swift/object-server.conf)
Starting object-replicator...(/etc/swift/object-server.conf)
Starting container-replicator...(/etc/swift/container-server.conf)
Starting container-sync...(/etc/swift/container-server.conf)
Starting object-updater...(/etc/swift/object-server.conf)
Starting account-auditor...(/etc/swift/account-server.conf)
Starting container-server...(/etc/swift/container-server.conf)
Starting account-server...(/etc/swift/account-server.conf)
Starting object-reconstructor...(/etc/swift/object-server.conf)
Starting container-updater...(/etc/swift/container-server.conf)
Starting proxy-server...(/etc/swift/proxy-server.conf)
Starting object-server...(/etc/swift/object-server.conf)
Starting account-reaper...(/etc/swift/account-server.conf)
```

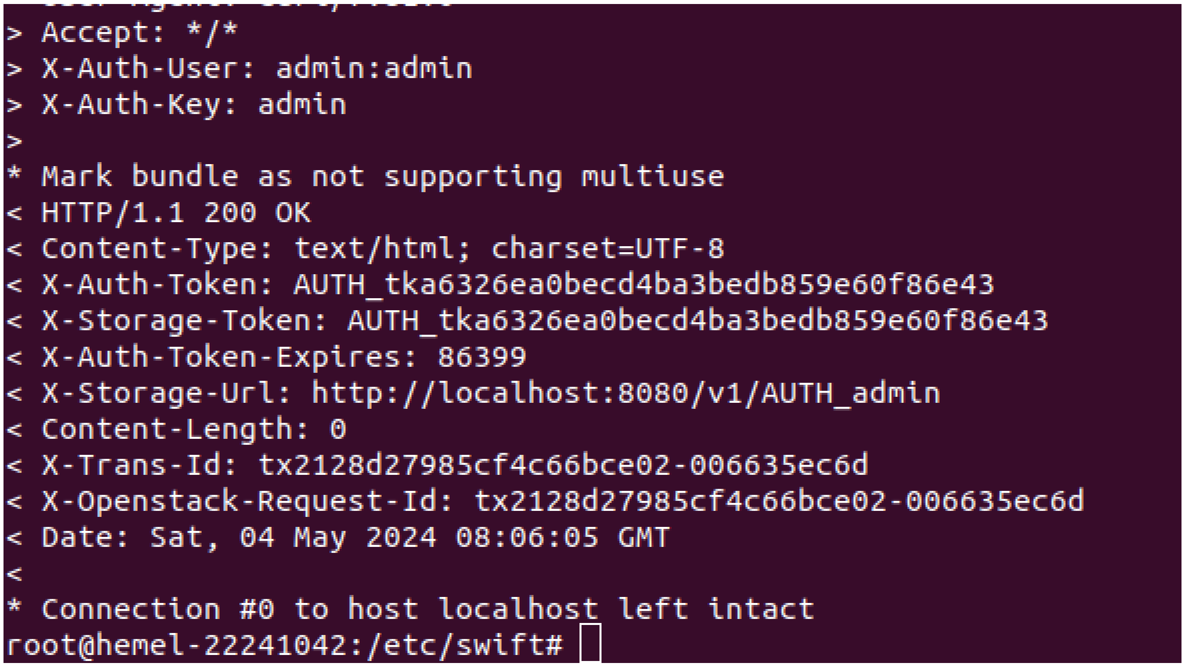
Test your authorization, authentication, and upload and download of a object successfully (Using both Curl command and Swift Client)

You can make changes to the proxy-server.conf file to add a new user. To access this file, use the code that follows.You can add a new user here. However, I will use the admin ID.
It's time to use the following code to test authentication.

```
sudo nano /etc/swift/proxy-server.conf
```

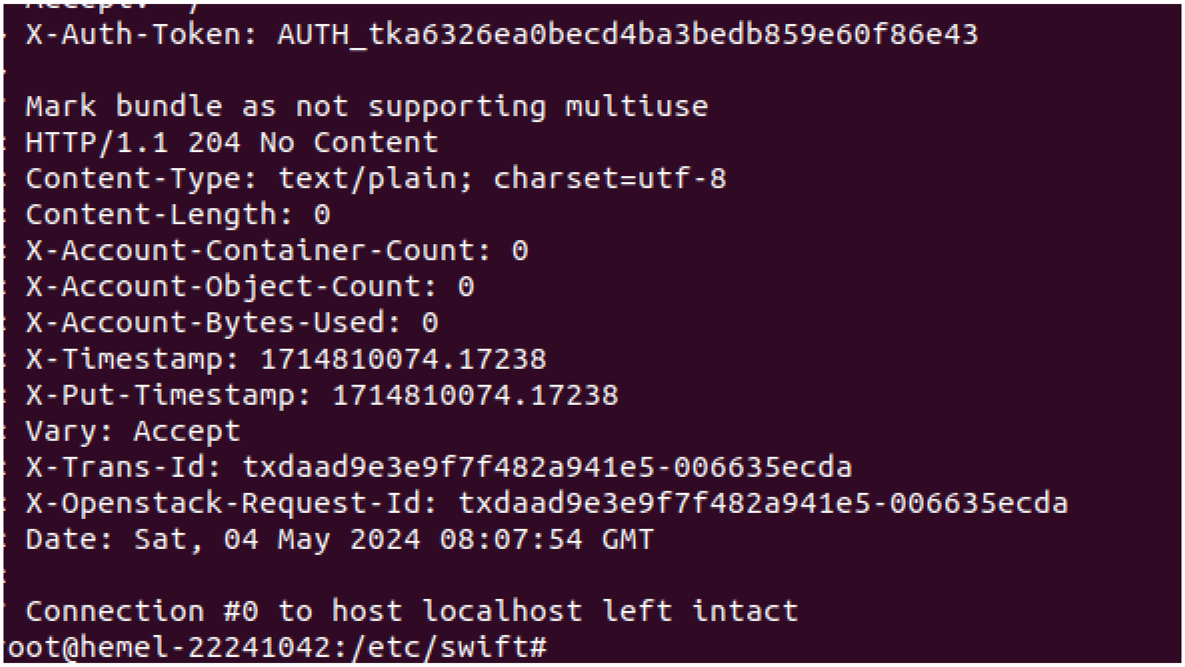
```
curl -v -H 'X-Auth-User: admin:admin' -H 'X-Auth-Key: admin' http://localhost:8080/auth/v1.0/
```

The authentication code will be needed in the future, so please make a copy of it. Additionally, keep in mind that the authentication code is reset each time the computer is restarted.
For me the token is : **X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43**
Which is collected from here.



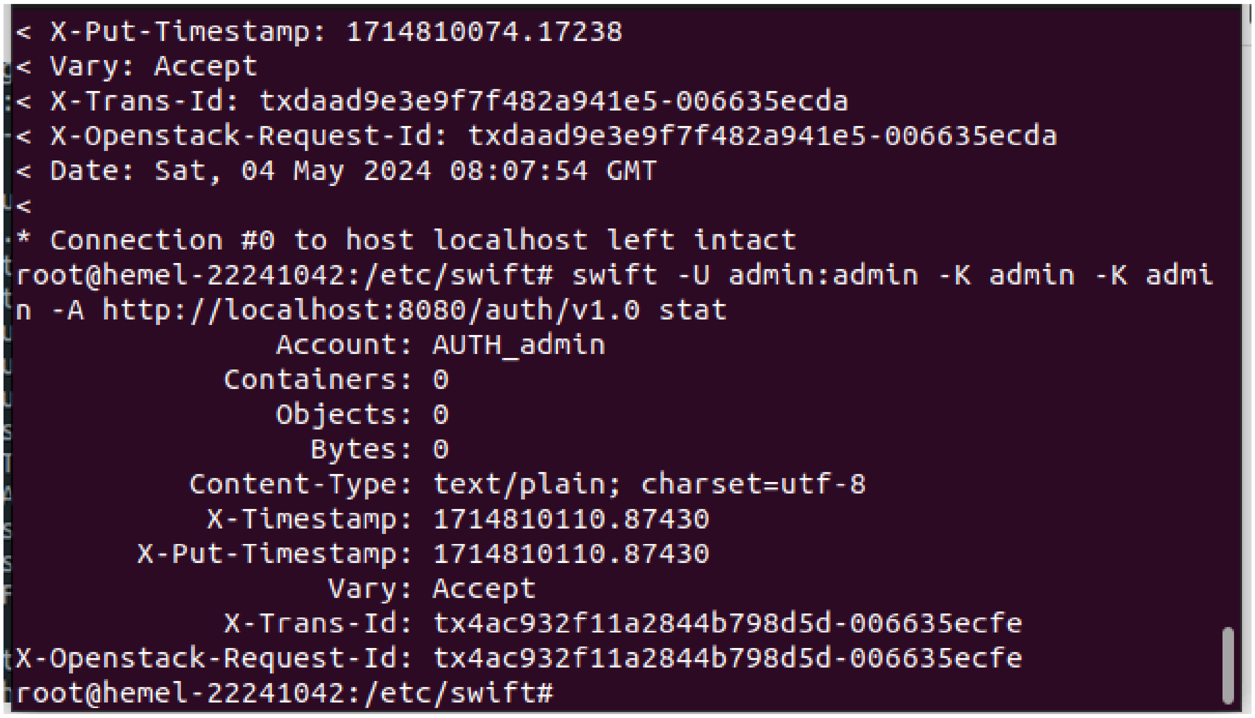
```
> Accept: */*
> X-Auth-User: admin:admin
> X-Auth-Key: admin
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: text/html; charset=UTF-8
< X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
< X-Storage-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
< X-Auth-Token-Expires: 86399
< X-Storage-Url: http://localhost:8080/v1/AUTH_admin
< Content-Length: 0
< X-Trans-Id: tx2128d27985cf4c66bce02-006635ec6d
< X-Openstack-Request-Id: tx2128d27985cf4c66bce02-006635ec6d
< Date: Sat, 04 May 2024 08:06:05 GMT
<
* Connection #0 to host localhost left intact
root@hemel-22241042:/etc/swift#
```

```
curl -v -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43'
http://localhost:8080/v1/AUTH_admin
```



```
 Accept: */*
 X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43

 Mark bundle as not supporting multiuse
 HTTP/1.1 204 No Content
 Content-Type: text/plain; charset=utf-8
 Content-Length: 0
 X-Account-Container-Count: 0
 X-Account-Object-Count: 0
 X-Account-Bytes-Used: 0
 X-Timestamp: 1714810074.17238
 X-Put-Timestamp: 1714810074.17238
 Vary: Accept
 X-Trans-Id: txdaad9e3e9f7f482a941e5-006635ecda
 X-Openstack-Request-Id: txdaad9e3e9f7f482a941e5-006635ecda
 Date: Sat, 04 May 2024 08:07:54 GMT

 Connection #0 to host localhost left intact
oot@hemel-22241042:/etc/swift#
```

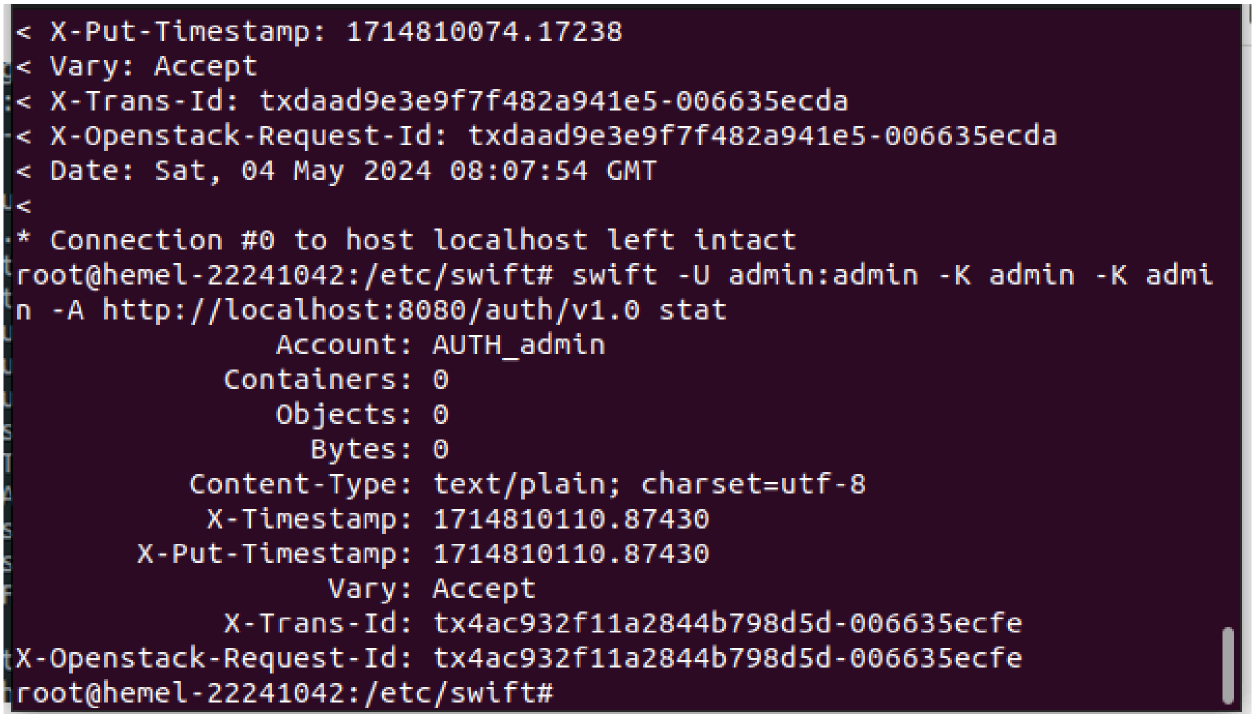
```
swift -U admin:admin -K admin -K admin -A http://localhost:8080/auth/v1.0 stat
```



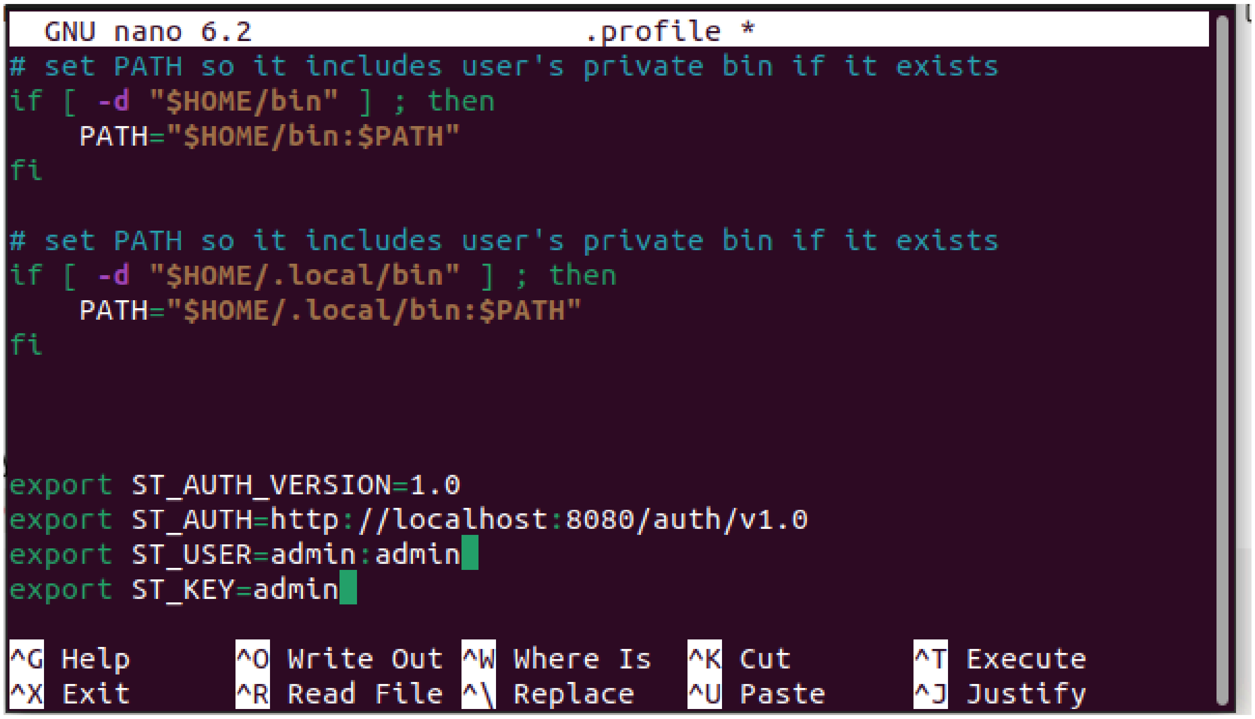
```
< X-Put-Timestamp: 1714810074.17238
< Vary: Accept
< X-Trans-Id: txdaad9e3e9f7f482a941e5-006635ecda
< X-Openstack-Request-Id: txdaad9e3e9f7f482a941e5-006635ecda
< Date: Sat, 04 May 2024 08:07:54 GMT
<
* Connection #0 to host localhost left intact
root@hemel-22241042:/etc/swift# swift -U admin:admin -K admin -K admi
n -A http://localhost:8080/auth/v1.0 stat
               Account: AUTH_admin
            Containers: 0
               Objects: 0
                 Bytes: 0
          Content-Type: text/plain; charset=utf-8
           X-Timestamp: 1714810110.87430
       X-Put-Timestamp: 1714810110.87430
                  Vary: Accept
            X-Trans-Id: tx4ac932f11a2844b798d5d-006635ecfe
X-Openstack-Request-Id: tx4ac932f11a2844b798d5d-006635ecfe
root@hemel-22241042:/etc/swift#
```

Go to /home/ and enter the following command to stop having to enter your username and password every time.Then, paste the code that follows at the end of the file.Paste the following code to make the change permanent. With the following command, the entire task can be completed.

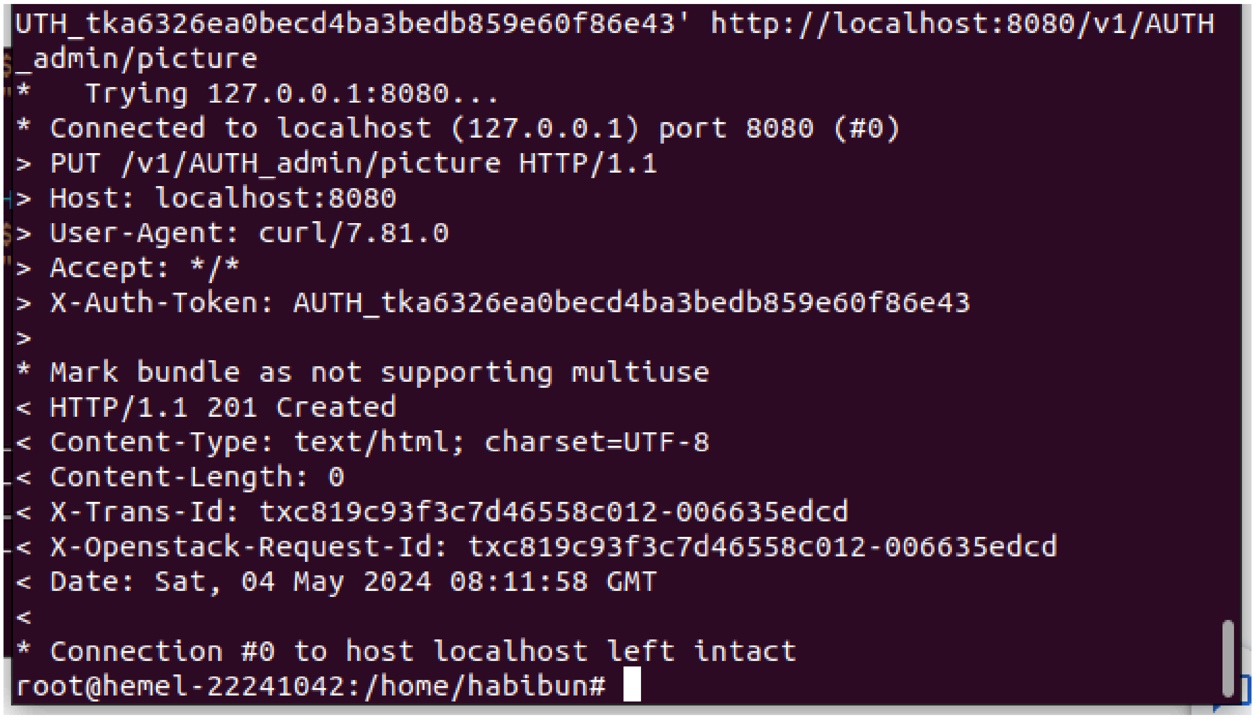
```
nano .profile


