



# **OPENSTACK STORAGE TUTORIAL: INSTALLING SWIFT-ALL-IN-ONE**

Me: Joshua McKenty

Twitter: @jmckenty

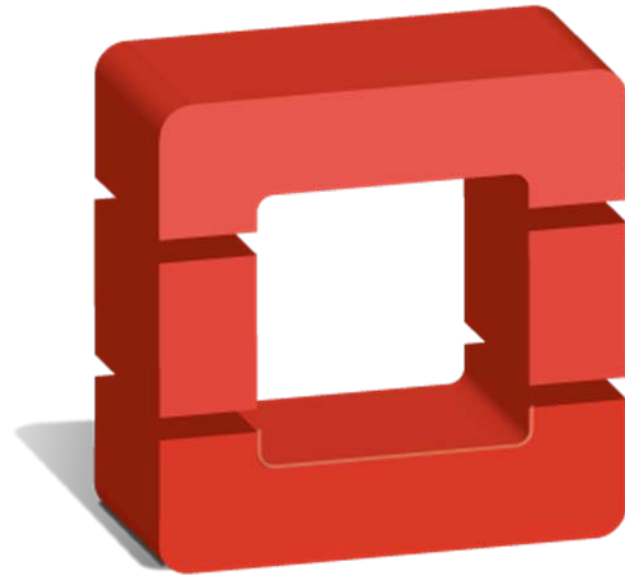
Email: joshua@piston.cc

Former Chief Architect, NASA Nebula

Founding Member, OpenStack

OpenStack Project Policy Board



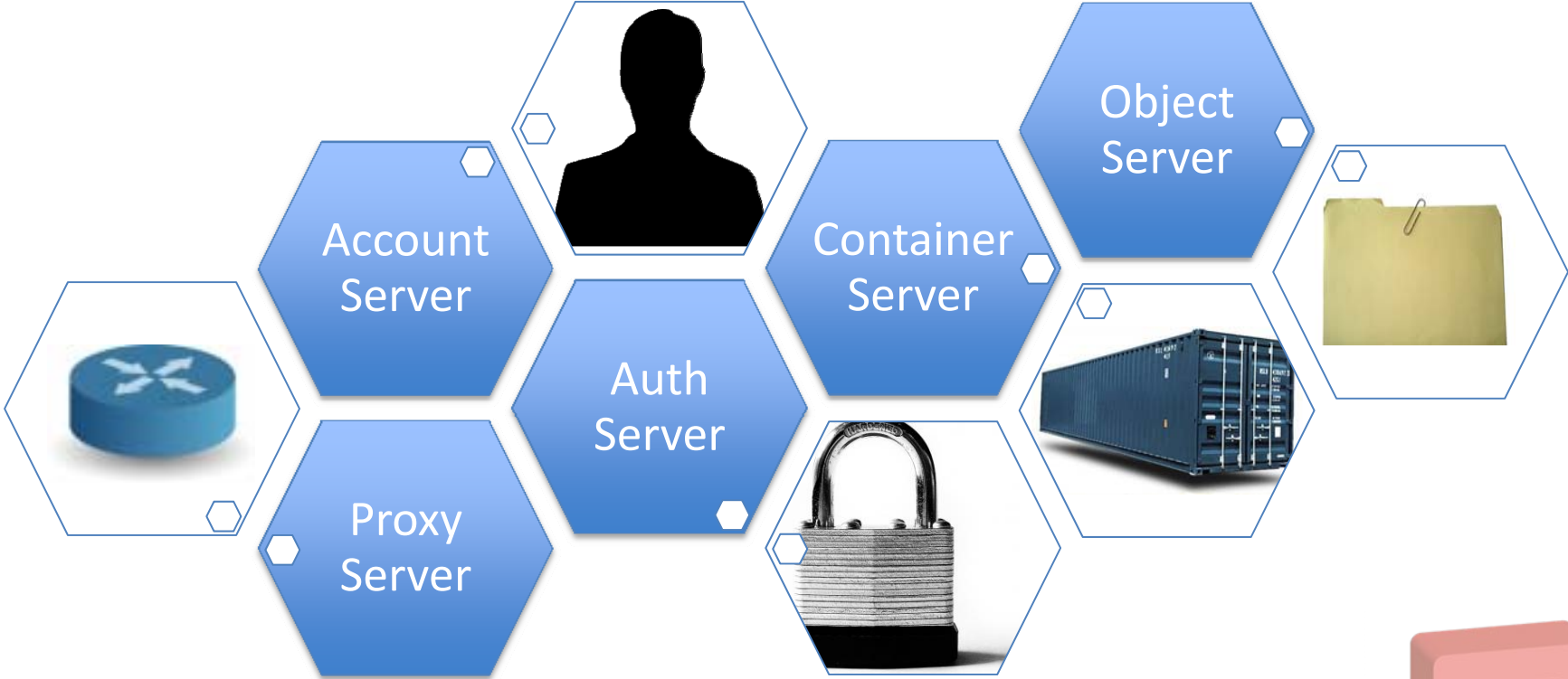


openstack™

```
> curl http://www.openstack.org
```



# SWIFT – What it is, in one slide



# Today's Menu

- One Virtual Machine
- Four Swift Nodes
- Loopback device for storage
- Self-signed SSL Certificate

# Before we begin...

- VMWare Fusion
  - <https://www.vmware.com/tryvmware/?p=vmware-fusion31&lp=1>
- Ubuntu Disk Image
  - <http://releases.ubuntu.com/lucid/ubuntu-10.04.2-server-amd64.iso>
- Swift Administrator's Guide
  - <http://docs.openstack.org/cactus/openstack-compute/admin/os-compute-adminguide-cactus.pdf>
- Cyberduck (for testing)
  - <http://cyberduck.ch/Cyberduck-4.0.2.zip>

Step One:



# Step Two: Get Remote

```
<log in>
```

```
sudo su -
```

```
apt-get install -y openssh-  
server
```

```
ifconfig #get your ip address
```

```
<ssh to this ip>
```



# Step Three: Apt Packages

```
sudo su -  
apt-get -y install python-software-properties  
add-apt-repository ppa:swift-core/ppa  
apt-get update  
apt-get -y install curl gcc bzip2 memcached python-  
configobj python-coverage python-dev python-nose  
python-setuptools python-simplejson python-xattr  
sqlite3 xfsprogs python-webob python-eventlet  
python-greenlet python-pastedeploy python-netifaces  
screen vim
```

