openstack devstack脚本安装（多结点，控制结点安装）

来自DEVSTACK官方文档，加入了网易的更新源，电信的速度1m/s以上，不用一个小时搞掂  
volume在装系统时分了个lvm，/dev/sda6  
swift在装系统时分了个lvm， /dev/sda7  
[localrc](http://blog.163.com/yyyhhhffflove@126/blog/static/373319932013138331202/)和local.sh要放在devstack目录下，  
  
/etc/network/interfaces:  
##  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto eth0  
iface eth0 inet static  
address 10.50.9.230  
netmask 255.255.255.0  
gateway 10.50.9.254  
dns-nameservers 8.8.8.8  
##  
  
  
  
###################################################################  
cd /etc/apt/  
wget http://mirrors.163.com/.help/sources.list.precise  
mv sources.list sources.list\_copy  
mv sources.list.precise sources.list  
cat >>sources.list<<EOF  
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-proposed/folsom main  
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/folsom main  
EOF  
cat sources.list\_copy >> sources.list   
########################################  
更新操作:  
  
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5EDB1B62EC4926EA  
apt-get update   
  
apt-get install -y git  
  
  
  
groupadd stack  
useradd -g stack -s /bin/bash -d /opt/stack -m stack  
echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers  
  
用stack登陆  
  
mkdir ~/.ssh; chmod 700 ~/.ssh  
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCyYjfgyPazTvGpd8OaAvtU2utL8W6gWC4JdRS1J95GhNNfQd657yO6s1AH5KYQWktcE6FO/xNUC2reEXSGC7ezy+sGO1kj9Limv5vrvNHvF1+wts0Cmyx61D2nQw35/Qz8BvpdJANL7VwP/cFI/p3yhvx2lsnjFE3hN8xRB2LtLUopUSVdBwACOVUmH2G+2BWMJDjVINd2DPqRIA4Zhy09KJ3O1Joabr0XpQL0yt/I9x8BVHdAx6l9U0tMg9dj5+tAjZvMAFfye3PJcYwwsfJoFxC8w/SLtqlFX7Ehw++8RtvomvuipLdmWCy+T9hIkl+gHYE4cS3OIqXH7f49jdJf jesse@spacey.local" > ~/.ssh/authorized\_keys  
  
  
git clone git://github.com/openstack-dev/devstack.git  
cd devstack  
  
  
############  
[localrc](http://blog.163.com/yyyhhhffflove@126/blog/static/373319932013138331202/) 放到/devstack下, local.sh 在example文件夹中，也把它拿到devstack目录下  
##########  
In the multi-node configuration the first 10 or so IPs in the private subnet are usually reserved. Add this to local.sh to have it run after every stack.sh run:  
  
for i in `seq 2 10`; do /opt/stack/nova/bin/nova-manage fixed reserve 10.4.128.$i; done  
  
cinder:  
pvcreate /dev/sda6  
vgcreate stack-volumes /dev/sda6  
  
  
Fire up OpenStack:  
  
./stack.sh  
  
A stream of activity ensues. When complete you will see a summary of stack.sh's work, including the relevant URLs, accounts and passwords to poke at your shiny new OpenStack. The most recent log file is available in stack.sh.log.  
Configure Compute Nodes  
  
The compute nodes only run the OpenStack worker services. For additional machines, create a localrc with:  
  
HOST\_IP=192.168.42.12 # change this per compute node  
FLAT\_INTERFACE=eth0  
FIXED\_RANGE=10.4.128.0/20  
FIXED\_NETWORK\_SIZE=4096  
FLOATING\_RANGE=192.168.42.128/25  
MULTI\_HOST=1  
LOGFILE=/opt/stack/logs/stack.sh.log  
ADMIN\_PASSWORD=labstack  
MYSQL\_PASSWORD=supersecret  
RABBIT\_PASSWORD=supersecrete  
SERVICE\_PASSWORD=supersecrete  
SERVICE\_TOKEN=xyzpdqlazydog  
MYSQL\_HOST=192.168.42.11  
RABBIT\_HOST=192.168.42.11  
GLANCE\_HOSTPORT=192.168.42.11:9292  
ENABLED\_SERVICES=n-cpu,n-net,n-api,c-sch,c-api,c-vol  
  
Fire up OpenStack:  
  
./stack.sh  
  
A stream of activity ensues. When complete you will see a summary of stack.sh's work, including the relevant URLs, accounts and passwords to poke at your shiny new OpenStack. The most recent log file is available in stack.sh.log.  
Cleaning Up After DevStack  
  
Shutting down OpenStack is now as simple as running the included unstack.sh script:  
  