export ST_AUTH_VERSION=1.0
export ST_AUTH=http://localhost:8080/auth/v1.0
export ST_USER=admin:admin
export ST_KEY=admin



source .profile
```



```
< X-Put-Timestamp: 1714810074.17238
< Vary: Accept
< X-Trans-Id: txdaad9e3e9f7f482a941e5-006635ecda
< X-Openstack-Request-Id: txdaad9e3e9f7f482a941e5-006635ecda
< Date: Sat, 04 May 2024 08:07:54 GMT
<
* Connection #0 to host localhost left intact
root@hemel-22241042:/etc/swift# swift -U admin:admin -K admin -K admi
n -A http://localhost:8080/auth/v1.0 stat
               Account: AUTH_admin
            Containers: 0
               Objects: 0
                 Bytes: 0
          Content-Type: text/plain; charset=utf-8
           X-Timestamp: 1714810110.87430
       X-Put-Timestamp: 1714810110.87430
                  Vary: Accept
            X-Trans-Id: tx4ac932f11a2844b798d5d-006635ecfe
X-Openstack-Request-Id: tx4ac932f11a2844b798d5d-006635ecfe
root@hemel-22241042:/etc/swift#
```



```
  GNU nano 6.2                   .profile *
# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi




export ST_AUTH_VERSION=1.0
export ST_AUTH=http://localhost:8080/auth/v1.0
export ST_USER=admin:admin
export ST_KEY=admin
```



```
UTH_tka6326ea0becd4ba3bedb859e60f86e43' http://localhost:8080/v1/AUTH
_admin/picture
*   Trying 127.0.0.1:8080...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> PUT /v1/AUTH_admin/picture HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.81.0
> Accept: */*
> X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 201 Created
< Content-Type: text/html; charset=UTF-8
< Content-Length: 0
< X-Trans-Id: txc819c93f3c7d46558c012-006635edcd
< X-Openstack-Request-Id: txc819c93f3c7d46558c012-006635edcd
< Date: Sat, 04 May 2024 08:11:58 GMT
<
* Connection #0 to host localhost left intact
root@hemel-22241042:/home/habibun# 
```

Creating container and objects

Using curl

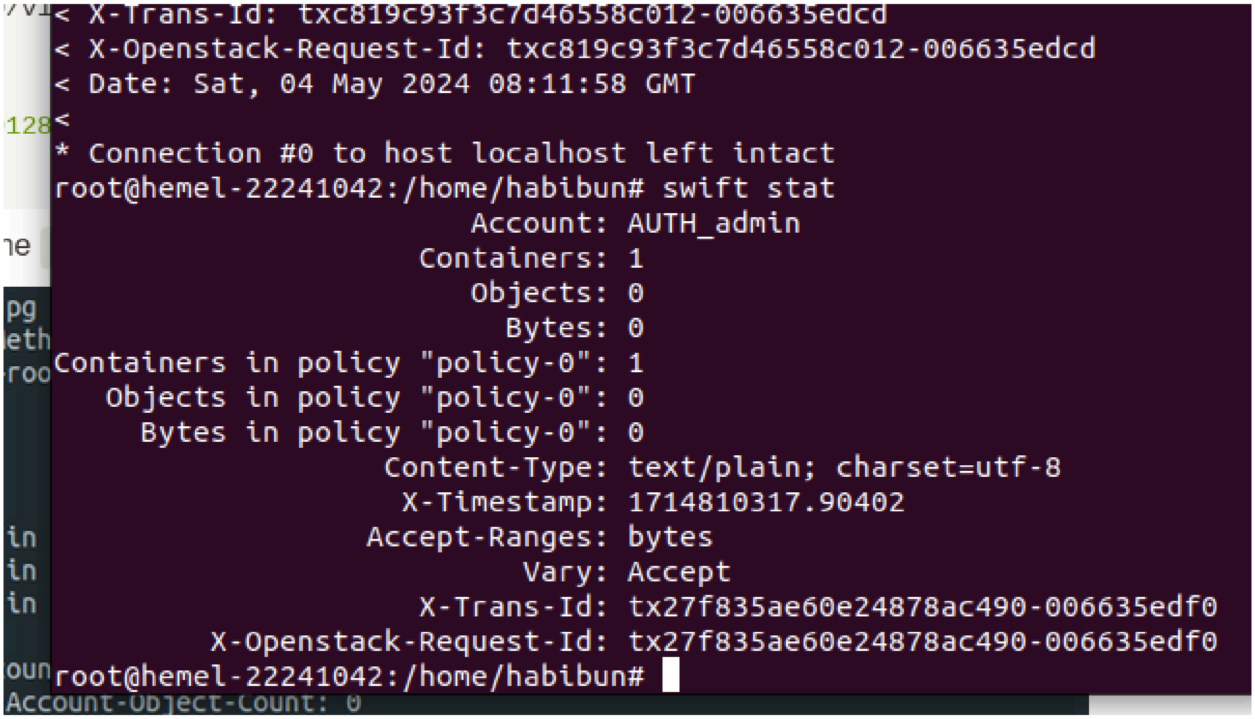
Here is the code to **create a container** using curl.

```
# for all users
curl -v -X PUT -H 'Token' http://localhost:8080/v1/AUTH_admin/container_name

#for me the change will be
curl -v -X PUT -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43'
http://localhost:8080/v1/AUTH_admin/picture
```

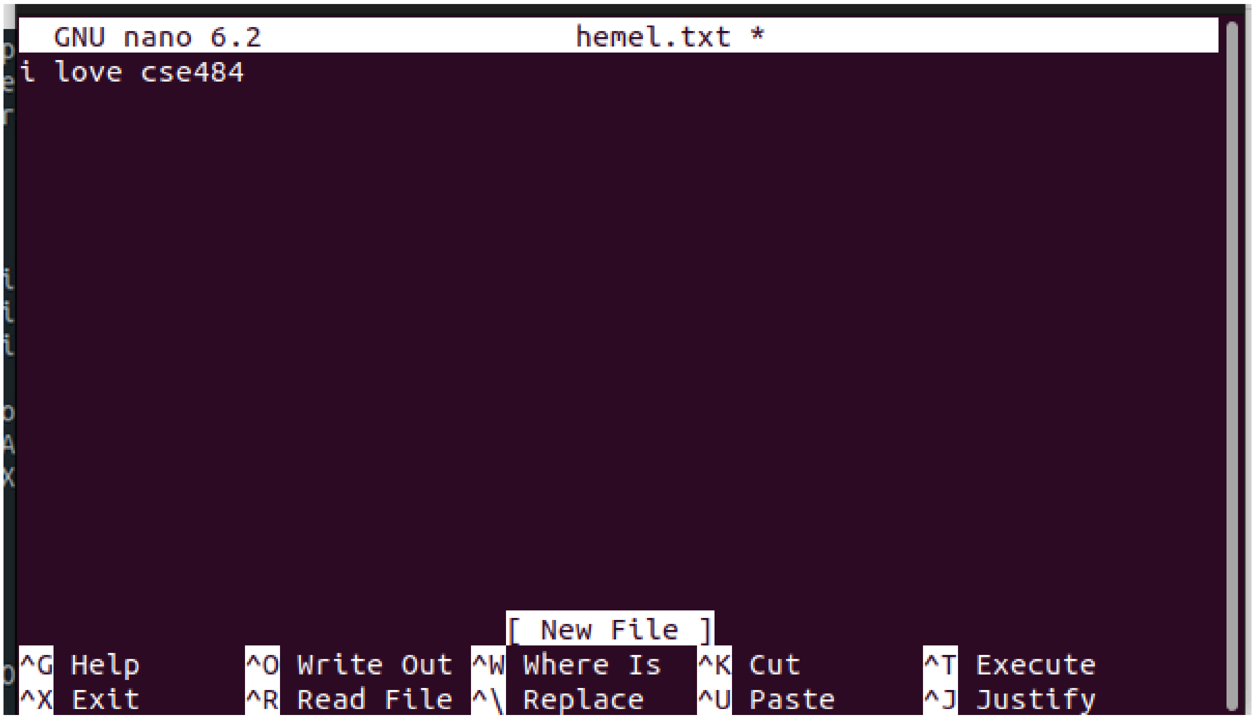
After executing the swift stat command, you will notice the container count has increased to

```
swift stat
```

Then create a text file called hemel.txt



```
< X-Trans-Id: txc819c93f3c7d46558c012-006635edcd
< X-Openstack-Request-Id: txc819c93f3c7d46558c012-006635edcd
< Date: Sat, 04 May 2024 08:11:58 GMT
<
* Connection #0 to host localhost left intact
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 1
                        Objects: 0
                          Bytes: 0
Containers in policy "policy-0": 1
   Objects in policy "policy-0": 0
     Bytes in policy "policy-0": 0
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx27f835ae60e24878ac490-006635edf0
         X-Openstack-Request-Id: tx27f835ae60e24878ac490-006635edf0
root@hemel-22241042:/home/habibun#
```

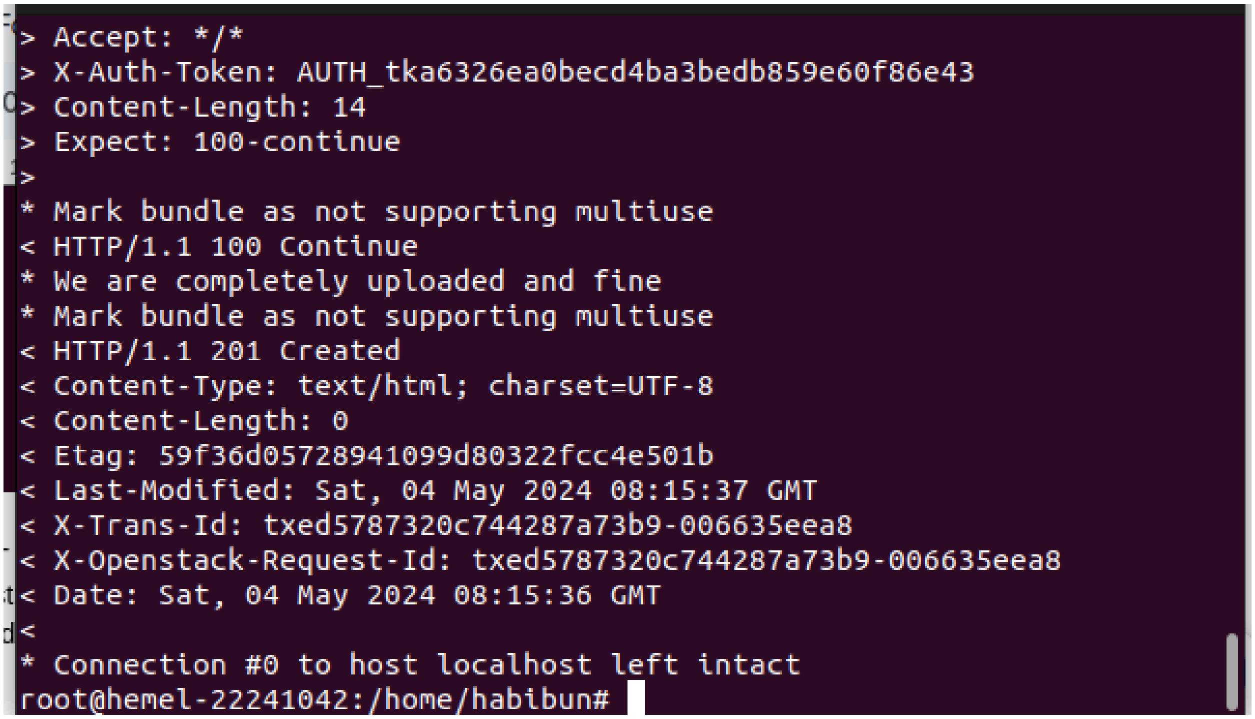


```
  GNU nano 6.2                  hemel.txt *
i love cse484





                             [ New File ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

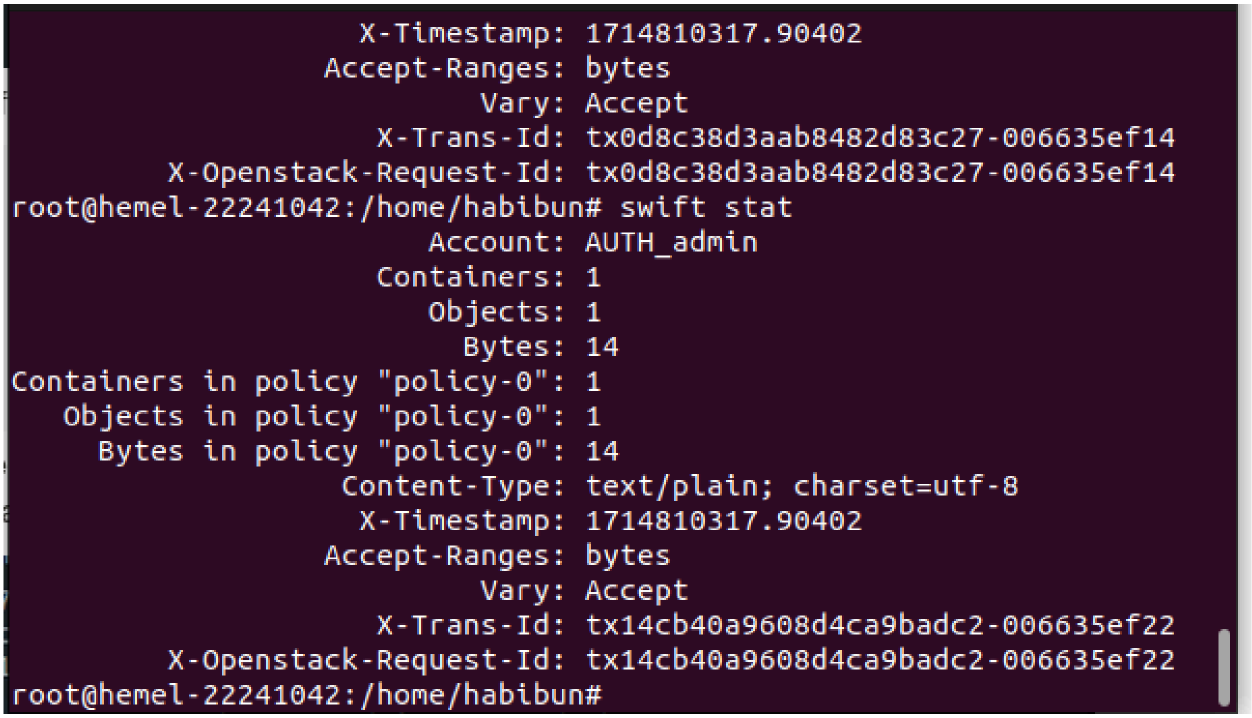
Upload it to Swift using the following command.

```
curl -v -X PUT -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43'
http://localhost:8080/v1/AUTH_admin/picture/hemel.txt -T hemel.txt
```

Navigating to the file location before executing the upload command would be a good decision. This way, you won't have to explicitly mention the location.



