

Creating a Multi-node Cassandra Cluster on Centos 6.5.

The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. Linear scalability and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's support for replicating across multiple datacenters is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive regional outages.

Table of Contents

- Initial Server Setup
 - Hardware Information
 - Setting Host for `cassandra`
 - Updating `hostname` on all servers.
 - Creating `cassandra` user with `sudo` permissions.
 - Creating passwordless entry from SEED (`CASSANDRA01`) to other servers.
- Extracting Files.
- Updating Configuration File.
 - Setting `initial_token` as below.
 - On Node `CASSANDRA01`
 - On Node `CASSANDRA02`
 - On Node `CASSANDRA03`
- Starting `cassandra`.
 - Checking Cluster Information.
 - Logging into CQL Shell.
 - Data Location on `CASSANDRA01`, `CASSANDRA02`, `CASSANDRA03`
- Performace Tuning.
 - Updating `cassandra.yaml` file.
 - Updating `cassandra-env.sh` file.
 - Updating `cassandra-topology.properties` file.
- Installing OpsCenter Monitoring for Cassandra.
 - Download `opscenter` Archive.
 - Extracting `opscenter`.
 - Configure `opscenter`
 - Configuring Agent.
 - Starting opsCenter.
 - Starting Agent Manually.
- Useful Links

This is a basic multi-node cassandra setup.

Initial Server Setup

Hardware Information

All the server were with below configuration.

CPU : 40 Cores
RAM : 192GB

Setting Host for cassandra

Setting up the servers and update `/etc/hosts` as below.

```
#Adding CASSANDRA NODES
10.130.18.35    CASSANDRA01    #SEED
10.130.18.93    CASSANDRA02    #Worker
10.130.18.98    CASSANDRA03    #Worker
```

Updating hostname on all servers.

Update `hostnames` as required.

```
sudo vim /etc/sysconfig/network
```

Update `hostname` as below, do the same in all servers [CASSANDRA01, CASSANDRA02,CASSANDRA03].

```
NETWORKING=yes
HOSTNAME=CASSANDRA01
```

To update the `hostname` without a reboot execute below command.

```
sudo hostname CASSANDRA01
```

NOTE : `hostname` command will keep the `hostname` till the next reboot. So its required that we update `/etc/sysconfig/network` file.

Creating cassandra user with sudo permissions.

Have a script which will create a user on server.

```
wget https://raw.githubusercontent.com/zubayr/create_user_script/master/create_user_script.sh
sh create_user_script.sh -s cassandra
```

This will create a `cassandra` user, with `sudo` permissions.

Creating passwordless entry from SEED (CASSANDRA01) to other servers.

Create a rsa key on CASSANDRA01

```
ssh-keygen -t rsa
```

Create .ssh directory on other 2 servers.

```
ssh cassandra@CASSANDRA02 mkdir -p .ssh
ssh cassandra@CASSANDRA03 mkdir -p .ssh
```

Add the id_rsa.pub to authorized_keys

```
cat ~/.ssh/id_rsa.pub | ssh cassandra@CASSANDRA02 'cat >> .ssh/authorized_keys'
cat ~/.ssh/id_rsa.pub | ssh cassandra@CASSANDRA03 'cat >> .ssh/authorized_keys'
```

Make sure we have the right permissions.

```
ssh cassandra@CASSANDRA02 chmod 744 -R .ssh
ssh cassandra@CASSANDRA03 chmod 744 -R .ssh
```

Testing.

```
ssh cassandra@CASSANDRA02
ssh cassandra@CASSANDRA03
```

Extracting Files.

Extracting Files to opt and creating a link.

```
sudo tar xvzf apache-cassandra-2.1.3-bin.tar.gz -C /opt
sudo ln -s /opt/apache-cassandra-2.1.3 /opt/cassandra
sudo chown cassandra:cassandra -R /opt/cassandra
sudo chown cassandra:cassandra -R /opt/apache-cassandra-2.1.3
```

Creating Required Directories.

```
sudo mkdir -p /data1/cassandra/commitlog
sudo mkdir -p /data1/cassandra/data
sudo mkdir -p /data1/cassandra/saved_cahes
```

Updating Configuration File.

