## Environment

On CloudLab, Ubuntu 18.0.4 LTS .

Python 2.7 ,OpenStack Pike, Python-swiftclient 3.6.0, TempAuth v1.0

## Node:

**Client Node(cp-3): 192.168.0.9**

Run as a client. Use python-swiftclient as CLI.

**Database server Node(cp-2): 192.168.0.7**

It’s the database which stores key used for encryption. It can only be accessed by backend node(Storage server).

**Storage Server Node(ctl): 192.168.0.1**

Take charge of storing objects(files).Process account and container method.

**Proxy Server(cp-4): 192.168.0.5**

Proxy server can access to the backend node, bridging communication between client and storage server.

③Query the key

② ①

**192.168.0.7**

**192.168.0.1**

④Return the key

①Object Initialize:

Choose whether to encrypt and which level it uses

②Handle object:

Proxy process the encryption info and put it to the backend

Fig 1.

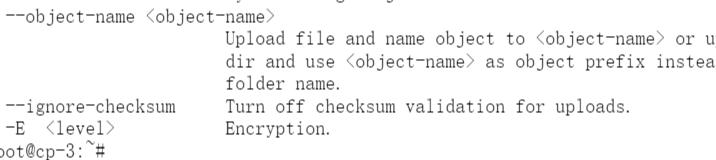
**192.168.0.9**

**192.168.0.5**

Backend Node

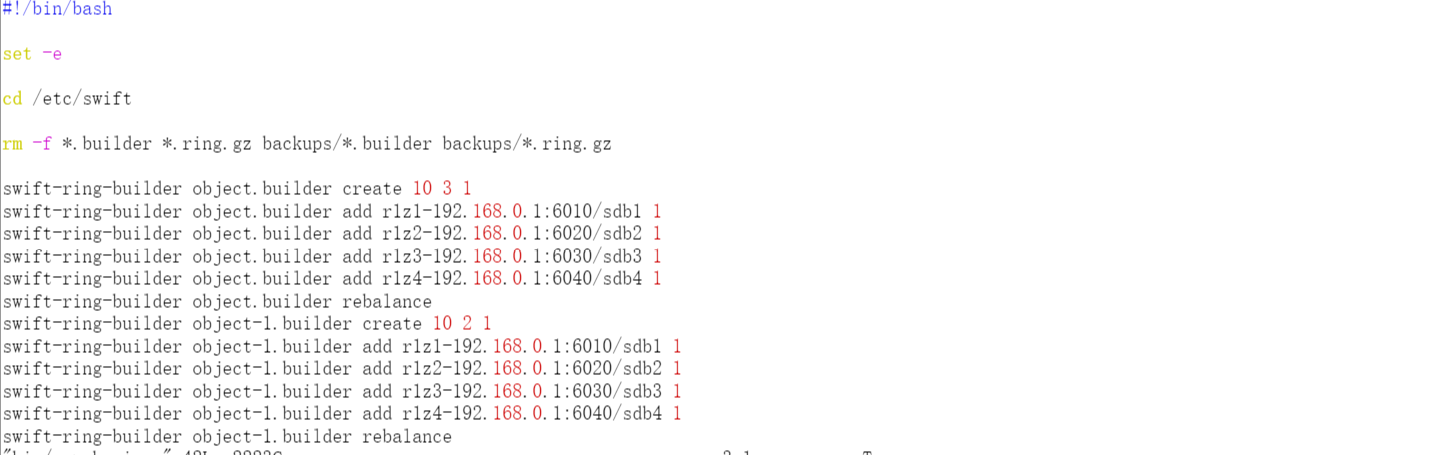
## Configuration

1.On client node(192.168.0.9),we install python-swiftclient v3.6.0. The CLI is modified so it can pass the encryption needs to proxy. We use command “-E” when uploading a file.



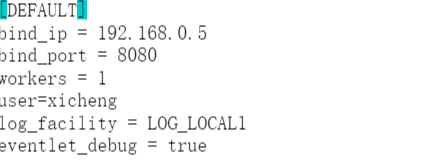
2.On proxy node, we set the information of ring like the following.

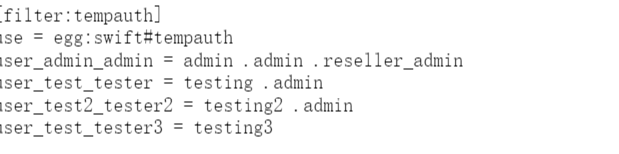
Ringmaker scripts:



Storage servers(including account, container and object-server) are all bind to ip 192.168.0.1 with different ports.

