**Creating a Replica Set**

Create a replica set rs-1 with one primary node, one secondary node & an arbitrary node. Demonstrate node failover in the replica set.

**7.1.1 Solution**

1. Shutdown any mongod and close terminals

2. Close all open terminals

3. Open 4 new terminals & set titles for each terminal as Node1, Node2, Arbiter& Client respectively. Use Terminal Set Title menu to do the same.

4. Goto $MONGODB\_HOME in all the terminals

5. Switch to Client terminal

6. Execute the below command and verify that ping is successful

$ ping localhost.localdomain

7. Switch user as root & create the below three folders & provide access to centos user

$ mkdir /data/node1

$ chown centos /data/node1

$ mkdir /data/node2

$ chown centos /data/node2

$ mkdir /data/arbiter

$ chown centos /data/arbiter

8. Exit from root

9. Switch to Node1 terminal and start the mongod service as given below

$ bin/mongod --replSet rs-1 --dbpath /data/node1 --port 40000

10. Switch to Node2 terminal and start the mongod service as given below

$ bin/mongod --replSet rs-1 --dbpath /data/node2 --port 40001

11. Switch to Arbiter terminal and start the mongod service as given below

$ bin/mongod --replSet rs-1 --dbpath /data/arbiter --port 40002

12. Switch to Client terminal and start the mongo shell connecting to Node1

$ bin/mongo localhost.localdomain:40000

13. Initiallize the Replica set &add Node2 & Arbiter nodes

>rs.initialize()

>rs.add("localhost.localdomain:40001")

>rs.add("localhost.localdomain:40002", {arbiterOnly: true})

14. Verify that Node1 is master by executing command below

>db.isMaster()

15. Display the status of the replication set and verify that Node1 is Primary

>rs.status()

16. Create a database called bookstore & insert a new book into Node1

> use bookstore

>db.books.insert({title: "MongoDB in Action"})

> show dbs

17. Exit mongo shell connected to Node1

18. Connect mongo shell to Node2 and verify that Node2 is Secondary

>mongo localhost.localdomain:40001

19. Set slave ok to true to enable accessing the data from slave node

>db.getMongo().setSlaveOk()

20. Verify that bookstore database is present in the datastore

> show dbs

21. Use bookstore database & find all the data in books collection. Verify that the book inserted in Node1 is appearing in Node2

> use bookstore

>db.books.find()

22. Switch to Node1 terminal and shutdown the mongod instance. Wait for a while and observe the traces which indicate that primary is down and makes Node2 as primary

23. Switch to Client terminal & connect mongo shell to Node2

24. Observe now that Node2 is Primary

25. Use bookstore database in insert a new book

>db.books.insert({title: "MongoDB : The Definitive Guide"})

26. Switch to Node1 terminal & bring up the mongod instance on Node1. Now the new book data will be replicated in Node1

27. Switch to Client terminal & connect to Node1

28. Observe that Node1 is Secondary

29. Set slave ok to true to enable accessing the data from slave node

>db.getMongo().setSlaveOk()

30. Verify that new book created in Node2 is accessible from Node1

31. Modify the new book title & verify that the modification is replicated in Node2

32. Remove the book from collection in Node2

33. Verify that the book is removed from Node1