OpenStack部署手册

[1 部署背景 5](#_Toc903269885)

[2 系统架构 6](#_Toc1019387016)

[3 环境准备 7](#_Toc639396070)

[3.1 软件准备 7](#_Toc883147624)

[3.1.1 OS系统 7](#_Toc395875939)

[3.1.2 虚拟化软件 7](#_Toc1302725955)

[3.2 服务准备 7](#_Toc810791466)

[3.3 配置准备 8](#_Toc1287533566)

[3.3.1 语言设置 8](#_Toc637560039)

[3.3.2 网络环境设置 8](#_Toc1413856601)

[3.3.3 安全设置 8](#_Toc702538972)

[4 系统安装 9](#_Toc1203463493)

[4.1 配置软件源 9](#_Toc1062131242)

[4.2 安装packstack工具 9](#_Toc697077394)

[4.3 生成answer-file文件 9](#_Toc1028165138)

[4.4 创建逻辑卷 10](#_Toc1849170387)

[4.5 安装Openstack 10](#_Toc773728213)

[4.6 安装完成 10](#_Toc1428662953)

[4.7 登录dashboard 10](#_Toc1810247409)

[5 服务检测 11](#_Toc1767314997)

[6 创建实例 13](#_Toc992939731)

[6.1 网络配置 13](#_Toc1111964620)

[6.2 新建实例 15](#_Toc1910817033)

[6.3 定制启动镜像 17](#_Toc622508594)

[7 系统配置 18](#_Toc2087821359)

[7.1 MariaDB配置 18](#_Toc928064450)

[7.2 Horizon配置 19](#_Toc140019601)

[7.3 Nova配置 20](#_Toc1789918178)

[7.4 存储后端配置 29](#_Toc2114187043)

[7.4.1 Nova配置 29](#_Toc682121298)

[7.4.2 Glance配置（可选） 30](#_Toc849312850)

[7.5 GPU Passthrough配置 31](#_Toc869973280)

[7.5.1 系统配置 31](#_Toc1701508314)

[7.5.2 nova配置 35](#_Toc1488708920)

[7.5.3 flavor配置 36](#_Toc1753120904)

[7.5.4 CPU Topologies配置 37](#_Toc2097384254)

[7.6 稳定性/性能配置 38](#_Toc643951228)

[7.6.1 关闭auditd服务 38](#_Toc416428723)

[7.6.2 开启tuned服务 38](#_Toc1237434172)

[7.6.3 开启ksm服务 38](#_Toc1281511267)

[8 新增计算节点 39](#_Toc1830285324)

[8.1 环境准备 39](#_Toc1939973144)

[8.2 系统安装 40](#_Toc337491113)

[8.3 系统验证 41](#_Toc744932918)

[8.4 系统调优 41](#_Toc489566890)

[9 新增存储节点 42](#_Toc1365656251)

[9.1 环境准备 42](#_Toc446619657)

[9.2 系统安装 43](#_Toc1263295103)

[9.3 系统验证 44](#_Toc646835556)

[10 常用命令行 45](#_Toc109383418)

[11 FAQ 46](#_Toc883126452)

# 部署背景

公司有5台服务器，用于各个部门的开发和测试，然而，由于无法做到“环境”和资源的隔离，会导致：

* 各个开发部门之间的运行环境出现冲突
* 各个开发部门对服务器的资源进行竞争

Openstack可以解决上述“环境”和“资源”隔离的痛点，其具有如下优点：

* 虚拟机的运行环境完全隔离
* 虚拟机资源可以根据需求定制，且可动态调整
* 充分有效地利用整个集群的资源

# 系统架构

为了快速、简单地部署OpenStack，采用了Packstack（部署Pike版本）工具来先部署两台机器，为了尽可能充分利用资源，服务器“21-server”将充当多个角色，如下图2-1所示：

21-server

网络节点

计算节点

存储节点

控制节点

计算节点

22-server

图2-1： 系统架构

# 环境准备

## 3.1 软件准备

### OS系统

本次安装将基于CentOS7.4，需要提前安装好系统。

### 虚拟化软件

OpenStack的底层需要使用虚拟化软件来提供支持，因此需要在计算节点上安装虚拟化软件：

|  |
| --- |
| # yum groups install "Virtualization Host" 🡸 |

## 服务准备

在每台机器上执行如下命令：

|  |
| --- |
| # systemctl disable firewalld  # systemctl stop firewalld  # systemctl disable NetworkManager  # systemctl stop NetworkManager  # systemctl enable network  # systemctl start network  # systemctl stop iptables  # systemctl disable iptables |

## 配置准备

### 语言设置

设置每台机器OS的语言如下：

|  |
| --- |
| # echo $LANG  en\_US.UTF-8 |

### 网络环境设置

* 配置每台机器之间可以无密码访问，为了简化，每台机器使用相同的公钥和私钥
* 修改/etc/hosts，使得每台之间机器可以互相使用hostname进行通信

### 安全设置

关闭每台机器上的SELinux设置:

|  |
| --- |
| # vim /etc/selinux/config  ……  SELINUX=disabled  … … |

注：需要重启机器才能生效，可以使用”setenforce 0”临时关闭。

清空每台机器上的iptables规则：

|  |
| --- |
| # iptables -F |

# 系统安装

## 配置软件源

在控制节点上执行：

|  |
| --- |
| # yum install -y centos-release-openstack-pike |

## 安装packstack工具

在控制节点上执行：

|  |
| --- |
| # yum install -y openstack-packstack |

## 生成answer-file文件

在控制节点上执行：

|  |
| --- |
| # packstack --gen-answer-file=./yilong\_answer-file.txt |

说明：

* 生成的yilong\_answer-file.txt是个模板文件，根据需求对相关参数进行修改（如：IP地址）
* 本次安装的yilong\_answer-file.txt可以从repo：[https://github.com/kulong015/packstack-config](http://git.sky-data.cn/yilong.ren/packstack-config) 获取

## 创建逻辑卷

本次部署的“21-server”机器也将作为存储节点，cinder服务将使用cinder-volumes逻辑卷作为存储后端，默认情况下cinder将使用loop设备自动创建cinder-volumes逻辑卷来作为存储后端。为了利用”21-server“上的磁盘来存储数据，需要通过LVM工具来创建cinder-volume逻辑卷：

|  |
| --- |
| # pvcreate /dev/sdb /dev/sdc  # vgcreate cinder-volumes /dev/sdb /dev/sdc |

## 安装Openstack

在控制节点上执行：

|  |
| --- |
| # packstack --answer-file=./yilong\_answer-file.txt |

注：整个安装比较耗时，跟网络带宽、机器性能、安装机器的数量等有关。

## 安装完成

当出现如下信息，表示Openstack安装完成：

|  |
| --- |
| … …  … …  … …  \* Time synchronization installation was skipped. Please note that unsynchronized time on server instances might be problem for some OpenStack components.  … …  \* To access the OpenStack Dashboard browse to http://10.0.2.21/dashboard .  Please, find your login credentials stored in the keystonerc\_admin in your home directory.  \* The installation log file is available at: /var/tmp/packstack/20180109-010608-1aT85n/openstack-setup.log  \* The generated manifests are available at: /var/tmp/packstack/20180109-010608-1aT85n/manifests |

## 登录dashboard

根据安装结束打印出的URL，在浏览器上输入对应的URL，确认是否可以正常访问。

# 服务检测

使用Packstack安装OpenStack，默认不会安装相关工具，为了方便检测各个服务的状态，可以安装openstack-utils工具包：

|  |
| --- |
| [root@21-server ~]# yum list | grep openstack-utils  openstack-utils.noarch 2017.1-1.el7 centos-openstack-pike  [root@21-server ~]# yum install openstack-utils  ......  [root@21-server ~]# rpm -ql openstack-utils  /usr/bin/openstack-config  /usr/bin/openstack-service  /usr/bin/openstack-status  /usr/share/doc/openstack-utils-2017.1  ......  [root@21-server ~]# |

* 检测各个服务状态

在控制节点上，执行如下命令：

|  |
| --- |
| [root@21-server ~]# source ~/keystonerc\_admin  [root@21-server ~(keystone\_admin)]# openstack-status  == Nova services ==  openstack-nova-api: active  openstack-nova-compute: active  openstack-nova-network: inactive (disabled on boot)  openstack-nova-scheduler: active  openstack-nova-conductor: active  openstack-nova-console: inactive (disabled on boot)  openstack-nova-consoleauth: active  openstack-nova-xvpvncproxy: inactive (disabled on boot)  == Glance services ==  openstack-glance-api: active  openstack-glance-registry: active  == Keystone service ==  openstack-keystone: inactive (disabled on boot)  == Horizon service ==  openstack-dashboard: active  == neutron services ==  neutron-server: active  neutron-dhcp-agent: active  neutron-l3-agent: active  neutron-metadata-agent: active  neutron-openvswitch-agent: active  neutron-metering-agent: active  == Cinder services ==  openstack-cinder-api: active  openstack-cinder-scheduler: active  openstack-cinder-volume: active  openstack-cinder-backup: inactive (disabled on boot)  == Support services ==  openvswitch: active  dbus: active  target: active  rabbitmq-server: active  memcached: active  ...... |

注：上面有几个处于inactive状态的服务是其实正常现象，因为使用packstack安装时默认没有安装这些服务。

* 重启所有服务

|  |
| --- |
| [root@21-server ~]# source ~/keystonerc\_admin  [root@21-server ~(keystone\_admin)]# openstack-service restart |

* 关闭所有服务

|  |
| --- |
| [root@21-server ~]# source ~/keystonerc\_admin  [root@21-server ~(keystone\_admin)]# openstack-service stop |

# 创建实例

## 网络配置

在创建VM前，需要先配置网络，网络分为”外网“和”内网”：

* 外网：表示可以通过该子网连接internet
* 内网：表示在VM之间进行通信

在Web上，“外网”应该在Admin菜单项创建，在创建“外网”时，“Provider Network Type”需要选择Flat模式，”Physical Network“需要填写extnet，如下图7-1所示：

