# kolla部署——horizon配置

kolla部署的容器场景，主要分控制节点和计算节点。控制节点有horzion（资源调度）。

# /etc/httpd/conf.d/horizon.conf

## Kolla 默认配置项

# 监听IP和端口

Listen 192.168.102.15:80

<VirtualHost \*:80>

LogLevel warn

ErrorLog /var/log/kolla/horizon/horizon.log

CustomLog /var/log/kolla/horizon/horizon-access.log combined

WSGIScriptReloading On

WSGIDaemonProcess horizon-http processes=5 threads=1 user=horizon group=horizon display-name=%{GROUP} python-path=/var/lib/kolla/venv/lib/python2.7/site-packages

WSGIProcessGroup horizon-http

# 配置django的wsgi配置文件

WSGIScriptAlias / /var/lib/kolla/venv/lib/python2.7/site-packages/openstack\_dashboard/wsgi/django.wsgi

WSGIPassAuthorization On

<Location "/">

Require all granted

</Location>

Alias /static /var/lib/kolla/venv/lib/python2.7/site-packages/static

<Location "/static">

SetHandler None

</Location>

</VirtualHost>

# NOTE(Jeffrey4l): Only enable deflate when tls is disabled until the

# OSSN-0037 is fixed.

# see https://wiki.openstack.org/wiki/OSSN/OSSN-0037 for more information.

<IfModule mod\_deflate.c>

# Compress HTML, CSS, JavaScript, Text, XML and fonts

AddOutputFilterByType DEFLATE application/javascript

AddOutputFilterByType DEFLATE application/rss+xml

AddOutputFilterByType DEFLATE application/vnd.ms-fontobject

AddOutputFilterByType DEFLATE application/x-font

AddOutputFilterByType DEFLATE application/x-font-opentype

AddOutputFilterByType DEFLATE application/x-font-otf

AddOutputFilterByType DEFLATE application/x-font-truetype

AddOutputFilterByType DEFLATE application/x-font-ttf

AddOutputFilterByType DEFLATE application/x-javascript

AddOutputFilterByType DEFLATE application/xhtml+xml

AddOutputFilterByType DEFLATE application/xml

AddOutputFilterByType DEFLATE font/opentype

AddOutputFilterByType DEFLATE font/otf

AddOutputFilterByType DEFLATE font/ttf

AddOutputFilterByType DEFLATE image/svg+xml

AddOutputFilterByType DEFLATE image/x-icon

AddOutputFilterByType DEFLATE text/css

AddOutputFilterByType DEFLATE text/html

AddOutputFilterByType DEFLATE text/javascript

AddOutputFilterByType DEFLATE text/plain

AddOutputFilterByType DEFLATE text/xml

</IfModule>

<IfModule mod\_expires.c>

<Filesmatch "\.(jpg|jpeg|png|gif|js|css|swf|ico|woff)$">

ExpiresActive on

ExpiresDefault "access plus 1 month"

ExpiresByType application/javascript "access plus 1 year"

ExpiresByType text/css "access plus 1 year"

ExpiresByType image/x-ico "access plus 1 year"

ExpiresByType image/jpg "access plus 1 year"

ExpiresByType image/jpeg "access plus 1 year"

ExpiresByType image/gif "access plus 1 year"

ExpiresByType image/png "access plus 1 year"

Header merge Cache-Control public

Header unset ETag

</Filesmatch>

</IfModule>

/var/lib/kolla/venv/lib/python2.7/site-packages/openstack\_dashboard/wsgi/django.wsgi

代码内容如下

import logging

import os

import sys

from django.core.wsgi import get\_wsgi\_application

from django.conf import settings

# Add this file path to sys.path in order to import settings

sys.path.insert(0, os.path.normpath(os.path.join(os.path.dirname(os.path.realpath(\_\_file\_\_)), '../..')))

#调用openstack\_dashboard.settings环境变量的设置

os.environ['DJANGO\_SETTINGS\_MODULE'] = 'openstack\_dashboard.settings'

sys.stdout = sys.stderr

DEBUG = False

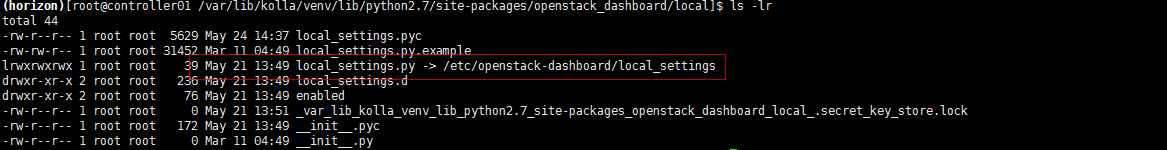
application = get\_wsgi\_application()

