ironic部署物理机

> ironic是openstack的帐篷项目之一，主要用来部署裸机。本文以tecs3.0为例，介绍ironic部署裸机的流程。

ironic安装

> 使用rpm -ivh安装rpms目录下所有rpm包

权限配置

> 下面--project后面的services要根据实际情况填写。开源社区默认是service, tecs3.0使用的是services。具体可以通过
> `openstack project list`命令查看。

```
openstack user create --password IRONIC_PASSWORD \
--email ironic@example.com ironic
openstack role add --project services --user ironic admin
```

```
openstack service create --name ironic --description \
"Ironic baremetal provisioning service" baremetal
```

```
# 这里IRONIC_NODE替换成自己要安装ironic的机器的ip
openstack endpoint create --region RegionOne \
--publicurl http://IRONIC_NODE:6385 \
--internalurl http://IRONIC_NODE:6385 \
--adminurl http://IRONIC_NODE:6385 \
baremetal
```

3. 创建数据库

> 数据库密码IRONIC_DBPASSWORD根据实际情况替换，也可以不替换。

```
# mysql -u root -p
mysql> CREATE DATABASE ironic CHARACTER SET utf8;
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'localhost' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
mysql> GRANT ALL PRIVILEGES ON ironic.* TO 'ironic'@'%' \
IDENTIFIED BY 'IRONIC_DBPASSWORD';
```

4. 配置裸机服务

- ironic-api配置

> 修改`/etc/ironic/ironic.conf`数据库连接用户名，密码，DB_IP要和前面的对应，另外mysql端口号如果不是3306需要在DB_IP之后加上端口号。

eg:
`connection = mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP:29998/ironic?charset=utf8`

> 确认mysql的端口号可以通过查看`/etc/my.cnf.d/server.cnf`的[mysqld]部分的port确认。

```
[database]
...
# The SQLAlchemy connection string used to connect to the
# database (string value)
connection = mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?charset=utf8
```

```
[DEFAULT]
...
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
#rpc_backend=rabbit

[oslo_messaging_rabbit]
...
# The RabbitMQ broker address where a single node is used
# (string value)
rabbit_host=RABBIT_HOST

# The RabbitMQ userid (string value)
#rabbit_userid=guest

# The RabbitMQ password (string value)
#rabbit_password=guest
```

> admin_tenant_name后的services还是要跟openstack project list看到的对应。
> IDENTITY_IP替换成实际IP

```
[DEFAULT]
...
# Authentication strategy used by ironic-api: one of
# "keystone" or "noauth". "noauth" should not be used in a
# production environment because all authentication will be
# disabled. (string value)
#auth_strategy=keystone

[keystone_authtoken]
...
# Complete public Identity API endpoint (string value)
auth_uri=http://IDENTITY_IP:5000/

# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
identity_uri=http://IDENTITY_IP:35357/

# Service username. (string value)
admin_user=ironic

# Service account password. (string value)
admin_password=IRONIC_PASSWORD

# Service tenant name. (string value)
admin_tenant_name=services
```

- 创建数据库表

> 创建命令如下，如果前面已经创建过表，把create_schema换成update

```
# $MARIADB_PASSWORD换成实际数据库密码
mysql -uroot -p$MARIADB_PASSWORD -h127.0.0.1 -e "show databases;" | grep ironic
if [ $? -eq 0 ];then
    ironic-dbsync --config-file $ironic_conf upgrade
else
    ironic-dbsync --config-file $ironic_conf create_schema
fi
```

- 重启**ironic-api**服务

```
systemctl restart openstack-ironic-api
```

- **ironic-conductor**配置

> HOST_IP替换成实际ip(ironic-conductor服务所在节点ip)enabled_drivers后的驱动根据实际添加，这里使用pxe_ipmitool

```
[DEFAULT]
...
# IP address of this host. If unset, will determine the IP
# programmatically. If unable to do so, will use "127.0.0.1".
# (string value)
my_ip = HOST_IP

# Specify the list of drivers to load during service
# initialization. Missing drivers, or drivers which fail to
# initialize, will prevent the conductor service from
# starting. The option default is a recommended set of
# production-oriented drivers. A complete list of drivers
# present on your system may be found by enumerating the
# "ironic.drivers" entrypoint. An example may be found in the
# developer documentation online. (list value)
# 这里需要添加pxe_ztetoool驱动
enabled_drivers=pxe_ipmitool,pxe_ztetoool
```

> api_url后的IRONIC_API_IP换成实际ip

```
[conductor]
...
# URL of Ironic API service. If not set ironic can get the
# current value from the keystone service catalog. (string
# value)
api_url=http://IRONIC_API_IP:6385
```