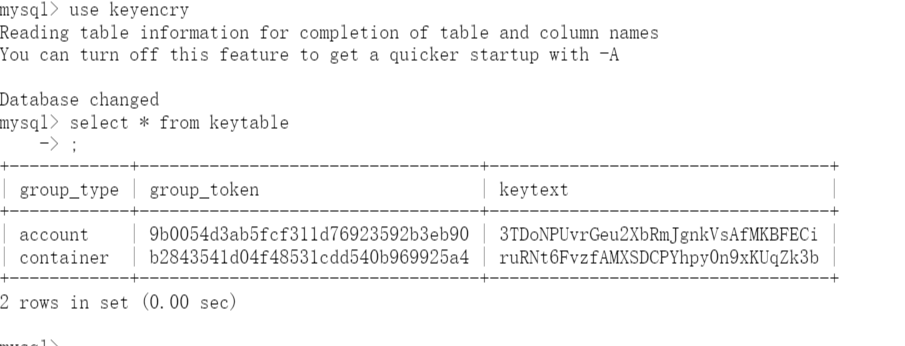
In /etc/swift/proxy-server.conf:



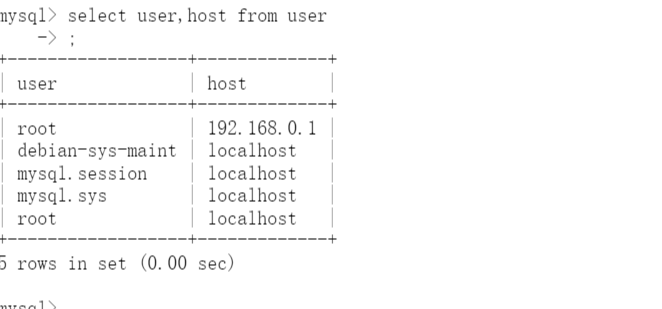


Using tempauth and ip is 192.168.0.5

3.On Database server node, we deploy MYSQL as our database server. The key is stored in table “keytable” of database “keyencry”.



The primary key is the combination of group\_type and group\_token.

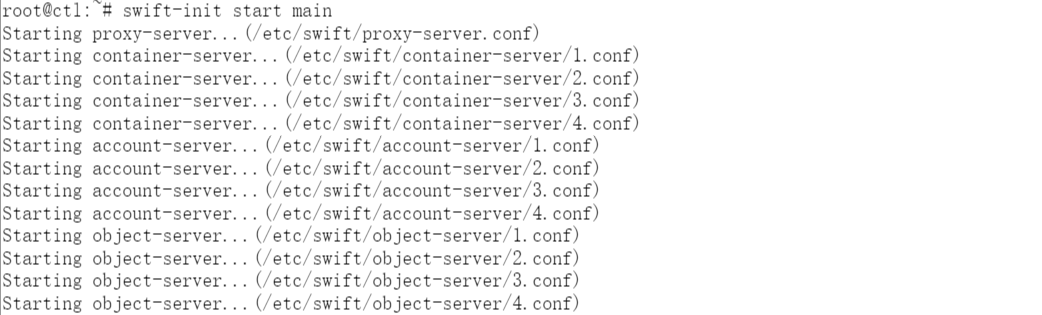


Database can only be accessed by localhost(192.168.0.7) and storage server (192.168.0.1) with grant on SELECT,INSERT. If further security is needed, we can deploy more regulations on firewall of database server.

4. On Storage Server node, we run object, account and container server. Also, we run proxy-server on this node, so a client can either contact to proxy server or to storage server directly to put or get an object.

## Run

1.Start main services on 192.168.0.1:



Note that proxy-server is optional.

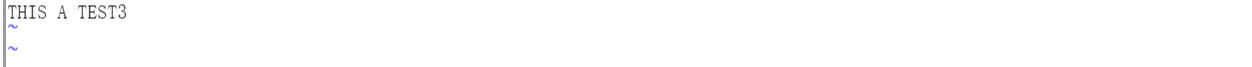
2.Start proxy-server on 192.168.0.5



3.Start MYSQL service on 192.168.0.7

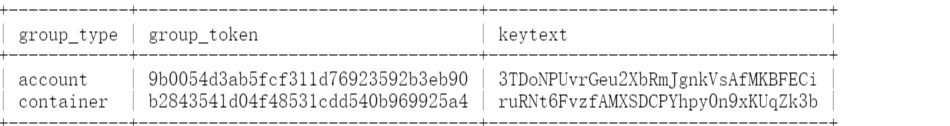
4.Upload and download test on client(192.168.0.5)

First, by accessing with proxy server(192.168.0.5) upload a file with encryption-level of account.



This file is a text file called test3.

In Database,the key of this account is already exist, so server will use this key to encrypt it and decrypt it.

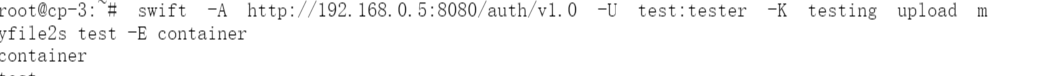


But what if we delete or alter this key?

And then we try to download this file. And now it turns like this, because it didn’t get the key using while encrypted ,so no one could know what the original file is.



Now,we re-upload this file but using encryption level container.



And a new key will generate for encrypting this level’s containers.

All token is the hash of the name of account (or container etc.). We use AES\_CTR to encrypt and md5 to do the hash. In the future, it may change to more strong encryption algorithm like SHA-1.