

- 云计算

- OpenStack

- Keystone

- 克隆虚拟机

- 基础设置

- Network

- NTP client

- Keystone Software install

- Base Node

- Ctrl Node

## 云计算

## OpenStack

### Keystone

克隆虚拟机

```
virt-clone -o a260c73 -n 21ctrl -f /var/lib/libvirt/images/21ctrl.qcow2
```

基础设置

### Network

1. 计算机名
2. IP地址

```
nmcli general hostname ctrl.blue.edu
```

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.91/24 ipv4.gateway 192.168.1.1 ipv4.dns "192.168.1.90 119.29.29.29" autoconnect yes ipv4.method manual ipv6.method ignore
```

```
vim /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
TYPE=Ethernet
BOOTPROTO=none
NAME=eth1
UUID=b29bc655-ddcb-402a-bdec-ab4711cda640
DEVICE=eth1
ONBOOT=yes
```

### NTP client

```
/etc/chrony.conf
```

```
server base.blue.edu
```

```
rpm -qa |grep chrony
```

```
cp /etc/chrony.conf /etc/chrony.conf.ds.bak  
cat /etc/chrony.conf.ds.bak | grep -v ^# | uniq > /etc/chrony.conf  
sed -i '/server/d' /etc/chrony.conf  
echo -e "server base.blue.edu" >> /etc/chrony.conf
```

```
systemctl enable chronyd.service  
systemctl restart chronyd.service
```

```
chronyc sources -v
```

## Keystone Software install

### Base Node

```
mysql -uroot -proot  
CREATE DATABASE keystone;  
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'KEYSTONE_DBPASS';  
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'KEYSTONE_DBPASS';  
SHOW DATABASES;  
EXIT
```

### Ctrl Node

```
yum install python-openstackclient  
yum install openstack-selinux
```

```
yum install openstack-keystone httpd mod_wsgi
```

```
/etc/keystone/keystone.conf
```

```
[database]  
connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@base.blue.edu/keystone  
  
[token]  
provider = fernet
```

```
cp /etc/keystone/keystone.conf /etc/keystone/keystone.conf.ds.bak  
cat /etc/keystone/keystone.conf.ds.bak | grep -v ^# | uniq > /etc/keystone/keystone.conf  
  
sed -i '/^\[database\]$/a connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@base.blue.edu/keystone' /etc/keystone/keystone.conf  
sed -i '/^\[token\]$/a provider = fernet' /etc/keystone/keystone.conf
```

```
su -s /bin/sh -c "keystone-manage db_sync" keystone (38)
```

```
keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone  
keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

```
keystone-manage bootstrap --bootstrap-password ADMIN_PASS \  
--bootstrap-admin-url http://ctrl.blue.edu:35357/v3/ \  
--bootstrap-internal-url http://ctrl.blue.edu:5000/v3/ \  
--bootstrap-public-url http://ctrl.blue.edu:5000/v3/ \  
--bootstrap-region-id RegionOne
```

## http

```
/etc/httpd/conf/httpd.conf
```

```
ServerName ctrl.blue.edu
```

```
cp /etc/httpd/conf/httpd.conf /etc/httpd/conf/httpd.conf.ds.bak  
cat /etc/httpd/conf/httpd.conf | grep -v ^# | uniq | grep ServerName  
echo -e "ServerName ctrl.blue.edu" >> /etc/httpd/conf/httpd.conf
```

```
ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/
```

```
systemctl enable httpd.service  
systemctl restart httpd.service
```

## 环境变量

```
export OS_USERNAME=admin
export OS_PASSWORD=ADMIN_PASS
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_AUTH_URL=http://ctrl.blue.edu:35357/v3
export OS_IDENTITY_API_VERSION=3
```

```
openstack project create --domain default --description "Service Project" service
openstack project create --domain default --description "Demo Project" demo
openstack user    create --domain default --password DEMO_PASS demo

openstack role create user
openstack role add --project demo --user demo user
```

/etc/keystone/keystone-paste.ini

```
[pipeline:public_api]
pipeline = healthcheck cors sizelimit http_proxy_to_wsgi osprofiler url_normaliz
e request_id build_auth_context token_auth json_body ec2_extension public_service

[pipeline:admin_api]
pipeline = healthcheck cors sizelimit http_proxy_to_wsgi osprofiler url_normaliz
e request_id build_auth_context token_auth json_body ec2_extension s3_extension ad
min_service

[pipeline:api_v3]
pipeline = healthcheck cors sizelimit http_proxy_to_wsgi osprofiler url_normaliz
e request_id build_auth_context token_auth json_body ec2_extension_v3 s3_extensio
n service_v3
```

```
cp /etc/keystone/keystone-paste.ini /etc/keystone/keystone-paste.ini.ds.bak
```

## Verify

```
unset OS_AUTH_URL OS_PASSWORD
```

```
openstack --os-auth-url http://ctrl.blue.edu:35357/v3 \
--os-project-domain-name default \
--os-user-domain-name default \
--os-project-name admin \
--os-username admin token issue
ADMIN_PASS
```

```
openstack --os-auth-url http://ctrl.blue.edu:5000/v3 \
--os-project-domain-name default \
--os-user-domain-name default \
--os-project-name demo \
--os-username demo token issue
DEMO_PASS
```

## Script

```
vim ~/admin-openrc

export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=ADMIN_PASS
export OS_AUTH_URL=http://ctrl.blue.edu:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

```
echo -e "export OS_PROJECT_DOMAIN_NAME=Default\nexport OS_USER_DOMAIN_NAME=Default
\nexport OS_PROJECT_NAME=admin\nexport OS_USERNAME=admin\nexport OS_PASSWORD=ADMIN
_PASS\nexport OS_AUTH_URL=http://ctrl.blue.edu:35357/v3\nexport OS_IDENTITY_API_V
ERSION=3\nexport OS_IMAGE_API_VERSION=2" > ~/admin-openrc
```

```
vim ~/demo-openrc

export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=DEMO_PASS
export OS_AUTH_URL=http://ctrl.blue.edu:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

```
echo -e "export OS_PROJECT_DOMAIN_NAME=Default\nexport OS_USER_DOMAIN_NAME=Default
\nexport OS_PROJECT_NAME=demo\nexport OS_USERNAME=demo\nexport OS_PASSWORD=DEMO_PA
SS\nexport OS_AUTH_URL=http://ctrl.blue.edu:5000/v3\nexport OS_IDENTITY_API_VERSIO
N=3\nexport OS_IMAGE_API_VERSION=2" > ~/demo-openrc
```

```
. admin-openrc  
openstack token issue
```