./unstack.sh  
  
Sometimes running instances are not cleaned up. DevStack attempts to do this when it runs but there are times it needs to still be done by hand:  
  
sudo rm -rf /etc/libvirt/qemu/inst\*  
sudo virsh list | grep inst | awk '{print $1}' | xargs -n1 virsh destroy  
  
Options pimp your stack  
Additional Users  
  
DevStack creates two users and two tenants, admin and demo. admin is exactly what it sounds like, a priveleged administrative account that is a member of both the admin and demo tenants. demo is a normal user account that is only a menber of the demo tenant. Creating additional OpenStack users can be done through the dashboard, sometimes it is easier to do them in bulk from a script, especially since they get blown away every time stack.sh runs. The following steps are ripe for scripting:  
  
# Get admin creds  
. openrc admin admin  
          
# List existing tenants  
keystone tenant-list  
  
# List existing users  
keystone user-list  
  
# Add a user and tenant  
NAME=bob  
PASSWORD=BigSecrete  
TENANT=$NAME  
keystone tenant-create --name=$NAME  
keystone user-create --name=$NAME --pass=$PASSWORD  
keystone user-role-add --user-id=<bob-user-id> --tenant-id=<bob-tenant-id> --role-id=<member-role-id>  
# member-role-id comes from the existing member role created by stack.sh  
# keystone role-list  
  
Swift  
  
Swift requires a significant amount of resources and is disabled by default in DevStack. The support in DevStack is geared toward a minimal installation but can be used for testing. To implement a true multi-node test of Swift required more than DevStack provides. Enabling it is as simple as enabling the swift service in localrc:  
  
enable\_service swift  
  
Swift will put its data files in SWIFT\_DATA\_DIR (default /opt/stack/data/swift). The size of the data 'partition' created (really a loop-mounted file) is set by SWIFT\_LOOPBACK\_DISK\_SIZE. The Swift config files are located in SWIFT\_CONFIG\_DIR (default /etc/swift). All of these settings can be overridden in (wait for it...) localrc.  
Volumes  
  
DevStack will automatically use an existing LVM volume group named stack-volumes to store cloud-created volumes. If stack-volumes doesn't exist, DevStack will set up a 5Gb loop-mounted file to contain it. This obviously limits the number and size of volumes that can be created inside OpenStack. The size can be overridden by setting VOLUME\_BACKING\_FILE\_SIZE in localrc.  
  
stack-volumes can be pre-created on any physical volume supported by Linux's LVM. The name of the volume group can be changed by setting VOLUME\_GROUP in localrc. stack.sh deletes all logical volumes in VOLUME\_GROUP that begin with VOLUME\_NAME\_PREFIX as part of cleaning up from previous runs. It is recommended to not use the root volume group as VOLUME\_GROUP.  
  
The details of creating the volume group depends on the server hardware involved but looks something like this:  
  
pvcreate /dev/sdc  
vgcreate stack-volumes /dev/sdc  
  
Syslog  
  
DevStack is capable of using rsyslog to agregate logging across the cluster. It is off by default; to turn it on set SYSLOG=True in localrc. SYSLOG\_HOST defaults to HOST\_IP; on the compute nodes it must be set to the IP of the cluster controller to send syslog output there. In the example above, add this to the compute node localrc:  
  
SYSLOG\_HOST=192.168.42.11  
  
Using Alternate Repositories/Branches  
  
The git repositories for all of the OpenStack services are defined in stackrc. Since this file is a part of the DevStack package changes to it will probably be overwritten as updates are applied. Every setting in stackrc can be redefined in localrc.  
  
To change the repository or branch that a particular OpenStack service is created from, simply change the value of \*\_REPO or \*\_BRANCH corresponding to that service.  
  
After making changes to the repository or branch, if RECLONE is not set in localrc it may be necessary to remove the corresponding directory from /opt/stack to force git to re-clone the repository.  
  
For example, to pull Nova from a proposed release candidate in the primary Nova repository:  
  
NOVA\_BRANCH=rc-proposed  
  
To pull Glance from an experimental fork:  
  
GLANCE\_BRANCH=try-something-big  
GLANCE\_REPO=https://github.com/mcuser/glance.git  
  
Notes stuff you might need to know  
Reset the Bridge  
  
How to reset the bridge configuration:  
  
sudo brctl delif br100 eth0.926  
sudo ip link set dev br100 down  
sudo brctl delbr br100  
  
Set MySQL Password  
  
If you forgot to set the root password you can do this:  
  
mysqladmin -u root -pnova password 'supersecret'  
  
? Openstack Foundation 2011-2013 — this is not an official OpenStack project...