# Step Four: Storage Environment

```
adduser swiftdemo
adduser swiftdemo admin
mkdir /srv
dd if=/dev/zero of=/srv/swift-disk bs=1024 count=0 seek=1000000
mkfs.xfs -i size=1024 /srv/swift-disk
echo "/srv/swift-disk /mnt/sdb1 xfs loop,noatime,nodiratime,nobarrier,logbufs=8 0" >> /etc/fstab

mkdir /mnt/sdb1
mount /mnt/sdb1
mkdir /mnt/sdb1/1 /mnt/sdb1/2 /mnt/sdb1/3 /mnt/sdb1/4
chown -R swiftdemo:swiftdemo /mnt/sdb1
for x in {1..4}; do ln -s /mnt/sdb1/$x /srv/$x; done

mkdir -p /etc/swift/object-server /etc/swift/container-server /etc/swift/account-server
mkdir -p /srv/1/node/sdb1 /srv/2/node/sdb2 /srv/3/node/sdb3 /srv/4/node/sdb4
mkdir -p /var/run/swift

chown -R swiftdemo:swiftdemo /etc/swift /mnt/sdb1 /srv/[1-4]/ /var/run/swift
sed -e "s/exit 0/mkdir \\/var\\/run\\/swift\\nchown swiftdemo:swiftdemo \\/var\\/run\\/swift\\nexit 0/g" /etc/rc.local -i
```

# Step Five: RSYNC

```
cat << EOF > /etc/rsyncd.conf
uid = swiftdemo
gid = swiftdemo
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
address = 127.0.0.1
```

EOF

# Step Five: RSYNC

```
cat << EOF >> /etc/rsyncd.conf
[account6012]
max connections = 25
path = /srv/1/node/
read only = false
lock file = /var/lock/account6012.lock

[account6022]
max connections = 25
path = /srv/2/node/
read only = false
lock file = /var/lock/account6022.lock

[account6032]
max connections = 25
path = /srv/3/node/
read only = false
lock file = /var/lock/account6032.lock

[account6042]
max connections = 25
path = /srv/4/node/
read only = false
lock file = /var/lock/account6042.lock

EOF
```