/var/lib/kolla/venv/lib/python2.7/site-packages/openstack\_dashboard/settings.py

这个文件中的setting中会去引入local\_settings这个py类

from local.local\_settings import \*

查看发现，local\_settings.py是/etc/openstack-dashboard/local\_settings的软连接。这个文件是主要的配置文件。



# /etc/openstack-dashboard/local\_settings

Kolla默认配置

import os

from django.utils.translation import ugettext\_lazy as \_

from openstack\_dashboard import exceptions

from openstack\_dashboard.settings import HORIZON\_CONFIG

DEBUG = False

TEMPLATE\_DEBUG = DEBUG

COMPRESS\_OFFLINE = True

#webroot的相对路径

WEBROOT = '/'

#允许被连接的所有hosts

ALLOWED\_HOSTS = ['\*']

#OpenStack的API版本

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

}

LOCAL\_PATH = '/tmp'

SECRET\_KEY='BeCFgYgL5lKYi1xFF77kx5YhB2wK9HiZSI78WTxJ'

SESSION\_ENGINE = 'django.contrib.sessions.backends.cache'

# 配置cache

CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION': ['192.168.102.15:11211'] }

}

EMAIL\_BACKEND = 'django.core.mail.backends.console.EmailBackend'

# host地址

OPENSTACK\_HOST = "192.168.102.47"

#keystone的url

OPENSTACK\_KEYSTONE\_URL = "http://192.168.102.47:5000/v3"

#keystone默认的角色

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "\_member\_"

OPENSTACK\_KEYSTONE\_BACKEND = {

'name': 'native',

'can\_edit\_user': True,

'can\_edit\_group': True,

'can\_edit\_project': True,

'can\_edit\_domain': True,

'can\_edit\_role': True,

}

OPENSTACK\_HYPERVISOR\_FEATURES = {

'can\_set\_mount\_point': False,

'can\_set\_password': False,

'requires\_keypair': False,

'enable\_quotas': True

}

#启用cinder的backup功能

OPENSTACK\_CINDER\_FEATURES = {

'enable\_backup': True,

}

#启动neutron的一些特性

OPENSTACK\_NEUTRON\_NETWORK = {

'enable\_router': True,

'enable\_quotas': True,

'enable\_ipv6': True,

'enable\_distributed\_router': False,

'enable\_ha\_router': False,

'enable\_lb': True,

'enable\_firewall': True,

'enable\_vpn': True,

'enable\_fip\_topology\_check': True,

# Default dns servers you would like to use when a subnet is

# created. This is only a default, users can still choose a different

# list of dns servers when creating a new subnet.

# The entries below are examples only, and are not appropriate for

# real deployments

# 'default\_dns\_nameservers': ["8.8.8.8", "8.8.4.4", "208.67.222.222"],

# The profile\_support option is used to detect if an external router can be

# configured via the dashboard. When using specific plugins the

# profile\_support can be turned on if needed.

'profile\_support': None,

#'profile\_support': 'cisco',

# Set which provider network types are supported. Only the network types

# in this list will be available to choose from when creating a network.

# Network types include local, flat, vlan, gre, vxlan and geneve.

# 'supported\_provider\_types': ['\*'],

# You can configure available segmentation ID range per network type

# in your deployment.

# 'segmentation\_id\_range': {

# 'vlan': [1024, 2048],

# 'vxlan': [4094, 65536],

# },

# You can define additional provider network types here.

# 'extra\_provider\_types': {

# 'awesome\_type': {

# 'display\_name': 'Awesome New Type',

# 'require\_physical\_network': False,

# 'require\_segmentation\_id': True,

# }

# },

# Set which VNIC types are supported for port binding. Only the VNIC

# types in this list will be available to choose from when creating a

# port.

# VNIC types include 'normal', 'macvtap' and 'direct'.

# Set to empty list or None to disable VNIC type selection.

'supported\_vnic\_types': ['\*'],

}

#启用Heat的stack，可用密码

OPENSTACK\_HEAT\_STACK = {

'enable\_user\_pass': True,

}

IMAGE\_CUSTOM\_PROPERTY\_TITLES = {

"architecture": \_("Architecture"),

"kernel\_id": \_("Kernel ID"),

"ramdisk\_id": \_("Ramdisk ID"),

"image\_state": \_("Euca2ools state"),

"project\_id": \_("Project ID"),

"image\_type": \_("Image Type"),

}

IMAGE\_RESERVED\_CUSTOM\_PROPERTIES = []

# 内部url的endpoint

OPENSTACK\_ENDPOINT\_TYPE = "internalURL"