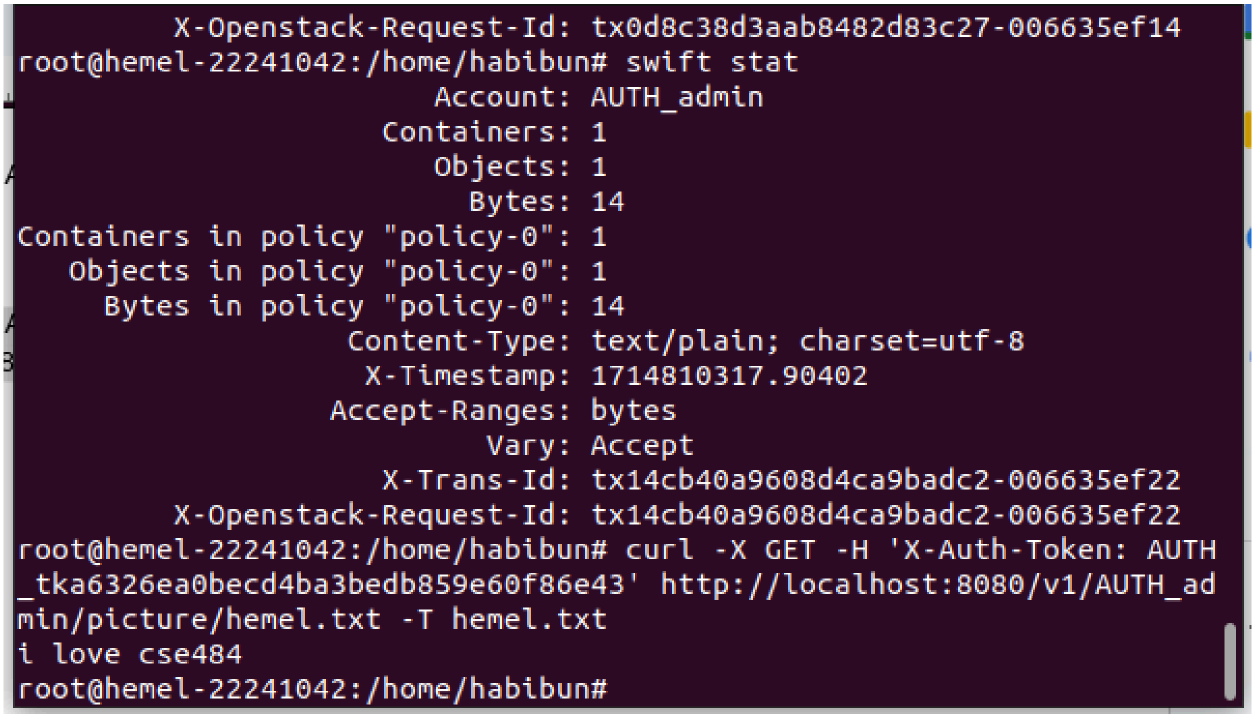
```
> Accept: */*
> X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
> Content-Length: 14
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 100 Continue
* We are completely uploaded and fine
* Mark bundle as not supporting multiuse
< HTTP/1.1 201 Created
< Content-Type: text/html; charset=UTF-8
< Content-Length: 0
< Etag: 59f36d05728941099d80322fcc4e501b
< Last-Modified: Sat, 04 May 2024 08:15:37 GMT
< X-Trans-Id: txed5787320c744287a73b9-006635eea8
< X-Openstack-Request-Id: txed5787320c744287a73b9-006635eea8
< Date: Sat, 04 May 2024 08:15:36 GMT
<
* Connection #0 to host localhost left intact
root@hemel-22241042:/home/habibun#
```



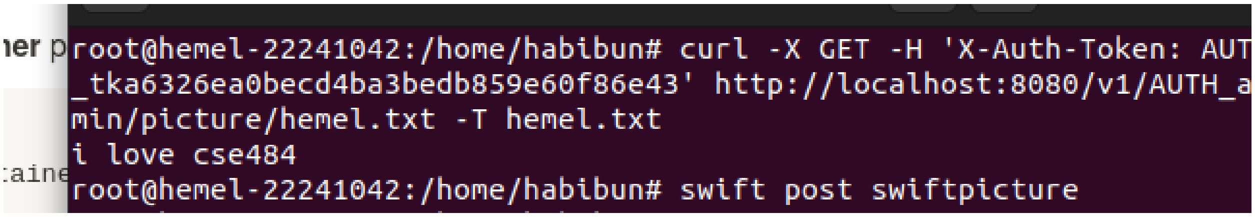
```
                  X-Timestamp: 1714810317.90402
                Accept-Ranges: bytes
                         Vary: Accept
                   X-Trans-Id: tx0d8c38d3aab8482d83c27-006635ef14
         X-Openstack-Request-Id: tx0d8c38d3aab8482d83c27-006635ef14
root@hemel-22241042:/home/habibun# swift stat
                      Account: AUTH_admin
                   Containers: 1
                      Objects: 1
                        Bytes: 14
Containers in policy "policy-0": 1
   Objects in policy "policy-0": 1
     Bytes in policy "policy-0": 14
                 Content-Type: text/plain; charset=utf-8
                  X-Timestamp: 1714810317.90402
                Accept-Ranges: bytes
                         Vary: Accept
                   X-Trans-Id: tx14cb40a9608d4ca9badc2-006635ef22
         X-Openstack-Request-Id: tx14cb40a9608d4ca9badc2-006635ef22
root@hemel-22241042:/home/habibun#
```

Now, let's download the hemel.txt file that we just uploaded.

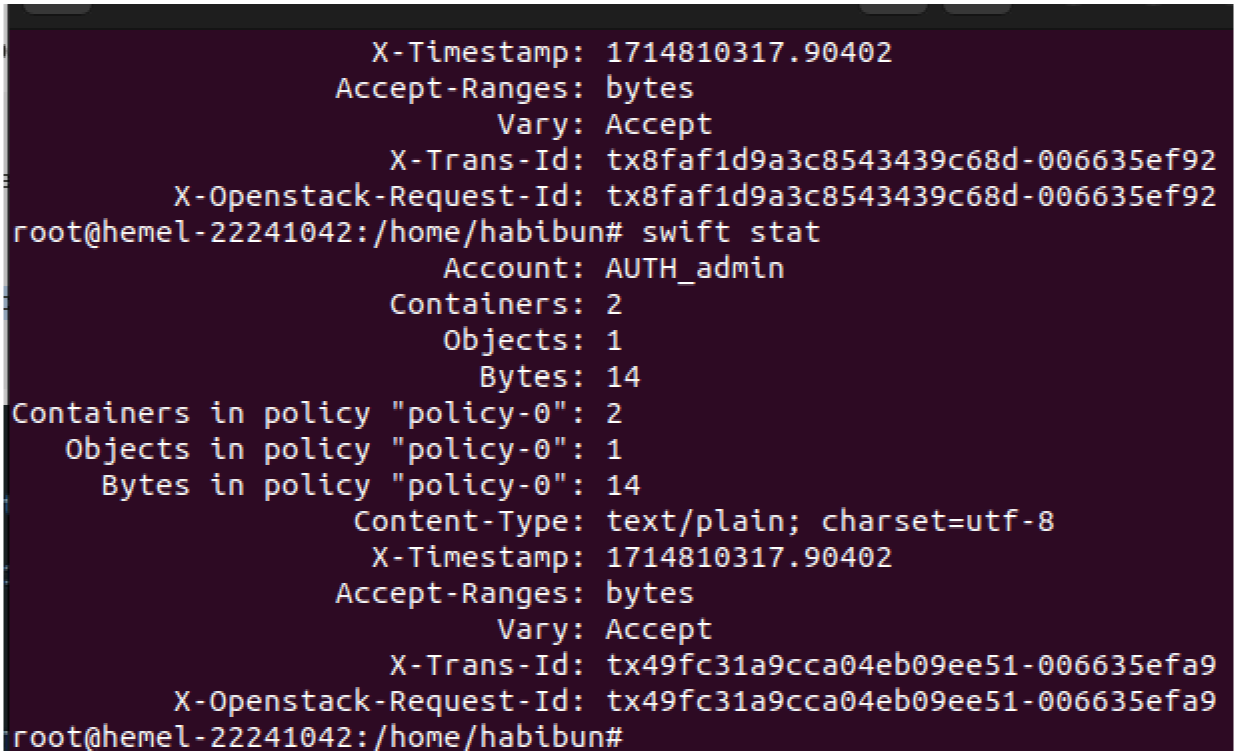
```
curl -X GET -H 'X-Auth-Token' http://localhost:8080/v1/AUTH_admin/container/object -o
<where it will be saved>

#For me
curl -X GET -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43'
http://localhost:8080/v1/AUTH_admin/picture/hemel.txt -T hemel.txt
```



```
         X-Openstack-Request-Id: tx0d8c38d3aab8482d83c27-006635ef14
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 1
                        Objects: 1
                          Bytes: 14
Containers in policy "policy-0": 1
   Objects in policy "policy-0": 1
     Bytes in policy "policy-0": 14
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx14cb40a9608d4ca9badc2-006635ef22
         X-Openstack-Request-Id: tx14cb40a9608d4ca9badc2-006635ef22
root@hemel-22241042:/home/habibun# curl -X GET -H 'X-Auth-Token: AUTH
_tka6326ea0becd4ba3bedb859e60f86e43' http://localhost:8080/v1/AUTH_ad
min/picture/hemel.txt -T hemel.txt
i love cse484
root@hemel-22241042:/home/habibun#
```



```
root@hemel-22241042:/home/habibun# curl -X GET -H 'X-Auth-Token: AUT
_tka6326ea0becd4ba3bedb859e60f86e43' http://localhost:8080/v1/AUTH_a
min/picture/hemel.txt -T hemel.txt
i love cse484
root@hemel-22241042:/home/habibun# swift post swiftpicture
```

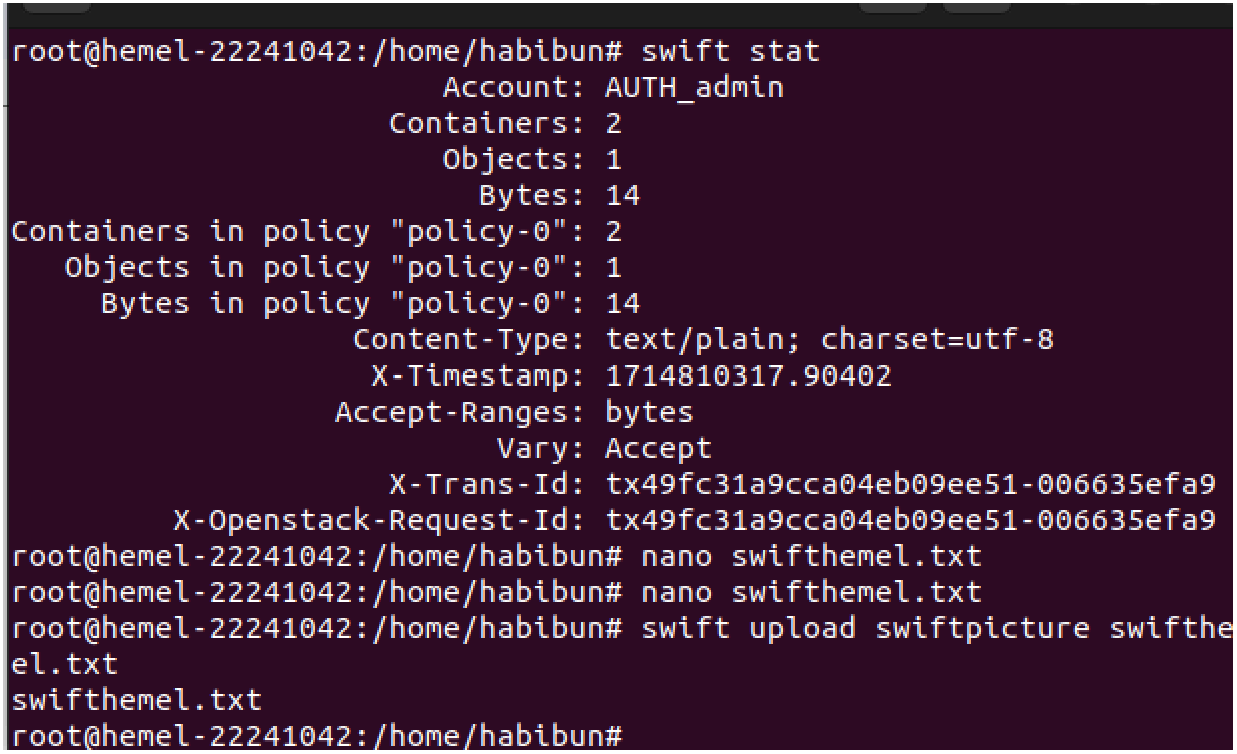


```
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx8faf1d9a3c8543439c68d-006635ef92
         X-Openstack-Request-Id: tx8faf1d9a3c8543439c68d-006635ef92
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 2
                        Objects: 1
                          Bytes: 14
Containers in policy "policy-0": 2
   Objects in policy "policy-0": 1
     Bytes in policy "policy-0": 14
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx49fc31a9cca04eb09ee51-006635efa9
         X-Openstack-Request-Id: tx49fc31a9cca04eb09ee51-006635efa9
root@hemel-22241042:/home/habibun#
```

Using swift CLI

To create an container paste the following command

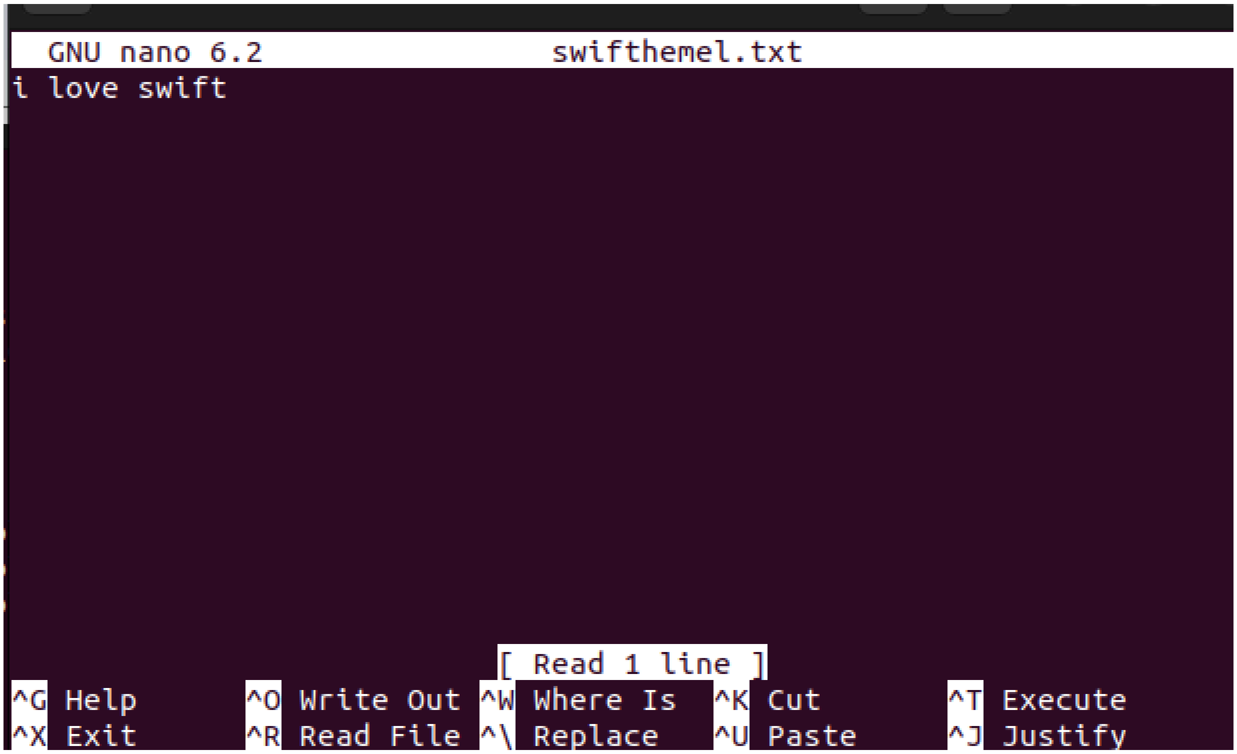
```
swift post swiftpic
```



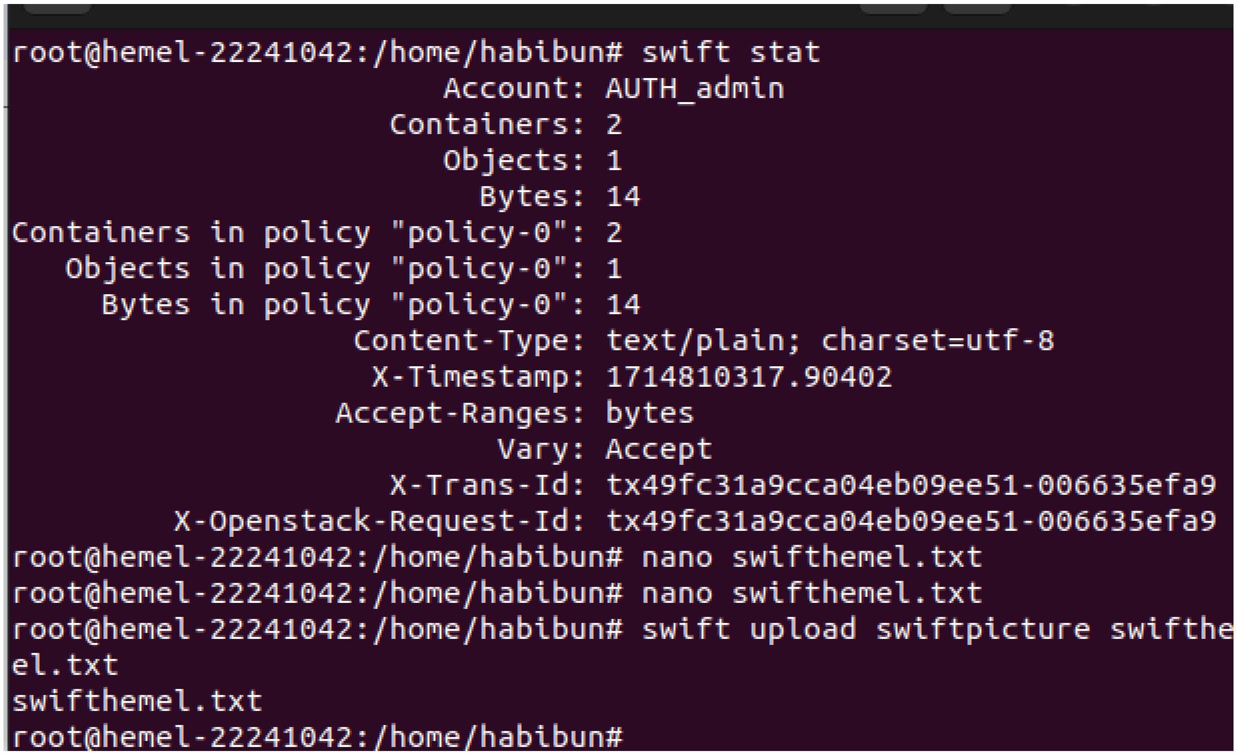
```
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 2
                        Objects: 1
                          Bytes: 14
Containers in policy "policy-0": 2
   Objects in policy "policy-0": 1
     Bytes in policy "policy-0": 14
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx49fc31a9cca04eb09ee51-006635efa9
         X-Openstack-Request-Id: tx49fc31a9cca04eb09ee51-006635efa9
root@hemel-22241042:/home/habibun# nano swifthemel.txt
root@hemel-22241042:/home/habibun# nano swifthemel.txt
root@hemel-22241042:/home/habibun# swift upload swiftpicture swifthe
el.txt
swifthemel.txt
root@hemel-22241042:/home/habibun#
```

Now, create a new text file named swifthemel.txt and upload it to the newly created container.