# Step Five: RSYNC

```
cat << EOF >> /etc/rsyncd.conf
[container6011]
max connections = 25
path = /srv/1/node/
read only = false
lock file = /var/lock/container6011.lock

[container6021]
max connections = 25
path = /srv/2/node/
read only = false
lock file = /var/lock/container6021.lock

[container6031]
max connections = 25
path = /srv/3/node/
read only = false
lock file = /var/lock/container6031.lock

[container6041]
max connections = 25
path = /srv/4/node/
read only = false
lock file = /var/lock/container6041.lock

EOF
```



# Step Five: RSYNC

```
cat << EOF >> /etc/rsyncd.conf
[object6010]
max connections = 25
path = /srv/1/node/
read only = false
lock file = /var/lock/object6010.lock

[object6020]
max connections = 25
path = /srv/2/node/
read only = false
lock file = /var/lock/object6020.lock

[object6030]
max connections = 25
path = /srv/3/node/
read only = false
lock file = /var/lock/object6030.lock

[object6040]
max connections = 25
path = /srv/4/node/
read only = false
lock file = /var/lock/object6040.lock

EOF

sed -e "s/RSYNC_ENABLE=false/RSYNC_ENABLE=true/g" /etc/default/rsync -i
```



# Step Six: Install Swift

```
su - swiftdemo
```

```
mkdir ~/bin
```

```
bzr init-repo swift
```

```
cd ~/swift; bzr branch lp:swift trunk
```

```
cd ~/swift/trunk; sudo python setup.py develop
```

```
cat << EOF >> ~/.bashrc
```

```
export
```

```
SWIFT_TEST_CONFIG_FILE=/etc/swift/func_test.conf
```

```
export PATH=${PATH}:~/bin
```

```
EOF
```

```
. ~/.bashrc
```



# Step Seven: SSL Certificates

```
cd /etc/swift
openssl req -new -x509 -nodes -
out cert.crt -keyout cert.key
# USE IP for COMMON NAME
```



# Step Eight: Configurations

```
sudo su -
export SWIFT_IP=172.16.81.129
cat << EOF >> /etc/swift/proxy-server.conf
[DEFAULT]
bind_port = 443
bind_ip = $SWIFT_IP
cert_file = /etc/swift/cert.crt
key_file = /etc/swift/cert.key
user = swiftdemo
log_facility = LOG_LOCAL1

[pipeline:main]
pipeline = healthcheck cache swauth proxy-server

[app:proxy-server]
use = egg:swift#proxy
allow_account_management = true

[filter:swauth]
use = egg:swift#swauth
super_admin_key = swauthkey
default_swift_cluster = local#https://$SWIFT_IP:443/v1

[filter:healthcheck]
use = egg:swift#healthcheck

[filter:cache]
use = egg:swift#memcache
EOF
```



# Step Eight: Configurations

```
cat << EOF >> /etc/swift/swift.conf
[swift-hash]
# random unique string that can never change
(DO NOT LOSE)
swift_hash_path_suffix = msst-swift-tutorial
EOF
```

# Step Eight: Account Server Configs

```
cd /etc/swift/account-server
cat << EOF > 1.conf
[DEFAULT]
devices = /srv/1/node
mount_check = false
bind_port = 6012
user = swiftdemo
log_facility = LOG_LOCAL2

[pipeline:main]
pipeline = account-server

[app:account-server]
use = egg:swift#account

[account-replicator]
vm_test_mode = yes

[account-auditor]

[account-reaper]
EOF

sed -e "s/srv\1/srv\2/" -e "s/6012/6022/" -e "s/LOG_LOCAL2/LOG_LOCAL3/" 1.conf > 2.conf
sed -e "s/srv\1/srv\3/" -e "s/6012/6032/" -e "s/LOG_LOCAL2/LOG_LOCAL4/" 1.conf > 3.conf
sed -e "s/srv\1/srv\4/" -e "s/6012/6042/" -e "s/LOG_LOCAL2/LOG_LOCAL5/" 1.conf > 4.conf
```



