

kolla-ansible部署OpenStack (allinone方式)

参考文档: <https://docs.openstack.org/kolla-ansible/yoga/user/quickstart.html>

1. 设置主机名、IP地址、关闭防火墙和SELinux

```
# 设置主机名
hostnamectl set-hostname openstack
# 设置IP地址（两块网卡）
# 网卡规划（至少两个网络，最好三个网络）
ens160（NAT/桥接）-配置地址
ens224（配置仅主机-专门去连接到内网）-不配置地址
# 查看网络连接
nmcli connection show
# 修改NAT网卡（ens160）并生效
nmcli connection modify "ens160" ipv4.addresses "192.168.8.10/24" ipv4.gateway
"192.168.8.2" ipv4.dns "223.5.5.5" ipv4.method manual connection.autoconnect yes
nmcli connection up ens160
```

2. 关闭防火墙和selinux

```
# 关闭防火墙
systemctl stop firewalld && systemctl disable firewalld
# 关闭selinux
vi /etc/selinux/config
# 修改以下内容
SELINUX=disabled
```

3. 去aliyun换相关yum源

```
# 基础源替换
sed -e 's|^mirrorlist=|#mirrorlist=|g' \
    -e 's|^#
baseurl=https://repo.almalinux.org|baseurl=https://mirrors.aliyun.com|g' \
    -i.bak \
    /etc/yum.repos.d/almalinux*.repo
# epel源
dnf install epel-release -y
sed -i
's|^#baseurl=https://download.example/pub|baseurl=https://mirrors.aliyun.com|'
/etc/yum.repos.d/epel*
sed -i 's|^metalink|#metalink|' /etc/yum.repos.d/epel*
```

4. 安装network-scripts并设置为首选网络管理服务

```
# 安装network-scripts
dnf install network-scripts -y
# 设置network-scripts为首选网络管理服务
systemctl stop NetworkManager && systemctl disable NetworkManager && systemctl
start network && systemctl enable network
```

5. 安装kolla-ansible支持的基础软件包

```
dnf install git python3-devel libffi-devel gcc openssl-devel python3-libselinux  
vim -y
```

6. 设置pip以及venv

```
# 替换pip源-准备工作  
mkdir .pip  
cd .pip  
vi pip.conf  
# 替换pip源-插入以下内容  
[global]  
index-url = http://mirrors.aliyun.com/pypi/simple/  
  
[install]  
trusted-host=mirrors.aliyun.com  
# 创建venv环境  
mkdir /kolla-ansible-env  
python3 -m venv /kolla-ansible-env/  
# 注意：这个部分是加载env环境的，只要你系统做过关机重启，就需要重新运行以下命令加载  
source /kolla-ansible-env/bin/activate
```

7. 升级pip版本

```
pip install --upgrade pip
```

8. 安装ansible和相关依赖包

```
pip install 'ansible>=4,<6'  
pip install selinux
```

9. 拉取kolla-ansible代码（有时候网不好，要多试几次）

```
pip install git+https://opendev.org/openstack/kolla-ansible@unmaintained/yoga
```

10. 创建/etc/kolla目录并设置权限

```
mkdir -p /etc/kolla  
chown $USER:$USER /etc/kolla
```

11. 拷贝模板文件到/etc/kolla目录

```
cp -r /kolla-ansible-env/share/kolla-ansible/etc_examples/kolla/* /etc/kolla
```

12. 拷贝主机清单文件并设置python解析器

```
# 拷贝文件  
cp /kolla-ansible-env/share/kolla-ansible/ansible/inventory/* .  
# 设置解析器  
vi all-in-one  
# 在文件首行插入以下内容  
localhost ansible_python_interpreter=python
```

13. 安装kolla-ansible相关galaxy

```
# 检查文件是否是如下内容
cat /kolla-ansible-env/share/kolla-ansible/requirements.yml
---
collections:
  - name: https://opendev.org/openstack/ansible-collection-kolla
    type: git
    version: unmaintained/yoga
# 安装依赖项
kolla-ansible install-deps
```