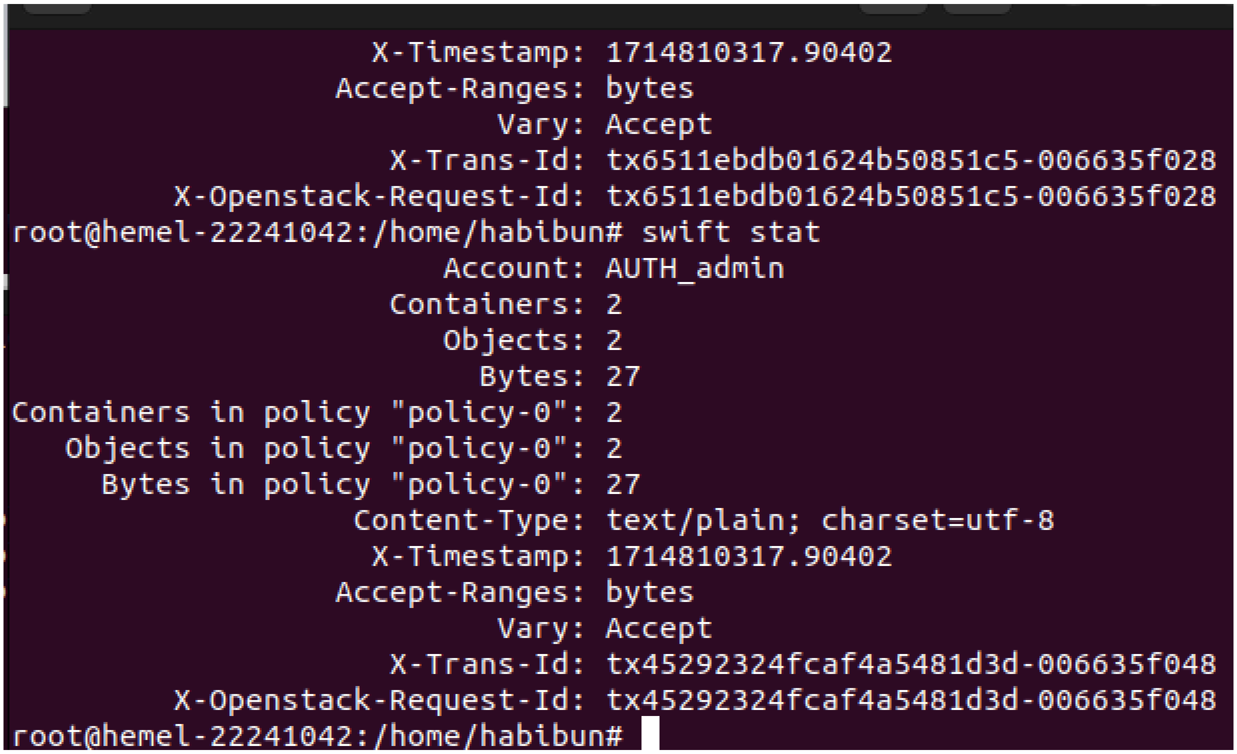
```
swift upload swiftpicture swifthemel.txt
```



```
  GNU nano 6.2                       swifthemel.txt
i love swift








                                   [ Read 1 line ]
^G Help         ^O Write Out ^W Where Is  ^K Cut          ^T Execute
^X Exit         ^R Read File ^\ Replace   ^U Paste        ^J Justify
```



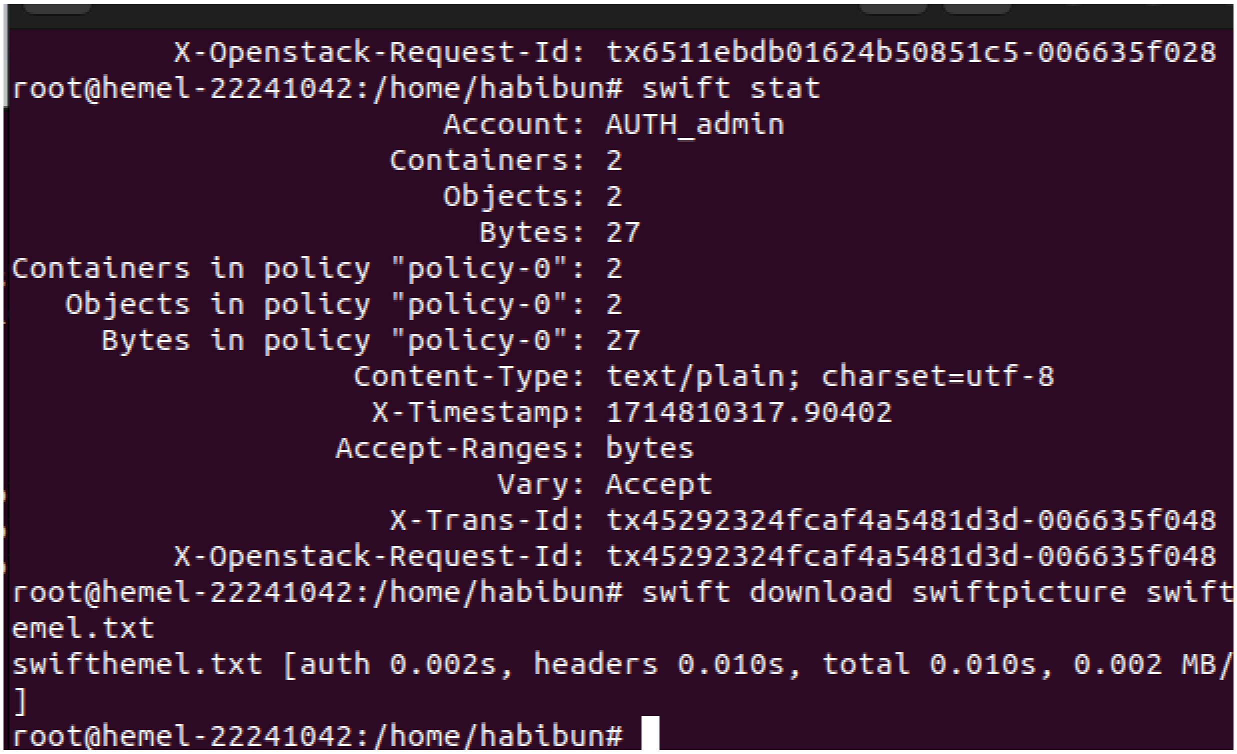
```
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 2
                        Objects: 1
                          Bytes: 14
Containers in policy "policy-0": 2
   Objects in policy "policy-0": 1
     Bytes in policy "policy-0": 14
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx49fc31a9cca04eb09ee51-006635efa9
         X-Openstack-Request-Id: tx49fc31a9cca04eb09ee51-006635efa9
root@hemel-22241042:/home/habibun# nano swifthemel.txt
root@hemel-22241042:/home/habibun# nano swifthemel.txt
root@hemel-22241042:/home/habibun# swift upload swiftpicture swifthe
el.txt
swifthemel.txt
root@hemel-22241042:/home/habibun#
```



```
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx6511ebdb01624b50851c5-006635f028
         X-Openstack-Request-Id: tx6511ebdb01624b50851c5-006635f028
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 2
                        Objects: 2
                          Bytes: 27
Containers in policy "policy-0": 2
   Objects in policy "policy-0": 2
     Bytes in policy "policy-0": 27
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx45292324fcaf4a5481d3d-006635f048
         X-Openstack-Request-Id: tx45292324fcaf4a5481d3d-006635f048
root@hemel-22241042:/home/habibun# 
```

Downloading the swifthemel.txt file.

```
swift download swiftpicture swifthemel.txt
```



```
         X-Openstack-Request-Id: tx6511ebdb01624b50851c5-006635f028
root@hemel-22241042:/home/habibun# swift stat
                        Account: AUTH_admin
                     Containers: 2
                        Objects: 2
                          Bytes: 27
Containers in policy "policy-0": 2
   Objects in policy "policy-0": 2
     Bytes in policy "policy-0": 27
                   Content-Type: text/plain; charset=utf-8
                    X-Timestamp: 1714810317.90402
                  Accept-Ranges: bytes
                           Vary: Accept
                     X-Trans-Id: tx45292324fcaf4a5481d3d-006635f048
         X-Openstack-Request-Id: tx45292324fcaf4a5481d3d-006635f048
root@hemel-22241042:/home/habibun# swift download swiftpicture swift
emel.txt
swifthemel.txt [auth 0.002s, headers 0.010s, total 0.010s, 0.002 MB/
]
root@hemel-22241042:/home/habibun# 
```

Test Other commands such as head/get/post etc. Test all thecommands of Swift Client.

Testing Few More Commands

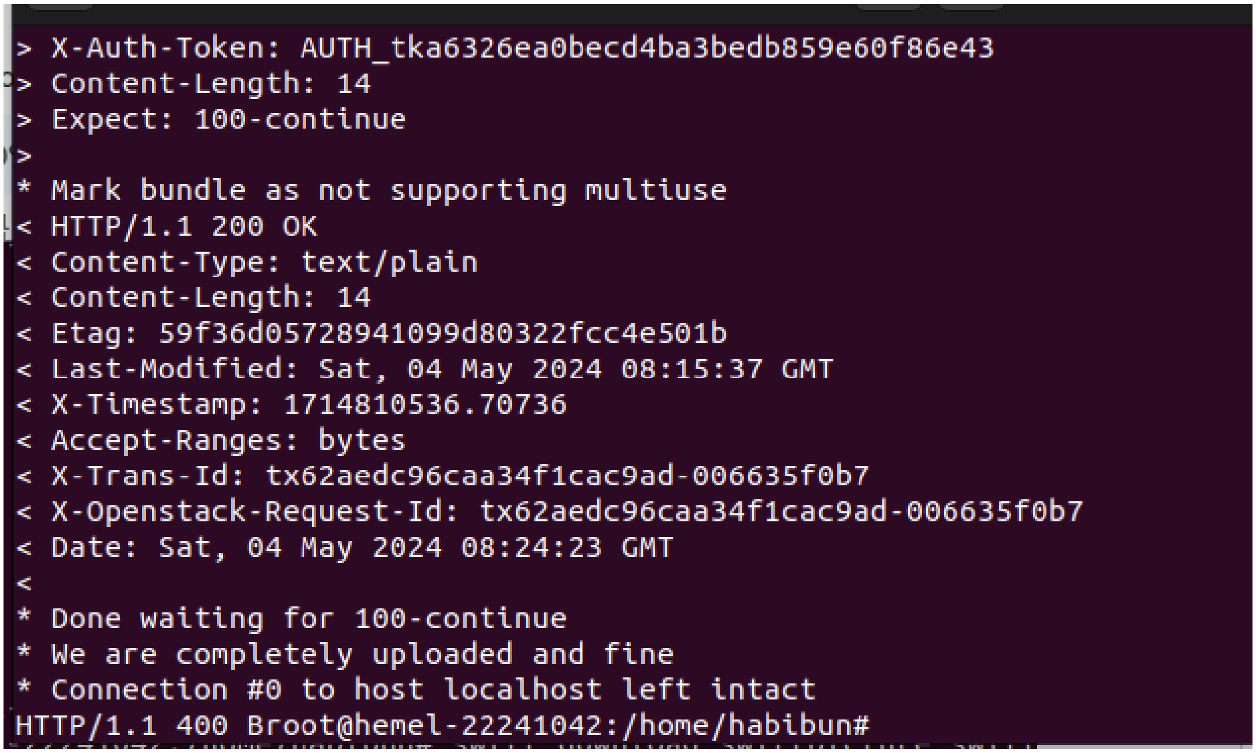
Using curl

1. HEAD request:

This curl command performs a verbose (-v) HTTP HEAD request (-X HEAD) to check the headers of a file named "hemel.txt" located at the specified URL. It includes an authentication token (-H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43') for authorization.

Additionally, it uploads (-T) the local file "hemel.txt" to the specified location. This command is typically used in scenarios involving file management and authentication within a web service or API, allowing users to interact with remote resources using the command line.

```
curl -v -X HEAD -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43'
http://localhost:8080/v1/AUTH_admin/picture/hemel.txt -T hemel.txt
```

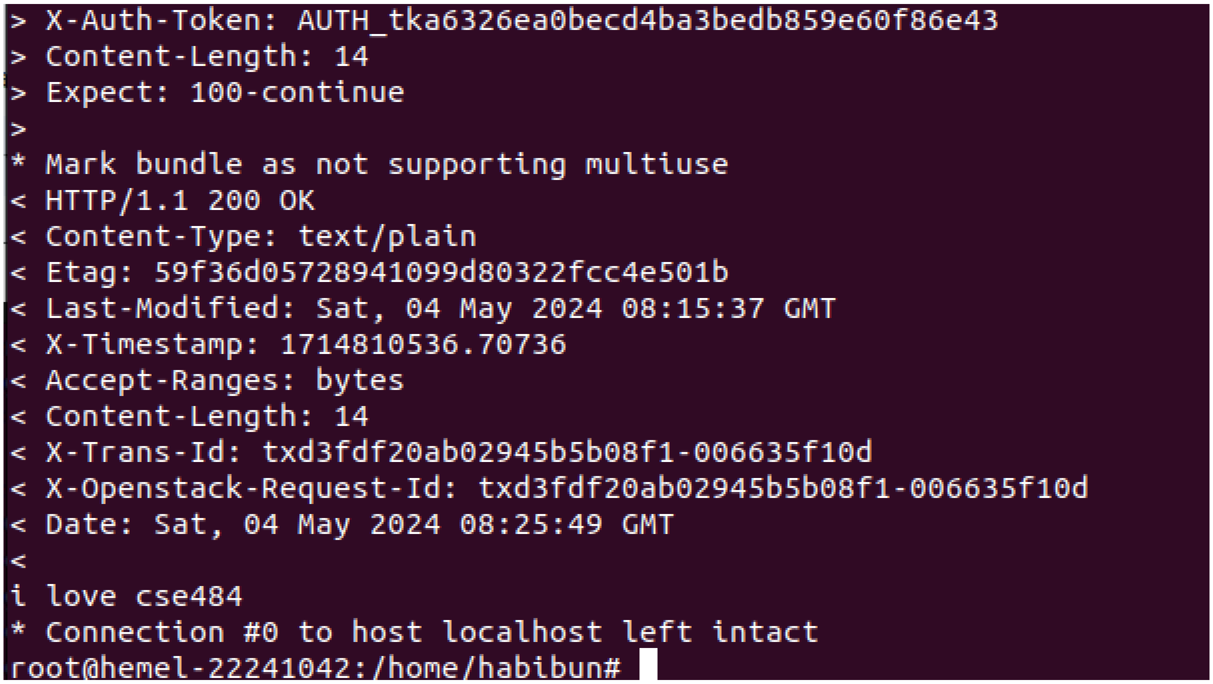


```
> X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
> Content-Length: 14
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: text/plain
< Content-Length: 14
< Etag: 59f36d05728941099d80322fcc4e501b
< Last-Modified: Sat, 04 May 2024 08:15:37 GMT
< X-Timestamp: 1714810536.70736
< Accept-Ranges: bytes
< X-Trans-Id: tx62aedc96caa34f1cac9ad-006635f0b7
< X-Openstack-Request-Id: tx62aedc96caa34f1cac9ad-006635f0b7
< Date: Sat, 04 May 2024 08:24:23 GMT
<
* Done waiting for 100-continue
* We are completely uploaded and fine
* Connection #0 to host localhost left intact
HTTP/1.1 400 Broot@hemel-22241042:/home/habibun#
```

2. GET request:

This verbose curl command (-v) sends an HTTP GET request (-X GET) to retrieve the contents of a file named "hemel.txt" located at the specified URL. It includes an authentication token (-H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43') for authorization. Additionally, it uploads (-T) the local file "hemel.txt" to the same location. This command is commonly used to fetch and transfer files between local and remote systems while ensuring authentication and security measures are in place.

```
curl -v -X GET -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43'
http://localhost:8080/v1/AUTH_admin/picture/hemel.txt -T hemel.txt
```