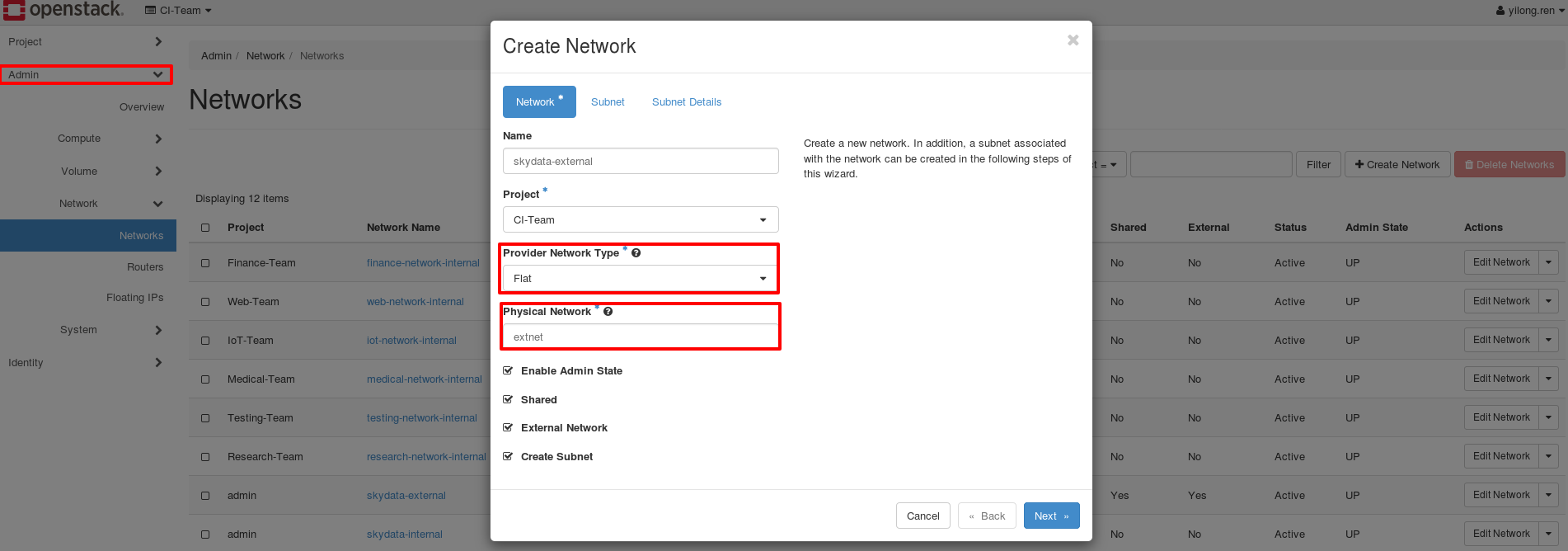


图7-1: 外网创建图

在Web上，“内网”应该在Project菜单项创建，如下图7-2所示：

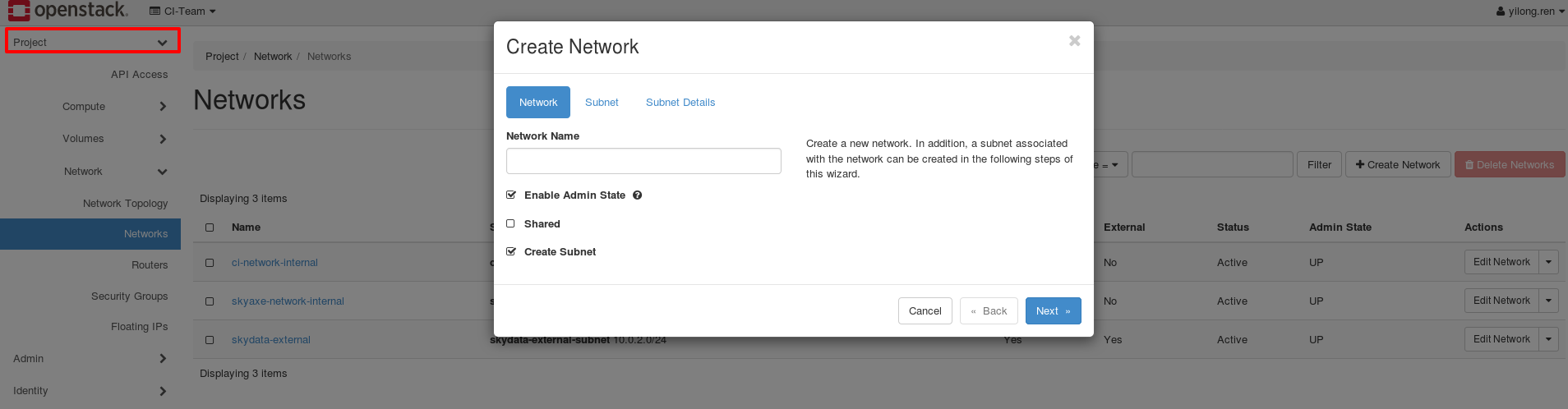


图7-2: 内网创建图

”外网“和”内网“创建好，需要创建一个路由器去连接它们。需要注意的是，在Project菜单项创建路由时，需要同时选择外网的Interface，否则 创建新增外网Interface时会占用网关，导致网络不通，如下图7-3所示：

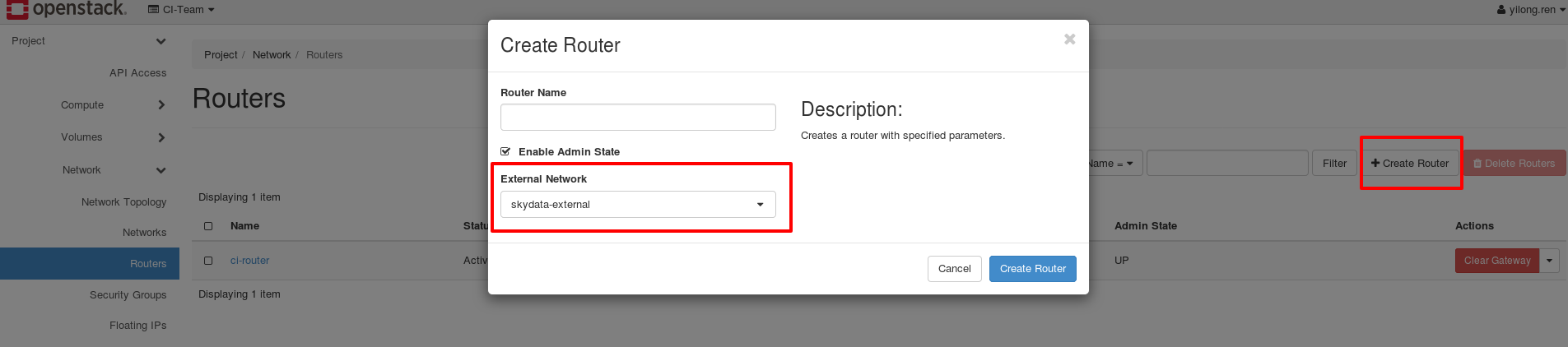
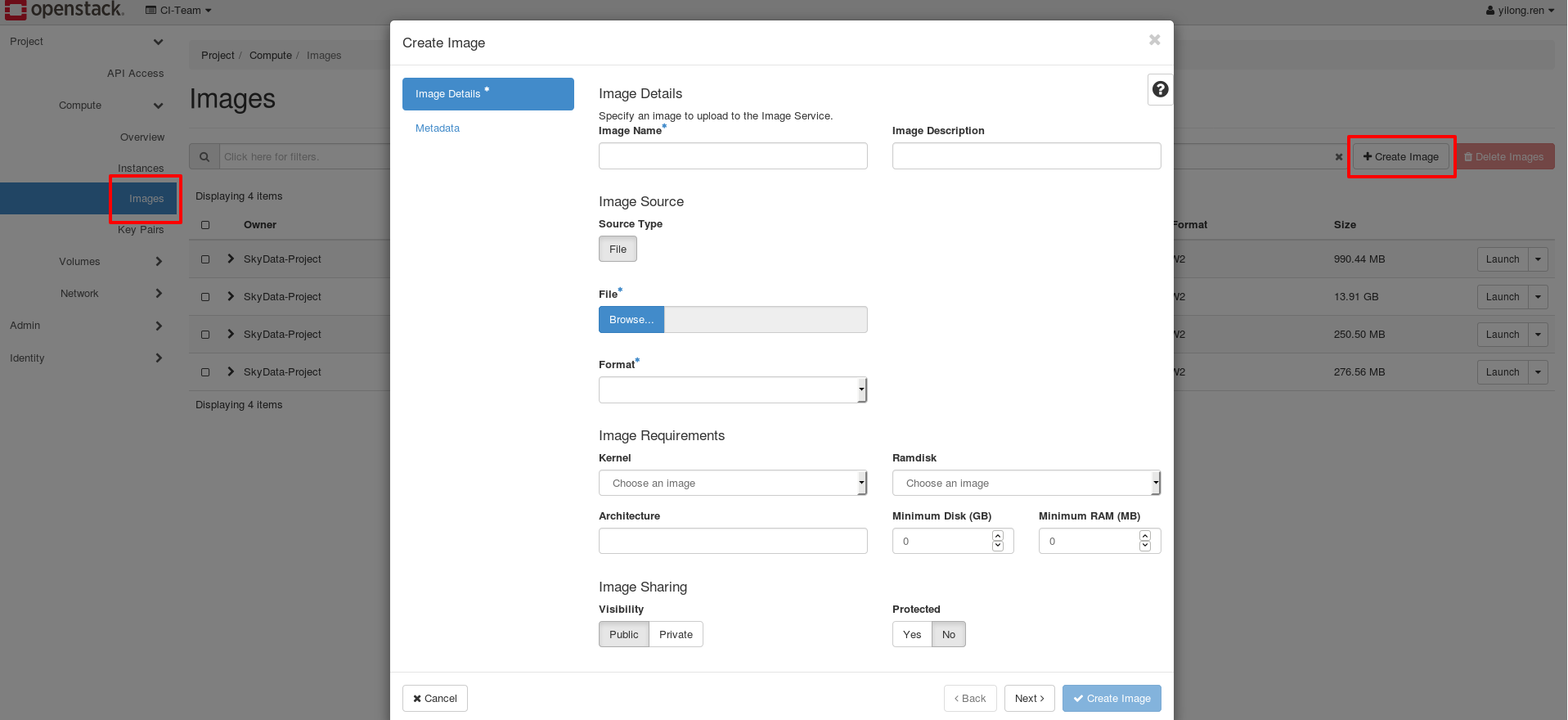


图7-3: 内网路由图

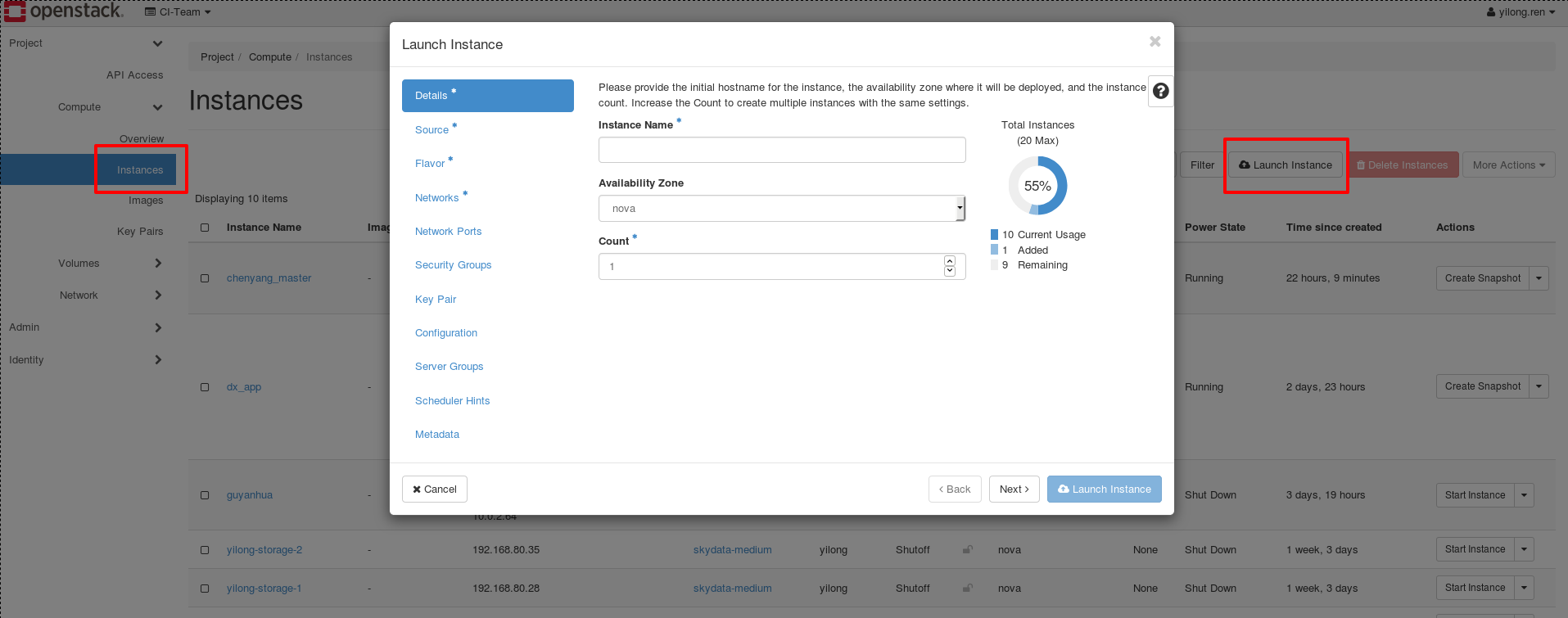
## 新建实例

1. 从Centos官网下载image文件: <http://cloud.centos.org/centos/7/images/>
2. 通过Web或命令行导入到glance仓库，如下图7-2所示：