> 配置同ironic-api，如果ironic-api和ironic-conductor在一个节点，只需要配置一次。

```
[database]
...
# The SQLAlchemy connection string to use to connect to the
# database. (string value)
connection = mysql+pymysql://ironic:IRONIC_DBPASSWORD@DB_IP/ironic?charset=utf8
```

```
[DEFAULT]
...
# The messaging driver to use, defaults to rabbit. Other
# drivers include qpid and zmq. (string value)
#rpc_backend=rabbit

[oslo_messaging_rabbit]
...
# The RabbitMQ broker address where a single node is used.
# (string value)
rabbit_host=RABBIT_HOST

# The RabbitMQ userid. (string value)
#rabbit_userid=guest

# The RabbitMQ password. (string value)
#rabbit_password=guest
```

```
[glance]
...
# Default glance hostname or IP address. (string value)
glance_host=GLANCE_IP
```

```
[neutron]
...
# URL for connecting to neutron. (string value)
url=http://NEUTRON_IP:9696
```

```
[keystone_authtoken]
...
# Complete public Identity API endpoint (string value)
auth_uri=http://IDENTITY_IP:5000/

# Complete admin Identity API endpoint. This should specify
# the unversioned root endpoint e.g. https://localhost:35357/
# (string value)
identity_uri=http://IDENTITY_IP:35357/

# Service username. (string value)
admin_user=ironic

# Service account password. (string value)
admin_password=IRONIC_PASSWORD

# 这里services注意根据实际情况替换
# Service tenant name. (string value)
admin_tenant_name=services
```

```
systemctl restart openstack-ironic-conductor
```

> 到这里ironic就已经安装完成了，后面就是nova和neutron的一些配置。

共管配置（控制节点修改）

> 由于一个nova-compute一次只能使用一种驱动，起虚机是用的libvirt驱动，而部署物理机是需要使用ironic驱动。
>
> 目前nova已经提供了共管的方式来同时管理裸机和虚机，参考：nova共管配置
>
> 需要说明的是共管方式是:

- 使用共管，需要设置nova_and_ironic_host_manager_enabled=True
- 共管方式只是用拿出一个节点提供ironic驱动，专门用来部署裸机
- 提供ironic驱动的机器不能用来起虚机
- 裸机使用的flavor必须要设置hw:compute_type=ironic属性

nova配置

```
[default]
compute_driver=ironic.IronicDriver

firewall_driver=nova.virt.firewall.NoopFirewallDriver

scheduler_host_manager=nova.scheduler.ironic_host_manager.IronicHostManager

ram_allocation_ratio=1.0

reserved_host_memory_mb=0

compute_manager=ironic.nova.compute.manager.ClusteredComputeManager

scheduler_use_baremetal_filters=True

scheduler_tracks_instance_changes=False

memcached_servers=localhost:11211

# 这个选项在控制节点配置
nova_and_ironic_host_manager_enabled=True
```

```
[ironic]
# Ironic keystone admin name
admin_username=ironic

#Ironic keystone admin password.
admin_password=IRONIC_PASSWORD

# keystone API endpoint
admin_url=http://IDENTITY_IP:35357/v2.0

# 这里services根据实际情况指定
# Ironic keystone tenant name.
admin_tenant_name=services

# URL for Ironic API endpoint.
api_endpoint=http://IRONIC_NODE:6385/v1

[keystone_authtoken]
memcached_servers=localhost:11211
```

```
systemctl restart openstack-nova-scheduler
systemctl restart openstack-nova-compute
```

neutron配置

> 部署裸机和虚机用的neutron配置是一样的，目前tecs3.0有些neutron默认没有设置，需要我们自己设置一下。
>
> 编辑 `/etc/neutron/plugins/ml2/ml2_conf.ini`,配置如下：
> ovs部分可能在`/etc/neutron/plugins/ml2/openvswitch_agent.ini`

```
[ml2]
type_drivers = flat
tenant_network_types = flat
mechanism_drivers = openvswitch

[ml2_type_flat]
flat_networks = physnet1

[ml2_type_vlan]
network_vlan_ranges = physnet1

[securitygroup]
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
enable_security_group = True

[ovs]
bridge_mappings = physnet1:br-eth2
# Replace eth2 with the interface on the neutron node which you
# are using to connect to the bare metal server
```

```
ovs-vsctl add-br br-int
ovs-vsctl add-br br-eth2
ovs-vsctl add-port br-eth2 eth2
```

```
systemctl restart neutron-plugin-openvswitch-agent
```

```
ovs-vsctl show

    Bridge br-int
        fail_mode: secure
        Port "int-br-eth2"
            Interface "int-br-eth2"
                type: patch
                options: {peer="phy-br-eth2"}
        Port br-int
            Interface br-int
                type: internal
    Bridge "br-eth2"
        Port "phy-br-eth2"
            Interface "phy-br-eth2"
                type: patch
                options: {peer="int-br-eth2"}
        Port "eth2"
            Interface "eth2"
        Port "br-eth2"
            Interface "br-eth2"
                type: internal
    ovs_version: "2.3.0"
```