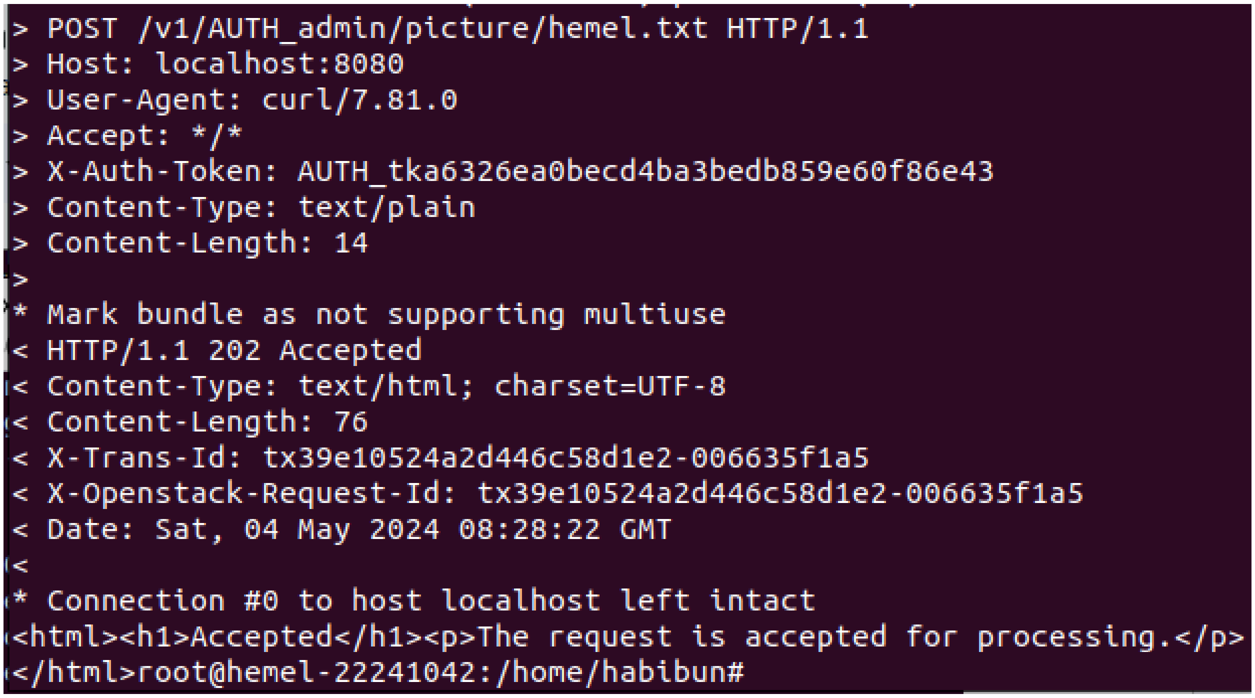


```
> X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
> Content-Length: 14
> Expect: 100-continue
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: text/plain
< Etag: 59f36d05728941099d80322fcc4e501b
< Last-Modified: Sat, 04 May 2024 08:15:37 GMT
< X-Timestamp: 1714810536.70736
< Accept-Ranges: bytes
< Content-Length: 14
< X-Trans-Id: txd3fdf20ab02945b5b08f1-006635f10d
< X-Openstack-Request-Id: txd3fdf20ab02945b5b08f1-006635f10d
< Date: Sat, 04 May 2024 08:25:49 GMT
<
i love cse484
* Connection #0 to host localhost left intact
root@hemel-22241042:/home/habibun#
```

3.POST request:

This verbose curl command (-v) initiates an HTTP POST request (-X POST) to the specified URL, which likely represents a storage service or API endpoint. It includes an authentication token (-H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43') for authorization. Additionally, it specifies the content type of the data being sent as text/plain (-H 'Content-Type: text/plain'). The actual data being sent is the content of the local file "hemel.txt", which is uploaded to the specified URL as binary data (--data-binary @hemel.txt). This command is commonly used to upload files or submit data to a server-side application, often within the context of data storage or manipulation operations.

```
curl -v -X POST -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43' -H
'Content-Type: text/plain' --data-binary @hemel.txt
http://localhost:8080/v1/AUTH_admin/picture/hemel.txt
```

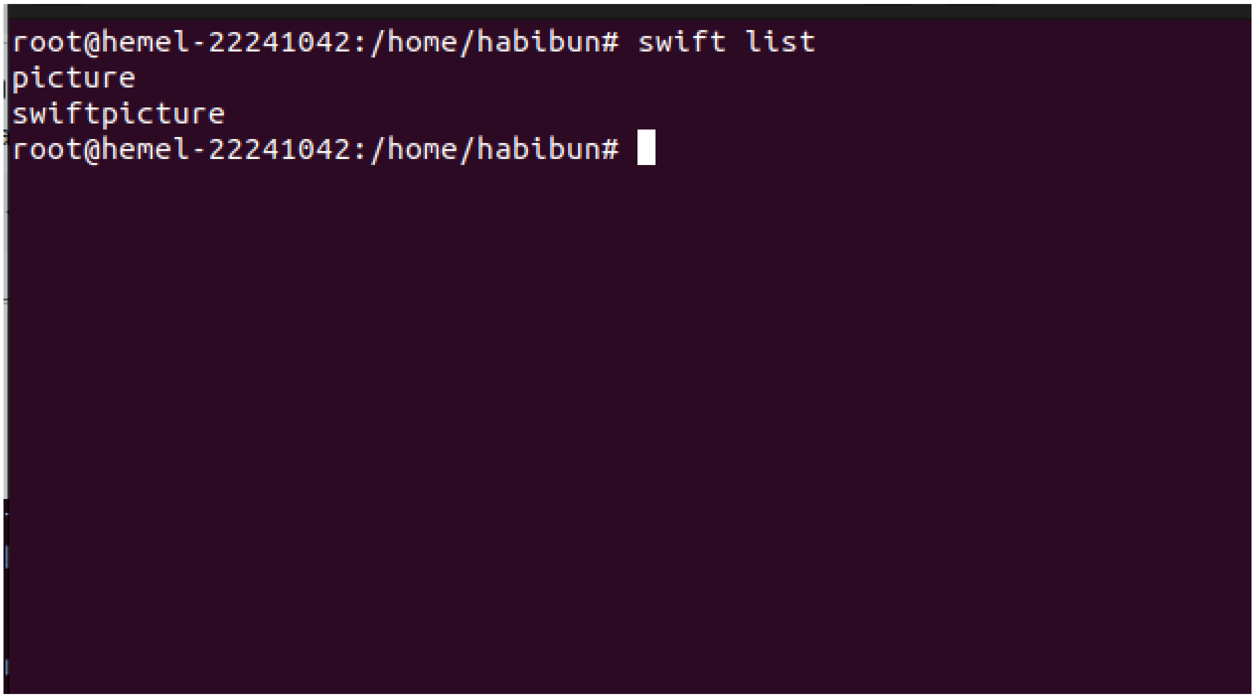


```
> POST /v1/AUTH_admin/picture/hemel.txt HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.81.0
> Accept: */*
> X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43
> Content-Type: text/plain
> Content-Length: 14
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 202 Accepted
< Content-Type: text/html; charset=UTF-8
< Content-Length: 76
< X-Trans-Id: tx39e10524a2d446c58d1e2-006635f1a5
< X-Openstack-Request-Id: tx39e10524a2d446c58d1e2-006635f1a5
< Date: Sat, 04 May 2024 08:28:22 GMT
<
* Connection #0 to host localhost left intact
<html><h1>Accepted</h1><p>The request is accepted for processing.</p>
</html>root@hemel-22241042:/home/habibun#
```

Using Swift CLI

1. Container List:

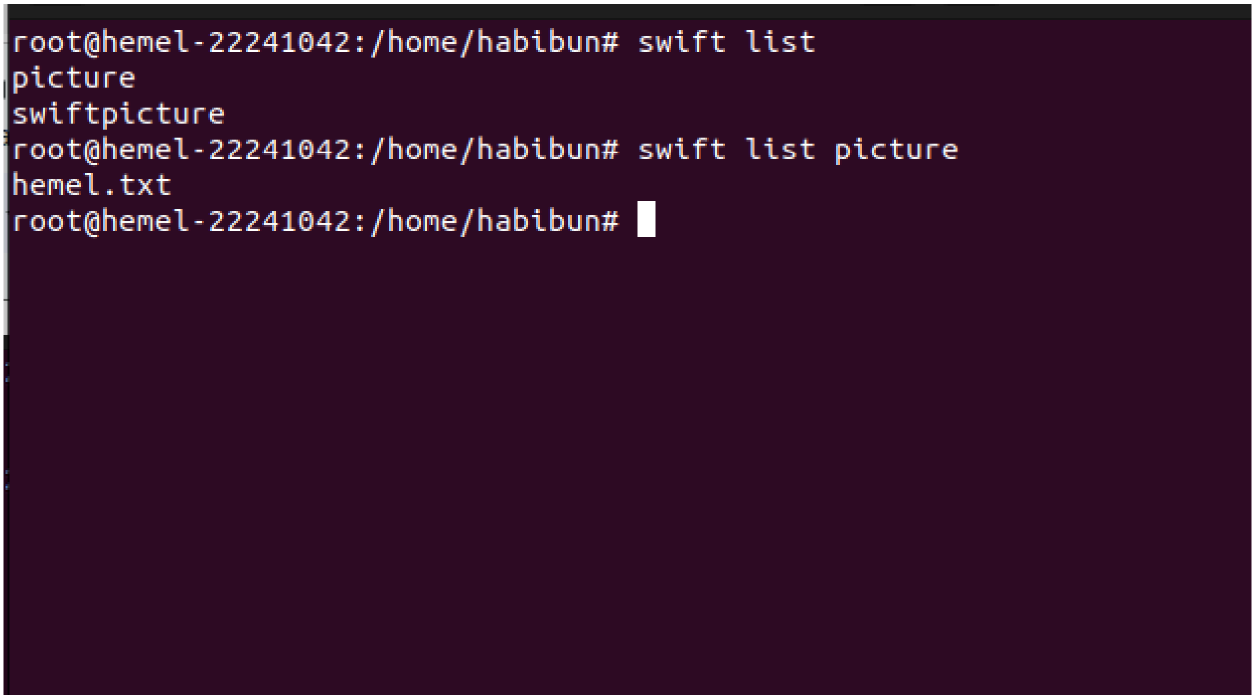
```
swift list
```



```
root@hemel-22241042:/home/habibun# swift list
picture
swiftpicture
root@hemel-22241042:/home/habibun# 
```

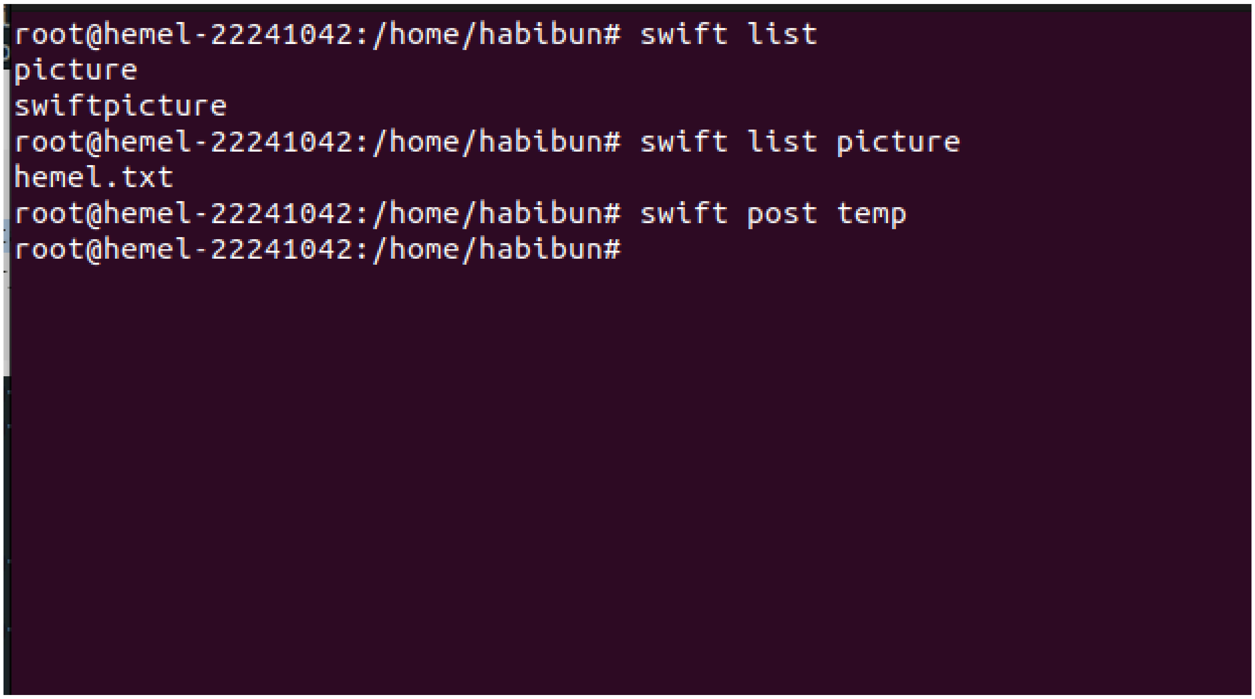
2. list picture

```
swift list picture
```



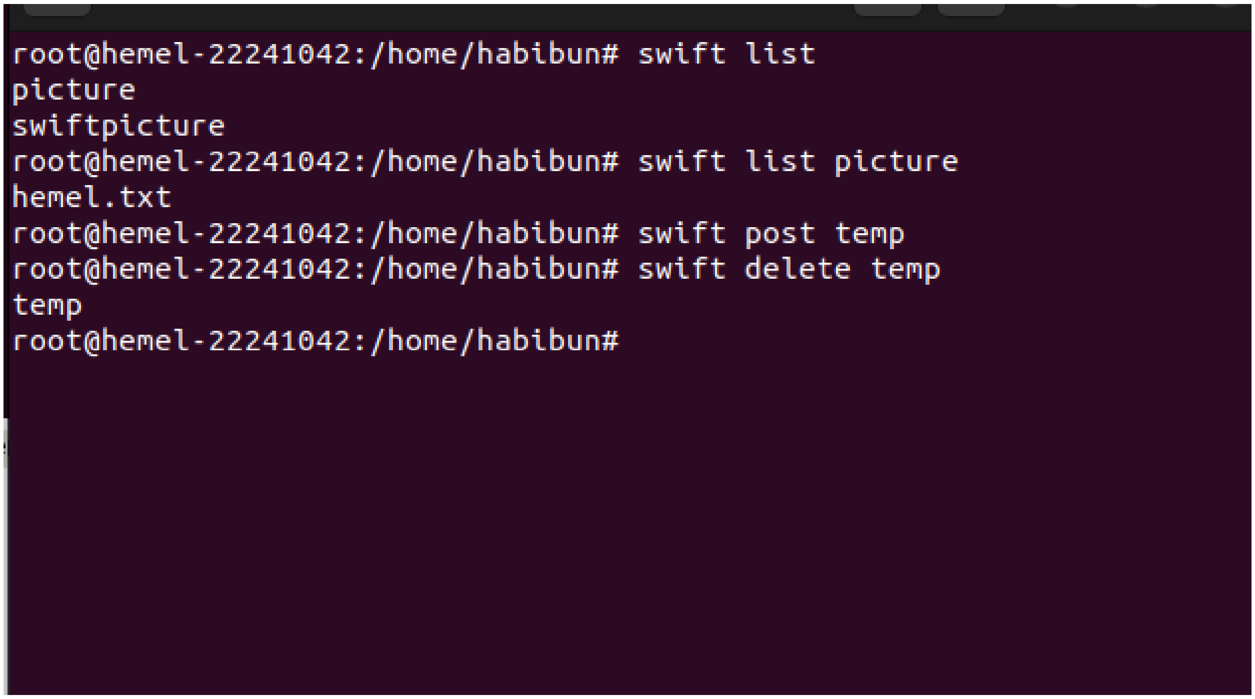
```
root@hemel-22241042:/home/habibun# swift list
picture
swiftpicture
root@hemel-22241042:/home/habibun# swift list picture
hemel.txt
root@hemel-22241042:/home/habibun# 
```

3.Creating unnecessary container to delete



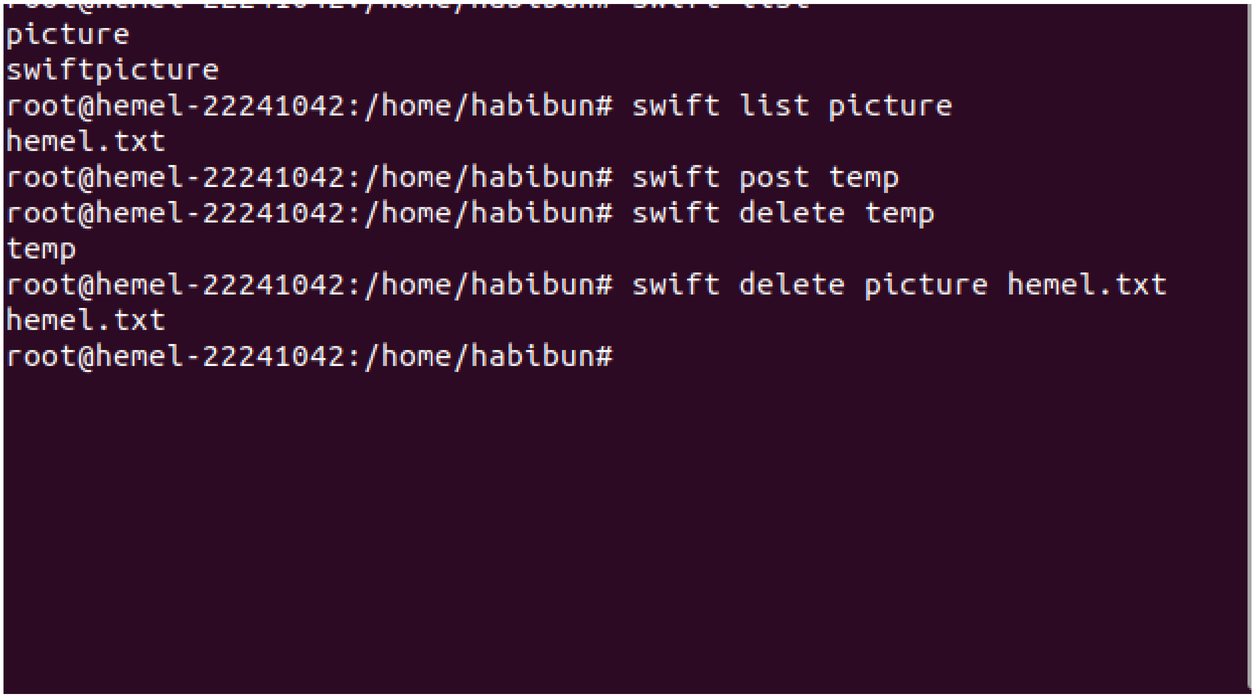
```
root@hemel-22241042:/home/habibun# swift list
picture
swiftpicture
root@hemel-22241042:/home/habibun# swift list picture
hemel.txt
root@hemel-22241042:/home/habibun# swift post temp
root@hemel-22241042:/home/habibun#
```