图：7-2 上传image

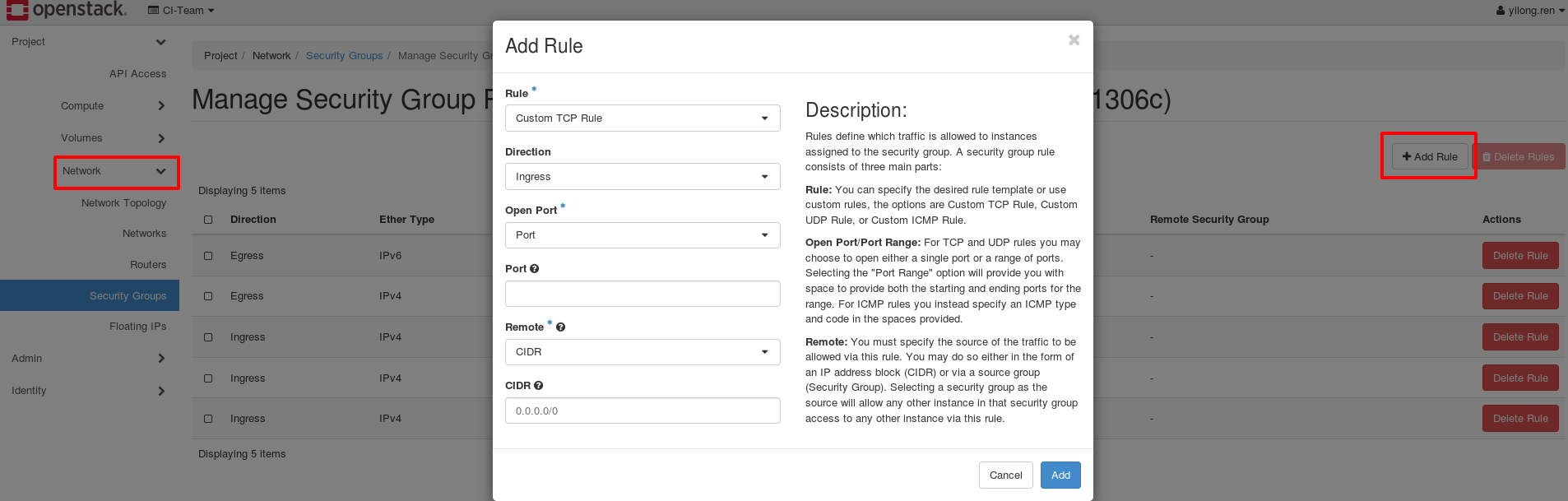
1. 通过Web使用这个image去创建一个VM，如下图7-3所示：



图：7-3 创建VM

4）修改security group 规则

Openstack默认的security group规则禁止ping和ssh，所以创建好的VM在分配floating ip之后，无法从外网ping/ssh这个floating IP，需要增加ping和ssh规则，如下图7-4所示：



图：7-4 增加rules

参考：

<https://docs.openstack.org/image-guide/obtain-images.html>

<https://docs.openstack.org/image-guide/centos-image.htm>

## 定制启动镜像

1. 从Centos/Ubuntu官网下载image文件，并通过Web导入
2. 通过Web使用该image启动一个VM（创建新Volume方式）
3. 通过Web，做相应的定制（如：安装相关的包）
4. 删除这个VM（并不会删除Volume）
5. 到volume选项中，使用刚刚启动VM所创建的Volume 点击“上传” 到 image仓库

注：

1. 创建VM时，选择flavor时不要选择太大的Disk，否则创建的volume上传到image后，size太大
2. 启动一个实例时，选择的Disk size必须要大于 image 的size（不是真实的size，是image的虚拟size)

|  |
| --- |
| [root@21-server images(keystone\_admin)]# qemu-img info b4649994-fb19-4a82-a1ce-2261c07ecb94  image: b4649994-fb19-4a82-a1ce-2261c07ecb94  file format: qcow2  virtual size: 8.0G (8589934592 bytes)  disk size: 815M  cluster\_size: 65536  ...... |

1. /etc/ssh/sshd 文件在VM启动时会自动被覆盖，所以，不管怎么在原始的image里面修改，都无法通过密码方式进行登陆，解决方法： a) 在Web上导入ssh-key； b）在创建VM时使用，插入如下代码到configuration选项中：

|  |
| --- |
| #!/bin/sh  passwd centos<<EOF  rootroot  rootroot  EOF  sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g'  /etc/ssh/sshd\_config  service sshd restart |

参考：

<https://docs.openstack.org/image-guide/obtain-images.html>

<https://docs.openstack.org/image-guide/centos-image.htm>

# 系统配置

## MariaDB配置

* 问题描述：

mariadb服务默认可“open file”的最大值为1024，而Openstack的大量服务使用mariadb进行存储，使得这个默认的设置值远远不能满足需求，从而导致各种错误，mariadb的log清晰的反应出这个错误：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# vim /var/log/mariadb/mariadb.log  ...  17512 2018-03-06 17:26:21 139816720410816 [ERROR] Error in accept: Bad file descriptor  17513 2018-03-06 17:26:21 139816720410816 [ERROR] Error in accept: Bad file descriptor  ... |

* 解决方法:

在控制节点上，修改/lib/systemd/system/ mariadb.service文件并重启mariadb服务：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# vim /lib/systemd/system/ mariadb.service  ......  [Service]  LimitNOFILE = infinity 🡸  LimitMEMLOCK = infinity  ......  [root@21-server ~(keystone\_admin)]# systemctl daemon-reload && systemctl restart mariadb.service  [root@21-server ~(keystone\_admin)]# cat /proc/$(pgrep mysqld$)/limits | grep files  Max open files 65536 65536 files  [root@21-server ~(keystone\_admin)]# |

参考：

<https://stackoverflow.com/questions/22495124/cannot-set-limit-of-mysql-open-files-limit-from-1024-to-65535>

## Horizon配置

* 问题描述：

当多用户同时登陆Dashboard，可能出现 “Too many open files“ 错误，导致页面无法正常加载。

* 原因调查：

同mariadb配置一样，修改service文件即可解决该问题，但是，在系统中却找不到 horizon.service文件。

检查系统进程(ps faux)，发现keystone进程是作为httpd的子进程在运行。

Openstack官方的文档中，描述了当前keystone服务被嵌入到httpd服务中：

|  |
| --- |
| This guide uses the Apache HTTP server with mod\_wsgi to serve Identity service requests on ports 5000 and 35357. By default, the keystone service still listens on these ports. Therefore, this guide manually disables the keystone service. |

注：引用自<https://docs.openstack.org/keystone/pike/install/keystone-install-rdo.html>

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# netstat -ntlp | grep 5000  tcp6 0 0 :::5000 :::\* LISTEN 62409/httpd  [root@21-server ~(keystone\_admin)]# netstat -ntlp | grep 35357  tcp6 0 0 :::35357 :::\* LISTEN 62409/httpd  [root@21-server ~(keystone\_admin)]# |

因此，为了解决该问题，可以通过修改 httpd.service 文件。

* 解决方法：

在控制节点上，修改/lib/systemd/system/ httpd.service文件并重启httpd服务：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# vim /usr/lib/systemd/system/httpd.service  ......  [Service]  LimitNOFILE = 65536  Type=notify  EnvironmentFile=/etc/sysconfig/httpd  ......  [root@21-server ~(keystone\_admin)]# systemctl daemon-reload && systemctl restart httpd.service  [root@21-server ~(keystone\_admin)]# cat /proc/$(pgrep httpd$ | head -1)/limits | grep files  Max open files 65536 65536 files  [root@21-server ~(keystone\_admin)]# |

## Nova配置

* 1问题描述：

当启动一个VM时，可能会出现如下错误：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# vim /var/log/nova/nova-compute.log  ....  2018-03-12 20:53:22.504 11084 ERROR nova.compute.manager [req-9fc22aeb-3673-448e-a901-77cce33f40e6 2aea832a4f7340e5a3829e25720db794 b7c054c329f24f5783296bb601847781 - default default] [instance: 66b5c215-3594-4828-840f-22ebed3eeaa8] Instance failed block device setup: VolumeNotCreated: Volume 5c44ece4-da3b-4449-933a-f6bbbddbeff9 did not finish being created even after we waited 187 seconds or 61 attempts. And its status is downloading.  2018-03-12 20:53:22.504 11084 ERROR nova.compute.manager [instance: 66b5c215-3594-4828-840f-22ebed3eeaa8] Traceback (most recent call last):  2018-03-12 20:53:22.504 11084 ERROR nova.compute.manager [instance: 66b5c215-3594-4828-840f-22ebed3eeaa8] File "/usr/lib/python2.7/site-packages/nova/compute/manager.py", line 1598, in \_prep\_block\_device  2018-03-12 20:53:22.504 11084 ERROR nova.compute.manager [instance: 66b5c215-3594-4828-840f-22ebed3eeaa8] wait\_func=self.\_await\_block\_device\_map\_created)  ...... |

可能是由于网络原因，导致Volume没有在”规定”的时间内被下载完毕，从而导致启动失败。

* 1解决方法：

在所有计算节点上：修改block\_device\_allocate\_retries参数为600，从而扩大默认的timeout：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# vim /etc/nova/nova.conf  ...  730 # \* 60 (default)  731 # \* If value is 0, then one attempt is made.  732 # \* Any negative value is treated as 0.  733 # \* For any value > 0, total attempts are (value + 1)  734 # (integer value)  735 # block\_device\_allocate\_retries=60  736 block\_device\_allocate\_retries=600  ... |