# Step Eight: Container Server Configs

```
cd /etc/swift/container-server
cat << EOF > 1.conf
[DEFAULT]
devices = /srv/1/node
mount_check = false
bind_port = 6011
user = swiftdemo
log_facility = LOG_LOCAL2

[pipeline:main]
pipeline = container-server

[app:container-server]
use = egg:swift#container

[container-replicator]
vm_test_mode = yes

[container-updater]

[container-auditor]
EOF
```

```
sed -e "s/srv\1/srv\2/" -e "s/601/602/" -e "s/LOG_LOCAL2/LOG_LOCAL3/" 1.conf > 2.conf
sed -e "s/srv\1/srv\3/" -e "s/601/603/" -e "s/LOG_LOCAL2/LOG_LOCAL4/" 1.conf > 3.conf
sed -e "s/srv\1/srv\4/" -e "s/601/604/" -e "s/LOG_LOCAL2/LOG_LOCAL5/" 1.conf > 4.conf
```



# Step Eight: Object Server Configs

```
cd /etc/swift/object-server
cat << EOF > 1.conf
[DEFAULT]
devices = /srv/1/node
mount_check = false
bind_port = 6010
user = swiftdemo
log_facility = LOG_LOCAL2

[pipeline:main]
pipeline = object-server

[app:object-server]
use = egg:swift#object

[object-replicator]
vm_test_mode = yes

[object-updater]

[object-auditor]
EOF
```

```
sed -e "s/srv\1/srv\2/" -e "s/601/602/" -e "s/LOG_LOCAL2/LOG_LOCAL3/" 1.conf > 2.conf
sed -e "s/srv\1/srv\3/" -e "s/601/603/" -e "s/LOG_LOCAL2/LOG_LOCAL4/" 1.conf > 3.conf
sed -e "s/srv\1/srv\4/" -e "s/601/604/" -e "s/LOG_LOCAL2/LOG_LOCAL5/" 1.conf > 4.conf
```



# Step Nine: Binaries

```
su - swiftdemo
```

```
cat << EOF > ~/bin/resetswift  
#!/bin/bash
```

```
swift-init all stop  
# find /var/log/swift -type f -exec rm -f {} \  
sudo umount /mnt/sdb1  
sudo mkfs.xfs -f -i size=1024 /srv/swift-disk  
sudo mount /mnt/sdb1  
sudo mkdir /mnt/sdb1/1 /mnt/sdb1/2 /mnt/sdb1/3 /mnt/sdb1/4  
sudo chown swiftdemo:swiftdemo /mnt/sdb1/*  
mkdir -p /srv/1/node/sdb1 /srv/2/node/sdb2 /srv/3/node/sdb3 /srv/4/node/sdb4  
sudo chown swiftdemo:swiftdemo /mnt/sdb1/*  
sudo rm -f /var/log/debug /var/log/messages /var/log/rsyncd.log /var/log/syslog  
sudo service rsyslog restart  
sudo service memcached restart  
EOF
```

# Step Nine: Binaries

```
export SWIFT_IP=172.16.81.129
cat << EOF > ~/bin/remakerings
#!/bin/bash

cd /etc/swift

rm -f *.builder *.ring.gz backups/*.builder backups/*.ring.gz

swift-ring-builder object.builder create 18 3 1
swift-ring-builder object.builder add z1-$SWIFT_IP:6010/sdb1 1
swift-ring-builder object.builder add z2-$SWIFT_IP:6020/sdb2 1
swift-ring-builder object.builder add z3-$SWIFT_IP:6030/sdb3 1
swift-ring-builder object.builder add z4-$SWIFT_IP:6040/sdb4 1
swift-ring-builder object.builder rebalance
swift-ring-builder container.builder create 18 3 1
swift-ring-builder container.builder add z1-$SWIFT_IP:6011/sdb1 1
swift-ring-builder container.builder add z2-$SWIFT_IP:6021/sdb2 1
swift-ring-builder container.builder add z3-$SWIFT_IP:6031/sdb3 1
swift-ring-builder container.builder add z4-$SWIFT_IP:6041/sdb4 1
swift-ring-builder container.builder rebalance
swift-ring-builder account.builder create 18 3 1
swift-ring-builder account.builder add z1-$SWIFT_IP:6012/sdb1 1
swift-ring-builder account.builder add z2-$SWIFT_IP:6022/sdb2 1
swift-ring-builder account.builder add z3-$SWIFT_IP:6032/sdb3 1
swift-ring-builder account.builder add z4-$SWIFT_IP:6042/sdb4 1
swift-ring-builder account.builder rebalance
EOF
```