Setting initial_token as below.

Node 0: 0 Node 1: 3074457345618258602 Node 2: 6148914691236517205

On Node CASSANDRA01

```
cluster_name: 'MyCassandraCluster'  
initial_token: 0  
seed_provider:  
  - class_name: org.apache.cassandra.locator.SimpleSeedProvider  
    parameters:  
      - seeds: "10.130.18.35"  
listen_address: 10.130.18.35  
endpoint_snitch: SimpleSnitch  
  
data_file_directories:  
  - /data1/cassandra/data  
  
commitlog_directory: /data1/cassandra/commitlog  
saved_caches_directory: /data1/cassandra/saved_caches
```

On Node CASSANDRA02

```
cluster_name: 'MyCassandraCluster'  
initial_token: 3074457345618258602  
seed_provider:  
  - class_name: org.apache.cassandra.locator.SimpleSeedProvider  
    parameters:  
      - seeds: "10.130.18.35"  
listen_address: 10.130.18.93  
endpoint_snitch: SimpleSnitch  
  
data_file_directories:  
  - /data1/cassandra/data  
  
commitlog_directory: /data1/cassandra/commitlog  
saved_caches_directory: /data1/cassandra/saved_caches
```

On Node CASSANDRA03

```
cluster_name: 'MyCassandraCluster'  
initial_token: 6148914691236517205  
seed_provider:  
  - class_name: org.apache.cassandra.locator.SimpleSeedProvider  
    parameters:  
      - seeds: "10.130.18.35"  
listen_address: 10.130.18.98  
endpoint_snitch: SimpleSnitch  
  
data_file_directories:  
  - /data1/cassandra/data  
  
commitlog_directory: /data1/cassandra/commitlog  
saved_caches_directory: /data1/cassandra/saved_caches
```

Starting cassandra.

On Server CASSANDRA01.

```
sh /opt/cassandra/bin/cassandra
```

Wait till the server initialize and then start rest of nodes.

On Server CASSANDRA02.

```
sh /opt/cassandra/bin/cassandra
```

On Server CASSANDRA03.

```
sh /opt/cassandra/bin/cassandra
```

Checking Cluster Information.

```
[cassandra@CASSANDRA01 bin]$ ./nodetool status
```

```
Datacenter: datacenter1
```

```
=====
```

```
Status=Up/Down
```

```
||/ State=Normal/Leaving/Joining/Moving
```

--	Address	Load	Tokens	Owns (effective)	Host ID	Rack
UN	10.10.18.98	72.09 KB	1	33.3%	1a5a0c77-b5e6-4057-87b4-a8e788786244	rack1
UN	10.10.18.35	46.24 KB	1	83.3%	67de1b1f-8070-48c1-ad88-2c0d4dd7a988	rack1
UN	10.10.18.93	55.64 KB	1	83.3%	7fba7cd0-6f99-4ce8-8194-c9a8b23488cd	rack1

Logging into CQL Shell.

We need to export CQLSH_HOST

```
[cassandra@CASSANDRA01 bin]$ export CQLSH_HOST=10.10.18.35
```

```
[cassandra@CASSANDRA01 bin]$ cqlsh
```

```
Connected to CassandraJIOCluster at 10.10.18.35:9042.
```

```
[cqlsh 5.0.1 | Cassandra 2.1.3 | CQL spec 3.2.0 | Native protocol v3]
```

```
Use HELP for help.
```

```
cqlsh>
```

Data Location on CASSANDRA01, CASSANDRA02, CASSANDRA03

```
[cassandra@CASSANDRA01 bin]$ ls -l /data1/cassandra/
```

```
total 12
```

```
drwxr-xr-x 2 cassandra cassandra 4096 Mar 19 14:23 commitlog
```

```
drwxr-xr-x 4 cassandra cassandra 4096 Mar 19 14:23 data
```

```
drwxr-xr-x 2 cassandra cassandra 4096 Mar 19 13:18 saved_caches
```

```
[cassandra@CASSANDRA01 bin]$
```

Performace Tuning.