参考：

- <https://bugs.launchpad.net/kolla-ansible/+bug/1701416>

- <https://www-01.ibm.com/support/docview.wss?uid=nas8N1021384>

* 2问题描述：

当发现运行的VM资源不足想要扩大时(如：CPU和内存），可以使用Openstack提供的动态扩容：

- 命令行操作：nova resize / nova resize-confirm

- Web操作： 在Web上点击对应VM选项下的”resize instance“按钮

然而，当对VM进行resize时，会出现如下错误：

|  |
| --- |
| [root@21-server nova(keystone\_admin)]# vim nova-compute.log  ...  44648 Stderr: u'Load key "/etc/nova/migration/identity": Permission denied\r\nPermission denied (publickey,gssapi-keyex,gssapi-with-mic).\r\n'  44649 2018-03-23 10:35:35.966 44360 ERROR oslo\_messaging.rpc.server Traceback (most recent call last):  ......  44689 2018-03-23 10:35:35.966 44360 ERROR oslo\_messaging.rpc.server raise error.inner\_exception  44690 2018-03-23 10:35:35.966 44360 ERROR oslo\_messaging.rpc.server ResizeError: Resize error: not able to execute ssh command: Unexpected error while running command.  44691 2018-03-23 10:35:35.966 44360 ERROR oslo\_messaging.rpc.server Command: ssh -o BatchMode=yes 10.0.2.22 mkdir -p /var/lib/nova/instances/334c51df-fd07-4c45-85f7-73011a063206  44692 2018-03-23 10:35:35.966 44360 ERROR oslo\_messaging.rpc.server Exit code: 255  44693 2018-03-23 10:35:35.966 44360 ERROR oslo\_messaging.rpc.server Stdout: u''  ...... |

* 2解决方法：

在所有计算节点上：修改allow\_resize\_to\_same\_host参数为true：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# vim /etc/nova/nova.conf  ......  177 # Allow destination machine to match source for resize. Useful when  178 # testing in single-host environments. By default it is not allowed  179 # to resize to the same host. Setting this option to true will add  180 # the same host to the destination options. Also set to true  181 # if you allow the ServerGroupAffinityFilter and need to resize.  182 # (boolean value)  183 #allow\_resize\_to\_same\_host=false  184 allow\_resize\_to\_same\_host=true  ...... |

注：此方法是规避resize在不同机器上，并不是真正的解决问题。真正的原因和解决方法看下面的问题3

* 3问题描述：

Openstack提供了migrate的功能，还可以通过--live参数来设置迁移到指定的机器上：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# openstack server migrate -h  usage: openstack server migrate [-h] [--live <hostname>] [--shared-migration | --block-migration]  [--disk-overcommit | --no-disk-overcommit] [--wait] <server>  Migrate server to different host  positional arguments:  <server> Server (name or ID)  optional arguments:  -h, --help show this help message and exit  --live <hostname> Target hostname  --shared-migration Perform a shared live migration (default)  --block-migration Perform a block live migration  --disk-overcommit Allow disk over-commit on the destination host  --no-disk-overcommit Do not over-commit disk on the destination host (default)  --wait Wait for migrate to complete  [root@21-server ~(keystone\_admin)]# |

在实际的操作过程中，遇到了各种错误，如其下的错误信息：

|  |
| --- |
| Resize error: not able to execute ssh command: Unexpected error while running command. Command: ssh -o BatchMode=yes 10.0.2.25 mkdir -p /var/lib/nova/instances/de208d6d-c11d-44c0-aa59-813b640edac7 Exit code: 255 Stdout: u'' Stderr: u'Host key verification |

* 3 原因分析

在migrate过程中出现各种错误的主要原因: Openstack在migrate时会用nova进行操作，但实际登陆使用的是nova\_migration 用户，从而引起各种各样的权限错误：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]#su - nova  Last login: Tue Apr 10 17:57:44 CST 2018 on pts/11  bash-4.2$ whoami  nova  bash-4.2$ cat ~/.ssh/config  Host \*  User nova\_migration  UserKnownHostsFile /dev/null  IdentityFile /etc/nova/migration/identity  bash-4.2$ |

同时，可以发现使用Packstack安装后，ssh的配置文件被修改了，表明：当使用nova\_migration用户登陆到本机时，会使用 /etc/nova/migration/authorized\_keys文件进行授权认证

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# cat /etc/ssh/sshd\_config | tail  # X11Forwarding no  # AllowTcpForwarding no  # PermitTTY no  # ForceCommand cvs server  Match User nova\_migration  AllowTcpForwarding no  AuthorizedKeysFile /etc/nova/migration/authorized\_keys  ForceCommand /bin/nova-migration-wrapper  PasswordAuthentication no  X11Forwarding no  [root@21-server ~(keystone\_admin)]# |

* 3 解决方法

为了解决这个问题，需要对多处进行修改和配置，在所有计算节点按如下步骤进行修改：

1. 修改nova和nova\_migration用户，从/sbin/nologin修改为/binbash，使其可登陆系统：

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# grep nova /etc/passwd  nova:x:162:162:OpenStack Nova Daemons:/var/lib/nova:/bin/bash  nova\_migration:x:981:976:OpenStack Nova Migration:/var/lib/nova:/bin/bash  [root@21-server ~(keystone\_yilong)]# |

1. 修改ssh配置，注释ForceCommand 选项，使得可直接使用nova\_migration进行ssh登陆：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# cat /etc/ssh/sshd\_config | tail  # X11Forwarding no  # AllowTcpForwarding no  # PermitTTY no  # ForceCommand cvs server  Match User nova\_migration  AllowTcpForwarding no  AuthorizedKeysFile /etc/nova/migration/authorized\_keys  #ForceCommand /bin/nova-migration-wrapper  PasswordAuthentication no  X11Forwarding no  [root@21-server ~(keystone\_admin)]# |

1. 修改authorized\_keys(640)和identify(600)文件权限和拥有者如下所示：

|  |
| --- |
| [root@21-server migration]# pwd  /etc/nova/migration  [root@21-server migration]# ll  total 12  drwxr-xr-x 3 root root 84 Apr 4 22:49 .  drwxr-xr-x 3 root root 137 Apr 4 19:13 ..  -rw-r----- 1 root nova 1158 Apr 4 22:49 authorized\_keys  -rw------- 1 nova nova 1674 Mar 10 17:37 identity  -rw-r----- 1 root root 121 Nov 23 04:30 rootwrap.conf  drwxr-xr-x 2 root root 36 Mar 10 17:37 rootwrap.d  [root@21-server migration]# |

1. 修改nova账号的ssh配置， 新增StrictHostKeyChecking no选项

|  |
| --- |
| [root@21-server ~(keystone\_admin)]#su - nova  Last login: Tue Apr 10 17:57:44 CST 2018 on pts/11  bash-4.2$ whoami  nova  bash-4.2$ cat ~/.ssh/config  Host \*  User nova\_migration  UserKnownHostsFile /dev/null  StrictHostKeyChecking no  IdentityFile /etc/nova/migration/identity  bash-4.2$ |

1. 增加nova\_migration用户到nova组，增加nova用户到nova\_migration组

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# usermod -a nova\_migration -G nova  [root@21-server ~(keystone\_admin)]# usermod -a nova -G nova\_migration |

1. 修改 /etc/nova/migration/authorized\_key，使得计算节点之间可以无密码访问，测试方法：

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# ssh -i /etc/nova/migration/identity nova\_migration@10.0.2.22 "hostname"  22-server.localdomain  [root@21-server ~(keystone\_yilong)]# |

1. 修改/var/lib/nova/instance权限为777

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# chmod -R 777 /var/lib/nova/instance |

* 3 细节分析

从VM创建到迁移，整个过程大致可描述如下：

1. 创建VM： 将会在/var/lib/nova/instance/目录下创建个实例目录，拥有者为nova用户

|  |
| --- |
| [root@21-server instances]# pwd  /var/lib/nova/instances  [root@21-server instances]# ll  total 8  drwxrwxrwx 20 nova nova 4096 Apr 11 10:20 .  drwxr-xr-x 11 nova nova 176 Apr 8 10:16 ..  drwxr-xr-x 2 nova nova 25 Apr 9 15:51 183f6ad4-d9c0-4e13-abe3-e2ce1ddd838a  ...... |

1. 将VM从21-server迁移到22-server过程中
2. 在21-server上，将实例目录从$id变为 ${id}-resize
3. 在22-server上，使用nova\_migration用户创建实例目录${id}，拥有者为nova\_migration
4. 在21-server上，使用nova用户将实例目录${id}-resize删除

|  |
| --- |
| [root@22-server instances]# ll  total 4  drwxrwxrwx 10 nova nova 317 Apr 11 10:20 .  drwxr-xr-x 11 nova nova 176 Apr 8 10:16 ..  drwxr-xr-x 2 nova\_migration nova\_migration 25 Apr 9 10:49 183f6ad4-d9c0-4e13-abe3-e2ce1ddd838a  ...... |

1. 再将VM从22-server迁移到21-server上

如上述2)里面描述的三个步骤，前两个步骤都可以成功，第三个步骤将会失败，因为nova账户无法删除nova\_migraion拥有者的目录。为了解决该问题，需要在第二次migrate时提前修改目录权限。

另，尽管第三个步骤会失败，但实际上VM已经迁移到21-server上了，只是这个时候VM的状态是"ERROR"状态，为了解决这个，可以执行命令`nova stop $id && nova start $id`来使VM处于正常的ACTIVE/running状态。

参考：

- <https://docs.openstack.org/nova/pike/admin/migration.html>

- [https://docs.openstack.org/nova/pike/admin/ssh-configuration.html#cli-os-migrate-cfg-ssh](https://docs.openstack.org/nova/pike/admin/ssh-configuration.html" \l "cli-os-migrate-cfg-ssh)

* 4问题描述：

当使用Openstack的live迁移时，会出现如下错误：

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# vim /var/log/nova/nova-compute.log  ....  2018-04-04 23:36:43.503 107129 ERROR nova.virt.libvirt.driver [req-77de991a-c367-4b67-aa7f-06173eccc137 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 7bd41573-3f07-4db3-a56d-d44bb50bff30] Live Migration failure: operation failed: Failed to connect to remote libvirt URI qemu+ssh://nova\_migration@24-server.localdomain/system?keyfile=/etc/nova/migration/identity: authentication unavailable: no polkit agent available to authenticate action 'org.libvirt.unix.manage': libvirtError: operation failed: Failed to connect to remote libvirt URI qemu+ssh://nova\_migration@24-server.localdomain/system?keyfile=/etc/nova/migration/identity: authentication unavailable: no polkit agent available to authenticate action 'org.libvirt.unix.manage'  ...... |

* 4 问题重现

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# virsh -c qemu+ssh://nova\_migration@22-server.localdomain/system?keyfile=/etc/nova/migration/identity list --all  error: failed to connect to the hypervisor  error: authentication unavailable: no polkit agent available to authenticate action 'org.libvirt.unix.manage'  [root@21-server ~(keystone\_yilong)]#]# |

* 4 解决方法

通过增加 nova\_migration 用户到 libvirt 组，可以解决：

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# # usermod -a nova\_migration -G libvirt  [root@21-server ~(keystone\_yilong)]# # virsh -c qemu+ssh://nova\_migration@22-server.localdomain/system?keyfile=/etc/nova/migration/identity list --all  Id Name State  ----------------------------------------------------  2 instance-000000e1 running  6 instance-000000f2 running  7 instance-000000f3 running  [root@21-server ~(keystone\_yilong)]# |

