RDO Packstack 安装 Openstack Icehouse CentOS 6.5 单网卡

CentOS 6.5 使用Redhat RDO packstack 安装openstack Icehouse

上一篇实际是有点问题的，就是没有办法访问vm，哈哈。

基本配置： vmware workstation 单网卡（eth1），桥接到无线网卡上，安装CentOS6.4

参考了陈沙克的文章：[http://www.chenshake.com/centos6-4-single-card-all-in-one-install-havana/](http://www.chenshake.com/centos6-4-single-card-all-in-one-install-havana/" \t "_blank)

就是把br-ex 桥接到 eth1 上，然后就可以设置无线网络的floating IP给vm使用，这样就可以访问VM了。

<http://openstack.redhat.com/Quickstart>

Please name the host with a fully qualified domain name rather than a short-form nameto avoid DNS issues with Packstack.

# vi /etc/hosts

# 127.0.0.1

#

127.0.0.1 centos2

# sudo yum update -y

# sudo yum install -y # reboot

# sudo yum install -y openstack-packstack

# packstack --allinone

installation complete successful

Additional information:

\* Anew answerfile was created in: /root/packstack-answers-20140601-050232.txt

\*Time synchronization installation was skipped. Please note that unsynchronizedtime on server instances might be problem for some OpenStack components.

\*Did not create a cinder volume group, one already existed

\*File /root/keystonerc\_admin has been created on OpenStack client host192.168.1.20. To use the command line tools you need to source the file.

\* Toaccess the OpenStack Dashboard browse to http://192.168.1.20/dashboard .

Please, find your login credentials storedin the keystonerc\_admin in your home directory.

\* Touse Nagios, browse to http://192.168.1.20/nagios username : nagiosadmin,password : a35344dc01ad45ee

\*The installation log file is available at:/var/tmp/packstack/20140601-050232-nSNyWK/openstack-setup.log

\*The generated manifests are available at:/var/tmp/packstack/20140601-050232-nSNyWK/manifests

Cat keystonerc\_admin

export OS\_USERNAME=admin

export OS\_TENANT\_NAME=admin

export OS\_PASSWORD=00539d19e3b04a9d

exportOS\_AUTH\_URL=http://192.168.1.20:5000/v2.0/

# cat ifcfg-eth1

DEVICE=eth1

TYPE=OVSPort

DEVICETYPE=ovs

OVS\_BRIDGE=br-ex

ONBOOT=yes

HWADDR=00:0C:29:39:36:53

# IPV6INIT=no

# UUID=0e6e86b5-721d-4219-a9fd-2076990f9e1f

# BOOTPROTO=none

# IPADDR=192.168.1.20

# PREFIX=24

# GATEWAY=192.168.1.1

# DNS1=202.106.0.20

# DEFROUTE=yes

# IPV4\_FAILURE\_FATAL=yes

# LAST\_CONNECT=1401649435

# cat ifcfg-br-ex

DEVICE=br-ex

DEVICETYPE=ovs

TYPE=OVSBridge

BOOTPROTO=static

IPADDR=192.168.1.20

NETMASK=255.255.255.0

ONBOOT=yes

登录到dashboard.

1, 添加外部网络（就是我的无线网络） External\_Network

添加 External\_Subnet 192.168.1.0/24

2, 添加私有网络（就是VM所使用的网络） Private\_Network

添加Private\_Subnet 172.16.1.0/24

3, 添加路由器，router, 设置网关gateway为外部网络External\_Subnet.

在路由器router上, 添加 172.16.1.0/24 网络的接口

4, 启动实例, 选择 Private\_Subnet.

5，设置 floating IP. (192.168.1.131)

6，设置安全规则，允许icmp和ssh对内部网络的访问.