Updating cassandra.yaml file.

```
# For workloads with more data than can fit in memory, Cassandra's  
# bottleneck will be reads that need to fetch data from  
# disk. "concurrent_reads" should be set to (16 * number_of_drives) in  
# order to allow the operations to enqueue low enough in the stack
```

```

# that the OS and drives can reorder them. Same applies to
# "concurrent_counter_writes", since counter writes read the current
# values before incrementing and writing them back.
#
# On the other hand, since writes are almost never IO bound, the ideal
# number of "concurrent_writes" is dependent on the number of cores in
# your system; (8 * number_of_cores) is a good rule of thumb.

#concurrent_reads: 32
#concurrent_writes: 32

# Change as we had a 40core machine which calculates to 240.
concurrent_reads: 32
concurrent_writes: 240
concurrent_counter_writes: 32

```

Updating cassandra-env.sh file.

```

# Override these to set the amount of memory to allocate to the JVM at
# start-up. For production use you may wish to adjust this for your
# environment. MAX_HEAP_SIZE is the total amount of memory dedicated
# to the Java heap; HEAP_NEWSIZE refers to the size of the young
# generation. Both MAX_HEAP_SIZE and HEAP_NEWSIZE should be either set
# or not (if you set one, set the other).
#
# The main trade-off for the young generation is that the larger it
# is, the longer GC pause times will be. The shorter it is, the more
# expensive GC will be (usually).
#
# The example HEAP_NEWSIZE assumes a modern 8-core+ machine for decent pause
# times. If in doubt, and if you do not particularly want to tweak, go with
# 100 MB per physical CPU core.

# Important is the HEAP_NEWSIZE 100MB * number of Core (40 cores in our case)

#MAX_HEAP_SIZE="4G"
#HEAP_NEWSIZE="800M"
MAX_HEAP_SIZE="15G"
HEAP_NEWSIZE="4G"

```

Updating cassandra-topology.properties file.

If the server are in Data Center which in different location then we need to update this file as well. Also specify rack in that DC.

Cassandra

```
{{Node IP}}={{Data Center}}:{{Rack}}.
```

NOTE : This has to match with the cassandra-rackdc.properties file.

```

10.130.18.35=DC1:RAC1
10.130.18.93=DC2:RAC1
10.130.18.98=DC2:RAC2

```

When using this format we need to update `cassandra-rackdc.properties` and use `endpoint_snitch: as GossipingPropertyFileSnitch` in the `cassandra.yaml`

Installing OpsCenter Monitoring for Cassandra.

Setting up a opscenter for our cassandra cluster

Download opscenter Archive.

```
wget http://downloads.datastax.com/community/opscenter-5.0.tar.gz
```

Extracting opscenter.

Extracting, Create and Change owner.

```
sudo tar xvzf opscenter-5.0.2.tar.gz -C /opt/  
cd /opt/  
sudo ln -s opscenter-5.0.2 opscenter  
sudo chown cassandra:cassandra -R opscenter*
```

Configure opscenter

Update configuration file.

```
vim /opt/opscenter/conf/opscenterd.conf
```

Update the interface as below.

```
[webservice]  
port = 8888  
interface = 10.10.18.35
```

Configuring Agent.

Update the File below

```
vim /opt/opscenter/agent/conf/address.yaml
```

Add Below Line.

```
stomp_interface: "10.10.18.35"
```

Starting opsCenter.

`/opt/opscenter/bin/opscenter`

Open the browser with below URL.

`http://10.10.18.35:8888/opscenter/index.html`

1. In the UI Select, Manager Existing Cluster. (Manage an existing DataStax Enterprise or Cassandra cluster with OpsCenter.)
2. Add **Server** IPs as below. Our Cluster running on JMX 7199 port.

Newline is the Separator.

10.10.18.35

10.10.18.93

10.10.18.98

Starting Agent Manually.

Agent can be started from the `opscenter`. But if there is some issues then we can start it manually. (Make sure to update the `address.yaml` as above.)

`/opt/opscenter/agent/bin/datastax-agent`

Useful Links

[Digital Ocean](#)

[DataStax](#)

[RackSpace](#)

[DataStax](#)

[DataStax](#)

[Blog WhatIZee](#)