参考：

<https://www.devsblock.com/?p=205>

解决上述问题后，运行没有报任何错误：

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# openstack server migrate --live 22-server.localdomain --wait 7bd41573-3f07-4db3-a56d-d44bb50bff30  Complete  [root@21-server ~(keystone\_yilong)]# |

然而，使用nova show命令查看，发现VM没有迁移成功，检查log文件，可以发现如下新的错误信息：

|  |
| --- |
| [root@21-server ~(keystone\_yilong)]# vim /var/log/nova/nova-compute.log  ......  2018-04-08 14:12:10.553 125936 ERROR nova.virt.libvirt.driver [req-2b145396-8e67-4447-b69f-cebb9d51f3dd 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 7bd41573-3f07-4db3-a56d-d44bb50bff30] Live Migration failure: unsupported configuration: Unable to find security driver for model selinux: libvirtError: unsupported configuration: Unable to find security driver for model selinux  2018-04-08 14:12:10.554 125936 DEBUG nova.virt.libvirt.driver [req-2b145396-8e67-4447-b69f-cebb9d51f3dd 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 7bd41573-3f07-4db3-a56d-d44bb50bff30] Migration operation thread notification thread\_finished /usr/lib/python2.7/site-packages/nova/virt/libvirt/driver.py:6729  2018-04-08 14:12:10.819 125936 DEBUG nova.virt.libvirt.migration [req-2b145396-8e67-4447-b69f-cebb9d51f3dd 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 7bd41573-3f07-4db3-a56d-d44bb50bff30] VM running on src, migration failed find\_job\_type /usr/lib/python2.7/site-packages/nova/virt/libvirt/migration.py:227  2018-04-08 14:12:10.820 125936 DEBUG nova.virt.libvirt.driver [req-2b145396-8e67-4447-b69f-cebb9d51f3dd 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 7bd41573-3f07-4db3-a56d-d44bb50bff30] Fixed incorrect job type to be 4 \_live\_migration\_monitor /usr/lib/python2.7/site-packages/nova/virt/libvirt/driver.py:6546  2018-04-08 14:12:10.820 125936 ERROR nova.virt.libvirt.driver [req-2b145396-8e67-4447-b69f-cebb9d51f3dd 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 7bd41573-3f07-4db3-a56d-d44bb50bff30] Migration operation has aborted |

修改/etc/libvirt/qemu.conf 配置文件的 security\_driver 的选项为 none，可解决上述错误：

|  |
| --- |
| [root@21-server ~]# vim /etc/libvirt/qemu.conf  ......  327 # security\_driver = [ "selinux", "apparmor" ]  328 #  329 # Notes: The DAC security driver is always enabled; as a result, the  330 # value of security\_driver cannot contain "dac". The value "none" is  331 # a special value; security\_driver can be set to that value in  332 # isolation, but it cannot appear in a list of drivers.  333 #  334 #security\_driver = "selinux"  335 security\_driver = "none" |

再次尝试live迁移，出现新的错误，如下所示：

|  |
| --- |
| 2018-04-12 21:04:53.369 22632 ERROR nova.virt.libvirt.driver [req-41640a22-af3f-418d-b280-3822d09d290a 2aea832a4f7340e5a3829e25720db794 da1648efed414cf3a5c5e6db9523fb01 - default default] [instance: 645006da-1147-4787-a56a-b6bfe155d846] Live Migration failure: Requested operation is not valid: domain 'instance-0000017b' is already active: libvirtError: Requested operation is not valid: domain 'instance-0000017b' is already active |

Openstack的官方文档提到live-migration支持三种类型：

|  |
| --- |
| Live migration  The instance keeps running throughout the migration. This is useful when it is not possible or desirable to stop the application running on the instance.  Live migrations can be classified further by the way they treat instance storage:  Shared storage-based live migration. The instance has ephemeral disks that are located on storage shared between the source and destination hosts.  Block live migration, or simply block migration. The instance has ephemeral disks that are not shared between the source and destination hosts. Block migration is incompatible with read-only devices such as CD-ROMs and Configuration Drive (config\_drive).  Volume-backed live migration. Instances use volumes rather than ephemeral disks.  Block live migration requires copying disks from the source to the destination host. It takes more time and puts more load on the network. Shared-storage and volume-backed live migration does not copy disks. |

目前搭建的环境采用的是 Volume-backed 模式，但是cinder后端采用的是lvm，而非NFS/CEPH等共享存储方式，所以无法去做live迁移。

另外，尝试使用NFS作为存储后端，直接使用virsh命令进行迁移，可以完成live-migration

参考：

<https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/virtualization/sect-virtualization-kvm_live_migration-live_kvm_migration_with_virsh>

<https://docs.openstack.org/nova/pike/admin/configuring-migrations.html>

<https://docs.openstack.org/nova/pike/admin/live-migration-usage.html>

## 存储后端配置

### Nova配置

nova默认使用/var/lib/nova/instances作为存储后端，来存储Instance OS相关的数据文件。考虑使用LVM作为nova的存储后端：

* 在计算节点，创建image-volume逻辑卷

|  |
| --- |
| # pvcreate /dev/sdd  # vgcreate images-volumes /dev/sdd |

* 在计算节点，修改nova配置文件

|  |
| --- |
| # vim /etc/nova/nova.conf  … …  [libvirt]  images\_type=lvm  images\_volume\_group=image-volume  … … |

* 重启nova相关服务

|  |
| --- |
| # systemctl restart openstack-nova-api openstack-nova-compute openstack-nova-scheduler openstack-nova-conductor |

### Glance配置（可选）

Glance服务默认使用/var/lib/glance/images/来作为存储后端，考虑使用cinder作为glance的存储后端：

* 在控制节点上，修改/etc/glance/glance-api.conf文件并重启glance相关服务

|  |
| --- |
| [root@21-server glance(keystone\_admin)]# vim /etc/glance/glance-api.conf  ... ...  [glance\_store]  ... ...  stores=cinder  default\_store=cinder  cinder\_store\_auth\_address = http://10.0.2.21:8776/v2  cinder\_store\_user\_name = glance  cinder\_store\_password = rootroot  ....  [root@21-server glance(keystone\_admin)]# systemctl restart openstack-glance-api.service openstack-glance-registry.service |

注：

1）关于auth\_address的获取： Web UI -> ProjectAPI Access

2）关于username 的获取： Web -> Identity -> Users 或者在部署Packstack 的所用的answer 文件

3）关于password的获取：在部署OpenStack Glance服务时所创建的 password

* 确认修改生效

上传一个Image后，可以看到/var/lib/glance/images/并没有相关的文件，同时在Web 的Volume选项卡中上可以看到一个与Image ID相关的Volume存在。另外，使用命令行也可以看到相关的信息

|  |
| --- |
| [root@21-server glance(keystone\_admin)]# openstack image list  +--------------------------------------+------+--------+  | ID | Name | Status |  +--------------------------------------+------+--------+  | d16ee14d-c02e-4f00-9c3c-4c03eddb00db | 1704 | active |  +--------------------------------------+------+--------+  [root@21-server glance(keystone\_admin)]# openstack volume list  +--------------------------------------+--------------------------------------------+-----------+------+-------------+  | ID | Name | Status | Size | Attached to |  +--------------------------------------+--------------------------------------------+-----------+------+-------------+  | 15224a74-6d91-4e9f-b717-c7e467af7a82 | image-d16ee14d-c02e-4f00-9c3c-4c03eddb00db | available | 1 | |  ... ...  [root@21-server glance(keystone\_admin)]# |

## GPU Passthrough配置

为了让Instance能使用Host上的GPU，需要对Host上的GPU资源进行Passthrough。需要注意的是，Host上不需要安装GPU相关的驱动和软件，否则可能会使instance无法正常使用Passthrough的GPU。

### 系统配置

* Host上GPU设备确认

|  |
| --- |
| [root@22-server ~]# lspci -nn -v -D -k | grep -i NVIDIA  0000:82:00.0 VGA compatible controller [0300]: NVIDIA Corporation GP104 [GeForce GTX 1080] [10de:1b80] (rev a1) (prog-if 00 [VGA controller])  0000:82:00.1 Audio device [0403]: NVIDIA Corporation GP104 High Definition Audio Controller [10de:10f0] (rev a1)  [root@22-server ~]# |

可以得知：

1. GPU 型号: GeForce GTX 1080
2. vendor\_id: 10de
3. product\_id: 1b80
4. 地址总线：0000:82:00.0

* 打开CPU虚拟化特性

检查当前系统的CPU VT-D特性

|  |
| --- |
| [root@22-server ~]# cat /proc/cpuinfo | grep vmx | wc -l  48  [root@22-server ~]# |

如果输出结果为非0，表示CPU已经启用VT-D特性，否则，需要通过BIOS打开CPU的VT-D特性。

* 配置kernel启动参数

修改grub.cfg文件，追加相关启动参数

|  |
| --- |
| [root@22-server ~]# vim /boot/efi/EFI/centos/grub.cfg  ......  87 ### BEGIN /etc/grub.d/10\_linux ###  88 menuentry 'CentOS Linux (3.10.0-693.17.1.el7.x86\_64) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry\_id\_option 'gnulinux-3.10.0-693.17.1.el7.x86\_64-advanced-db4364 d2-6b6d-45c7-9014-8c41545e8999' {  89 set gfxpayload=text  90 insmod gzio  91 insmod part\_gpt  92 insmod xfs  93 set root='hd0,gpt2'  ......  99 linuxefi /vmlinuz-3.10.0-693.17.1.el7.x86\_64 root=/dev/mapper/centos00-root ro crashkernel=auto rd.lvm.lv=centos00/root rd.lvm.lv=centos00/swap rhgb quiet intel\_iommu=on nouveau.modeset=0 rd.driver.blacklist=nouveau video=vesa:off  ...... |

重启后，可使用如下命令查看是否生效：

|  |
| --- |
| [root@22-server ~]# cat /proc/cmdline  BOOT\_IMAGE=/vmlinuz-3.10.0-693.17.1.el7.x86\_64 root=/dev/mapper/centos00-root ro crashkernel=auto rd.lvm.lv=centos00/root rd.lvm.lv=centos00/swap rhgb quiet intel\_iommu=on nouveau.modeset=0 rd.driver.blacklist=nouveau video=vesa:off  [root@22-server ~]# |

如果通过 lspci 命令知道某个设备的地址总线后，可以通过以下方式来查看具体信息（如：0000:82:00.0）

|  |
| --- |
| [root@22-server 0000:82:00.0(keystone\_admin)]# pwd  /sys/bus/pci/devices/0000:82:00.0  [root@22-server 0000:82:00.0(keystone\_admin)]# cat vendor  0x10de  [root@22-server 0000:82:00.0(keystone\_admin)]# cat device  0x1b80  [root@22-server 0000:82:00.0(keystone\_admin)]# |

同时，通过以下信息可以知道GPU GTX1080 和GPU Audio控制器属于同一个组：

|  |
| --- |
| [root@22-server devices(keystone\_admin)]# pwd  /sys/kernel/iommu\_groups/39/devices  [root@22-server devices(keystone\_admin)]# ls  0000:82:00.0 0000:82:00.1  [root@22-server devices(keystone\_admin)]# |

