# kolla部署——neutron配置

kolla部署的容器场景，主要分控制节点和计算节点。控制节点主要有neutron-server（管理入口）。计算节点主要有neutron\_openvswitch\_agent（neutron的L2 Agent）, openvswitch\_vswitchd（ovs服务），openvswitch\_db（ovs的db服务，不确定是否需要，可以直接用网络节点上的db）。网络节点除了计算节点的三个容器外，还有neutron\_metadata\_agent（元数据服务，获取数据后发给nova的metadata服务），neutron\_l3\_agent(L3 agent,虚拟路由相关的服务)，neutron\_dhcp\_agent（dhcp agent）

# neutron.conf

所有neutron相关容器都需要配置

## Kolla 默认配置项

[DEFAULT]

debug = False

log\_dir = /var/log/kolla/neutron

use\_stderr = False

#neutron Server的监听地址，可以配置为本地ip，只对neutron server服务有效，其他服务可以不用管。

bind\_host = 192.168.102.15

#neutron Server的监听端口

bind\_port = 9696

#paste.deploy 配置文件 api service

api\_paste\_config = /usr/share/neutron/api-paste.ini

endpoint\_type = internalURL

# neutron sever的后端worker进程数

api\_workers = 5

# neutron\_metadata\_agent服务的后端进程数

metadata\_workers = 5

# neutron-ns-metadata-proxy和neutron-metadata-agent两个进程之间交互metadata数据的

# unix domain socket

metadata\_proxy\_socket = /var/lib/neutron/kolla/metadata\_proxy

# 管理虚拟interface的驱动

interface\_driver = openvswitch

# 允许重叠子网

allow\_overlapping\_ips = true

#开启核心插件是ml2

core\_plugin = ml2

#开启服务插件router。这里是个list配置，还可以配置虚拟机负载均衡和虚拟防火墙

service\_plugins = router

#消息中间的地址

transport\_url = rabbit://openstack:cYhhGc6Gj0XMmY1p4Spj83Ms7cafzYT4wdM6y6d9@192.168.102.15:5672,openstack:cYhhGc6Gj0XMmY1p4Spj83Ms7cafzYT4wdM6y6d9@192.168.102.21:5672

[nova]

#nova的用户和keystone 的admin段的endpoint配置

auth\_url = http://192.168.102.47:35357/v3

auth\_type = password

project\_domain\_id = default

user\_domain\_id = default

region\_name = RegionOne

project\_name = service

username = nova

password = L9Cgvp2kAp0I0aMBgp60QG08T4aCTCm3xEfahjuZ

endpoint\_type = internal

[oslo\_concurrency]

lock\_path = /var/lib/neutron/tmp

[agent]

#配置rootwrap

#下面是必须的，neutron调用OVS的时候是通过命令行完成的，它会执行sudo尝试切换到#root用户执行，如果没有下面的配置你会看到ovs-agent满屏幕的错误信息

root\_helper = sudo neutron-rootwrap /etc/neutron/rootwrap.conf

[database]

#数据库连接信息，最大连接次数不限制

connection = mysql+pymysql://neutron:OB139WSGV96QrhiGIFKU8NAsWYWOU2DVxobjDHaX@192.168.102.47:3306/neutron

max\_retries = -1

[keystone\_authtoken]

auth\_uri = http://192.168.102.47:5000/v3

auth\_url = http://192.168.102.47:35357/v3

auth\_type = password

project\_domain\_id = default

user\_domain\_id = default

project\_name = service

username = neutron

password = Hhzv0jyo36DMvf9kagE2AqIOqE6nC8z4iQXu0Fhg

# (Optional) If defined, indicate whether token data should be authenticated or# authenticated #and encrypted. If MAC, token data is authenticated (with HMAC)# in the cache. If ENCRYPT, #token data is encrypted and authenticated in the# cache. If the value is not one of these options #or empty, auth\_token will# raise an exception on initialization. (string value)# Allowed values: #None, MAC, ENCRYPT

memcache\_security\_strategy = ENCRYPT

memcache\_secret\_key = WqfQMiQZ3ZZ7NeAoVwcL8k9ECHm5RgZgSr9VgpBZ

memcached\_servers = 192.168.102.15:11211

[oslo\_messaging\_notifications]

driver = noop

## 其他重要配置项

参考 <https://docs.openstack.org/ocata/config-reference/networking/samples/neutron.conf.html>

router\_distributed = false

#是否要启用分布式路由，默认是false。

# ml2\_conf.ini

所有neutron相关容器都需要配置，计算节点可能用不到（待确认），因为只处理api流程。

[ml2]

