studioidefix

here be dragons

Home
About
Archive
Devlog
GitHub
Currently v1.0.7

© 2016. All rights reserved.

Written by Anish Bhatt

Multi Node devstack setup

12 Jun 2015

> Note : A detailed guide to a Multi-Node devstack setup can be found in openstack documentation, this guide simply elaborates on the node configuration.

This guide covers setting up a separate controller, cinder and compute node via devstack. Configuration of each node is addressed individually.

Controller Node

The controller node must be brought up first as the cinder/compute node rely on it for services. You would bring up devstack regularly using the following configuration :

```
# Controller configuration
[[local|localrc]]

SERVICE_TOKEN=azertytoken
ADMIN_PASSWORD=foobar
MYSQL_PASSWORD=foobar
SERVICE_PASSWORD=foobar
DATABASE_PASSWORD=foobar
RABBIT_PASSWORD=foobar

HOST_IP=10.192.194.1
PUBLIC_INTERFACE=ens0

MULTI_HOST=True
```

```
disable_service c-vol
disable_service n-cpu
disable_service n-net
```

We disable `c-vol` , `n-cpu` & `n-net` because we do not want to run the cinder volume, nova compute or nova network on the controller node, as we have separate nodes for this. Setting `MULTI_HOST=True` is required for a multi node setup.

Once devstack has been launched, remember to allow inbound amqp & mysql connections via the following iptables rules :

```
# keystone
$ sudo iptables -I INPUT 1 -p tcp --dport 5000 -j ACCEPT

# amqp
$ sudo iptables -I INPUT 1 -p tcp --dport 5672 -j ACCEPT
$ sudo iptables -I INPUT 1 -p tcp --dport 5673 -j ACCEPT
$ sudo iptables -I INPUT 1 -p tcp --dport 15672 -j ACCEPT

# mysql
$ sudo iptables -I INPUT 1 -p tcp --dport 3306 -j ACCEPT

# glance
$ sudo iptables -I INPUT 1 -p tcp --dport 9292 -j ACCEPT

# cinder api
$ sudo iptables -I INPUT 1 -p tcp --dport 8776 -j ACCEPT
```

While the cinder and compute node can be brought up simulatenously, ensure that the controller node is completely up and running before bringing up the other nodes.

Cinder Node

Next we bring up the cinder node. The configuration here is slightly more complicated.

```
# Cinder configuration
[[local|localrc]]

SERVICE_TOKEN=azertytoken
ADMIN_PASSWORD=foobar
MYSQL_PASSWORD=foobar
SERVICE_PASSWORD=foobar
DATABASE_PASSWORD=foobar
RABBIT_PASSWORD=foobar

SERVICE_HOST=10.192.194.1 # controller ip
HOST_IP=10.192.194.2
PUBLIC_INTERFACE=ens0
```

```
MULTI_HOST=True

MYSQL_HOST=$SERVICE_HOST
RABBIT_HOST=$SERVICE_HOST
CINDER_SERVICE_HOST=$SERVICE_HOST
GLANCE_HOSTPORT=$SERVICE_HOST:9292
DATABASE_TYPE=mysql

ENABLED_SERVICES=c-sch,c-vol

CINDER_ENABLED_BACKENDS=lvm:lvmdriver-1

[[post-config|$CINDER_CONF]]
[DEFAULT]
my_ip=$HOST_IP

[lvmdriver-1]
VOLUME_GROUP="stack-volumes"
VOLUME_NAME_PREFIX="volume-"
```

We only enable the required services, `c-sch` and `c-vol` here as this will be a cinder only node. Note the difference in format while disabling individual services and enabling an array.

We also add a post config section to cinder to make some changes required for a multi node setup. The `my\_ip` value, used to indicate iscsi target address defaults to `SERVICE\_HOST` which normally indicates the devstack host ip. For a multi-node setup we need to change it to `HOST\_IP` . We also declare an lvm backend with a backend name (lvmdriver-1 in this case). When the cinder service with an lvm backend is started, it creates a new volume group named in the format `\<VOLUME\_GROUP\>-\<backend_name\>` . By specifying the `VOLUME\_GROUP` under the named lvm backend configuration section, we can control the name to be `stack-volumes-lvmdriver-1` . By creating this volume group before devstack is brought up, when can make devstack use our preconfigured VG.

Once devstack has been launched, remember to allow inbound iscsi connections via the following iptables rules :

```
# iscsi
$ sudo iptables -I INPUT 1 -p tcp --dport 3260 -j ACCEPT
```

Compute Node

For the compute node, the configuration is much simpler.

```
# Compute configuration
[[local|localrc]]

SERVICE_TOKEN=azertytoken
ADMIN_PASSWORD=foobar
MYSQL_PASSWORD=foobar
```

```
SERVICE_PASSWORD=foobar
DATABASE_PASSWORD=foobar
RABBIT_PASSWORD=foobar

SERVICE_HOST=10.192.194.1 # controller ip
HOST_IP=10.192.194.3
PUBLIC_INTERFACE=ens0

MULTI_HOST=True

MYSQL_HOST=$SERVICE_HOST
RABBIT_HOST=$SERVICE_HOST
CINDER_SERVICE_HOST=$SERVICE_HOST
GLANCE_HOSTPORT=$SERVICE_HOST:9292
DATABASE_TYPE=mysql

ENABLED_SERVICES=n-cpu,n-net
```

For the compute node, we similarly enable only the required services. Once you have all three nodes up and running, you should have a fully working multi-node setup deployed. Happy stacking !

> Note : devstack code changes quite rapidly and might be sometimes unstable. This instructions were verified working at the time of the Kilo Release, but YMMV.

0 Comments **studioidefix** 1 Login

Recommend Share Sort by Best

Start the discussion…

Be the first to comment.

ALSO ON **STUDIOIDEFIX**

Replying to a thread on lkml via git send-email

1 comment • 2 years ago•

saurabh sengar — I am not able to see the Message Id as it is coming empty for all the threads I am seeing

Compiling ergodox firmware

3 comments • 2 years ago•

putin bush — How do you create the picture of your layout using the py file that is in

Solving dnsmasq/vpn resolv.conf contention with openresolv

2 comments • 2 years ago•

Anish Bhatt — Sorry for the late reply Brian, yes. That's exactly what you do

Building a DCS keycap set for an Ergodox

2 comments • 2 years ago•

Anish Bhatt — They used to be sold as singles by Signature Plastics, unfortunately they are no longer

Subscribe Add Disqus to your site Add Disqus Add Privacy

Related Posts

- Connect to slack via irc/irssi 31 Jul 2015
- Copying large blocks of text between different vim instances 03 Nov 2014
- Migrate from RHEL7 to CentOS7 repos 25 Oct 2014