**openstack devstack 脚本安装（多结点，计算结点安装）**

Install a couple of packages to bootstrap configuration:  
  
apt-get install -y git sudo  
  
Network Configuration  
  
The first iteration of the lab uses OpenStack's FlatDHCP network controller so only a single network will be required. It should be on its own subnet without DHCP; the host IPs and floating IP pool(s) will come out of this block. This example uses the following:  
  
    Gateway: 192.168.42.1  
    Physical nodes: 192.168.42.11-192.168.42.99  
    Floating IPs: 192.168.42.128-192.168.42.254  
  
Configure each node with a static IP in /etc/network/interfaces:  
  
   auto eth0  
   iface eth0 inet static  
           address 192.168.42.11  
           netmask 255.255.255.0  
           gateway 192.168.42.1  
  
Installation shake and bake  
Add the DevStack User  
  
OpenStack runs as a non-root user that has sudo access to root. There is nothing special about the name, we'll use stack here. Every node must use the same name and preferably uid. If you created a user during the OS install you can use it and give it sudo priviledges below. Otherwise create the stack user:  
  
groupadd stack  
useradd -g stack -s /bin/bash -d /opt/stack -m stack  
  
This user will be making many changes to your system during installation and operation so it needs to have sudo priviledges to root without a password:  
  
echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers  
  
From here on use the stack user. Logout and login as the stack user.  
Set Up Ssh  
  
Set up the stack user on each node with an ssh key for access:  
  
mkdir ~/.ssh; chmod 700 ~/.ssh  
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCyYjfgyPazTvGpd8OaAvtU2utL8W6gWC4JdRS1J95GhNNfQd657yO6s1AH5KYQWktcE6FO/xNUC2reEXSGC7ezy+sGO1kj9Limv5vrvNHvF1+wts0Cmyx61D2nQw35/Qz8BvpdJANL7VwP/cFI/p3yhvx2lsnjFE3hN8xRB2LtLUopUSVdBwACOVUmH2G+2BWMJDjVINd2DPqRIA4Zhy09KJ3O1Joabr0XpQL0yt/I9x8BVHdAx6l9U0tMg9dj5+tAjZvMAFfye3PJcYwwsfJoFxC8w/SLtqlFX7Ehw++8RtvomvuipLdmWCy+T9hIkl+gHYE4cS3OIqXH7f49jdJf jesse@spacey.local" > ~/.ssh/authorized\_keys  
###################################################################  
cd /etc/apt/  
wget http://mirrors.163.com/.help/sources.list.precise  
mv sources.list sources.list\_copy  
mv sources.list.precise sources.list  
cat >>sources.list<<EOF  
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-proposed/folsom main  
deb http://ubuntu-cloud.archive.canonical.com/ubuntu precise-updates/folsom main  
EOF  
cat sources.list\_copy >> sources.list   
########################################  
更新操作:  
  
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 5EDB1B62EC4926EA  
apt-get update   
Download DevStack  
  
Grab the latest version of DevStack from github:  
  
git clone git://github.com/openstack-dev/devstack.git  
cd devstack  
  
Up to this point all of the steps apply to each node in the cluster. From here on there are some differences between the cluster controller (aka 'head node') and the compute nodes.  
  
  
####################  
  
Configure Compute Nodes  
  
The compute nodes only run the OpenStack worker services. For additional machines, create a localrc with:  
  
HOST\_IP=192.168.42.12 # change this per compute node  
FLAT\_INTERFACE=eth0  
FIXED\_RANGE=10.4.128.0/20  
FIXED\_NETWORK\_SIZE=4096  
FLOATING\_RANGE=192.168.42.128/25  
MULTI\_HOST=1  
LOGFILE=/opt/stack/logs/stack.sh.log  
ADMIN\_PASSWORD=labstack  
MYSQL\_PASSWORD=supersecret  
RABBIT\_PASSWORD=supersecrete  
SERVICE\_PASSWORD=supersecrete  
SERVICE\_TOKEN=xyzpdqlazydog  
MYSQL\_HOST=192.168.42.11  
RABBIT\_HOST=192.168.42.11  
GLANCE\_HOSTPORT=192.168.42.11:9292  
ENABLED\_SERVICES=n-cpu,n-net,n-api,c-sch,c-api,c-vol  
  
  
cinder:  
pvcreate /dev/sda6  
vgcreate stack-volumes /dev/sda6  
  
  
Fire up OpenStack:  
  
./stack.sh  
  
A stream of activity ensues. When complete you will see a summary of stack.sh's work, including the relevant URLs, accounts and passwords to poke at your shiny new OpenStack. The most recent log file is available in stack.sh.log.