API\_RESULT\_LIMIT = 1000

API\_RESULT\_PAGE\_SIZE = 20

SWIFT\_FILE\_TRANSFER\_CHUNK\_SIZE = 512 \* 1024

DROPDOWN\_MAX\_ITEMS = 30

TIME\_ZONE = "UTC"

# policy相关的文件路径

POLICY\_FILES\_PATH = '/etc/openstack-dashboard'

LOGGING = {

'version': 1,

# When set to True this will disable all logging except

# for loggers specified in this configuration dictionary. Note that

# if nothing is specified here and disable\_existing\_loggers is True,

# django.db.backends will still log unless it is disabled explicitly.

'disable\_existing\_loggers': False,

'formatters': {

'operation': {

# The format of "%(message)s" is defined by

# OPERATION\_LOG\_OPTIONS['format']

'format': '%(asctime)s %(message)s'

},

},

'handlers': {

'null': {

'level': 'DEBUG',

'class': 'logging.NullHandler',

},

'console': {

# Set the level to "DEBUG" for verbose output logging.

'level': 'INFO',

'class': 'logging.StreamHandler',

},

'operation': {

'level': 'INFO',

'class': 'logging.StreamHandler',

'formatter': 'operation',

},

},

'loggers': {

# Logging from django.db.backends is VERY verbose, send to null

# by default.

'django.db.backends': {

'handlers': ['null'],

'propagate': False,

},

'requests': {

'handlers': ['null'],

'propagate': False,

},

'horizon': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'horizon.operation\_log': {

'handlers': ['operation'],

'level': 'INFO',

'propagate': False,

},

'openstack\_dashboard': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'novaclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'cinderclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'keystoneclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'glanceclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'neutronclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'heatclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'ceilometerclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'swiftclient': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'openstack\_auth': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'nose.plugins.manager': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'django': {

'handlers': ['console'],

'level': 'DEBUG',

'propagate': False,

},

'iso8601': {

'handlers': ['null'],

'propagate': False,

},

'scss': {

'handlers': ['null'],

'propagate': False,

},

},

}

SECURITY\_GROUP\_RULES = {

'all\_tcp': {

'name': \_('All TCP'),

'ip\_protocol': 'tcp',

'from\_port': '1',

'to\_port': '65535',

},

'all\_udp': {

'name': \_('All UDP'),

'ip\_protocol': 'udp',

'from\_port': '1',

'to\_port': '65535',

},

'all\_icmp': {

'name': \_('All ICMP'),

'ip\_protocol': 'icmp',

'from\_port': '-1',

'to\_port': '-1',

},

'ssh': {

'name': 'SSH',

'ip\_protocol': 'tcp',

'from\_port': '22',

'to\_port': '22',

},

'smtp': {

'name': 'SMTP',

'ip\_protocol': 'tcp',

'from\_port': '25',

'to\_port': '25',

},

'dns': {

'name': 'DNS',

'ip\_protocol': 'tcp',

'from\_port': '53',

'to\_port': '53',

},

'http': {

'name': 'HTTP',

'ip\_protocol': 'tcp',

'from\_port': '80',

'to\_port': '80',

},

'pop3': {

'name': 'POP3',

'ip\_protocol': 'tcp',

'from\_port': '110',

'to\_port': '110',

},

'imap': {

'name': 'IMAP',

'ip\_protocol': 'tcp',

'from\_port': '143',

'to\_port': '143',

},

'ldap': {

'name': 'LDAP',

'ip\_protocol': 'tcp',

'from\_port': '389',

'to\_port': '389',

},

'https': {

'name': 'HTTPS',

'ip\_protocol': 'tcp',

'from\_port': '443',

'to\_port': '443',

},

'smtps': {

'name': 'SMTPS',

'ip\_protocol': 'tcp',

'from\_port': '465',

'to\_port': '465',

},

'imaps': {

'name': 'IMAPS',

'ip\_protocol': 'tcp',

'from\_port': '993',

'to\_port': '993',

},

'pop3s': {

'name': 'POP3S',

'ip\_protocol': 'tcp',

'from\_port': '995',

'to\_port': '995',

},

'ms\_sql': {

'name': 'MS SQL',

'ip\_protocol': 'tcp',

'from\_port': '1433',

'to\_port': '1433',

},

'mysql': {

'name': 'MYSQL',

'ip\_protocol': 'tcp',

'from\_port': '3306',

'to\_port': '3306',

},

'rdp': {

'name': 'RDP',

'ip\_protocol': 'tcp',

'from\_port': '3389',

'to\_port': '3389',

},

}

REST\_API\_REQUIRED\_SETTINGS = ['OPENSTACK\_HYPERVISOR\_FEATURES',

'LAUNCH\_INSTANCE\_DEFAULTS',

'OPENSTACK\_IMAGE\_FORMATS']