Installation procedure

Release brahmaputra.1.0 (33355d5)

OPNFV

August 15, 2016

CONTENTS

CHAPTER ONE

COLORADO 1.0

This document will give the user instructions on how to deploy available scenarios verified for the Colorado release of OPNFV platform.

1.1 Multisite.Kingbird installation instruction

1.1.1 Preparing the installation

Kingbird is centralized synchronization service for multi-region OpenStack deployments. In OPNFV Colorado release, Kingbird provides centralized quota management feature. At least two OpenStack regions with shared KeyStone should be installed first.

Kingbird includes kingbird-api and kingbird-engine, kingbird-api and kingbird-engine which talk to each other through message bus, and both services access the database. Kingbird-api receives the RESTful API request for quota management and forward the request to kingbird-engine to do quota synchronization etc task.

Therefore install Kingbird on the controller nodes of one of the OpenStack region, these two services could be deployed in same node or different node. Both kingbird-api and kingbird-engine can run in multiple nodes with multi-workers mode. It's up to you how many nodes you want to deploy kingbird-api and kingbird-engine and they can work in same node or different nodes.

1.1.2 HW requirements

No special hardware requirements

1.1.3 Installation instruction

In colorado release, Kingbird is recommended to be installed in a python virtual environment. So install and activate virtualenv first.

```
sudo pip install virtualenv
virtualenv venv
source venv/bin/activate
```

Get the latest code of Kingbird from git repository:

```
git clone https://github.com/openstack/kingbird.git
cd kingbird/
pip install -e .
```

or get the stable release from PyPI repository:

```
pip install kingbird
```

In case of the database package are not installed, you may need to install:

```
pip install mysql
pip install pymysql
```

In the Kingbird root folder, where you can find the source code of Kingbird, generate the configuration sample file for Kingbird:

```
oslo-config-generator --config-file=./tools/config-generator.conf
```

prepare the folder used for cache, log and configuration for Kingbird:

```
sudo rm -rf /var/cache/kingbird
sudo mkdir -p /var/cache/kingbird
sudo chown `whoami` /var/cache/kingbird
sudo rm -rf /var/log/kingbird
sudo mkdir -p /var/log/kingbird
sudo chown `whoami` /var/log/kingbird
sudo rm -rf /etc/kingbird
sudo mkdir -p /etc/kingbird
sudo chown `whoami` /etc/kingbird
```

Copy the sample configuration to the configuration folder /etc/kingbird:

```
cp etc/kingbird/kingbird.conf.sample /etc/kingbird/kingbird.conf
```

Before editing the configuration file, prepare the database info for Kingbird.

```
mysql -uroot -e "CREATE DATABASE $kb_db CHARACTER SET utf8;"
mysql -uroot -e "GRANT ALL PRIVILEGES ON $kb_db.* TO '$kb_db_user'@'%' IDENTIFIED BY '$kb_db_pwd';"
```

For example, the following command will create database "kingbird", and grant the privilege for the db user "kingbird" with password "password":

```
mysql -uroot -e "CREATE DATABASE kingbird CHARACTER SET utf8;"
mysql -uroot -e "GRANT ALL PRIVILEGES ON kingbird.* TO 'kingbird'@'%' IDENTIFIED BY 'password';"
```

Create the service user in OpenStack:

```
source openrc admin admin
openstack user create --project=service --password=$kb_svc_pwd $kb_svc_user
openstack role add --user=$kb_svc_user --project=service admin
```

For example, the following command will create service user "kingbird", and grant the user "kingbird" with password "password" the role of admin in service project:

```
source openrc admin admin
openstack user create --project=service --password=password kingbird
openstack role add --user=kingbird --project=service admin
```

Then edit the configuration file for Kingbird:

```
vim /etc/kingbird/kingbird.conf
```

By default, the bind_host of kingbird-api is local_host(127.0.0.1), and the port for the service is 8118, you can leave it as the default if no port conflict happened.