> 最终要保证如下几个服务是正常的：

```
neutron-l3-agent.service
neutron-dhcp-agent.service
neutron-openvswitch-agent.service
neutron-server.service
```

> 如果l3-agent启动失败，检查下`/etc/neutron/l3_agent.ini`是否配置了：

```
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver`
```

控制节点修改`/etc/neutron/neutron.conf`文件

```
# 这里确保firewall配置了
service_plugins =lbaasv2,router,metering,firewall
```

> 控制节点修改`/etc/neutron/fwaas_driver.ini`文件，内容如下：

```
[fwaas]
#
driver = neutron.services.firewall.drivers.linux.iptables_fwaas.IptablesFwaasDriver

# Enable FWaaS (boolean value)
enabled = True
```

> 部署裸机时哪个节点提供**dhcp**服务，就需要在哪个节点上配置**neutron dhcp**

> 修改`/etc/neutron/dhcp_agent.ini`，配置dhcp_driver和interface_driver

```
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
```

创建网络

```
neutron net-create --tenant-id $TENANT_ID deploynet1 --shared \
--provider:network_type flat --provider:physical_network physnet1
```

```
neutron subnet-create deploynet1 $NETWORK_CIDR --name $SUBNET_NAME \
--ip-version=4 --gateway=$GATEWAY_IP --allocation-pool \
start=$START_IP,end=$END_IP --enable-dhcp
```

镜像添加

> 目前提供版本使用的是iso镜像，仅测试过centos7。
>
> 镜像上传方式如下：

```
glance image-create --name deploy-vmlinuz --visibility public --disk-format aki --container-format aki < ironic-deploy.kernel

glance image-create --name deploy-initrd --visibility public --disk-format ari --container-format ari < ironic-deploy.initramfs

glance image-create --name my-centos --tag "os:linux" "os_type:CentOS 7  (64-bit)" --visibility public \
--disk-format iso --container-format bare \
--property kernel_id=$(glance image-list | awk '/deploy-vmlinuz/ {print $2}') \
--property ramdisk_id=$(glance image-list | awk '/deploy-initrd/ {print $2}') \
< CentOS-7-x86_64-Minimal-1611.iso
```

创建**flavor**

> 这里我们先创建一个flavor，flavor里的硬件信息我们先手动填写，里面的cpu，内存，硬盘和要部署的裸机硬件信息学保持一致。
>
> 如果是实际大批量部署，需要用到ironic-inspector来收集硬件信息。

```
RAM_MB=55784
CPU=48
DISK_GB=100
ARCH=x86_64
nova flavor-create my-baremetal-flavor 20 $RAM_MB $DISK_GB $CPU
nova flavor-key my-baremetal-flavor set cpu_arch=$ARCH
nova flavor-key my-baremetal-flavor set capabilities:boot_option="local"
nova flavor-key my-baremetal-flavor set hw:compute_type=ironic
```

> 当然，在创建flavor之前，必须先确保quota额度够了。通过nova quota-show可以看一下，不够可以分别通过下面命令进行cores和ram的扩展（后面的ID是tenant为admin的ID）

```
# openstack project list可以查看admin的ID
nova  quota-update --cores 48 817176c38de94b32af9e49a5bb792e3d
nova  quota-update --ram 55784 817176c38de94b32af9e49a5bb792e3d
```

PXE配置

> 在部署之前，我们需要配置tftp服务，先安装tftp-serve和xinet的rpm包。

```
rpm -ivh tftp-server*.rpm
rpm -ivh xinetd*.rpm
systemctl enable xinetd.service
systemctl start xinetd.service
```

> 创建对应的目录，并修改一下其所有者。

```
mkdir -p /tftpboot
mkdir -p /tftpboot/pxelinux.cfg
chown -R ironic /tftpboot
mkdir /home/install_share
chown -R ironic /home/install_share

mkdir -p /linuxinstall/linuxinstall_0
mkdir -p /linuxinstall/linuxinstall_1
mkdir -p /linuxinstall/linuxinstall_2
mkdir -p /linuxinstall/linuxinstall_3
mkdir -p /linuxinstall/linuxinstall_4
mkdir -p /linuxinstall/linuxinstall_5
mkdir -p /linuxinstall/linuxinstall_6
mkdir -p /linuxinstall/linuxinstall_7
mkdir -p /linuxinstall/linuxinstall_8
mkdir -p /linuxinstall/linuxinstall_9
chown -R ironic /linuxinstall
```

