Exercise 6 – Node

In this exercise, you will:

- Understand what Apache Cassandra™ nodes are.
- Understand core hardware/software requirements of a node.

Nodes are the building blocks of Apache Cassandra™'s clusters. Therefore, it is useful to understand the care and feeding of nodes. These exercises will do just that.

Steps

1) Open a terminal and navigate to your `/home/ubuntu/node/resources/cassandra/bin/` folder.

2) Execute `nodetool` with the `help` command to list all possible commands.

```
./nodetool help
```

```
ubuntu@ds201-node1:~/node/resources/cassandra/bin$ ./nodetool help
usage: nodetool [(-u <username> | --username <username>)]
        [(-pw <password> | --password <password>)] [(-h <host> | --host <host>)]
        [(-p <port> | --port <port>)]
        [(-pwf <passwordFilePath> | --password-file <passwordFilePath>)] <command>
        [<args>]

The most commonly used nodetool commands are:
    abortrebuild                Abort a currently running rebuild operation.
                                Currently active streams will finish but no new
                                streams will be started.
    assassinate                 Forcefully remove a dead node without re-replicating
                                any data.  Use as a last resort if you cannot
                                removenode
    bootstrap                   Monitor/manage node's bootstrap process
    cleanup                     Triggers the immediate cleanup of keys no longer
                                belonging to a node. By default, clean all keyspaces
    clearsnapshot               Remove the snapshot with the given name from the
                                given keyspaces. If no snapshotName is specified we
                                will remove all snapshots
    compact                     Force a (major) compaction on one or more tables or
                                user-defined compaction on given SSTables
    compactionhistory           Print history of compaction
    compactionstats             Print statistics on compactions
    decommission                Decommission the *node I am connecting to*
```

```
describecluster             Print the name, snitch, partitioner and schema
                            version of a cluster
describering                Shows the token ranges info of a given keyspace
disableautocompaction       Disable autocompaction for the given keyspace and
                            table
disablebackup               Disable incremental backup
disablebinary               Disable native transport (binary protocol)
disablegossip               Disable gossip (effectively marking the node down)
disablehandoff              Disable storing hinted handoffs
disablehintsfordc           Disable hints for a data center
drain                       Drain the node (stop accepting writes and flush all
                            tables)
enableautocompaction        Enable autocompaction for the given keyspace and
                            table
enablebackup                Enable incremental backup
enablebinary                Reenable native transport (binary protocol)
enablegossip                Reenable gossip
enablehandoff               Reenable future hints storing on the current node
enablehintsfordc            Enable hints for a data center that was previsouly
                            disabled
failuredetector             Shows the failure detector information for the
                            cluster
flush                       Flush one or more tables
garbagecollect              Remove deleted data from one or more tables
gcstats                     Print GC Statistics
getbatchlogreplaythrottle   Print batchlog replay throttle in KB/s. This is
                            reduced proportionally to the number of nodes in the
                            cluster.
getcompactionthreshold      Print min and max compaction thresholds for a given
                            table
getcompactionthroughput     Print the MB/s throughput cap for compaction in the
                            system
getconcurrentcompactors     Get the number of concurrent compactors in the
                            system.
getendpoints                Print the end points that owns the key
getinterdcstreamthroughput  Print the Mb/s throughput cap for inter-datacenter
                            streaming in the system
getloggginglevels           Get the runtime logging levels
getmaxhintwindow            Print the max hint window in ms
getsstables                 Print the sstable filenames that own the key
getstreamthroughput         Print the Mb/s throughput cap for streaming in the
                            system
gettimeout                  Print the timeout of the given type in ms
gettraceprobability         Print the current trace probability value
gossipinfo                  Shows the gossip information for the cluster
handoffwindow               Print current hinted handoff window
help                        Display help information
info                        Print node information (uptime, load, ...)
inmemorystatus              Returns a list of the in-memory tables for this node
                            and the amount of memory each table is using, or
                            information about a single table if the keyspace and
                            columnfamily are given.
invalidatecountercache      Invalidate the counter cache
invalidatekeycache          Invalidate the key cache
invalidaterowcache          Invalidate the row cache
```

```
join                            Join the ring
listsnapshots                   Lists all the snapshots along with the size on disk
                                and true size.
mark_unrepaired                 Mark all SSTables of a table or keyspace as
                                unrepaired. Use when no longer running incremental