# Step Nine: Binaries

```
cat << EOF > ~/bin/startmain
#!/bin/bash
```

```
swift-init main start
EOF
```

```
cat << EOF > ~/bin/recreateaccounts
#!/bin/bash
```

```
# Replace swauthkey with whatever your super_admin key is (recorded in
# /etc/swift/proxy-server.conf).
swauth-prep -K swauthkey -A https://$SWIFT_IP:443/auth/
swauth-add-user -K swauthkey -A https://$SWIFT_IP:443/auth/ -a test tester
testing
swauth-add-user -K swauthkey -A https://$SWIFT_IP:443/auth/ -a test2 tester2
testing2
swauth-add-user -K swauthkey -A https://$SWIFT_IP:443/auth/ test tester3 testing3
swauth-add-user -K swauthkey -A https://$SWIFT_IP:443/auth/ -a -r reseller
reseller reseller
EOF
```





# Step Nine: Binaries

```
cat << EOF > ~/bin/startrest  
#!/bin/bash
```

```
swift-init rest start  
EOF
```

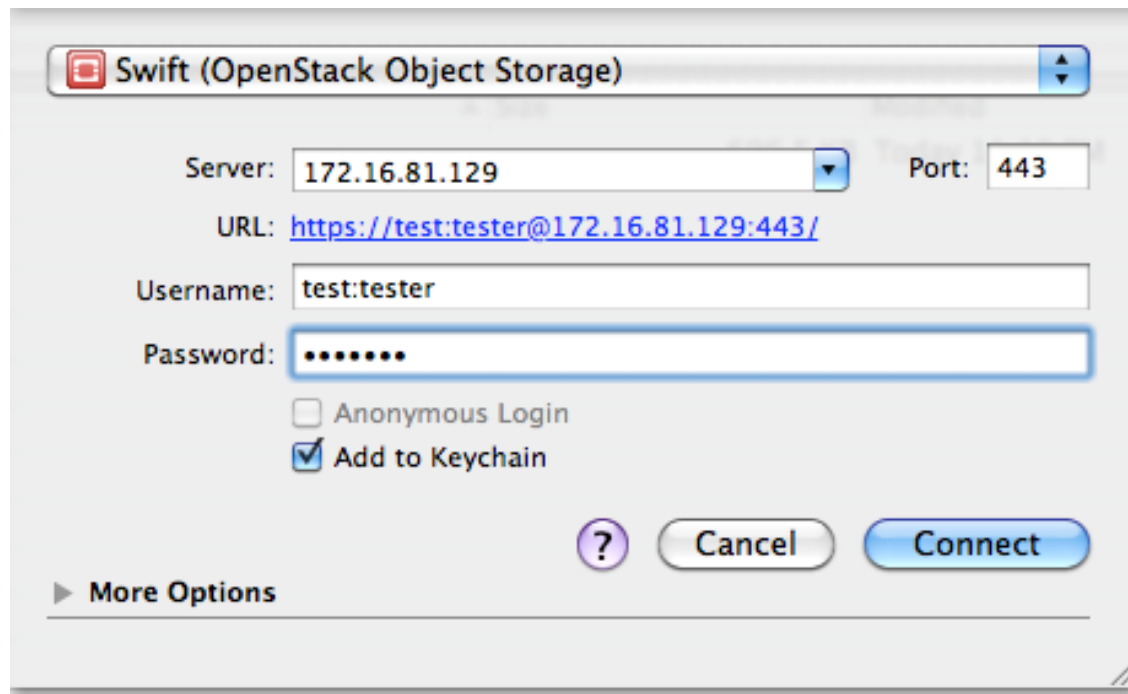
```
chmod +x ~/bin/*
```

# Step Ten: Running It

```
. ~/.bashrc  
remakerings  
cd ~/swift/trunk; ./unittests  
sudo bin/startmain  
recreateaccounts
```

# Fire up Cyberduck!

```
defaults write ch.sudo.cyberduck  
cf.authentication.context /auth/v1.0
```



The image shows a dialog box titled "Swift (OpenStack Object Storage)". It contains the following fields and options:

- Server: 172.16.81.129
- Port: 443
- URL: <https://test:tester@172.16.81.129:443/>
- Username: test:tester
- Password: [masked with dots]
- Anonymous Login
- Add to Keychain
- Buttons: ? (help), Cancel, Connect
- More Options: ▶ More Options

# QUESTIONS AND ANSWERS



Me: Joshua McKenty

Twitter: @jmckenty

Email: [joshua@piston.cc](mailto:joshua@piston.cc)

Cell: 650-283-6846





“Wilber is probably taking this Cloud computing too seriously.”