To make the Kingbird work normally, you have to edit these configuration items. The [cache] section is used by kingbird engine to access the quota information of Nova, Cinder, Neutron in each region, replace the auth_uri to the keystone service in your environment, especially if the keystone service is not located in the same node, and also for the account to access the Nova, Cinder, Neutron in each region, in the following configuration, user “admin” with password “password” of the tenant “admin” is configured to access other Nova, Cinder, Neutron in each region:

```
[cache]
auth_uri = http://127.0.0.1:5000/v3
admin_tenant = admin
admin_password = password
admin_username = admin
```

Configure the database section with the service user “kingbird” and its password, to access database “kingbird”. For detailed database section configuration, please refer to http://docs.openstack.org/developer/oslo.db/opts.html, and change the following configuration accordingly based on your environment.

```
[database]
connection = mysql+pymysql://$kb_db_user:$kb_db_pwd@127.0.0.1/$kb_db?charset=utf8
```

For example, if the database is “kingbird”, and the db user “kingbird” with password “password”, then the configuration is as following:

```
[database]
connection = mysql+pymysql://kingbird:password@127.0.0.1/kingbird?charset=utf8
```

The [keystone_authtoken] section is used by keystonemiddleware for token validation during the API request to the kingbird-api, please refer to http://docs.openstack.org/developer/keystonemiddleware/middlewarearchitecture.html on how to configure the keystone_authtoken section for the keystonemiddleware in detail, and change the following configuration accordingly based on your environment:

please specify the region_name where you want the token will be validated if the KeyStone is deployed in multiple regions

```
[keystone_authtoken]
signing_dir = /var/cache/kingbird
cafile = /opt/stack/data/ca-bundle.pem
auth_uri = http://127.0.0.1:5000/v3
project_domain_name = Default
project_name = service
user_domain_name = Default
password = $kb_svc_pwd
username = $kb_svc_user
auth_url = http://127.0.0.1:35357/v3
auth_type = password
region_name = RegionOne
```

For example, if the service user is “kingbird, and the password for the user is “password”, then the configuration will look like this:

```
[keystone_authtoken]
signing_dir = /var/cache/kingbird
cafile = /opt/stack/data/ca-bundle.pem
auth_uri = http://127.0.0.1:5000/v3
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = kingbird
auth_url = http://127.0.0.1:35357/v3
auth_type = password
region_name = RegionOne
```

And also configure the message bus connection, you can refer to the message bus configuration in Nova, Cinder, Neutron configuration file.

```
[DEFAULT]
rpc_backend = rabbit
control_exchange = openstack
transport_url = None

[oslo_messaging_rabbit]
rabbit_host = 127.0.0.1
rabbit_port = 5671
rabbit_userid = guest
rabbit_password = guest
rabbit_virtual_host = /
```

After these basic configuration items configured, now the database schema of “kingbird” should be created:

```
python kingbird/cmd/manage.py --config-file=/etc/kingbird/kingbird.conf db_sync
```

And create the service and endpoint for Kingbird, please change the endpoint url according to your cloud planning:

```
openstack service create --name=kingbird synchronization
openstack endpoint create --region=RegionOne \
--publicurl=http://127.0.0.1:8118/v1.0 \
--adminurl=http://127.0.0.1:8118/v1.0 \
--internalurl=http://127.0.0.1:8118/v1.0 kingbird
```

Now it’s ready to run kingbird-api and kingbird-engine:

```
nohup python kingbird/cmd/api.py --config-file=/etc/kingbird/kingbird.conf &
nohup python kingbird/cmd/engine.py --config-file=/etc/kingbird/kingbird.conf &
```

Run the following command to check whether kingbird-api and kingbird-engine are running:

```
ps aux|grep python
```

1.1.4 Post-installation activities

Run the following commands to check whether kingbird-api is serving, please replace $token to the token you get from “openstack token issue”:

```
openstack token issue
curl  -H "Content-Type: application/json"  -H "X-Auth-Token: $token" \
http://127.0.0.1:8118/
```

If the response looks like following: {“versions”: [{“status”: “CURRENT”, “updated”: “2016-03-07”, “id”: “v1.0”, “links”: [{“href”: “http://127.0.0.1:8118/v1.0/”, “rel”: “self”}]}]}, then that means the kingbird-api is working normally.

Run the following commands to check whether kingbird-engine is serving, please replace $token to the token you get from “openstack token issue”, and the $admin_project_id to the admin project id in your environment:

```
curl  -H "Content-Type: application/json"  -H "X-Auth-Token: $token" \
-H  "X_ROLE: admin"  -X PUT \
http://127.0.0.1:8118/v1.0/$admin_project_id/os-quota-sets/$admin_project_id/sync
```

If the response looks like following: “triggered quota sync for 0320065092b14f388af54c5bd18ab5da”, then that means the kingbird-engine is working normally.