Initializing a Cassandra Cluster

Wednesday, October 22, 2014 12:51

1. **Initializing a multiple node cluster (single data center)**

   You can initialize a Cassandra cluster with a single data center.

   a. **Prerequisites**

   Each node must be correctly configured before starting the cluster. You must determine or perform the following before starting the cluster:

   - Install Cassandra on each node.
   - Choose a name for the cluster.
   - Get the IP address of each node.
   - Determine which nodes will be seed nodes. (Cassandra nodes use the seed node list for finding each other and learning the topology of the ring.)
   - Determine the snitch.
   - If using multiple data centers, determine a naming convention for each data center and rack, for example: DC1, DC2 or 100, 200 and RAC1, RAC2 or R101, R102. Choose the name carefully; renaming a data center is not possible.
   - Other possible configuration settings are described in The cassandra.yaml configuration file.

   This example describes installing a 6 node cluster spanning 2 racks in a single data center. Each node is configured to use the RackInferringSnitch (multiple rack aware) and 256 virtual nodes (vnodes). It is recommended to have more than one seed node per data center.

   In Cassandra, the term data center is a grouping of nodes. Data center is synonymous with replication group, that is, a grouping of nodes configured together for replication purposes.

   b. **Procedure**

   - Suppose you install Cassandra on these nodes:
     - □ node0 110.82.155.0 (seed1)
     - □ node1 110.82.155.1
     - □ node2 110.82.155.2
     - □ node3 110.82.156.3 (seed2)
     - □ node4 110.82.156.4
     - □ node5 110.82.156.5

     It is a best practice to have more than one seed node per data center.

   - If you have a firewall running on the nodes in your cluster, you must open certain ports to allow communication between the nodes. See Configuring firewall port access.

   - If Cassandra is running:

     ```
     Packaged installs:
     Stop Cassandra:
     $ sudo service cassandra stop

     Clear the data:
     $ sudo rm -rf /var/lib/cassandra/*

     Tarball installs:
     Stop Cassandra:
     $ ps auwx | grep cassandra
     $ sudo  kill pid

     Clear the data:
     $ cd install_location
     $ sudo rm -rf /var/lib/cassandra/*
     ```

   - Set the properties in the cassandra.yaml file for each node:

     Note: After making any changes in the cassandra.yaml file, you must restart the node for the changes to take effect.

     Location:
     - □ Packaged installs: /etc/cassandra/conf
     - □ Binary installs: install_location/conf

     Properties to set:

     Note: If the nodes in the cluster are identical in terms of disk layout, shared libraries, and so on, you can use the same copy of the cassandra.yaml file on all of them.
     - □ **num_token**s: recommended value: 256
     - □ -**seeds**: internal IP address of each seed node
     - □ **listen_address**: empty

       If not set, Cassandra asks the system for the local address, the one associated with its hostname. In some cases Cassandra doesn't produce the correct address and you must specify the listen_address.
     - □ **endpoint_snitch**: name of snitch (See endpoint_snitch.)
     - □ **auto_bootstrap**: false (Add this setting only when initializing a fresh cluster with no data.)

     Example:

     ```
     cluster_name: 'MyDemoCluster'
        num_tokens: 256
        seed_provider:
          - class_name: org.apache.cassandra.locator.SimpleSeedProvider
            parameters:
                 - seeds: "110.82.155.0,110.82.155.3"
        listen_address:
        rpc_address: 0.0.0.0
        endpoint_snitch: RackInferringSnitch
     ```

- After you have installed and configured Cassandra on all nodes, start the seed nodes one at a time, and then start the rest of the nodes.

  If the node has restarted because of automatic restart, you must stop the node and clear the data directories, as described above.

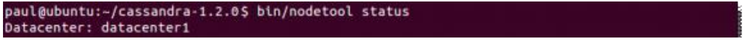
  For packaged installs, run the following command:

  ```
  $ sudo service cassandra start
  ```

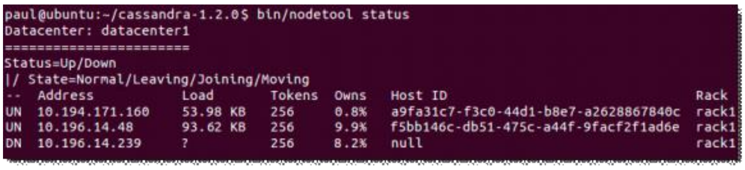
  For binary installs, run the following commands:

  ```
  $ cd install_location
  $ bin/cassandra
  ```

  To check that the ring is up and running, run the nodetool status command.

  ```
  paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
  Datacenter: datacenter1
  ```



```
paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens  Owns   Host ID                               Rack
UN  10.194.171.160   53.98 KB   256     0.8%   a9fa31c7-f3c0-44d1-b8e7-a2628867840c  rack1
UN  10.196.14.48     93.62 KB   256     9.9%   f5bb146c-db51-475c-a44f-9facf2f1ad6e  rack1
DN  10.196.14.239    ?          256     8.2%   null                                  rack1
```

2. **Initializing a multiple node cluster (multiple data centers)**
   You can initialize a Cassandra cluster with multiple data centers.