4. swift delete temp



```
root@hemel-22241042:/home/habibun# swift list
picture
swiftpicture
root@hemel-22241042:/home/habibun# swift list picture
hemel.txt
root@hemel-22241042:/home/habibun# swift post temp
root@hemel-22241042:/home/habibun# swift delete temp
temp
root@hemel-22241042:/home/habibun#
```

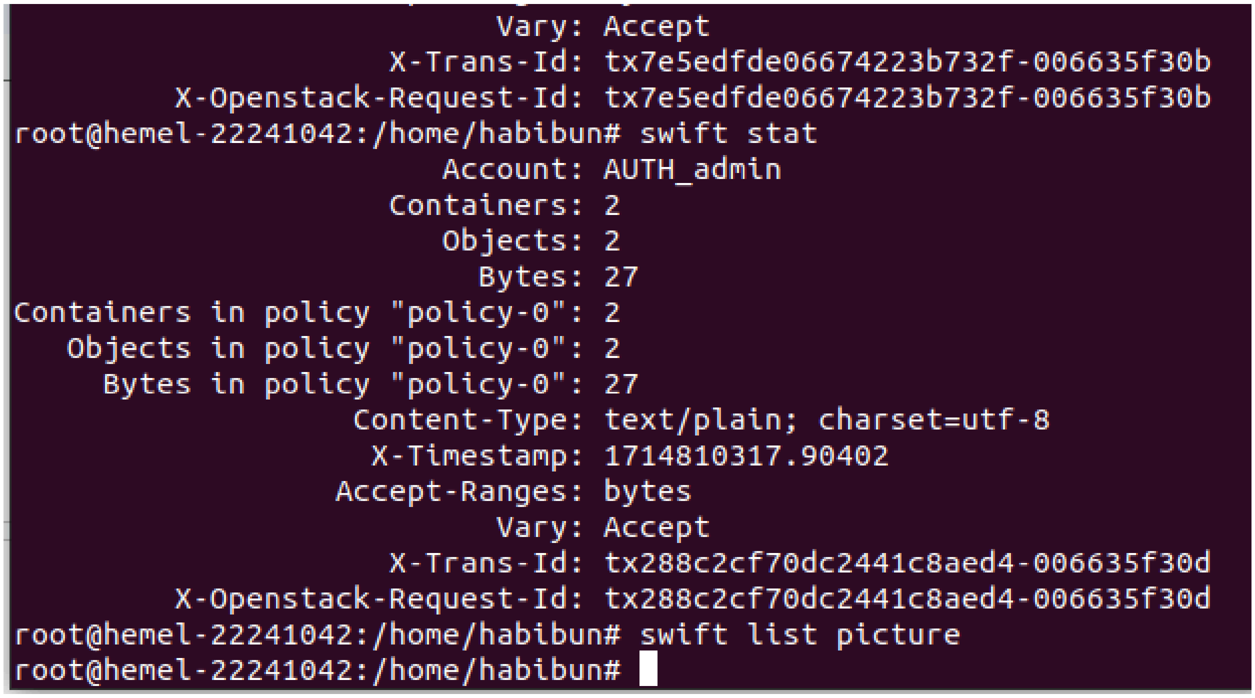
5. swift delete picture hemel.txt



```
root@hemel-22241042:/home/habibun# swift list
picture
swiftpicture
root@hemel-22241042:/home/habibun# swift list picture
hemel.txt
root@hemel-22241042:/home/habibun# swift post temp
root@hemel-22241042:/home/habibun# swift delete temp
temp
root@hemel-22241042:/home/habibun# swift delete picture hemel.txt
hemel.txt
root@hemel-22241042:/home/habibun#
```

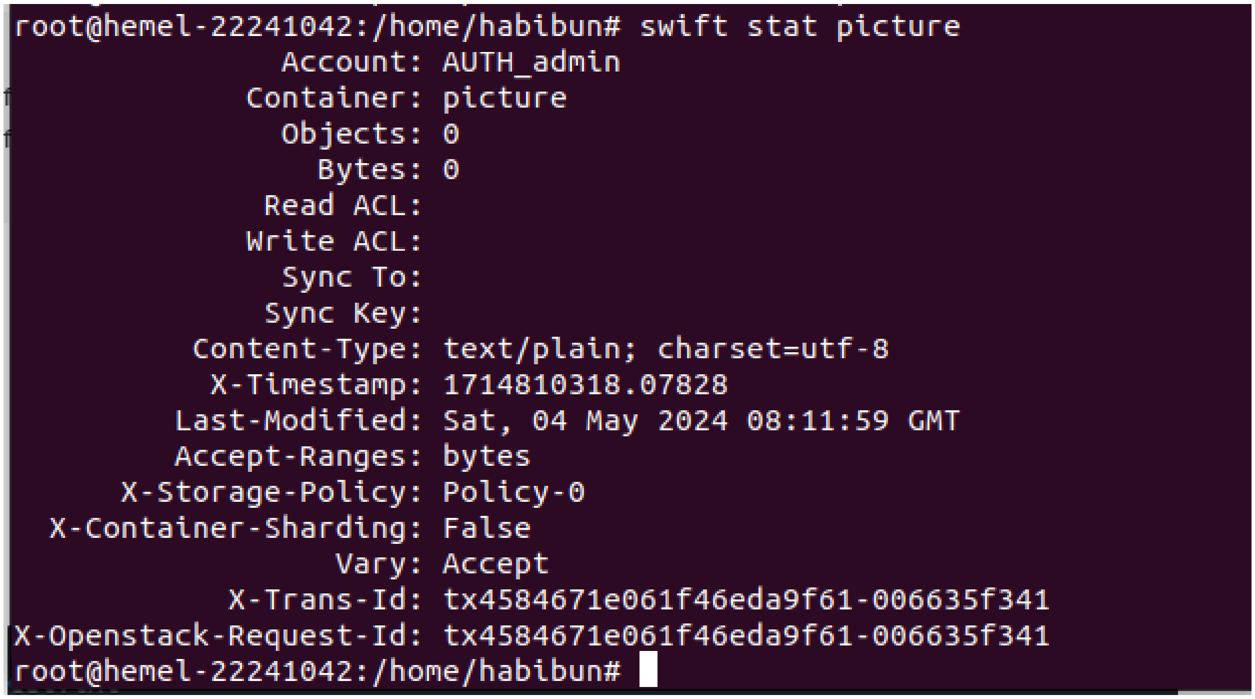
6. Status:

```
swift stat

swift stat your_container_name

swift stat your_container_name your_file_name
```



```
                          Vary: Accept
                    X-Trans-Id: tx7e5edfde06674223b732f-006635f30b
        X-Openstack-Request-Id: tx7e5edfde06674223b732f-006635f30b
root@hemel-22241042:/home/habibun# swift stat
                       Account: AUTH_admin
                    Containers: 2
                       Objects: 2
                         Bytes: 27
Containers in policy "policy-0": 2
   Objects in policy "policy-0": 2
     Bytes in policy "policy-0": 27
                  Content-Type: text/plain; charset=utf-8
                   X-Timestamp: 1714810317.90402
                 Accept-Ranges: bytes
                          Vary: Accept
                    X-Trans-Id: tx288c2cf70dc2441c8aed4-006635f30d
        X-Openstack-Request-Id: tx288c2cf70dc2441c8aed4-006635f30d
root@hemel-22241042:/home/habibun# swift list picture
root@hemel-22241042:/home/habibun# 
```

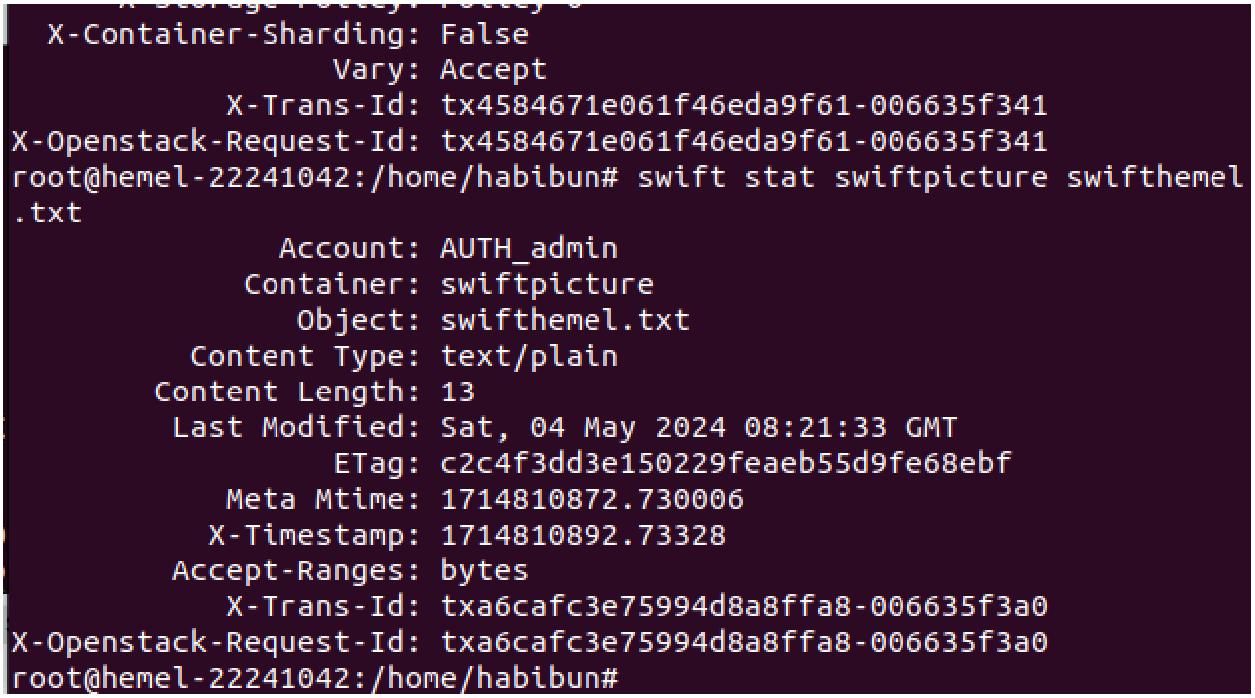
7. swift stat picture



```
root@hemel-22241042:/home/habibun# swift stat picture
               Account: AUTH_admin
             Container: picture
               Objects: 0
                 Bytes: 0
              Read ACL:
             Write ACL:
               Sync To:
              Sync Key:
          Content-Type: text/plain; charset=utf-8
           X-Timestamp: 1714810318.07828
         Last-Modified: Sat, 04 May 2024 08:11:59 GMT
         Accept-Ranges: bytes
      X-Storage-Policy: Policy-0
  X-Container-Sharding: False
                  Vary: Accept
            X-Trans-Id: tx4584671e061f46eda9f61-006635f341
X-Openstack-Request-Id: tx4584671e061f46eda9f61-006635f341
root@hemel-22241042:/home/habibun# 
```

8.

swift stat swiftpicture swifthemel.txt

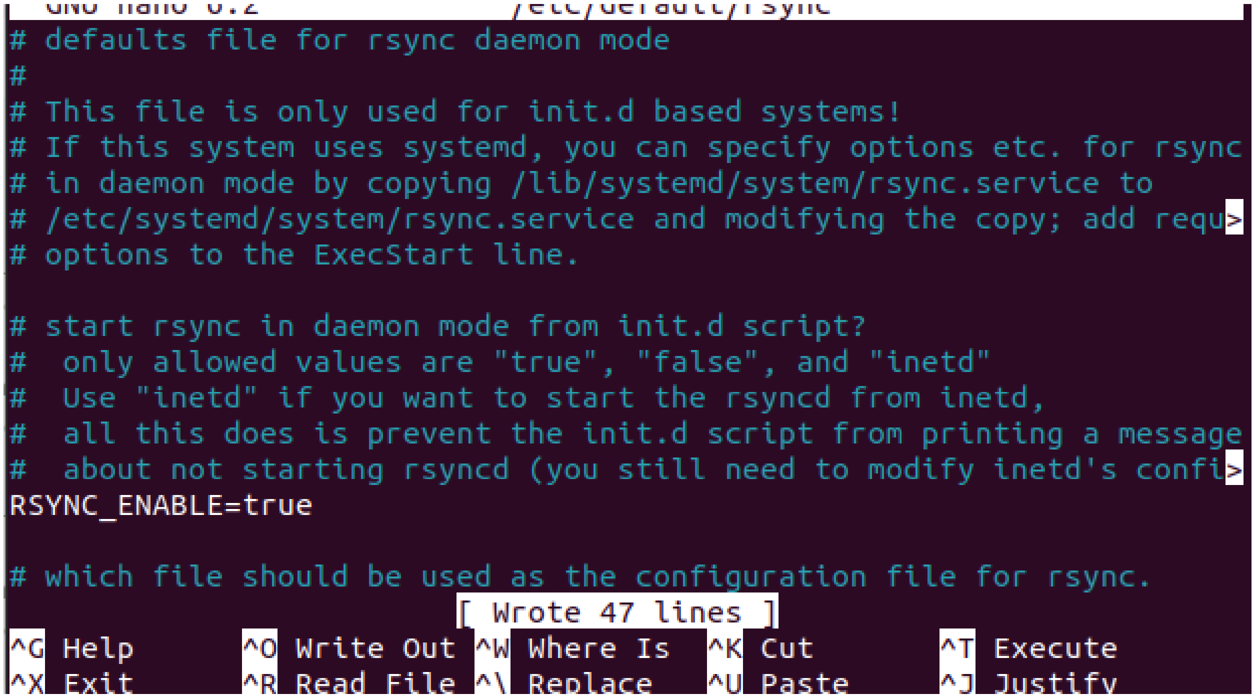


```
  X-Container-Sharding: False
                  Vary: Accept
            X-Trans-Id: tx4584671e061f46eda9f61-006635f341
X-Openstack-Request-Id: tx4584671e061f46eda9f61-006635f341
root@hemel-22241042:/home/habibun# swift stat swiftpicture swifthemel
.txt
               Account: AUTH_admin
             Container: swiftpicture
                Object: swifthemel.txt
          Content Type: text/plain
        Content Length: 13
         Last Modified: Sat, 04 May 2024 08:21:33 GMT
                  ETag: c2c4f3dd3e150229feaeb55d9fe68ebf
            Meta Mtime: 1714810872.730006
           X-Timestamp: 1714810892.73328
         Accept-Ranges: bytes
            X-Trans-Id: txa6cafc3e75994d8a8ffa8-006635f3a0
X-Openstack-Request-Id: txa6cafc3e75994d8a8ffa8-006635f3a0
root@hemel-22241042:/home/habibun#
```

Test replication is working in your system.

The RSYNC_ENABLE variable should first be set to true. This variable is located in the etc/default/rsync file. The command to access this file is as follows:This command will open the rsync configuration file in the Nano text editor with administrative privileges, allowing you to modify the RSYNC_ENABLE variable to true. Once you've made the change, save the file and exit the text editor.

```
nano /etc/default/rsync
```



```
  GNU nano 6.2                /etc/default/rsync
# 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 requ>
# 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 confi>
RSYNC_ENABLE=true

# which file should be used as the configuration file for rsync.
                              [ Wrote 47 lines ]
^G Help        ^O Write Out  ^W Where Is   ^K Cut         ^T Execute
^X Exit        ^R Read File  ^\ Replace    ^U Paste       ^J Justify
```

You must now produce a configuration file. Paste the following command to get started.The command "nano /etc/rsyncd.conf" opens the rsync daemon configuration file (/etc/rsyncd.conf) in the Nano text editor. This file contains settings and options for configuring the behavior of the rsync daemon, allowing you to customize how rsync operates on your system.

```
nano /etc/rsyncd.conf
```

insert the codespnipet that follow into the configuration file.This configuration file for the rsync daemon sets up rules for synchronizing files between computers. It defines the user and group IDs rsync will use, where it will store log and process ID files, and specifies different modules or directories it can synchronize. Each module has settings like maximum connections allowed, directory path, and whether it's read-only or not. Additionally, lock files are specified to prevent conflicts when multiple instances of rsync are accessing the same files simultaneously, ensuring data integrity.

