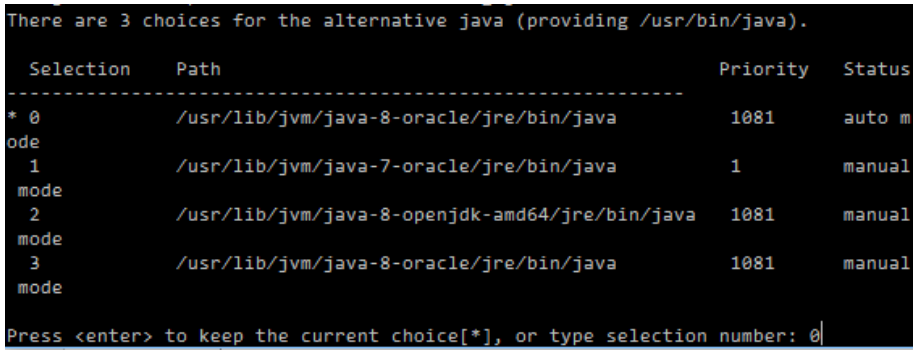
Cassandra Tutorial

Jinlai Xu

Prerequisites

- Java 8
  - sudo update-alternatives --config java
- Python 2.7 for CQL client
  - sudo apt-get install python2.7



Install Cassandra (on all nodes) 3.11

- echo "deb http://www.apache.org/dist/cassandra/debian 311x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list
- curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add -
- sudo apt-get update
- sudo apt-get install cassandra

Configure (on all nodes)

- sudo nano /etc/cassandra/cassandra.yaml
- Edit the lines below (red denotes the changes):
  - - seeds: “CC-demo-01, CC-demo-02, CC-demo-03” (on all nodes)
  - listen_address: CC-demo-1 (on master node)
  - listen_address: CC-demo-2 (on slave node 1)
  - listen_address: CC-demo-3 (on slave node 2)
  - …
  - rpc_address: CC-demo-1 (on master node)
  - rpc_address : CC-demo-2 (on slave node 1)
  - rpc_address : CC-demo-3 (on slave node 2)
  - …

Start the services (on all the nodes)

- Firstly, stop the Cassandra service:
  - sudo service cassandra stop
- Then, start the Cassandra process manually to monitor the service status and logs:
  - sudo cassandra -Rf

Start CQL client

- Initial a new terminal and log into one of the VM
- Use nodetool to see the Cassandra cluster's status
- nodetool status

```
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address         Load       Tokens       Owns (effective)  Host ID
                    Rack
DN  138.197.42.93   ?          256          100.0%            a646a4d2-feae-4daf
-b65f-8d9f8a701108  rack1
UN  162.243.40.202  167.27 KiB  256          100.0%            eaad9ec8-6a73-40d
9-83f2-d463dc49815b  rack1
```

- Start the CQL client
  - cqlsh CC-demo-1
  - OR cqlsh CC-demo-2 (because both nodes start the CQL service)

CQL test

- CREATE KEYSPACE patient WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 1};
- CREATE TABLE patient.exam (patient_id int, id int, date timeuuid, details text, PRIMARY KEY (patient_id, id));
- USE patient;
- INSERT INTO exam (patient_id,id,date,details) values (1,1,now(),'first exam patient 1');
- INSERT INTO exam (patient_id,id,date,details) values (1,2,now(),'second exam patient 1');
- INSERT INTO exam (patient_id,id,date,details) values (2,1,now(),'first exam patient 2');
- INSERT INTO exam (patient_id,id,date,details) values (3,1,now(),'first exam patient 3');
- select * from exam where patient_id=1;

FAQ

- How to solve the ERROR that the cluster name is not matched?
  - If you change the cluster name, you need to delete the storage directory for every nodes in “/var/lib/cassandra/data”
- How to solve the ERROR “ReadTimeOut”?
  - Two possible conditions:
    - Your Cassandra cluster is not running well. Then you need to firstly check the status of your cluster by “nodetool status” and manually run the Cassandra service to debug it.
    - The process cannot run within the defined time out setting. You can change the socket time out option to solve this problem: https://stackoverflow.com/questions/29437517/cassandra-timeout-cqlsh-query-largeish-amount-of-data

FAQ

- What can I do if I met “address is already in use” error?
  - This is often caused by that there is a dead casssandra process taking the port on the VM
  - You can use “sudo service cassandra status” to see if the cassandra process is dead
  - Then find the PID of the cassandra by ‘sudo netstat -peanut | grep “:7199 “‘ to find the PID which is represented at the end of the result
  - kill -9 <PID>