                                repair on a table or keyspace.
move                            Move node on the token ring to a new token
netstats                        Print network information on provided host
                                (connecting node by default)
nodesyncservice                 Manage the NodeSync service on the connected node
pausehandoff                    Pause hints delivery process
proxyhistograms                 Print statistic histograms for network operations
rangekeysample                  Shows the sampled keys held across all keyspaces
rebuild                         Rebuild data by streaming from other nodes
                                (similarly to bootstrap)
rebuild_index                   A full rebuild of native secondary indexes for a
                                given table
refresh                         Load newly placed SSTables to the system without
                                restart
refreshsizeestimates            Refresh system.size_estimates
reloadlocalschema               Reload local node schema from system tables
reloadtriggers                  Reload trigger classes
relocatesstables                Relocates sstables to the correct disk
removenode                      Show status of current node removal, force
                                completion of pending removal or remove provided ID
repair                          Repair one or more tables
repair_admin                    list and fail incremental repair sessions
replaybatchlog                  Kick off batchlog replay and wait for finish
resetlocalschema                Reset node's local schema and resync
resumehandoff                   Resume hints delivery process
ring                            Print information about the token ring
scrub                           Scrub (rebuild sstables for) one or more tables
sequence                        Run multiple nodetool commands from a file, resource
                                or stdin in sequence. Common options (host, port,
                                username, password) are passed to child commands.
setbatchlogreplaythrottle       Set batchlog replay throttle in KB per second, or 0
                                to disable throttling. This will be reduced
                                proportionally to the number of nodes in the
                                cluster.
setcachecapacity                Set global key, row, and counter cache capacities
                                (in MB units)
setcachekeystosave              Set number of keys saved by each cache for faster
                                post-restart warmup. 0 to disable
setcompactionthreshold          Set min and max compaction thresholds for a given
                                table
setcompactionthroughput         Set the MB/s throughput cap for compaction in the
                                system, or 0 to disable throttling
setconcurrentcompactors         Set number of concurrent compactors in the system.
sethintedhandoffthrottlekb      Set hinted handoff throttle in kb per second, per
                                delivery thread.
setinterdcstreamthroughput      Set the Mb/s throughput cap for inter-datacenter
                                streaming in the system, or 0 to disable throttling
setlogginglevel                 Set the log level threshold for a given component or
                                class. Will reset to the initial configuration if
                                called with no parameters.
```

```
    setmaxhintwindow            Set the specified max hint window in ms
    setstreamthroughput         Set the Mb/s throughput cap for streaming in the
                                system, or 0 to disable throttling
    settimeout                  Set the specified timeout in ms, or 0 to disable
                                timeout
    settraceprobability         Sets the probability for tracing any given request
                                to value. 0 disables, 1 enables for all requests, 0
                                is the default
    sjk                         Run commands of 'Swiss Java Knife'. Run 'nodetool
                                sjk --help' for more information.
    snapshot                    Take a snapshot of specified keyspaces or a snapshot
                                of the specified table
    status                      Print cluster information (state, load, IDs, ...)
    statusautocompaction        status of autocompaction of the given keyspace and
                                table
    statusbackup                Status of incremental backup
    statusbinary                Status of native transport (binary protocol)
    statusgossip                Status of gossip
    statushandoff               Status of storing future hints on the current node
    stop                        Stop compaction
    stopdaemon                  Stop cassandra daemon
    tablehistograms             Print statistic histograms for a given table
    tablestats                  Print statistics on tables
    toppartitions               Sample and print the most active partitions for a
                                given column family
    tpstats                     Print usage statistics of thread pools
    truncatehints               Truncate all hints on the local node, or truncate
                                hints for the endpoint(s) specified.
    upgradesstables             Rewrite sstables (for the requested tables) that are
                                not on the current version (thus upgrading them to
                                said current version)
    verify                      Verify (check data checksum for) one or more tables
    version                     Print cassandra version
    viewbuildstatus             Show progress of a materialized view build
```