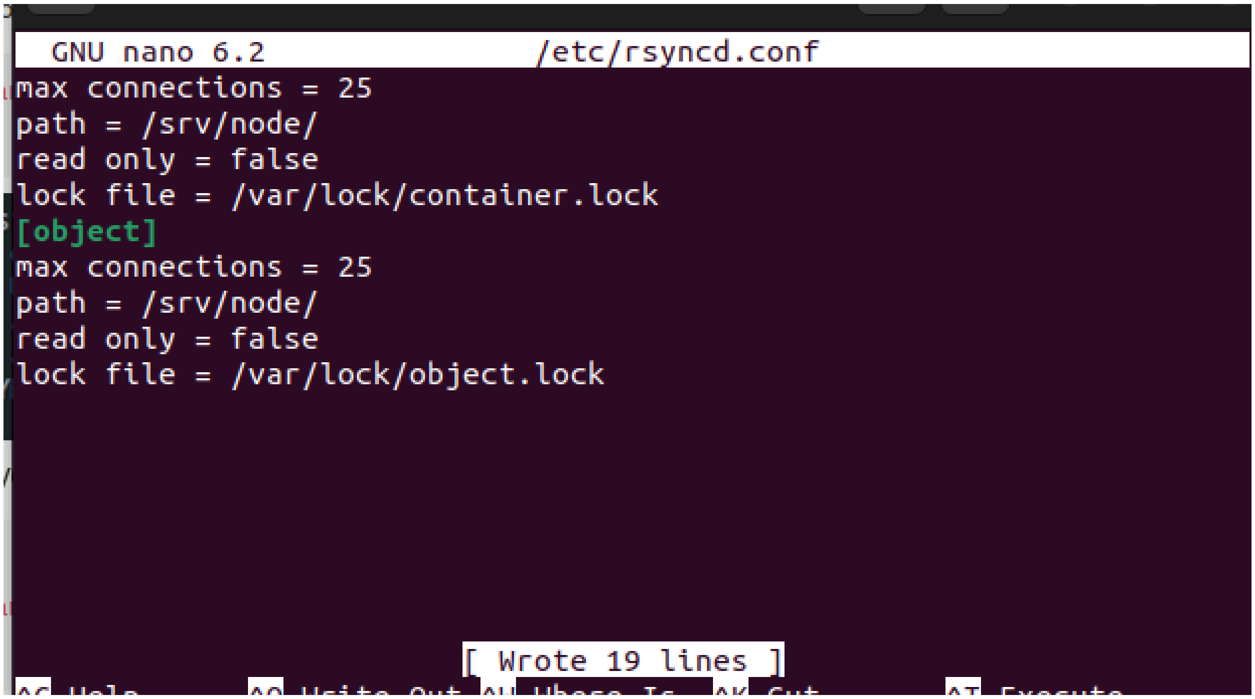
```
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
[account]
max connections = 25
path = /srv/node/
read only = false
lock file = /var/lock/account.lock
[container]
max connections = 25
```

```
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 pasting the code and saving it, restart rsync using the following command.

```
systemctl restart rsync
```

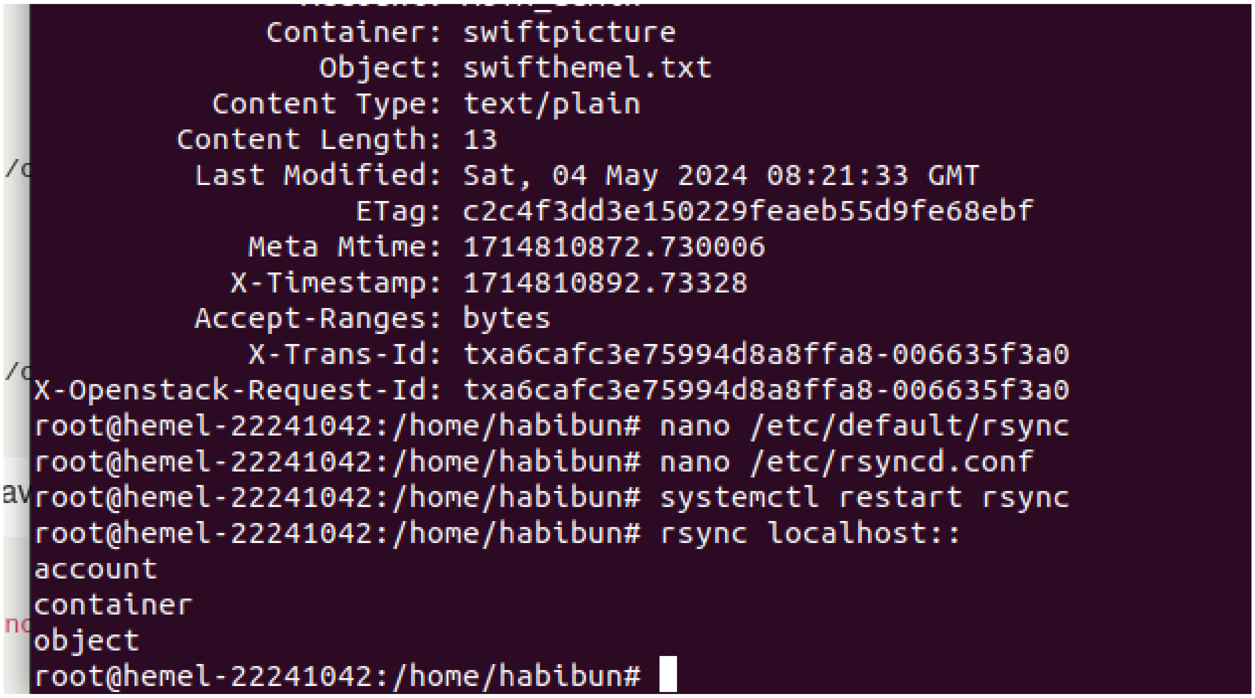
If everything is configured correctly, execute the following code. The output should resemble the subsequent image.



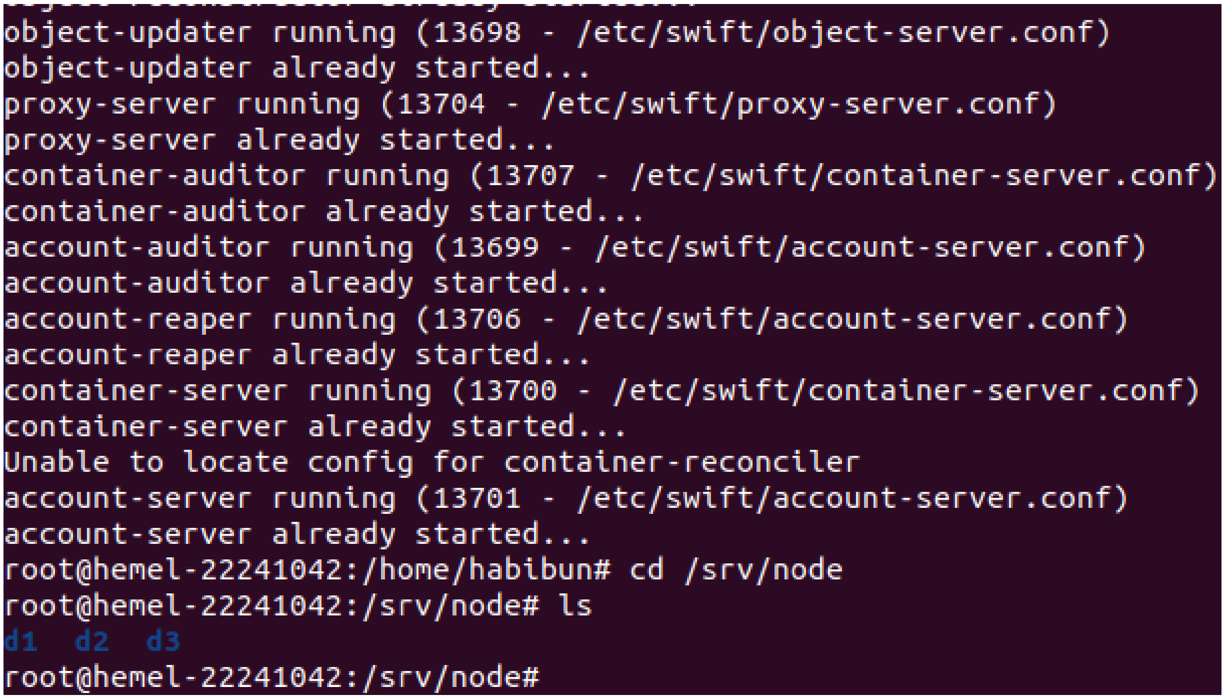
```
  GNU nano 6.2                /etc/rsyncd.conf
max connections = 25
path = /srv/node/
read only = false
lock file = /var/lock/container.lock
[object]
max connections = 25
path = /srv/node/
read only = false
lock file = /var/lock/object.lock




                         [ Wrote 19 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
```

The command "rsync localhost::" lists the available modules that can be synchronized from the localhost (the current machine) using the rsync protocol.

```
rsync localhost::
```



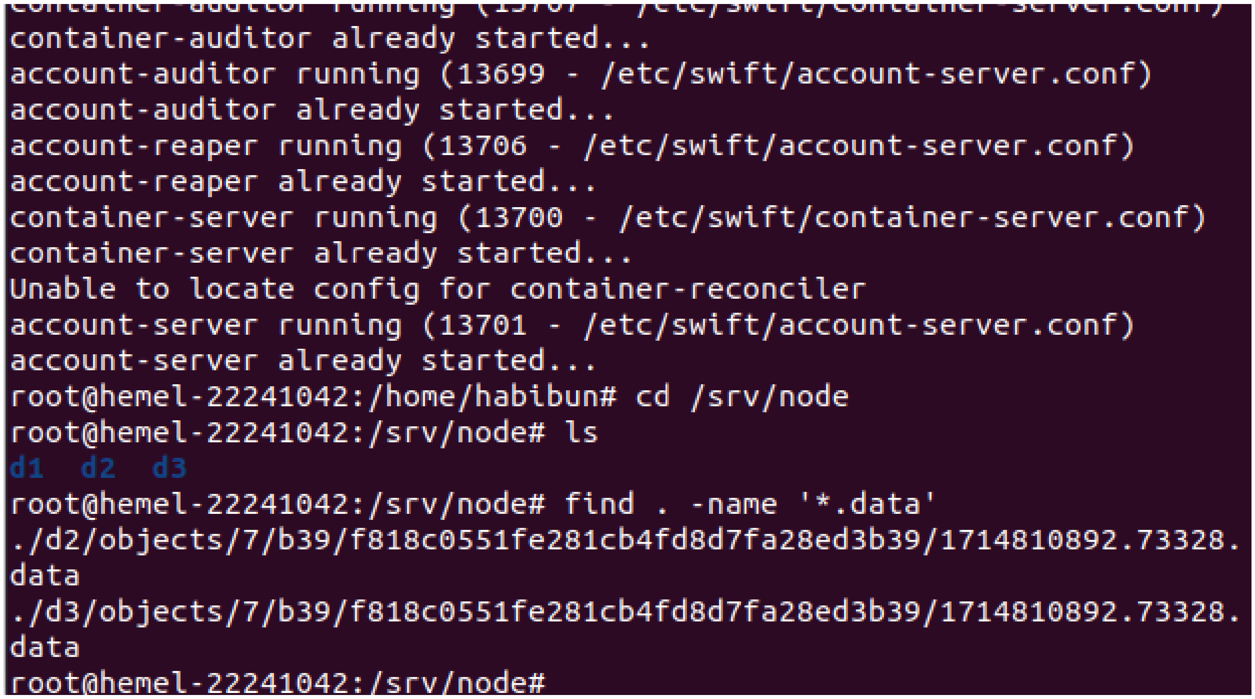
```
               Container: swiftpicture
                  Object: swifthemel.txt
            Content Type: text/plain
          Content Length: 13
           Last Modified: Sat, 04 May 2024 08:21:33 GMT
                    ETag: c2c4f3dd3e150229feaeb55d9fe68ebf
              Meta Mtime: 1714810872.730006
             X-Timestamp: 1714810892.73328
           Accept-Ranges: bytes
              X-Trans-Id: txa6cafc3e75994d8a8ffa8-006635f3a0
X-Openstack-Request-Id: txa6cafc3e75994d8a8ffa8-006635f3a0
root@hemel-22241042:/home/habibun# nano /etc/default/rsync
root@hemel-22241042:/home/habibun# nano /etc/rsyncd.conf
root@hemel-22241042:/home/habibun# systemctl restart rsync
root@hemel-22241042:/home/habibun# rsync localhost::
account
container
object
root@hemel-22241042:/home/habibun# 
```



```
object-updater running (13698 - /etc/swift/object-server.conf)
object-updater already started...
proxy-server running (13704 - /etc/swift/proxy-server.conf)
proxy-server already started...
container-auditor running (13707 - /etc/swift/container-server.conf)
container-auditor already started...
account-auditor running (13699 - /etc/swift/account-server.conf)
account-auditor already started...
account-reaper running (13706 - /etc/swift/account-server.conf)
account-reaper already started...
container-server running (13700 - /etc/swift/container-server.conf)
container-server already started...
Unable to locate config for container-reconciler
account-server running (13701 - /etc/swift/account-server.conf)
account-server already started...
root@hemel-22241042:/home/habibun# cd /srv/node
root@hemel-22241042:/srv/node# ls
d1  d2  d3
root@hemel-22241042:/srv/node#
```

```
swift-init all start
```

```
find . -name '*.data'
```



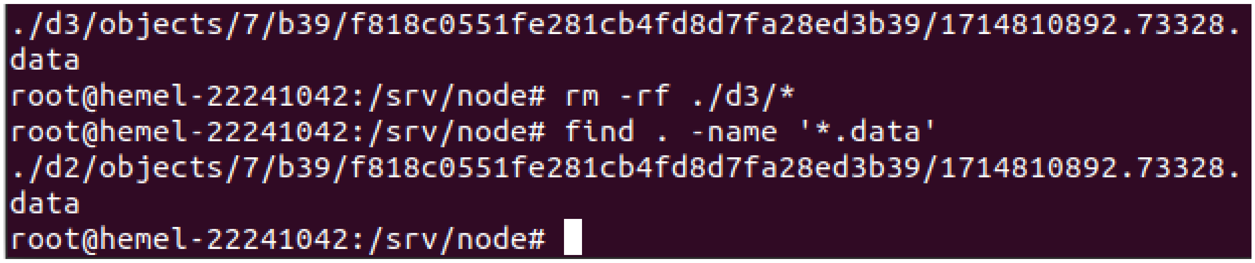
```
container-auditor running (13707 - /etc/swift/container-server.conf)
container-auditor already started...
account-auditor running (13699 - /etc/swift/account-server.conf)
account-auditor already started...
account-reaper running (13706 - /etc/swift/account-server.conf)
account-reaper already started...
container-server running (13700 - /etc/swift/container-server.conf)
container-server already started...
Unable to locate config for container-reconciler
account-server running (13701 - /etc/swift/account-server.conf)
account-server already started...
root@hemel-22241042:/home/habibun# cd /srv/node
root@hemel-22241042:/srv/node# ls
d1  d2  d3
root@hemel-22241042:/srv/node# find . -name '*.data'
./d2/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
./d3/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
root@hemel-22241042:/srv/node#
```

Lets delete all the files of d3 and see what will happen next.

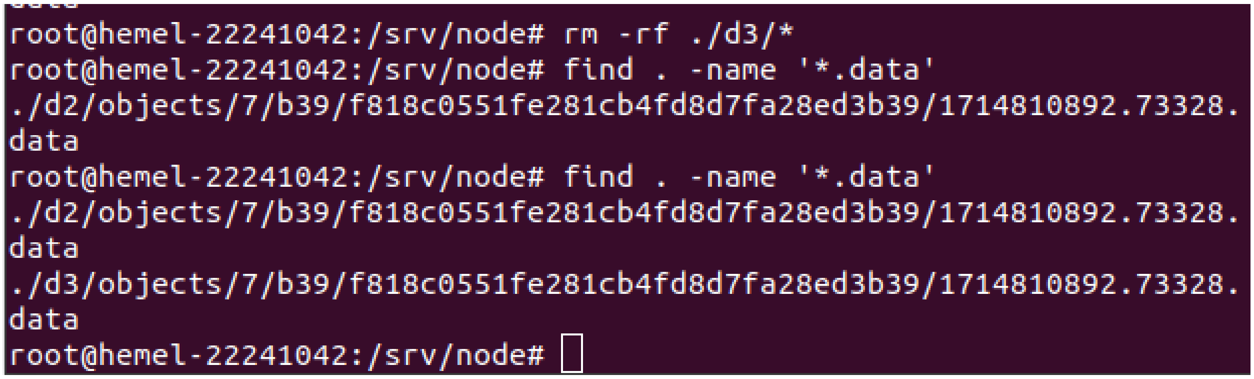
After removing all data from the "d3" drive using the command "rm -rf ./d3/*" and confirming that it's empty, you can witness the magic of rsync by running the command again. After a few seconds, you'll notice that the "d3" drive is magically restored, as if the data never disappeared. This demonstrates the effectiveness of using rsync for data synchronization and backup. With the completion of the OpenStack Swift installation, everything is now set up and functioning smoothly.

```
rm -rf ./d3/*
```

Removed the d3 disk



```
./d3/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
root@hemel-22241042:/srv/node# rm -rf ./d3/*
root@hemel-22241042:/srv/node# find . -name '*.data'
./d2/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
root@hemel-22241042:/srv/node# 
```



```
root@hemel-22241042:/srv/node# rm -rf ./d3/*
root@hemel-22241042:/srv/node# find . -name '*.data'
./d2/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
root@hemel-22241042:/srv/node# find . -name '*.data'
./d2/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
./d3/objects/7/b39/f818c0551fe281cb4fd8d7fa28ed3b39/1714810892.73328.
data
root@hemel-22241042:/srv/node# 
```

after a few time D3 restored.

BONUS:

After installation of the object storage, setup/find out the way to measure different performance metrics. What are the measurement metrics?

To view performance metrics in OpenStack Swift, you can utilize various command-line tools and utilities. Here are some examples

1. Using Swift CLI:
   This command displays information about your Swift account, including the number of containers, objects, and storage used

```
swift stat
```

This command provides more detailed information, including the number of bytes transferred, the number of requests made, and the response status codes.

```
swift stat -v
```

This command lists all objects in a container along with their metadata, which can be useful for performance analysis.