14. 配置ansible.cfg

```
mkdir /etc/ansible/
vi /etc/ansible/ansible.cfg
# 插入以下内容
[defaults]
host_key_checking=False
pipelining=True
forks=100
```

15. 生成密码文件

```
# 生成
kolla-genpwd
# 所有密码都在这里找，重点查看登录密码
cat /etc/kolla/passwords.yml | grep keystone_admin_password
.....
keystone_admin_password: xxxxxxxxxxxxxxxxxxxxxxxxx
.....
```

16. 修改globals.yml

```
vi /etc/kolla/globals.yml
# 只修改以下几个选项并取消注释
# 这行直接取消注释即可
kolla_base_distro: "centos"
# 这行修改成你的桥接/NAT的外网网卡
network_interface: "ens160"
# 这行修改为你的内网网卡即可
neutron_external_interface: "ens224"
# 配置虚拟 IP 地址（和外网一个网段）
kolla_internal_vip_address: "192.168.8.210"
```

17. 修改docker的daemon.json文件，防止拉取镜像出问题

```
mkdir -p /etc/docker
tee /etc/docker/daemon.json <<- 'EOF'
{
  "registry-mirrors": ["这里写你的镜像加速器地址，去阿里云里边去看"]
}
EOF
```

18. 软件更新/重启/做模板（给多节点部署做准备工作）

```
# 更新
dnf update
# 更新后重启
reboot
# 开机后关机->做快照
```

19. 下载需要部署的软件包

```
# 注意：因为docker国内被封禁，所以需要改这个kolla-ansible源码中的变量
# 转到这个目录
cd
/root/.ansible/collections/ansible_collections/openstack/kolla/roles/baremetal/default
faults
# 打开这个文件
vi main.yml
# 找到这条，修改以下内容
docker_yum_url: "https://mirrors.aliyun.com/docker-ce/linux/centos/"

# 注意：这个部分是加载env环境的，只要你系统做过关机重启，就需要重新运行以下命令加载
source /kolla-ansible-env/bin/activate
# 运行预部署（allinone单节点方式）
kolla-ansible -i ./all-in-one bootstrap-servers

# 多节点方式（需要配置清单）
# 先修改multinode里边的IP地址
vim multinode
# 测试这些节点是否可以通畅（前提各个节点都需要做免密登录）
ansible -i multinode all -m ping
# 运行预部署
kolla-ansible -i ./multinode bootstrap-servers
```

20. 检查机器是否符合要求

```
# 注意：因为这个工具不支持咱们的系统，但实际是可以的，所以需要改代码
cd /kolla-ansible-env/share/kolla-ansible/ansible/roles/prechecks/vars
# 编辑文件
vi main.yml
# 添加以下内容（注意跟前边的格式对齐）
AlmaLinux:
  - "8"
  - "9"
# 单节点
kolla-ansible -i ./all-in-one prechecks
# 多节点
kolla-ansible -i ./multinode prechecks
# 如果在单节点预检时候出现下列问题，检查/etc/host里的内容，删除多余hosts条目即可
```

```
failed: [localhost] (item=[{'changed': False, 'stdout': '172.25.250.10  STREAM
openstack\n172.25.250.10  DGRAM  \n172.25.250.10  RAW  ', 'stderr': '', 'rc':
0, 'cmd': ['getent', 'ahostsv4', 'openstack'], 'start': '2024-06-20
06:32:26.093954', 'end': '2024-06-20 06:32:26.096506', 'delta': '0:00:00.002552',
'msg': '', 'invocation': {'module_args': {'_raw_params': 'getent ahostsv4
openstack', '_uses_shell': False, 'warn': False, 'stdin_add_newline': True,
'strip_empty_ends': True, 'argv': None, 'chdir': None, 'executable': None,
'creates': None, 'removes': None, 'stdin': None}}, 'stderr_lines': [], 'failed':
False, 'item': 'localhost', 'ansible_loop_var': 'item'}, '172.25.250.10  STREAM
openstack']) => {"ansible_loop_var": "item", "changed": false, "item":
[{"ansible_loop_var": "item", "changed": false, "cmd": ["getent", "ahostsv4",
"openstack"], "delta": "0:00:00.002552", "end": "2024-06-20 06:32:26.096506",
"failed": false, "invocation": {"module_args": {"_raw_params": "getent ahostsv4
openstack", "_uses_shell": false, "argv": null, "chdir": null, "creates": null,
"executable": null, "removes": null, "stdin": null, "stdin_add_newline": true,
"strip_empty_ends": true, "warn": false}}, "item": "localhost", "msg": "", "rc":
0, "start": "2024-06-20 06:32:26.093954", "stderr": "", "stderr_lines": [],
"stdout": "172.25.250.10  STREAM openstack\n172.25.250.10  DGRAM
\n172.25.250.10  RAW  "}, "172.25.250.10  STREAM openstack"], "msg":
"Hostname has to resolve uniquely to the IP address of api_interface"}
```