   Data replicates across the data centers automatically and transparently; no ETL work is necessary to move data between different systems or servers. You can configure the number of copies of the data in each data center and Cassandra handles the rest, replicating the data for you.

   In Cassandra, the term data center is a grouping of nodes. Data center is synonymous with replication group, that is, a grouping of nodes configured together for replication purposes.

   a. **Prerequisites**
      Each node must be correctly configured before starting the cluster. You must determine or perform the following before starting the cluster:
      - Install Cassandra on each node.
      - Choose a name for the cluster.
      - Get the IP address of each node.
      - Determine which nodes will be seed nodes. (Cassandra nodes use the seed node list for finding each other and learning the topology of the ring.)
      - Determine the snitch.
      - If using multiple data centers, determine a naming convention for each data center and rack, for example: DC1, DC2 or 100, 200 and RAC1, RAC2 or R101, R102. Choose the name carefully; renaming a data center is not possible.
      - Other possible configuration settings are described in The cassandra.yaml configuration file.

   b. **Procedure**
      - Suppose you install Cassandra on these nodes:
        - □ node0 10.168.66.41 (seed1)
        - □ node1 10.176.43.66
        - □ node2 10.168.247.41
        - □ node3 10.176.170.59 (seed2)
        - □ node4 10.169.61.170
        - □ node5 10.169.30.138

      - If you have a firewall running on the nodes in your cluster, you must open certain ports to allow communication between the nodes. See Configuring firewall port access.

      - If Cassandra is running:

        ```
        Packaged installs:
        Stop Cassandra:
        $ sudo service cassandra stop

        Clear the data:
        $ sudo rm -rf /var/lib/cassandra/*

        Tarball installs:
        Stop Cassandra:
        $ ps auwx | grep cassandra
        $ sudo  kill pid

        Clear the data:
        $ cd install_location
        $ sudo rm -rf /var/lib/cassandra/*
        ```

      - Set the properties in the cassandra.yaml file for each node:
        Note: After making any changes in the cassandra.yaml file, you must restart the node for the changes to take effect.
        Location:
        - □ Packaged installs: /etc/cassandra/conf
        - □ Binary installs: install_location/conf

        Properties to set:
        Note: If the nodes in the cluster are identical in terms of disk layout, shared libraries, and so on, you can use the same copy of the cassandra.yaml file on all of them.
        - □ **num_token**s: recommended value: 256
        - □ -**seeds**: internal IP address of each seed node
        - □ **listen_address**: empty
          If not set, Cassandra asks the system for the local address, the one associated with its hostname. In some cases Cassandra doesn't produce the correct address and you must specify the listen_address.
        - □ **endpoint_snitch**: name of snitch (See endpoint_snitch.)
        - □ **auto_bootstrap**: false (Add this setting only when initializing a fresh cluster with no data.)

        Example:

        ```
        cluster_name: 'MyDemoCluster'
        num_tokens: 256
        seed_provider:
          - class_name: org.apache.cassandra.locator.SimpleSeedProvider
            parameters:
                 - seeds:  "10.168.66.41,10.176.170.59"
        listen_address:
        endpoint_snitch: PropertyFileSnitch
        ```

   - In the cassandra-topology.properties file, assign the data center and rack names you determined in the Prerequisites to the IP addresses of each node. For example:

     ```
     # Cassandra Node IP=Data Center:Rack
     10.168.66.41=DC1:RAC1
     10.176.43.66=DC2:RAC1
     10.168.247.41=DC1:RAC1
     ```

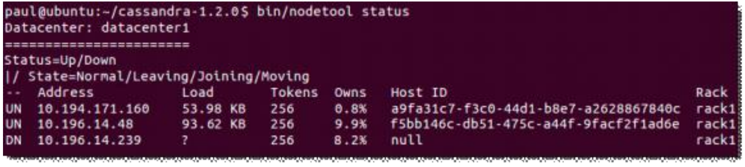
```
10.176.170.59=DC2:RAC1
10.169.61.170=DC1:RAC1
10.169.30.138=DC2:RAC1
```

- Also, in the cassandra-topologies.properties file, assign a default data center name and rack name for unknown nodes.

```
# default for unknown nodes
default=DC1:RAC1
```

- After you have installed and configured Cassandra on all nodes, start the seed nodes one at a time, and then start the rest of the nodes.
  If the node has restarted because of automatic restart, you must stop the node and clear the data directories, as described above.
  For packaged installs, run the following command:
  `$ sudo service cassandra start`
  For binary installs, run the following commands:
  `$ cd install_location`
  `$ bin/cassandra`

  To check that the ring is up and running, run the nodetool status command.



```
paul@ubuntu:~/cassandra-1.2.0$ bin/nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load       Tokens  Owns   Host ID                               Rack
UN  10.194.171.160   53.98 KB   256     0.8%   a9fa31c7-f3c0-44d1-b8e7-a2628867840c  rack1
UN  10.196.14.48     93.62 KB   256     9.9%   f5bb146c-db51-475c-a44f-9facf2f1ad6e  rack1
DN  10.196.14.239    ?          256     8.2%   null                                  rack1
```