# 驱动类型。可以从neutron.ml2.type\_drivers里面去加载。

type\_drivers = flat,vlan,vxlan

# 租户网络的驱动类型，默认是local.这里选择了vxlan。

tenant\_network\_types = vxlan

#网络机制的驱动，从neutron.ml2.mechanism\_drivers 里面去加载。l2population主要是给虚拟机交换机(如ovs)，减少广播阻塞。原话如下：

#L2 population is mechanism driver for ML2 plugin which tends to leverage the #implementation of overlay networks. By populating the forwarding tables of virtual #switches (LinuxBridge or OVS), l2population mech driver will decrease broadcast traffics #inside the physical networks fabric while using overlays networks (VXLan, GRE)

mechanism\_drivers = openvswitch,l2population

[ml2\_type\_vlan]

# List of <physical\_network>:<vlan\_min>:<vlan\_max> or <physical\_network># specifying #physical\_network names usable for VLAN provider and tenant# networks, as well as ranges of #VLAN tags on each available for allocation to# tenant networks. (list value)

#这里不配置，也没有用到

network\_vlan\_ranges =

[ml2\_type\_flat]

#扁平网络配置，physnet1是provider-physical-network的值，用于创建浮动ip地址。该项可#配置多个，如physnet1,physnet2。默认是\*,即名称可以任意。

flat\_networks = physnet1

[ml2\_type\_vxlan]

#VXLAN VNI IDs的范围，和vlan tag类似

vni\_ranges = 1:1000

# VXLAN的广播组，不配置表示禁用广播组。

vxlan\_group = 239.1.1.1

[securitygroup]

#L2 agent的防火墙驱动

firewall\_driver = neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

[agent]

#未知

tunnel\_types = vxlan

l2\_population = true

arp\_responder = true

[ovs]

# 这段创建br-ex网桥，其实只要在网络节点就行，计算节点和控制节点应该是多余的。

bridge\_mappings = physnet1:br-ex

ovsdb\_connection = tcp:192.168.102.22:6640

local\_ip = 192.168.102.22

## 其他重要配置项

参考 <https://docs.openstack.org/ocata/config-reference/networking/samples/ml2_conf.ini.html>

【ml2】

# An ordered list of extension driver entrypoints to be loaded from the

# neutron.ml2.extension\_drivers namespace. For example: extension\_drivers =

# port\_security,qos (list value)

#外部扩展驱动，如port\_security,qos

extension\_drivers =

# Maximum size of an IP packet (MTU) that can traverse the underlying physical

# network infrastructure without fragmentation when using an overlay/tunnel

# protocol. This option allows specifying a physical network MTU value that

# differs from the default global\_physnet\_mtu value. (integer value)

#物理网卡的mtu值，默认是1500bytes，网络好的话可以适当增加该值

#参考<http://baike.baidu.com/item/mtu/508920?fr=aladdin>

path\_mtu = 0

# A list of mappings of physical networks to MTU values. The format of the

# mapping is <physnet>:<mtu val>. This mapping allows specifying a physical

# network MTU value that differs from the default global\_physnet\_mtu value.# (list value)

#修改指定网络的mtu值的物理网络，这个配置比上面那个更细化。

physical\_network\_mtus =

# metadata\_agent.ini

网络节点neutron\_metadata\_agent容器的配置。

OpenStack的metadata服务机制：<https://www.ibm.com/developerworks/cn/cloud/library/1509_liukg_openstackmeta/index.html>

neutron\_metadata\_agent服务把收集到的metadata信息会发送到nova metadata api。

[DEFAULT]

#nova metadata服务的ip和端口配置

nova\_metadata\_ip = 192.168.102.47

nova\_metadata\_port = 8775

#和nova metadata服务交互的协定的secret，在nova.conf的[neutron]中也配置了相同的secret。

metadata\_proxy\_shared\_secret = q8FqPL6i9ctlkJOWCI8ju8eGQ2ag27XQ9fktp14Z

## 其他重要配置项

参考 <https://docs.openstack.org/ocata/config-reference/networking/samples/metadata_agent.ini.html>