注：无法passthrough同组GPU中的不同GPU到不同的VM中，即只能映射同组GPU到一个VM中。

* vfio-pci模块配置

1. 增加GPU设备相关信息到/etc/modprobe.d/vfio.conf文件

|  |
| --- |
| [root@22-server ~]# cat /etc/modprobe.d/vfio.conf  options vfio-pci ids=10de:1b80,10de:10f0  [root@22-server ~]# |

注： 上述数字为GPU相关的vender\_ID和Product\_ID

1. 增加vfio-pci到/etc/modules-load.d/modules.conf文件

|  |
| --- |
| [root@22-server ~]# cat /etc/modules-load.d/modules.conf  vfio-pci  [root@22-server ~]# |

1. 重启系统，确认GPU的driver为vfio-pci

|  |
| --- |
| [root@22-server ~]# lspci -nnv –D  … …  0000:82:00.0 VGA compatible controller [0300]: NVIDIA Corporation GP104 [GeForce GTX 1080] [10de:1b80] (rev a1) (prog-if 00 [VGA controller])  Subsystem: Gigabyte Technology Co., Ltd Device [1458:3702]  Capabilities: [900] #19  Kernel driver in use: vfio-pci  Kernel modules: nouveau  0000:82:00.1 Audio device [0403]: NVIDIA Corporation GP104 High Definition Audio Controller [10de:10f0] (rev a1)  Capabilities: [100] Advanced Error Reporting  Kernel driver in use: vfio-pci  Kernel modules: snd\_hda\_intel  ......  [root@22-server ~] |

* KVM配置

由于Nvidia对GPU做了限制：当GPU driver检测到当前环境是虚拟机环境，其driver将不能被正确加载。为了解决这个，需要“欺骗” GPU driver当前环境是物理机环境，非虚拟机环境。

默认情况下，可能出现如下错误：

|  |
| --- |
| [root@localhost ~ ]# nvidia-smi  Unable to determine the device handle for GPU 0000:00:08.0: Unknown Error  [root@localhost ~ ]# |

为了解决这个问题，可以添加 <hidden state='on'/> 选项到KVM的xml文件，如下：

|  |
| --- |
| <features>  <kvm>  <hidden state='on'/>  </kvm>  </features> |

由于Openstack启动VM时会自动”重写“KVM的xml文件，并不是直接调用“virsh start”，所以这种手工修改方式不仅仅是不方便，而是完全不可行。调查后，发现可以通过“重写”启动VM的qemu-kvm文件来解决:

1. 重命名源文件/usr/libexec/qemu-kvm

|  |
| --- |
| [root@21-server ~]# mv /usr/libexec/qemu-kvm /usr/libexec/qemu-kvm.orig |

1. 生成新的/usr/libexec/qemu-kvm文件

|  |
| --- |
| [root@21-server ~]# vim /usr/libexec/qemu-kvm  #!/usr/bin/python  import os  import sys  new\_args = []  # only change the "-cpu" options (inject kvm=off and hv\_vendor\_id=MyFake\_KVM)  for i in range(len(sys.argv)):  if i<=1:  new\_args.append(sys.argv[i])  continue  if sys.argv[i-1] != "-cpu":  new\_args.append(sys.argv[i])  continue  subargs = sys.argv[i].split(",")  subargs.insert(1,"kvm=off")  new\_arg = ",".join(subargs)  new\_args.append(new\_arg)  os.execv('/usr/libexec/qemu-kvm.orig', new\_args) |

参考：

- <https://gist.github.com/claudiok/890ab6dfe76fa45b30081e58038a9215>

- [https://wiki.archlinux.org/index.php/PCI\_passthrough\_via\_OVMF#Enabling\_IOMMU](https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF" \l "Enabling_IOMMU)

- [https://libvirt.org/formatdomain.html#elementsNICSHostdev](https://libvirt.org/formatdomain.html" \l "elementsNICSHostdev)

### nova配置

1. 在控制节点，增加PciPassthroughFilter到配置文件/etc/nova/nova.conf的 enabled\_filters选项

|  |
| --- |
| [root@21-server ~]# vim /etc/nova/nova.conf  … …  [filter\_scheduler]  enabled\_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,DiskFilter,ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter,CoreFilter,PciPassthroughFilter  ...... |

注：scheduler\_default\_filters和scheduler\_available\_filterseh选项已经被废弃。

1. 在控制节点，修改文件/etc/nova/nova.conf增加alias行

* 配置单个GPU

|  |
| --- |
| [root@21-server nova(keystone\_admin)]# vim /etc/nova/nova.conf  ......  [pci]  alias = { "vendor\_id": "10de", "product\_id": "1b80", "device\_type": "type-PCI", "name": "a1" }  ...... |

* 配置多个GPU

|  |
| --- |
| [root@21-server nova(keystone\_admin)]# vim /etc/nova/nova.conf  ......  [pci]  alias = { "vendor\_id": "10de", "product\_id": "1b80", "device\_type": "type-PCI", "name": "a1" }  alias = {"vendor\_id": "10de", "product\_id": "10f0", "device\_type": "type-PCI", "name": "a2"}  ...... |

1. 在计算节点，修改文件/etc/nova/nova.conf增加alias和passthrough\_whitelist

|  |
| --- |
| [root@22-server ~(keystone\_admin)]# vim /etc/nova/nova.conf  ......  [pci]  alias = { "vendor\_id": "10de", "product\_id": "1b80", "device\_type": "type-PCI", "name": "a1" }  alias = {"vendor\_id": "10de", "product\_id": "10f0", "device\_type": "type-PCI", "name": "a2"}  ......  passthrough\_whitelist = [{"product\_id":"1b80", "vendor\_id":"10de"}, {"product\_id":"10f0", "vendor\_id":"10de"}]  ...... |

参考：

<https://docs.openstack.org/nova/pike/admin/pci-passthrough.html>

### flavor配置

Openstack通过flavor来定义资源（包括：CPU/Mem/Disk等），创建VM时必须指定某个flavor。为了让VM能够使用到GPU资源，需要给flavor加上特有的属性。

* 配置一个GPU资源

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# openstack flavor create --ram 4096 --disk 20 --vcpus 4 gpu\_1  [root@21-server ~(keystone\_admin)]# openstack flavor set --property "pci\_passthrough:alias"="a1:1" gpu\_1 🡸 |

注："a1:1": 指的是在控制节点定义的a1 alias ,以及一个GPU设备

* 配置多个GPU资源

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# openstack flavor create --ram 4096 --disk 20 --vcpus 4 gpu\_2  [root@21-server ~(keystone\_admin)]# openstack flavor set --property "pci\_passthrough:alias"="a1:1,a2:1" gpu\_2 |

注：这里gpu\_2 flavor 相当于定义使用两块GPU

* 查看配置的flavor

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# openstack flavor show gpu\_1 --format shell  +----------------------------+--------------------------------------+  | Field | Value |  +----------------------------+--------------------------------------+  | OS-FLV-DISABLED:disabled | False |  | OS-FLV-EXT-DATA:ephemeral | 0 |  | access\_project\_ids | None |  | disk | 20 |  | id | 631ba2e4-29b9-457e-ad37-215929c6255d |  | name | gpu\_1 |  | os-flavor-access:is\_public | True |  | properties | pci\_passthrough:alias='a1:1' |  | ram | 4096 |  | rxtx\_factor | 1.0 |  | swap | |  | vcpus | 4 |  +----------------------------+--------------------------------------+  [root@21-server ~(keystone\_admin)]# |

参考：

<https://docs.openstack.org/nova/pike/admin/pci-passthrough.html>

### CPU Topologies配置

* 问题描述：

创建好的虚拟机，CPU的Sockets和Core数一样

|  |
| --- |
| [renyl@localhost repo]$ ssh centos@10.0.2.75  Last login: Mon Apr 2 02:10:12 2018 from 192.168.50.155  [centos@yilong-app ~]$ lscpu  Architecture: x86\_64  CPU op-mode(s): 32-bit, 64-bit  Byte Order: Little Endian  CPU(s): 8  On-line CPU(s) list: 0-7  Thread(s) per core: 1  Core(s) per socket: 1  Socket(s): 8  NUMA node(s): 1  Vendor ID: GenuineIntel  CPU family: 6  Model: 94  Model name: Intel Core Processor (Skylake)  ......  [centos@yilong2-app-1 ~]$ |

官方文档描述，默认情况下新建的VM，其sockets等于core数

|  |
| --- |
| By default, when instance NUMA placement is not specified, a topology of N sockets, each with one core and one thread, is used for an instance, where N corresponds to the number of instance vCPUs requested. |

* 解决方法：

设置所有的flavor使用2个NUMA，2个Socket:

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# openstack flavor set skydata-small --property hw:cpu\_sockets=2  [root@21-server ~(keystone\_admin)]# openstack flavor set skydata-small --property hw:numa\_nodes=2 |

参考：

<https://docs.openstack.org/nova/pike/admin/cpu-topologies.html>

## 稳定性/性能配置

### 关闭auditd服务

Libvirtd服务默认使用auditd来记录信息，会产生数十GB的log文件，关闭auditd服务减少log文件：

|  |
| --- |
| [root@21-server ~]# systemctl kill auditd && systemctl disable auditd |

注：也可修改libvirtd服务的配置文件/etc/libvirt/libvirtd.conf的audit\_level选项为0

### 开启tuned服务

|  |
| --- |
| [root@21-server ~]# systemctl start tuned && systemctl enable auditd  [root@21-server ~]# tuned-adm list  Available profiles:  - balanced - General non-specialized tuned profile  - desktop - Optimize for the desktop use-case  - latency-performance - Optimize for deterministic performance at the cost of increased power consumption  ......  - virtual-guest - Optimize for running inside a virtual guest  - virtual-host - Optimize for running KVM guests  Current active profile: virtual-host  [root@21-server ~]# |

### 开启ksm服务

|  |
| --- |
| [root@21-server ~]# systemctl start ksm && systemctl enable ksm  [root@21-server ~]# systemctl start ksmtuned && systemctl enable ksmtuned |

# 新增计算节点

当前Openstack集群已经包含21-server和22-server两台机器，将新增24-server到现有集群，增加后的Openstack集群架构，如下图8-1所示：

计算节点

24-server

计算节点

22-server

21-server

网络节点

计算节点

存储节点

控制节点

图8-1: Openstack集群架构

## 环境准备

参考第3章节

## 系统安装

* 在控制节点，修改answer-file.txt文件的EXCLUDE\_SERVERS和CONFIG\_COMPUTE\_HOSTS选项

|  |
| --- |
| [root@21-server ~]# vim yilong\_answer-file.txt  .......  81 # Comma-separated list of servers to be excluded from the  82 # installation. This is helpful if you are running Packstack a second  83 # time with the same answer file and do not want Packstack to  84 # overwrite these server's configurations. Leave empty if you do not  85 # need to exclude any servers.  86 EXCLUDE\_SERVERS=10.0.2.21,10.0.2.22,  ......  96 # List the servers on which to install the Compute service.  97 CONFIG\_COMPUTE\_HOSTS=10.0.2.21,10.0.2.22,10.0.2.24  ...... |