> 配置tftp服务

/etc/xinetd.d/tftp配置文件如下：

```
service tftp
{
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -v -v -v -v -v --map-file /tftpboot/map-file /tftpboot
        disable                 = no
        flags                   = IPv4
        port                    = 69
}
```

```
cp /usr/share/syslinux/pxelinux.0 /tftpboot

echo 're ^(/tftpboot/) /tftpboot/\2' > /tftpboot/map-file
echo 're ^/tftpboot/ /tftpboot/' >> /tftpboot/map-file
echo 're ^(^/) /tftpboot/\1' >> /tftpboot/map-file
echo 're ^([^/]) /tftpboot/\1' >> /tftpboot/map-file
```

> ironic部署的时候会自动生成指定的镜像，以及pxe需要的default文件，并存放在/tftpboot目录下。根据ironic node的uuid和mac地址来区分。

环境准备

保证裸机的网络连通性

1 部署裸机与部署计算节点需要在同一个vlan中

2 裸机需要能访问neutron的DHCP服务
 2.1 组网方式 1 : 简单的组网方式, 裸机可以访问控制节点的ovs□
 2.2 组网方式2 : 如果限制裸机只能访问计算节点, 则需要在ironic 的conductor节点增加neutron 的dhcp agent 服务

- /etc/neutron/neutron.conf 中需要 dhcp_agents_per_network = 1 这里的1需要改成需要的数量
- neutron agent-list |grep DHCP 获取 DHCP_AGENT (对应节点的neutron-dhcp-agent服务必须已经安装启动. TECS 774版本安装计算节点, 默认不安装 openstack-neutron-dhcp-2016.3.16-1.1.774.noarch 这个包)
- neutron dhcp-agent-network-remove
- neutron dhcp-agent-network-add

3 裸机需要能访问 /etc/ironic/ironic.conf 中定义的 my_ip 这个地址(TFTP, NFS, Samba 服务均使用这个地址)

4 部署计算节点需要能访问 glance 镜像服务

5 部署计算节点需要能访问 neutron 网络服务

开始部署

ironic node节点创建

> 每一个ironic node对应一个裸机，每个ironic port对应ironic裸机的一个网卡。创建ironic node时需要指定ironic的驱动信息，裸机带外信息。

```
# 创建ironic node
NODE_UUID=$(ironic node-create -d  pxe_ztetoool | awk '/^\| uuid / {print $4}')

# 为node节点添加带外信息
ironic node-update $NODE_UUID add driver_info/ipmi_username=root
driver_info/ipmi_password=ossdbg1 driver_info/ipmi_address=10.43.200.161

DEPLOY_VMLINUZ_UUID=$(glance image-list | awk '/deploy-vmlinuz/ {if($4=="deploy-vmlinuz"){print
$2}}')
DEPLOY_INITRD_UUID=$(glance image-list | awk '/deploy-initrd/ {if($4=="deploy-initrd"){print
$2}}')

# 指定小镜像
ironic node-update $NODE_UUID add driver_info/deploy_kernel=$DEPLOY_VMLINUZ_UUID
driver_info/deploy_ramdisk=$DEPLOY_INITRD_UUID

# 指定裸机硬件信息
ironic node-update $NODE_UUID add properties/cpus=48 properties/memory_mb=108667
properties/local_gb=200 properties/cpu_arch=x86_64

# 创建ironic port
# 这里的mac地址是pxe网口，用来部署的
ironic port-create -n $NODE_UUID -a 98:40:bb:81:37:5b

# 设置启动方式
ironic node-update $NODE_UUID add properties/capabilities="boot_option:local"
```

裸机部署

> 在使用nova部署裸机之前需要，需要完成一些准备工作。并保证一下服务状态是正常的：

- xinetd.service

> 一般创建完ironic node之后需要等待大概1min才能进行nova boot操作，否则会出现no valid host错误。这是因为nova-compute会定期去同步ironic node的信息到nova数据库中。
>
> 检查nova hypervisor-list 和nova hypervisor-show $ID，确保信息按配置修改。没有则等待一段时间。

```
# 这里使用我们前面创建的neutron net id
NIC_UUID=$(neutron net-list | awk '/deploynet1/ {print $6}')

# flavor需要指定对应的
nova boot --flavor 20 --image my-centos --nic net-id=$NIC_UUID instance_partion_deploy
```

结果查看

> 裸机安装和普通的pxe流程差不多，都是通过日志的方式来确认安装进度的。当系统安装完成之后，ironic会自动把ironic的状态修改为active，nova会自动同步ironic的状态，将nova instance的状态设置为ACTIVE