# OpenStack实例自动迁移的实现

——操作文档

1. OpenStack平台搭建

OpenStack运行环境基础节点：控制节点、网络节点、计算节点

实例迁移环境：计算节点1、计算节点2

网络环境：管理网络、隧道网络、外网网络

服务器：Ubuntu14.04 TLS

* 1. NTP安装

1. 在各节点安装NTP；
2. 修改各节点NTP配置文件/etc/ntp.conf，并重启；
3. 执行命令watch ntpq –p，检验各节点是否同步成功。
   1. MySql数据库及RabbitMQ安装

在各节点上安装OpenStack库，在控制节点上安装MySql和RabbitMQ，并修改相应配置

* 1. 安装keystone

1. 在MySql中建立keystone数据表，配置访问权限；
2. 在控制节点上下载并安装keystone，并修改相关配置文件；
3. 同步数据库与修改设置；
4. 配置环境变量，创建keystone服务、租户及角色，验证keystone是否安装成功。
   1. 安装glance
5. 在MySql中建立glance数据表，配置访问权限；
6. 配置环境变量，通过keystone创建glance服务、角色；
7. 在控制节点上下载并安装glance，并修改相关配置文件；
8. 同步数据库与修改设置；
9. 下载官方验证镜像到本地，并将镜像文件上传到glance中，验证glance是否安装成功。
   1. 安装nova
10. 在MySql中建立nova数据表，配置访问权限；
11. 通过keystone创建nova服务、角色；
12. 在控制节点上安装nova控制组件，并配置文件，同步数据库；
13. 在计算节点上安装nova组件，配置文件并重启；
14. 通过控制节点查看计算节点nova服务，验证nova组件是否安装成功。
    1. 安装neutron
15. 在MySql中建立neutron数据表，配置访问权限；
16. 通过keystone创建neutron服务、角色；
17. 在控制节点上安装neutron服务组件，并配置文件，同步数据库；
18. 在网络节点上安装neutron组件，配置ml2、DHCP、l3代理，配置ovs服务，添加外部网桥，连接外部网络；
19. 在计算节点上安装并配置网络通用组件，配置ml2、ovs服务；
20. 在控制节点上查看各节点网络服务状态，创建外部网络、租户网络及子网，验证neutron是否安装成功。
    1. 配置dashboard

在控制节点上安装并配置dashboard，在浏览器上访问OpenStack界面，完成OpenStack基本搭建。

1. 热迁移环境配置

热迁移根据存储区别可分为三类:基于块存储的热迁移、共享存储的热迁移及volume存储的热迁移，本文档的热迁移实际指的是共享存储下的热迁移。热迁移配置如下：

1. 修改各节点nova.conf文件；
2. 配置控制节点和计算节点的nfs和挂载；
3. 修改计算节点配置文件，重启libvrit-bin服务。

可用如下命令测试迁移原主机与目的主机是否能通过libvrit连通：

virsh -c qemu+tcp://目的主机IP/system （若节点间未连通，则会导致迁移失败）

3. Ceilometer监控工具配置搭建

（1）在控制节点、计算节点上安装和配置ceilometer：

ubuntu14.04 openstack juno ceilometer服务安装文档链接：

http://www.bkjia.com/yjs/1039753.html

官网 opensstack kilo版安装和配置ceilometer链接：

http://docs.openstack.org/developer/ceilometer/install/manual.html#installing-the-api-server

安装后CLI基本命令总结：

ceilometer meter-list -查看meters

ceilometer sample-list -m <meter\_name> -查看某个 meter 收集的具体值

可通过创建alarm实现报警，本文中是通过设定的阈值进行报警迁移的，因此并没有用到ceilometer自带alarm的报警功能，列在此处为方便使用其功能。

ceilometer alarm-list -查看alarms

ceilometer alarm-show -a <alarm\_ID> -查看具体某个alarm信息

ceilometer alarm-threshold-create --name host\_test5 \

--description 'hardware' \

--meter-name hardware.system\_stats.cpu.idle --threshold 50.0 \

--comparison-operator lt --statistic avg \

--period 600 --evaluation-periods 1 \

--alarm-action 'log://' \

--query resource\_id=192.168.0.122

•name: 告警名称

•meter-name：meter 名称

•threshold: 阈值

•comparison\_operator: 这个参数确定了怎么和阈值进行比较，有6个可选：lt, le, eq, ne, ge, gt，默认是eq

•statistic: 这个参数确定了使用什么数据去和 threshold 比较，有5种可选：max, min, avg, sum, count，默认是avg

•period: 这个参数其实有两个作用，一个是确定了获取该监控指标的监控数据的时间范围，和下面的 evaluation\_periods 配合使用，另外一个作用就是它确定了两个点之间的时间间隔，默认是60s

•evaluation\_periods: 表示连续的监控间隔数目。和 period 参数相乘，可以确定获取监控数据的时间范围，默认是1。

•alarm-action：告警产生后的反应。

•query: 该参数一般用于过滤到监控指标下的某个资源，默认是[]

（2）增加硬件监控项

安装ceilometer后只包含部分实例相关的meters，这里参考官网

http://docs.openstack.org/admin-guide/telemetry-measurements.html#snmp-based-meters

其中包含可添加的meter项。

本文介绍其中snmp based meters添加过程：

①　安装snmp，snmpd包

②　修改配置文件/etc/snmp/snmpd.conf：

#开启监听端口和对应的ip

agentAddress udp:161,udp6:[::1]:161

#开启所有SNMP访问项,注释掉前两项，这样的话，我们就可以获取更多的节点信息，否则，我们只能获取两个已注释节点所包含的信息。

#view systemonly included .1.3.6.1.2.1.1

#view systemonly included .1.3.6.1.2.1.25.1

view systemonly included .1 80

③　重启snmpd服务

service snmpd restart

④　测试

snmpwalk -v 2c -c public localhost .1.3.6.1.4.1.2021.4.3.0

⑤　修改ceilometer的/etc/ceilometer/pipeline.yaml配置：

#添加监控项

sources: # 在sources段下增加如下配置：

- name: hardware\_source

interval: 600 #监控间隔时间 单位s

meters:

- "hardware.\*"

resources:

- snmp://\*.\*.\*.\* # 被监控的物理机snmpd服务ip，可以同时加入多行，表示同时监控多个物理机

- snmp://\*.\*.\*.\*

sinks:

- meter\_sink

⑥　测试新添加meter项

ceilometer sample-list -m hardware.memory.total

# 资料

[1] OpenStack平台搭建参考文档 <http://www.aboutyun.com/thread-11722-1-1.html>；

[2] OpenStack共享存储热迁移参考文档 <http://www.aboutyun.com/thread-7878-1-1.html>；

[3] ceilometer安装文档 <http://www.bkjia.com/yjs/1039753.html>；