21. 正式部署

```
# 单节点
kolla-ansible -i ./all-in-one deploy
# 多节点
kolla-ansible -i ./multinode deploy
```

22. 安装命令行管理

```
# 安装
pip install python-openstackclient -c
https://releases.openstack.org/constraints/upper/yoga
# 生成admin文件
kolla-ansible post-deploy
# 使用文件
cp /etc/kolla/admin-openrc.sh /root/admin-openrc
source /root/admin-openrc
```

23. (重要) 修改网络配置文件, 使得openstack能正常与外部通信

注意:

1. 一定要保证openstack中没有任何网络, 如果有, 需要把所有网络删除清空后在修改配置文件
2. 如果是多节点部署, 这个配置需要在控制节点/网络节点里修改 (如何找这个文件请自行研究)

```
# 修改此文件
vim /etc/kolla/neutron-server/ml2_conf.ini
# 清空文件中的所有行, 并复制以下的所有内容
[ml2]
type_drivers = flat,vlan,vxlan
tenant_network_types = vxlan
mechanism_drivers = openvswitch,l2population
extension_drivers = port_security,qos
```

```

[securitygroup]
enable_security_group=True

[m12_type_geneve]

[m12_type_gre]

[m12_type_flat]
flat_networks = physnet1

[m12_type_vxlan]
vni_ranges = 10000:19999

[m12_type_flat]
flat_networks=extnet

# 修改此文件
vim /etc/kolla/neutron-openvswitch-agent/openvswitch_agent.ini
# 清空文件中的所有行，并复制以下的所有内容
[agent]
tunnel_types = vxlan
vxlan_udp_port=4789
l2_population = false
arp_responder = true
drop_flows_on_start=false

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

[ovs]
bridge_mappings = extnet:br-ex
integration_bridge=br-int
tunnel_bridge=br-tun
datapath_type = system
ovsdb_connection = tcp:127.0.0.1:6640
ovsdb_timeout = 10
local_ip = 192.168.8.10
# 重启openstack虚拟机
reboot

```

24. 思考

- 多节点部署各个节点需不需要做免密登录？需不需要写hosts文件？如果需要的话可以参考此示例

```

# 设置hosts
echo "192.168.8.10 openstack" >> /etc/hosts
# 设置密钥
ssh-keygen -t rsa
# 配置免密
ssh-copy-id openstack

```

- 多节点清单如何更改？参考此链接学习

https://blog.csdn.net/qq_35485875/article/details/128868634

- 如果多节点，存储地址需要设置在哪？参考此链接学习

<https://docs.openstack.org/kolla-ansible/yoga/admin/production-architecture-guide.html#network-configuration>

