READY

```
%sh
STATUS="$(service cassandra status)"

if [[ $STATUS == *"is running"* ]]; then
    echo "Cassandra is running"
else
    echo " Cassandra not running .... Starting"
    service cassandra restart > /dev/null 2>&1 &
    echo " Started"
fi
```

READY

Exercise 7 – Gossip

READY

In this exercise, you will:

- Understand how Apache Cassandra™ uses gossip.
- Understand how gossip information propagates through a cluster.
- Understand how a gossip exchange works.

In a fully distributed system such as Apache Cassandra™, there is no single repository that contains the state of all the nodes in the cluster. Clearly, such a repository would be a single point of failure. Instead, Apache Cassandra™ uses the Gossip protocol to distribute nodes' status
amongst its peers.

In this exercise, we will examine the gossip information for our cluster.

READY

Steps

1. Be sure node is up and running using nodetool status. Start your nodes if necessary.

```
nodetool status
```

READY

07_Gossip_Solution

2. Execute the following command:

READY

```
nodetool gossipinfo
```

READY

Rerun your nodetool gossipinfo command a few times and notice the heartbeat values increasing for both nodes.

READY

```
nodetool gossipinfo
```

READY

Execute the following query of the system.peers table which stores some gossip data about a node’s peers.

READY

```
%cassandra

SELECT peer, data_center, host_id, preferred_ip, rack,
release_version, rpc_address, schema_version
FROM system.peers;
```

READY

Notice the values here are some of the same values you saw in the terminal. Also notice that a node does not store a row of peer data for itself. By default, cqlsh connects to 127.0.0.1

READY

READY

READY