Cache相关的配置比较用的到。

# dhcp\_agent.ini

网络节点neutron\_dhcp\_agent容器的配置

[DEFAULT]

#dnsmasq的配置文件路径

dnsmasq\_config\_file = /etc/neutron/dnsmasq.conf

# Allows for serving metadata requests coming from a dedicated metadata access

# network whose CIDR is 169.254.169.254/16 (or larger prefix), and is connected

# to a Neutron router from which the VMs send metadata:1 request. In this case

# DHCP Option 121 will not be injected in VMs, as they will be able to reach

# 169.254.169.254 through a router. This option requires

#激活通过dhcp方式发送neutron的metadata信息。参考metadata\_agent.ini配置章节开头的metadata服务原理章节的链接。

enable\_isolated\_metadata = true

# In some cases the Neutron router is not present to provide the metadata IP

# but the DHCP server can be used to provide this info. Setting this value will

# force the DHCP server to append specific host routes to the DHCP request. If

# this option is set, then the metadata service will be activated for all the

# networks. (boolean value)

#该选项配置后，虚拟机的会增加一条路由到dhcp这个namespace地址，来上报metadata信息。

force\_metadata = true

# dns服务器地址

dnsmasq\_dns\_servers = 8.8.8.8,8.8.4.4

## 其他重要配置项

【default】

# The driver used to manage the virtual interface. (string value)

#openvswitch 或linuxbridge都可以，这里在外层的neutron.conf文件中已经配置成了openvswitch了。

interface\_driver = <None>

# The driver used to manage the DHCP server. (string value)

#dhcp驱动，默认neutron.agent.linux.dhcp.Dnsmasq。

dhcp\_driver = neutron.agent.linux.dhcp.Dnsmasq

# dnsmasq.conf

网络节点neutron\_dhcp\_agent容器的配置

#只有一行配置，记录了Log

log-facility=/var/log/kolla/neutron/dnsmasq.log

# l3\_agent.ini

网络节点neutron\_l3\_agent容器的配置。

Kolla 有部署分布式路由的配置开关。globals.yml文件的enable\_neutron\_dvr选项。

#只有一行配置

[DEFAULT]

# The working mode for the agent. Allowed modes are: 'legacy' - this preserves

# the existing behavior where the L3 agent is deployed on a centralized

# networking node to provide L3 services like DNAT, and SNAT. Use this mode if

# you do not want to adopt DVR. 'dvr' - this mode enables DVR functionality and

# must be used for an L3 agent that runs on a compute host. 'dvr\_snat' - this

# enables centralized SNAT support in conjunction with DVR. This mode must be

# used for an L3 agent running on a centralized node (or in single-host# deployments, e.g. devstack) (string value)

# Allowed values: dvr, dvr\_snat, legacy

#配置了过期的模式，只的是集中式虚拟路由模式，流量都会到网络节点汇聚。还有其他两

#种模式，dvr和dvr\_snat。

#dvr指分布式路由，在计算节点也要部署l3 agent服务，适合在计算节点上配置。 dvr\_snat在dvr的基础上支持集中#的snat，适合在网络节点上配置。

agent\_mode = legacy

## 其他重要配置项

【DEFAULT】

# metadata 数据的监听端口，用于neutron-ns-metadata-proxy服务。参考https://www.ibm.com/developerworks/cn/cloud/library/1509\_liukg\_openstackmeta/index.html

端口最大值：65535

metadata\_port = 9697

# Allow running metadata proxy. (boolean value)。neutron-ns-metadata-proxy的开关

enable\_metadata\_proxy = true

# The driver used to manage the virtual interface. (string value)

#openvswitch 或linuxbridge都可以，这里在外层的neutron.conf文件中已经配置成了openvswitch了。

interface\_driver = <None>

# fwaas\_driver.ini（后续补充）

网络节点neutron\_l3\_agent容器的配置

# neutron\_lbaas.conf(后续补充)

# neutron\_vpnaas.conf(后续补充)

# 分布式路由场景下

相关配置修改：

1，neutron.conf文件

router\_distributed = True

2，l3\_agent.ini文件

网络节点

agent\_mode = dvr\_snat

计算节点

agent\_mode = dvr

3，m2\_conf.ini

[ml2]

enable\_distributed\_routing = True

[ovs]

bridge\_mappings = physnet1:br-ex