- 虚拟机镜像怎么选择，在哪里去下载openstack专用镜像？
- 可以尝试着创建一个虚拟机实例，将OpenStack中的虚拟机实例连接到外网中。关键点如下

a.neutron相关

1.给huaweicloud租户添加公有网络 (public-network)

```
# 查询租户ID (project-id/tenant-id, 后边会用到)
source /kolla-ansible-env/bin/activate
source admin-openrc
openstack project list

+-----+-----+
| ID                                | Name      |
+-----+-----+
| b8336aafeedf450da44b30878cb9c71b | admin     |
| ff8a5329175b45b492552485f00ee6fb | service   |
+-----+-----+

# 查询到admin ID为b8336aafeedf450da44b30878cb9c71b
# 使用openstack命令如何创建
openstack network create \
  --external \
  --project b8336aafeedf450da44b30878cb9c71b \
  --provider-network-type flat \
  --provider-physical-network extnet \
  public-network
```

2.添加公有子网 (外部网络, public-subnet01)

```
openstack subnet create \
  --network public-network \
  --subnet-range 192.168.8.0/24 \
  --project b8336aafeedf450da44b30878cb9c71b \
  --ip-version 4 \
  --allocation-pool start=192.168.8.110,end=192.168.8.150 \
  --gateway 192.168.8.2 \
  public-subnet01
```

3.用户添加私有网络-私有子网（内部网络，Openstack实例用的网络， private-network)

```
# 创建私有网络
openstack network create private-network
# 创建私有子网
openstack subnet create \
  --network private-network \
  --subnet-range 172.16.1.0/24 \
  --project b8336aafeedf450da44b30878cb9c71b \
  --ip-version 4 \
  --dhcp \
  --allocation-pool start=172.16.1.101,end=172.16.1.110 \
  --gateway 172.16.1.254 \
  private-subnet01
```

4.创建路由器 - 关联外部网络/内部网络

```
# 创建路由器
openstack router create router01
# 路由器跟外网关联
openstack router set --external-gateway public-network router01
# 路由器跟内网关联
openstack router add subnet router01 private-subnet01
```

5. 创建安全组并添加安全组规则

```
# 创建一个web安全组
openstack security group create web
# 入方向允许tcp 22通过
openstack security group rule create --ingress --ethertype IPv4 --protocol tcp --dst-port 22 web
# 入方向允许tcp 80通过
openstack security group rule create --ingress --ethertype IPv4 --protocol tcp --dst-port 80 web
# 入方向允许tcp 443通过
openstack security group rule create --ingress --ethertype IPv4 --protocol tcp --dst-port 443 web
```

b.glance相关

上传CentOS-7-x86_64-GenericCloud-2211.qcow2镜像到glance中

```
# 安装wget
yum install wget -y
# 下载镜像
wget https://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud-2211.qcow2
# 上传镜像
source /kolla-ansible-env/bin/activate
source admin-openrc
```



```
openstack image create --file CentOS-7-x86_64-GenericCloud-2211.qcow2 --disk-format qcow2 --container-format bare --public --progress centos7_2211
```

c.nova相关

1.添加一个规格，名称为flavor-zhang，分配内存为1GiB，磁盘大小为15GiB，vcpu为1核

```
# 查看nova规格（默认什么也没有）
openstack flavor list
# 创建规格
openstack flavor create --id auto --ram 1024 --disk 15 --vcpus 1 flavor-zhang
```

2.生成一个叫openstack_demo_zhang_key的密钥对

```
openstack keypair create --private-key ~/openstack_demo_zhang_key.pem
openstack_demo_zhang_key
```

3.创建一个叫cloud_server的服务器，使用规格为flavor-zhang，镜像为centos7_2211，密钥为openstack_demo_zhang_key，安全组为web，网络使用私有网络（创建实例）

```
openstack server create --flavor flavor-zhang --image centos7_2211 --key-name
openstack_demo_zhang_key --security-group web --network private-network
cloud_server
```

d.后续工作

1.绑定浮动IP

🔍

正在显示 1 项

示例 ID =

筛选

创建实例

删除实例

更多操作

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	cloud_server	centos7_2211	172.16.1.107	flavor-zhang	openstack_demo_zhang_key	运行	all nova	无	运行中	0 分钟	<div>创建快照</div> <div>绑定浮动IP</div>