Some commands display information about the entire cluster. Some commands show information only about the node that nodetool has connected to. Others are operations that can be run specifically on the connected node.

3) Try:

```
./nodetool status
```

```
Datacenter: Cassandra
=====================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address    Load       Owns Host ID                               Token  Rack
UN  127.0.0.1  174.89 KiB ?    6e125b89-eb17-42b1-907a-a78c7d290f70  0      rack1
```

The status command shows information about the entire cluster, particularly the state of each node, and information about each of those nodes: IP address, data load, number of tokens,

total percentage of data saved on each node, host ID, and datacenter and rack. We will discuss these in detail as the course progresses.

4) Try:

```
/home/ubuntu/node/bin/dsetool status
```

Take note as to the differences between `dsetool status` and `nodetool status`. Although both tools have a `status` command, `dsetool` works with DataStax Enterprise™ as a whole (Apache Cassandra™, Apache Spark™, Apache Solr™, Graph) whereas `nodetool` is specific to Apache Cassandra™. Their functionality diverges from here.

5) Try:

```
./nodetool info
```

```
ubuntu@ds201-node1:~/node/resources/cassandra/bin$ ./nodetool info
ID                     : 6e125b89-eb17-42b1-907a-a78c7d290f70
Gossip active          : true
Native Transport active: true
Load                   : 174.89 KiB
Generation No          : 1515627123
Uptime (seconds)       : 159937
Heap Memory (MB)       : 315.65 / 512.00
Off Heap Memory (MB)   : 0.00
Data Center            : Cassandra
Rack                   : rack1
Exceptions             : 0
Key Cache              : entries 0, size 0 bytes, capacity 25 MiB, 0 hits, 0 requests, NaN
recent hit rate, 14400 save period in seconds
Row Cache              : entries 0, size 0 bytes, capacity 0 bytes, 0 hits, 0 requests, NaN
recent hit rate, 0 save period in seconds
Counter Cache          : entries 0, size 0 bytes, capacity 12 MiB, 0 hits, 0 requests, NaN
recent hit rate, 7200 save period in seconds
Chunk Cache            : entries 445, size 13.4 MiB, capacity 2.38 GiB, 496 misses, 3027
requests, 0.836 recent hit rate, 214.016 microseconds miss latency
Percent Repaired       : 100.0%
Token                  : 0
```

The info command displays information about the connected node, which includes token information, host ID, protocol status, data load, node uptime, heap memory usage and capacity, datacenter and rack information, number of errors reported, cache usage, and percentage of SSTables that have been incrementally repaired. Again, we will cover most of these later.

6) Try:

./nodetool describecluster

```
Cluster Information:
```

```
        Name: Test Cluster
        Snitch: com.datastax.bdp.snitch.DseDelegateSnitch
        DynamicEndPointSnitch: enabled
        Partitioner: org.apache.cassandra.dht.Murmur3Partitioner
        Schema versions:
                264b8f23-db5e-377f-a14c-1722b3d51389: [127.0.0.1]
```

`describecluster` shows the settings that are common across all of the nodes in the cluster and the current schema version used by each node. We have a simple one node cluster currently but will add to it soon.

7) Try:

`./nodetool getlogginglevels`

```
ubuntu@ds201-node1:~/node/resources/cassandra/bin$ ./nodetool getlogginglevels

Logger Name                                                 Log Level
ROOT                                                             INFO
DroppedAuditEventLogger                                          INFO
SLF4JAuditWriter                                                 INFO
com.cryptsoft                                                     OFF
com.datastax.bdp.db                                             DEBUG
com.datastax.bdp.search.solr.metrics.SolrMetricsEventListener     DEBUG
com.datastax.driver.core.NettyUtil                              ERROR
org.apache.cassandra                                            DEBUG
org.apache.lucene.index                                          INFO
org.apache.solr.core.CassandraSolrConfig                         WARN
org.apache.solr.core.RequestHandlers                             WARN
org.apache.solr.core.SolrCore                                    WARN
org.apache.solr.handler.component                                WARN
org.apache.solr.search.SolrIndexSearcher                         WARN
org.apache.solr.update                                           WARN
org.apache.spark.rpc                                            ERROR
```

8) Also try:

`./nodetool setlogginglevel org.apache.cassandra TRACE`

The command `setlogginglevel` dynamically changes the logging level used by Apache Cassandra™ without the need for a restart. You can also look at the `/var/log/cassandra/system.log` afterwards to observe the changes.

9) Try:

`./nodetool settraceprobability 0.1`

The resultant value from the `settraceprobability` command represents a decimal describing the percentage of queries being saved, starting from 0 (0%) to 1 (100%). Saved traces can then be viewed in the `system_traces` keyspace.

10) Try:

```
./nodetool drain
```

The `drain` command stops writes from occurring on the node and flushes all data to disk. Typically, this command may be run before stopping an Apache Cassandra™ node.

11) Try:

./nodetool stopdaemon

The `stopdaemon` command stops a node's execution. Wait for it to complete.

12) Restart your node by running:

```
/home/ubuntu/node/bin/dse cassandra
```

Wait for Apache Cassandra™ to start before continuing.

13) We will now stress the node using a simple tool called Apache Cassandra(TM) Stress. Once your node has restarted, navigate to the `/home/ubuntu/node/resources/cassandra/tools/bin` directory in the terminal. Run `cassandra-stress` to populate the cluster with 50,000 partitions using 1 client thread and without any warmup using:

```
./cassandra-stress write n=50000 no-warmup -rate threads=1
```

Initially, we will see a long list of setting for the stress run. As Apache Cassandra™ stress executes, it logs several statistics to the terminal. Each line displays the statistics for the operations that occurred each second and shows number of partitions written, operations per second, latency information, and more.

14) Navigate back to `/home/ubuntu/node/resources/cassandra/bin` and run:

```
./nodetool flush
```

The `flush` command commits all written (memtable, discussed later) data to disk. Unlike `drain`, `flush` allows further writes to occur.

15) Check the new load on the node. Run:

```
./nodetool status
```

16) We will now examine the data `cassandra-stress` wrote to our node. Start `./cqlsh`. Execute the following CQLSH command to view the current keyspaces:

```
DESCRIBE KEYSPACES;
```

Notice the presence of `keyspace1` which `cassandra-stress` created.

17) Switch to that keyspace by executing the following:

```
USE keyspace1;
```

18) View the tables in `keyspace1` by executing the following:

```
DESCRIBE TABLES;
```

19) Query the first five rows from `standard1` by executing the following query:

```
SELECT *
FROM standard1
LIMIT 5;
```

The data that was written is not very meaningful, since they are all arbitrary BLOB values.