注：10.0.2.21和10.0.2.22是现有Openstack集群的IP，10.0.2.24是这次要新增的计算节点IP。

* 在控制节点，执行packstack进行安装

|  |
| --- |
| [root@21-server ~]# packstack --answer-file=./yilong\_answer-file.txt  ......  \* Time synchronization installation was skipped. Please note that unsynchronized time on server instances might be problem for some OpenStack components.  … …  \* To access the OpenStack Dashboard browse to http://10.0.2.21/dashboard .  Please, find your login credentials stored in the keystonerc\_admin in your home directory.  \* The installation log file is available at: /var/tmp/packstack/20180109-010608-1aT85n/openstack-setup.log  \* The generated manifests are available at: /var/tmp/packstack/20180109-010608-1aT85n/manifests  [root@21-server ~]# |

注：整个安装比较耗时，跟网络带宽、机器性能等有关。

参考：

<https://www.rdoproject.org/install/adding-a-compute-node/>

## 系统验证

在控制节点，执行如下命令:

|  |
| --- |
| [root@21-server ~]#source ~/keystonerc\_admin  [root@21-server ~(keystone\_admin)]# nova service-list  +--------------------------------------+------------------+-----------------------+----------+---------+-------+----------------------------+-----------------+-------------+  | Id | Binary | Host | Zone | Status | State | Updated\_at | Disabled Reason | Forced down |  +--------------------------------------+------------------+-----------------------+----------+---------+-------+----------------------------+-----------------+-------------+  | 67a1d086-e68e-41a0-8328-9d9a9ba52235 | nova-conductor | 21-server.localdomain | internal | enabled | up | 2018-03-24T03:17:13.000000 | - | False |  | 7f011121-419f-423c-92c2-bd67f759d85b | nova-scheduler | 21-server.localdomain | internal | enabled | up | 2018-03-24T03:17:10.000000 | - | False |  | 3c2901dd-4f24-4de7-b432-b017977c08bf | nova-consoleauth | 21-server.localdomain | internal | enabled | up | 2018-03-24T03:17:08.000000 | - | False |  | f3cb1178-dd5e-48c5-9206-376a141b44ce | nova-compute | 21-server.localdomain | nova | enabled | up | 2018-03-24T03:17:09.000000 | - | False |  | f9e3f0f2-6241-4fe7-85c9-a408eec022a7 | nova-compute | 22-server.localdomain | nova | enabled | up | 2018-03-24T03:17:17.000000 | - | False |  | ebf2c7df-8231-44f0-8144-6f9f74876837 | nova-compute | 24-server.localdomain | nova | enabled | up | 2018-03-24T03:17:15.000000 | - | False |  +--------------------------------------+------------------+-----------------------+----------+---------+-------+----------------------------+-----------------+-------------  [root@21-server ~(keystone\_admin)]# |

可以看到，24-server这台机器已经加入到Openstack集群。

## 系统调优

参考第7章节

# 新增存储节点

当前Openstack集群已经包含21-server/22-server/24-server三台机器，22-server和24-server作为计算节点。为了利用22-server和24-server上的磁盘，将在其上部署cinder服务，使其也作为存储节点，如下图9-1所示：

计算节点

24-server

存储节点

计算节点

22-server

存储节点

21-server

网络节点

计算节点

存储节点

控制节点

图9-1: Openstack集群架构

## 环境准备

参考第三章节

## 系统安装

以24-server为例，22-server上执行一样的操作即可

* 安装LVM包

|  |
| --- |
| [root@24-server ~]# yum install lvm2 device-mapper-persistent-data  [root@24-server ~]# systemctl enable lvm2-lvmetad.service  [root@24-server ~]# systemctl start lvm2-lvmetad.service |

* 创建逻辑卷cinder-volumes

|  |
| --- |
| [root@24-server ~]# pvcreate /dev/sd{d,e,f,g,h,i}  [root@24-server ~]# vgcreate cinder-volumes /dev/sd{d,e,f,g,h,i} |

* 安装cinder包

|  |
| --- |
| [root@24-server ~]#yum install openstack-cinder targetcli python-keystone |

* 创建cinder-volumes文件

|  |
| --- |
| [root@24-server ~]# touch /var/lib/cinder/cinder-volumes && chown cinder:cinder /var/lib/cinder/cinder-volumes |

注：必须创建这个文件，否则在启动cinder服务时会失败。这看起来像是cinder服务的一个bug。

* 配置cinder

为了简化配置，可直接从21-server拷贝：

|  |
| --- |
| [root@24-server ~]# scp [root@10.0.2.21:/etc/cinder/](mailto:root@10.0.2.23:/etc/cinder/)cinder.conf /etc/cinder/  [root@24-server ~]# vim /etc/cinder/cinder.conf  5175 [lvm]  5176 volume\_backend\_name=lvm  5177 volume\_driver = cinder.volume.drivers.lvm.LVMVolumeDriver  5178 iscsi\_ip\_address=10.0.2.21  5179 volume\_group = cinder-volumes  5180 iscsi\_protocol = iscsi  5181 iscsi\_helper = lioadm |

* 启动cinder服务

|  |
| --- |
| [root@24-server ~]# systemctl enable openstack-cinder-volume.service  [root@24-server ~]# target.service#systemctl start openstack-cinder-volume.service target.service |

## 系统验证

在控制节点上，执行如下命令：

|  |
| --- |
| [root@21-server ~(keystone\_admin)]# openstack volume service list  +------------------+---------------------------+------+---------+-------+----------------------------+  | Binary | Host | Zone | Status | State | Updated At |  +------------------+---------------------------+------+---------+-------+----------------------------+  | cinder-scheduler | 21-server.localdomain | nova | enabled | up | 2018-04-12T08:25:36.000000 |  | cinder-volume | 21-server.localdomain@lvm | nova | enabled | up | 2018-04-12T08:25:33.000000 |  | cinder-volume | 22-server.localdomain@lvm | nova | enabled | up | 2018-04-12T08:25:37.000000 |  | cinder-volume | 24-server.localdomain@lvm | nova | enabled | up | 2018-04-12T08:25:34.000000 |  +------------------+---------------------------+------+---------+-------+----------------------------+  [root@21-server ~(keystone\_admin)]# |

表明：22-server和24-server这两台机器现在已经作为存储节点了。

参考：

<https://docs.openstack.org/cinder/queens/install/cinder-storage-install-rdo.html>

# 常用命令行

* nova服务

|  |  |  |
| --- | --- | --- |
| 序号 | 命令 | 说明 |
| 1 | nova list | 查看当前有哪些VM |
| 2 | nova show $id | 查看特定VM的详细信息 |
| 3 | nova service-list | 查看nova service |
| 4 | nova hypervisor-list | 查看hypervisor list |
| 5 | nova hypervisor-stats | 查看所有hypervisor整体资源属性 |
| 6 | nova hypervisor-show $id | 查看单个hypervisor的详细属性 |
| 7 | nova hypervisor-servers $hostname | 查看特定的Host上有哪些VM |
| 8 | nova volume-attachments $instance\_id | 查看该instance\_ID下有哪些Volume |
| 9 | openstack flavor create --ram 4096 --disk 20 --vcpus 4 skydata-small | 创建一个flavor |
| 10 | openstack flavor set --property "pci\_passthrough:alias"="a1:1" $flavor\_id | 对一个flavor增加属性 |
| 11 | openstack host show $hostname | 查看单个hypervisor的资源使用情况 |

* cinder服务相关

|  |  |  |
| --- | --- | --- |
| 序号 | 命令 | 说明 |
| 1 | openstack volume service list / cinder service-list | 列举当前的volume service |
| 2 | openstack volume list / cinder list | 列举当前的volume |
| 3 | openstack volume show $volume\_id / cinder show $volume\_id | 查看volume的相关信息 |
| 4 | openstack volume create –size 30 yilong\_volume\_30GB | 创建一个30GB的volume |
| 5 | cinder force-delete $volume\_id | 强制删除volume |

* glance服务

|  |  |  |
| --- | --- | --- |
| 序号 | 命令 | 说明 |
| 1 | openstack image list | 查看当前有哪些image |
| 2 | openstack image show $image\_id | 查看某个image 的详细信息 |
| 3 | openstack image create --public --unprotected --disk-format qcow2 --file CentOS-7-x86\_64-GenericCloud-1708.qcow2 centos\_image | 上传一个新的image  名称为centos\_image |
| 4 | glance image-download --file ./local\_image $image\_id | 下载image到本地以文件的形式保存 |

# FAQ

* 1Q：GPU可以被多个Instance用吗？GPU目前可以支持虚拟化吗？
* 1A: GPU资源仅仅只能被一个Instance使用，不能共享。

如果其它Instance想用GPU资源，之前的instance必须释放GPU资源（即删除Instance）

* 2Q：新创建VM，会在Web UI上出现两个内部IP地址，怎么办？
* 2A：这可能是由于iptables导致的，清除所有iptables规则（iptables -F)
* 3Q：使用从Ubuntu官网下载的image，不能成功创建VM，会报如下错误：

|  |
| --- |
| ERROR (BadRequest): Invalid image metadata. Error: Architecture name '' is not valid (HTTP 400) (Request-ID: req-916e957e-04f3-411f-b698-f760b5f94d0d) |

* 3A：添加architecture=x86\_64即可解决

|  |
| --- |
| root@21-server tools(keystone\_admin)]# openstack image set Ubuntu16.04 --property architecture=x86\_64 |

* 4Q: 通过Dashboard来看CPU和MEM的资源， overcommit功能似乎没有生效，为什么？
* 4A: NOVA API只显示物理机器资源，overcommit相关的配置，是被nova在调度instance时采用<https://docs.openstack.org/newton/config-reference/compute/schedulers.html>

|  |
| --- |
| The Compute API always returns the actual number of CPU cores available on a compute node regardless of the value of the cpu\_allocation\_ratio configuration key. As a result changes to the cpu\_allocation\_ratio are not reflected via the command line clients or the dashboard. Changes to this configuration key are only taken into account internally in the scheduler. |

* 5Q: 在有48CPU机器上，设置cpu\_allocation\_ratio=16.0, 创建一个100 VCPU的instance会失败，为什么？
* 5A：单个instance的CPU数不能超过物理CPU数，所以创建一个100 VCPU的instance会失败，

但可以成功创建16个VCPU=48的instance

<https://docs.openstack.org/arch-design/design-compute/design-compute-overcommit.html>

* 6Q：创建 instances后， 在使用console时 VNC不能正常显示界面， 而出现错误code: 1006
* 6A：修改文件/etc/nova/nova.conf文件vncserver\_proxyclient\_address选项为IP地址即可解决
* 7Q：创建VM，报如下错误，什么原因？

|  |
| --- |
| Couldn't find iscsi sessions because iscsiadm err: iscsiadm: No active sessions. |

* 7A：使用`netstat -nltp | grep 3260`检测iscsi服务是否启动正常，以及iptables -F清空规则。
* 8Q：VM里面不能正常使用Passthrough的GPU，报如下错误：

|  |
| --- |
| [root@~]# nvidia-smi  Unable to determine the device handle for GPU 0000:00:08.0: Unknown Error  [root@~]# |

* 8A：参考7.5章节，修改qemu-kvm文件。
* 9Q：如何在指定的Host上创建Instance？
* 9A：通过以下脚本可实现在指定的Host创建Instance：