正在显示 1 项

绑定浮动IP

管理浮动IP的关联

IP 地址 *

没有分配浮动IP

+

请为选中的实例或端口选择要

待连接的端口 *

cloud_server: 172.16.1.107

分配浮动IP

资源池 *
public-network

描述:
从指定的浮动IP池中分配一个浮动IP。

项目配额
浮动IP 0 已使用, 共 50

描述

取消

分配IP

管理浮动IP的关联

IP 地址 *
192.168.1.139

待连接的端口 *
cloud_server: 172.16.1.107

请为选中的实例或端口选择要绑定的IP地址。

取消

关联

正在显示 1 项											
<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
<input type="checkbox"/>	cloud_server	centos7_2211	172.16.1.107 192.168.1.139	flavor_zhang	openstack_demo_zhang_key	运行	eu-nova	无	运行中	3 分钟	<div>创建快照</div>

2.测试云主机

```
# 加安全组允许icmp
openstack security group rule create --ingress --ethertype IPv4 --protocol icmp web

# ping分配的浮动IP
ping x.x.x.x(上边分配的浮动IP-192段的)

# 设置密钥
chmod 0400 openstack_demo_zhang_key.pem

# 连接到你创建的Openstack实例中
ssh -i openstack_demo_zhang_key.pem centos@192.168.8.119
The authenticity of host '192.168.8.119 (192.168.8.119)' can't be established.
ECDSA key fingerprint is SHA256:8qkRaa6/40ocjB9vIOVK+BBZ845CC7Uham0PWGQ4+ag.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added '192.168.8.119' (ECDSA) to the list of known hosts.
[centos@cloud-server ~]$

# 如何在这个openstack创建的实例使用root账户
[centos@cloud-server ~]$ sudo -i
[root@cloud-server ~]#

# 测试实例可不可以连接到外网
[root@cloud-server ~]# ping www.baidu.com
PING www.a.shifen.com (110.242.68.3) 56(84) bytes of data.
```

```
64 bytes from 110.242.68.3 (110.242.68.3): icmp_seq=1 ttl=127 time=19.1 ms
64 bytes from 110.242.68.3 (110.242.68.3): icmp_seq=2 ttl=127 time=26.2 ms
64 bytes from 110.242.68.3 (110.242.68.3): icmp_seq=3 ttl=127 time=18.6 ms
```

3.扩展

```
# 第一部分：如何解决网络问题
# 如果在物理机中ping在openstack中创建的浮动IP不通（上图192段），该如何做？
# 临时解决方案：将原有机主机网络也调整成NAT（后期部署多节点的时候将ens160和ens224网络类型调
换，即ens160为仅主机，ens224为NAT）-待测试
# 编辑br-ex文件
vim /etc/sysconfig/network-scripts/ifcfg-br-ex
# 直接输入以下内容
TYPE="OVSBridge"
DEVICETYPE=ovs
BOOTPROTO="static"
NAME="br-ex"
DEVICE="br-ex"
ONBOOT="yes"
# 此处改一个和NAT一样网段的IP
IPADDR=192.168.8.100
PREFIX=24
# 以下内容如果两张网卡都是VNET8就不填写
GATEWAY=192.168.8.2
DNS1=223.5.5.5
# 编辑ens224文件
# 此处文件名我的是ifcfg-wired_connection_1，你的可能不一定是ifcfg-wired_connection_1，
请自行辨别
vim /etc/sysconfig/network-scripts/ifcfg-wired_connection_1
TYPE="OVSPort"
BOOTPROTO="static"
NAME="ens224"
DEVICE="ens224"
ONBOOT="yes"
DEVICETYPE=ovs
OVS_BRIDGE=br-ex
# 重启网络服务
systemctl restart network
# 参考文档（98%好用）
https://blog.csdn.net/m0\_69088440/article/details/130272298
# 第二部分：如何改openstack自带的logo
# 到这个目录中
cd /usr/share/openstack-dashboard/static/dashboard/img
# 把你的logo图片用相关工具转换成svg
# 替换这两个文件即可
logo-splash.svg logo.svg
```