MongoDB Replica Set Setup & Replication Commands (Windows)

Section 1: Setting Up MongoDB Replica Set on Windows

This section provides a step-by-step guide to setting up a MongoDB replica set on Windows for testing replication commands.

Assume we are setting up a 3-node replica set using local ports: 27017, 27018, and 27019.

1. Create Data Directories

Open Command Prompt and run:

```
> mkdir C:\data\rs0

> mkdir C:\data\rs1

> mkdir C:\data\rs2
```

2. Start Mongod Instances

Start each instance in separate Command Prompt windows:

```
> mongod --replSet rs0 --port 27017 --dbpath C:\data\rs0 --bind_ip localhost --fork --logpath C:\data\rs0\log.txt

> mongod --replSet rs0 --port 27018 --dbpath C:\data\rs1 --bind_ip localhost --fork --logpath C:\data\rs1\log.txt

> mongod --replSet rs0 --port 27019 --dbpath C:\data\rs2 --bind_ip localhost --fork --logpath C:\data\rs2\log.txt
```

3. Connect and Initialize Replica Set

Connect to the Mongo shell on port 27017:

```
> mongo --port 27017
```

Then initiate the replica set:

```
> rs.initiate({
   _id: "rs0",
   members: [
     { _id: 0, host: "localhost:27017" },
     { _id: 1, host: "localhost:27018" },
     { _id: 2, host: "localhost:27019" }
   ]
})
```

4. Check Replica Set Status

Run:

```
> rs.status()
```

This confirms the members and their states (PRIMARY, SECONDARY).

Section 2: Replication Commands with Examples

This guide demonstrates Level 1 MongoDB replication commands using the 'Student' database on Windows.

Make sure MongoDB is installed and configured properly. To follow these steps, launch `mongod` with

replica set configuration and then open the MongoDB shell using `mongo`.

rs.initiate()

Initializes a new replica set. Run this on the primary node.

Example:

```
> rs.initiate()
{
    "ok" : 1,
    ...
}
```

rs.status()

Checks replica set status.

Example:

```
> rs.status()
{
    "set" : "rs0",
    "members" : [...],
    ...
}
```

rs.add('localhost:27018')

Adds a secondary node to the replica set.

Example:

```
> rs.add('localhost:27018')

{ "ok" : 1 }
```

rs.remove('localhost:27018')

Removes a node from the replica set.

Example:

```
> rs.remove('localhost:27018')

{ "ok" : 1 }
```

rs.conf()

Returns current configuration.

Example:

```
> rs.conf()

{

   "_id" : "rs0",

   "members" : [...]

}
```

rs.reconfig(config)

Applies a new configuration to the replica set.

Example:

```
> var cfg = rs.conf()

> cfg.members[0].priority = 2

> rs.reconfig(cfg)
```

rs.stepDown()

Forces the primary to step down.

Example:

```
> rs.stepDown()
```

rs.freeze(60)

Freezes a node from becoming primary for 60 seconds.

Example:

```
> rs.freeze(60)
```

rs.printReplicationInfo()

Displays replication details.

Example:

```
> rs.printReplicationInfo()

configured oplog size: 990MB

...
```

rs.printSlaveReplicationInfo()

Displays slave replication info.

Example:

```
> rs.printSlaveReplicationInfo()

source: ...
```