```
swift list -l
```

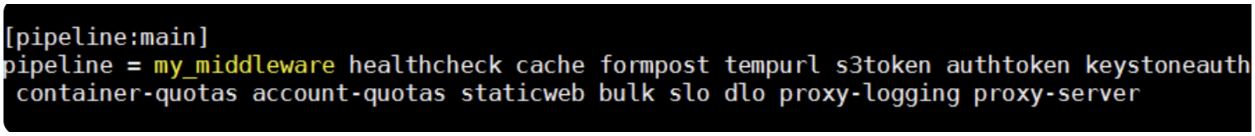
2. Using curl with Swift API:You can use curl commands to directly access Swift's RESTful API endpoints and retrieve performance-related data. For example:This command retrieves information about the Swift cluster, including the number of accounts, containers, and objects.
   curl -v -X GET -H 'X-Auth-Token: <YOUR_AUTH_TOKEN>' http://<SWIFT_ENDPOINT>/driveperf: This command retrieves drive performance metrics, including read and write throughput, latency, and error rates.

```
curl -v -X GET -H 'X-Auth-Token: AUTH_tka6326ea0becd4ba3bedb859e60f86e43’
```

3. To measure performance metrics in Ubuntu after setting up object storage, use commands like top for system monitoring, iostat for disk I/O statistics, and Swift-specific tools like swift stat for monitoring Swift clusters. Additionally, consider logging and monitoring services like Prometheus and Grafana for comprehensive analysis.

A new middleware in OpenStack Swift that can do something new!!!

step 1: go to file /etc/swift/proxy-server.conf
This is is the configuration file which contains details about your middleware, you can see the pipeline with different middleware like healthcheck, cache, formpost, tempurl and so on

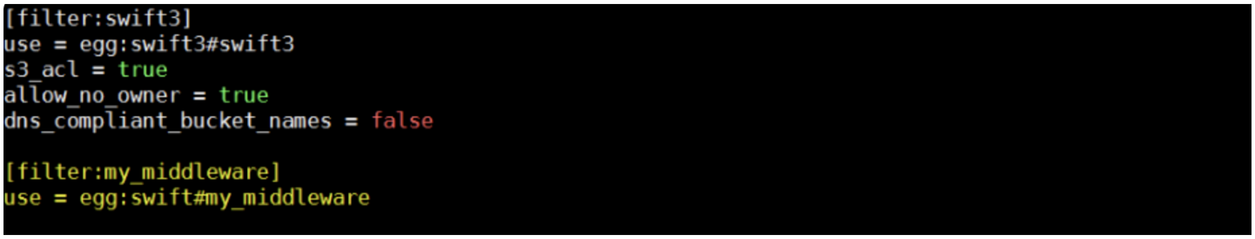


```
[pipeline:main]
pipeline = my_middleware healthcheck cache formpost tempurl s3token authtoken keystoneauth
 container-quotas account-quotas staticweb bulk slo dlo proxy-logging proxy-server
```

Now I added my_middleware here which I am going to define soon

step 2: Configure your middleware
You can do it in two ways

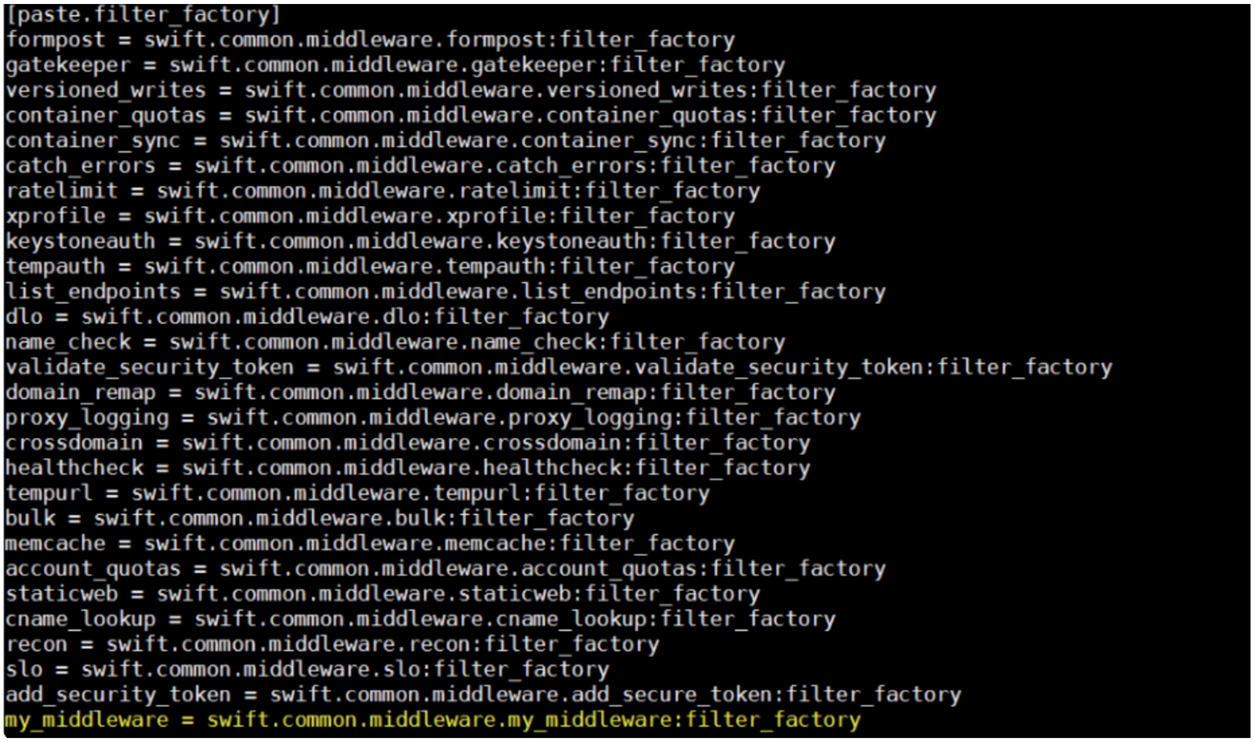
one way
Add the below two lines which are highlighted to /etc/swift/proxy-server.conf file



```
[filter:swift3]
use = egg:swift3#swift3
s3_acl = true
allow_no_owner = true
dns_compliant_bucket_names = false

[filter:my_middleware]
use = egg:swift#my_middleware
```

The use = egg:swift#my_middleware lines here are PasteDeploy entrypoints. Entrypoints are references to Python objects in packages that are named, require certain arguments, and expect a specific return value.

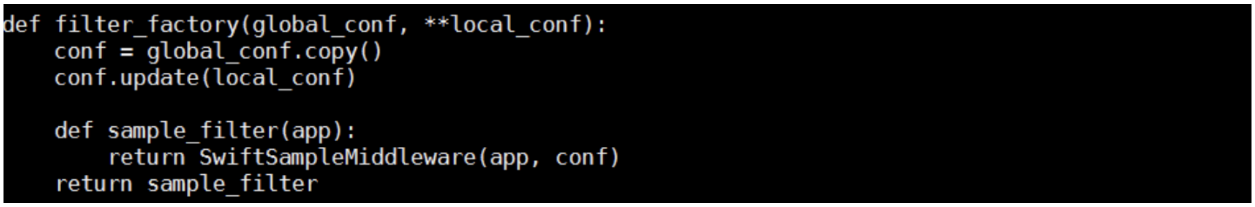
The entrypoint has to be specified in the /usr/lib/python2.7/site-packages/swift-2.7.2.dev33-py2.7.egg-info/entry_points.txt under the [paste.filter_factory] section



```
[paste.filter_factory]
formpost = swift.common.middleware.formpost:filter_factory
gatekeeper = swift.common.middleware.gatekeeper:filter_factory
versioned_writes = swift.common.middleware.versioned_writes:filter_factory
container_quotas = swift.common.middleware.container_quotas:filter_factory
container_sync = swift.common.middleware.container_sync:filter_factory
catch_errors = swift.common.middleware.catch_errors:filter_factory
ratelimit = swift.common.middleware.ratelimit:filter_factory
xprofile = swift.common.middleware.xprofile:filter_factory
keystoneauth = swift.common.middleware.keystoneauth:filter_factory
tempauth = swift.common.middleware.tempauth:filter_factory
list_endpoints = swift.common.middleware.list_endpoints:filter_factory
dlo = swift.common.middleware.dlo:filter_factory
name_check = swift.common.middleware.name_check:filter_factory
validate_security_token = swift.common.middleware.validate_security_token:filter_factory
domain_remap = swift.common.middleware.domain_remap:filter_factory
proxy_logging = swift.common.middleware.proxy_logging:filter_factory
crossdomain = swift.common.middleware.crossdomain:filter_factory
healthcheck = swift.common.middleware.healthcheck:filter_factory
tempurl = swift.common.middleware.tempurl:filter_factory
bulk = swift.common.middleware.bulk:filter_factory
memcache = swift.common.middleware.memcache:filter_factory
account_quotas = swift.common.middleware.account_quotas:filter_factory
staticweb = swift.common.middleware.staticweb:filter_factory
cname_lookup = swift.common.middleware.cname_lookup:filter_factory
recon = swift.common.middleware.recon:filter_factory
slo = swift.common.middleware.slo:filter_factory
add_security_token = swift.common.middleware.add_secure_token:filter_factory
my_middleware = swift.common.middleware.my_middleware:filter_factory
```

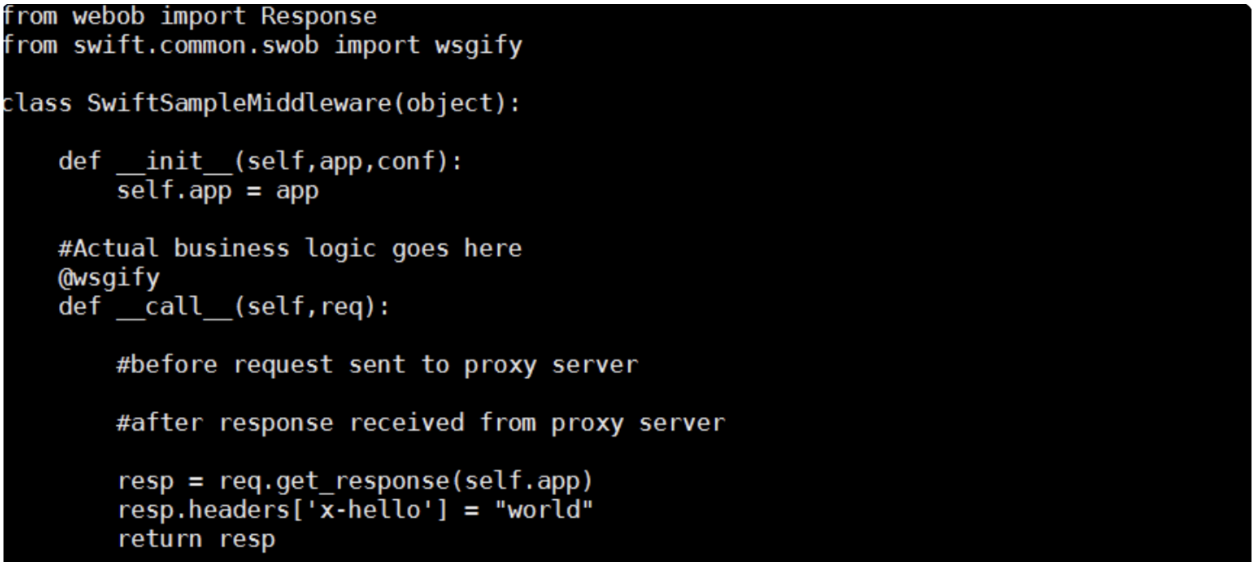
step 3: create your new middleware file
Create a python file in the path we have specified in the configuration file /usr/lib/python2.7/site-packages/swift/commomn/middleware where all the midleware are defined.

I created my_middleware.py file in the specified path. Add the filter_factory function to your program



```
def filter_factory(global_conf, **local_conf):
    conf = global_conf.copy()
    conf.update(local_conf)

    def sample_filter(app):
        return SwiftSampleMiddleware(app, conf)
    return sample_filter
```

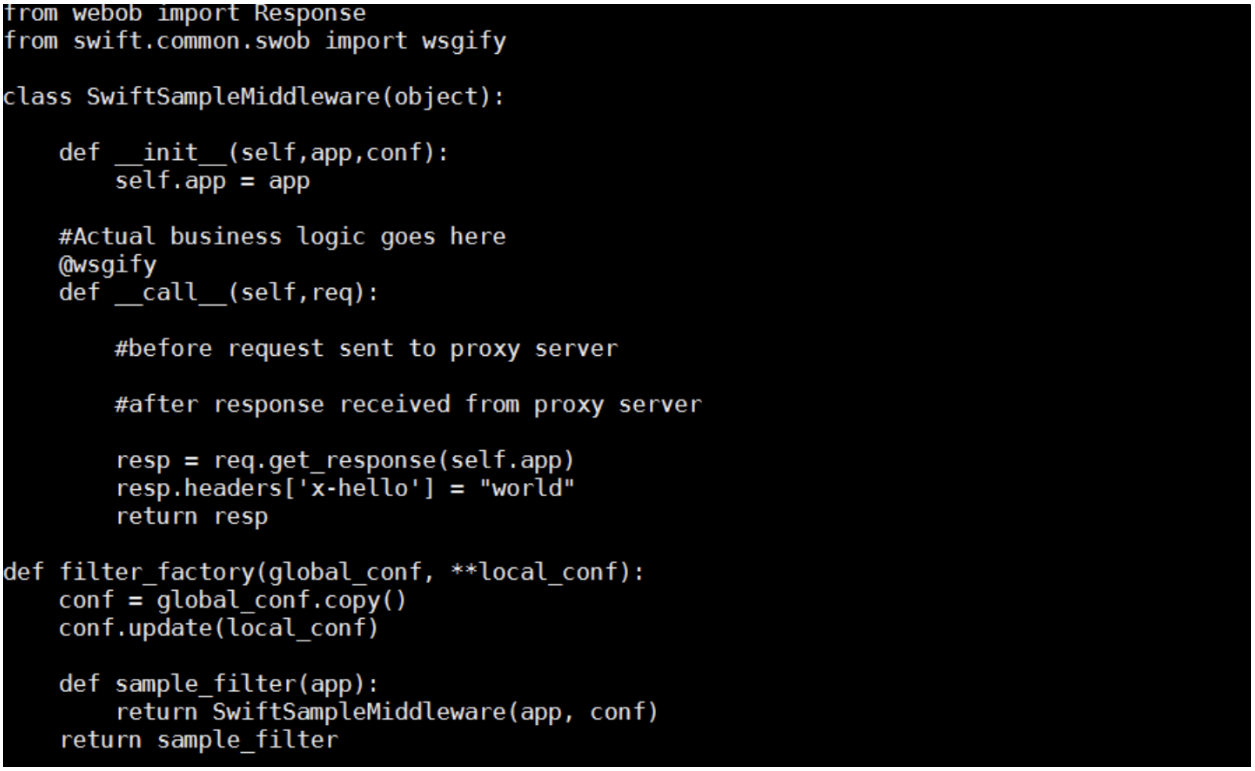
Now define the class SwiftSampleMiddleWare where actual changes reside



```
from webob import Response
from swift.common.swob import wsgify

class SwiftSampleMiddleware(object):

    def __init__(self,app,conf):
        self.app = app

    #Actual business logic goes here
    @wsgify
    def __call__(self,req):

        #before request sent to proxy server

        #after response received from proxy server

        resp = req.get_response(self.app)
        resp.headers['x-hello'] = "world"
        return resp
```

In this class your actual business logic reside. The import statements one is for accessing response and the other is WSGI(Web Server Gateway Interface). I am altering the response headers by adding an additional attribute to my actual response headers 'x-hello' which has value 'world'. similarly you can alter request/response according to your needs.

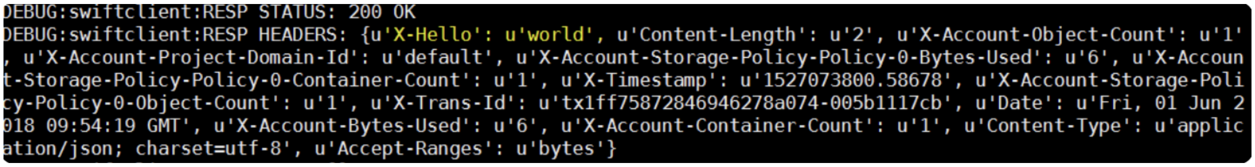
so finally your python file will be something like this



```
from webob import Response
from swift.common.swob import wsgify

class SwiftSampleMiddleware(object):

    def __init__(self,app,conf):
        self.app = app

    #Actual business logic goes here
    @wsgify
    def __call__(self,req):

        #before request sent to proxy server

        #after response received from proxy server

        resp = req.get_response(self.app)
        resp.headers['x-hello'] = "world"
        return resp

def filter_factory(global_conf, **local_conf):
    conf = global_conf.copy()
    conf.update(local_conf)

    def sample_filter(app):
        return SwiftSampleMiddleware(app, conf)
    return sample_filter
```

step 4: check the output:

Now you have created your middleware file and configured in the swift pipeline so lets try to check whether our variable is getting added to the response headers
Restart the swift proxy server using command service openstack-swift-proxy restart
Next source /root/openrc then run swift list --debug



```
DEBUG:swiftclient:RESP STATUS: 200 OK
DEBUG:swiftclient:RESP HEADERS: {u'X-Hello': u'world', u'Content-Length': u'2', u'X-Account-Object-Count': u'1'
, u'X-Account-Project-Domain-Id': u'default', u'X-Account-Storage-Policy-Policy-0-Bytes-Used': u'6', u'X-Accoun
t-Storage-Policy-Policy-0-Container-Count': u'1', u'X-Timestamp': u'1527073800.58678', u'X-Account-Storage-Poli
cy-Policy-0-Object-Count': u'1', u'X-Trans-Id': u'tx1ff75872846946278a074-005b1117cb', u'Date': u'Fri, 01 Jun 2
018 09:54:19 GMT', u'X-Account-Bytes-Used': u'6', u'X-Account-Container-Count': u'1', u'Content-Type': u'applic
ation/json; charset=utf-8', u'Accept-Ranges': u'bytes'}
```

We can see that our attribute is added to the response headers. If you have any errors in between you can check in the /var/log/swift/proxy-server.log file for details. If you want more about middle implementation you can visit the official site. Thank You

Reference for doing the bonus part: click here

THE END