打完收工。

# ip netns

qrouter-6b53c093-9504-416a-af7e-6d2296def3a1

qdhcp-272f94df-64d0-492d-bfac-5f9b57888faa

# ip netns exec qrouter-6b53c093-9504-416a-af7e-6d2296def3a1 ip addr

10: lo: mtu16436 qdisc noqueue state UNKNOWN

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

13: qr-86d8f14a-3c: mtu 1500 qdisc noqueue state UNKNOWN

link/ether fa:16:3e:be:c5:c1 brd ff:ff:ff:ff:ff:ff

inet 172.16.1.1/24 brd 172.16.1.255 scope global qr-86d8f14a-3c

inet6 fe80::f816:3eff:febe:c5c1/64 scope link

valid\_lft forever preferred\_lft forever

14: qg-b04ec080-fd: mtu 1500 qdisc noqueue state UNKNOWN

link/ether fa:16:3e:fa:f5:b7 brd ff:ff:ff:ff:ff:ff

inet 192.168.1.130/24 brd 192.168.1.255 scope global qg-b04ec080-fd

inet 192.168.1.131/32 brd 192.168.1.131 scope global qg-b04ec080-fd

inet6 fe80::f816:3eff:fefa:f5b7/64 scope link

valid\_lft forever preferred\_lft forever

# ip netns exec qrouter-6b53c093-9504-416a-af7e-6d2296def3a1 ip route

192.168.1.0/24 dev qg-b04ec080-fd proto kernel scope link src 192.168.1.130

172.16.1.0/24 dev qr-86d8f14a-3c proto kernel scope link src 172.16.1.1

default via 192.168.1.1 dev qg-b04ec080-fd

# ip netns exec qrouter-6b53c093-9504-416a-af7e-6d2296def3a1 iptables -t nat -S

-P PREROUTING ACCEPT

-P POSTROUTING ACCEPT

-P OUTPUT ACCEPT

-N neutron-l3-agent-OUTPUT

-N neutron-l3-agent-POSTROUTING

-N neutron-l3-agent-PREROUTING

-N neutron-l3-agent-float-snat

-N neutron-l3-agent-snat

-N neutron-postrouting-bottom

-A PREROUTING -jneutron-l3-agent-PREROUTING

-A POSTROUTING -jneutron-l3-agent-POSTROUTING

-A POSTROUTING -jneutron-postrouting-bottom

-A OUTPUT -j neutron-l3-agent-OUTPUT

-A neutron-l3-agent-OUTPUT -d192.168.1.131/32 -j DNAT --to-destination 172.16.1.2

-A neutron-l3-agent-POSTROUTING ! -iqg-b04ec080-fd ! -o qg-b04ec080-fd -m conntrack ! --ctstate DNAT -j ACCEPT

-A neutron-l3-agent-PREROUTING -d169.254.169.254/32 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 9697

-A neutron-l3-agent-PREROUTING -d192.168.1.131/32 -j DNAT --to-destination 172.16.1.2

-A neutron-l3-agent-float-snat -s172.16.1.2/32 -j SNAT --to-source 192.168.1.131

-A neutron-l3-agent-snat -jneutron-l3-agent-float-snat

-A neutron-l3-agent-snat -s 172.16.1.0/24-j SNAT --to-source 192.168.1.130

-A neutron-postrouting-bottom -jneutron-l3-agent-snat

# ping 192.168.1.131

PING 192.168.1.131 (192.168.1.131) 56(84)bytes of data.

64 bytes from 192.168.1.131: icmp\_seq=1ttl=63 time=193 ms

64 bytes from 192.168.1.131: icmp\_seq=2ttl=63 time=2.99 ms

64 bytes from 192.168.1.131: icmp\_seq=3ttl=63 time=0.519 ms

64 bytes from 192.168.1.131: icmp\_seq=4ttl=63 time=1.38 ms

^C

--- 192.168.1.131 ping statistics ---

4 packets transmitted, 4 received, 0%packet loss, time 3082ms

rtt min/avg/max/mdev =0.519/49.512/193.150/82.934 ms