<http://192.168.20.14/yilong.ren/packstack-config/blob/master/tools/create_boot_instance.sh>

* 10Q：当在Web UI点击“删除”Volume按钮后，Web UI上却一直显示deleting，无法成功删除，怎么办？
* 10A：通过以下脚本可强制删除:

<http://192.168.20.14/yilong.ren/packstack-config/blob/master/tools/change-volume-status.sh>

* 11Q：如何批量创建用户？
* 11A：通过以下脚本可批量创建用户：<http://192.168.20.14/yilong.ren/packstack-config/>
* 12Q通过Web去创建一个Instance时，却发现OS数据盘仍然使用的是cinder提供的cinder-volumes ，并没有使用nova配置好的image-volume，为什么？
* 12A：参考: [https://docs.openstack.org/nova/pike/admin/arch.html#block-storage](https://docs.openstack.org/nova/queens/admin/arch.html" \l "block-storage)

首先：cinder-volumes和image-volume两种提供存储数据的方式不同：

1. cinder-volume下的磁盘数据将会长期的存储，即使Instance被删除也不会删除这个Volume数据
2. image-volume下的磁盘数据是有生命周期的，它随着Instance的删除而删除

其次：在Web UI上“Project -> Compute -> Instances -> Launch Instance -> Flavor”中的“Ephemeral Disk”就是指image-volume下的数据磁盘空间的大小（即：临时磁盘），如下图11-1所示:

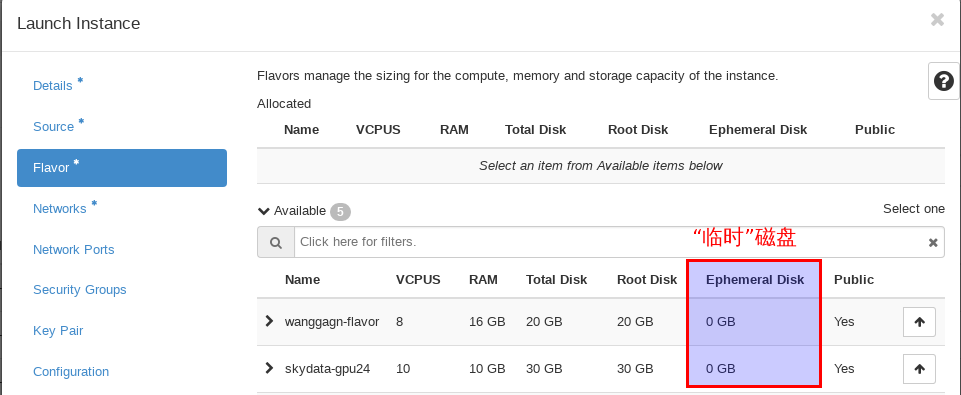


图11-1 Flavor中的“临时”磁盘

最后：为了让新创建的Instance使用image-volume来存储OS数据，在创建时选择不创建新的Volume即可，如下图11-2所示：

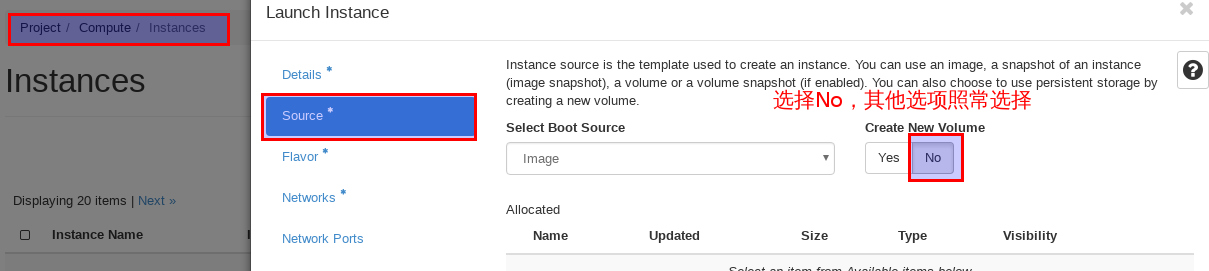


图11-2 创建Instance的选项

* 13Q：我已经在机器中成功的将使用软件配置好了，仍然会出现某个端口在OpenStack内不可访问，然而在真实的物理机器中是可以的，为什么？
* 13A：这是由于OpenStack默认的安全组规则禁用了有些端口和协议，因此需要手动更新安全组规则。图11-3表示的是如何更新安全组规则。在选择默认的安全组规则后，选择“Add Rule”，然后分别对Ingress和Egrees的数据进行更新。

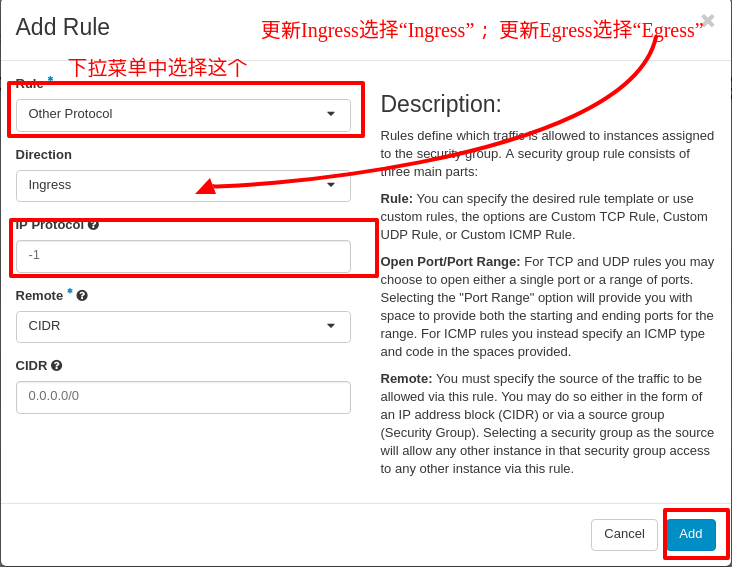


图11-3 更改安全组规则

如图11-4，对OpenStack默认的规则进行更新后的最终结果。

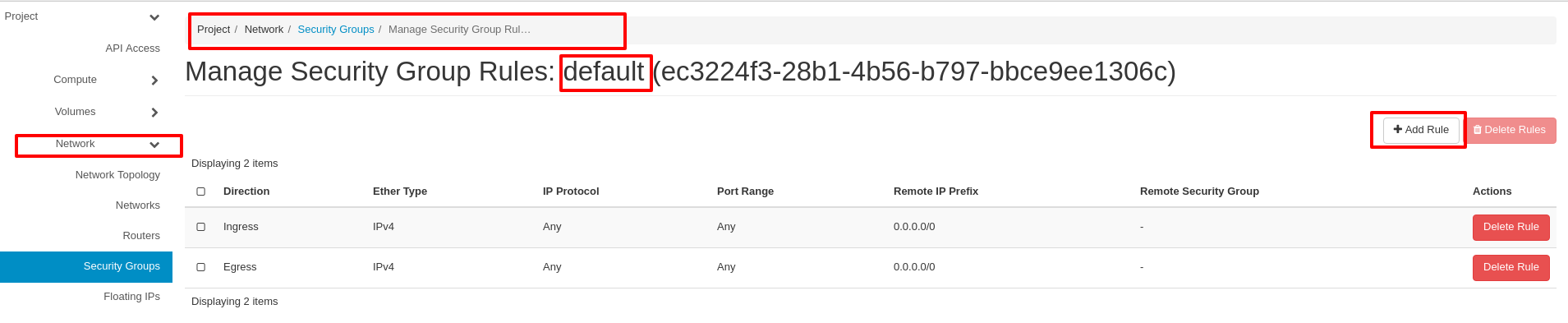


图11-4 更新后的最终结果

* 14Q：我使用keepalived想要搭建HA，但是发现配置好的VIP在不同节点上不能够相互ping通，然而在真实物理机器上测试是可以ping通的，为什么？
* 14A：由于VM中所连接的port没有允许某些IP与之相连接，导致在OpenStack环境下，keepalived的集群环境不能够正常使用，需要对port进行配置更新。有两种方式可以对port进行更新，从而得到目的。

**第一种：命令行模式。**

1）使用ifconfig 或者ip addr 命令获得集群环境的所有VM 的内部 IP 地址，如 192.168.60.31 ... ... ； 准备好 keepavlied 的VIP 地址，如 ：192.168.60.120

2）分别使用以下命令来更新port（以MASTER节点IP为例）。

|  |
| --- |
| $ neutron port-list | grep -E "192.168.60.31"  # 红色字体标注为获得的port ID  | **a45b9fd3-d367-474a-ba62-e3affd552ed8** | | da1648efed414cf3a5c5e6db9523fb01 | fa:16:3e:ad:91:9b | {"subnet\_id": "a256cbcd-db7b-485b-b8f4-d0096f225d6f", "ip\_address": "192.168.60.31"} |  # 增加参数 --allowed-address-pair； 另外，也支持CIDR形式，如--allowed-address-pair ip\_address=192.168.60.0/24  $ neutron port-update **--allowed-address-pair ip\_address=192.168.60.120** **a45b9fd3-d367-474a-ba62-e3affd552ed8** |

3）更新剩余其他VM port 的配置。

**第二种：Web UI模式。**

以VIP=192.168.60.120，MASTER节点IP=192.168.60.31，BACKUP节点IP=192.168.60.14为例。登陆到OpenStack Web界面后，分别选择“ **Project —— > Network —— > Networks —— > 当前项目的内部网络名称 —— > Ports ——> VM 的内部IP 地址**”，如图6-3-1、图6-3-2所示来修改MASTER节点的网络配置。

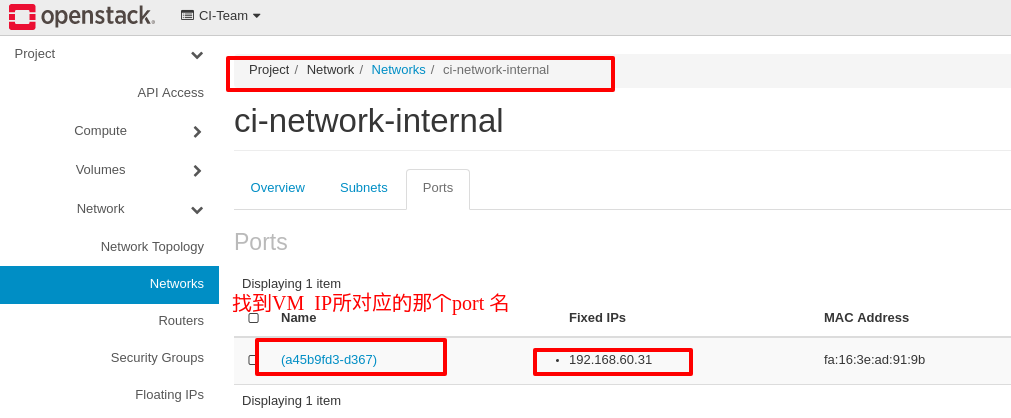


图6-3-1 选择当前VM 的port



图6-3-2 修改port

注意：以上网络配置均要在**所有的keepalived所在的机器环境**中进行修改，否则仍然会出现ping不通的错误现象。

参考：<https://blog.codecentric.de/en/2016/11/highly-available-vips-openstack-vms-vrrp/>