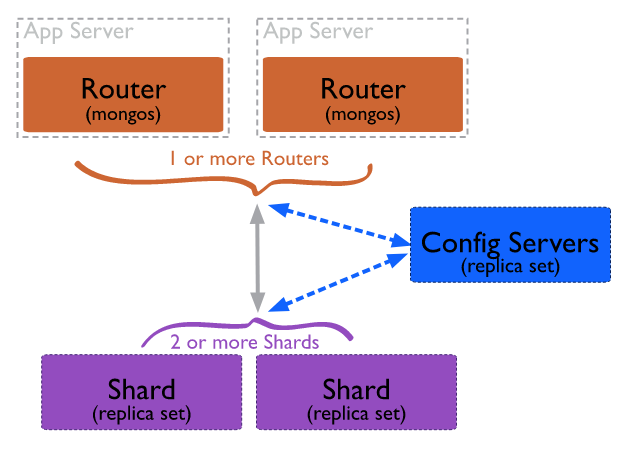
## **How to Setup MongoDB Sharding**



We will use the following setup for setting up MongoDB Sharding:

We want 4 servers

**Config Server**

**Query Router (mongos) Server**

**Shard-1 Server**

**Shard-2 Server**

## **Step-1: Install MongoDB on all Servers**

First you need to update and upgrade your systems repository in order to install MongoDB. Type the following command in your terminal and then press Enter.

|  |
| --- |
| sudo apt update && sudo apt upgrade -y |

First, you will need to install the [MongoDB community server](https://cloudinfrastructureservices.co.uk/how-to-setup-mongodb-community-server-on-azure-aws/) package on all four servers. Perform the below steps on all servers to install the MongoDB community server:

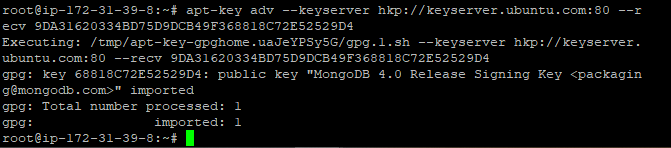
First, run the following command to install all required dependencies.

|  |
| --- |
| apt-get install gnupg2 wget tree -y |

Next, import the MongoDB GPG key using the command below:

|  |
| --- |
| apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9DA31620334BD75D9DCB49F368818C72E52529D4 |

Sample output:



Next, add the MongoDB repository to APT source file:

|  |
| --- |
| echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.0 multiverse" | tee /etc/apt/sources.list.d/mongodb-org.list |

Next, update the repository and install the MongoDB server package:

|  |
| --- |
| apt-get update -y  apt-get install mongodb-org -y |

Once the MongoDB server package is installed, verify the MongoDB version using the following command:

|  |
| --- |
| mongo --version |

Sample output:

Text

Description automatically generated

## **Step-2: Configure Config Server**

In this section, we will configure the config server to be a replica set. First, create a directory structure using the following command:

|  |
| --- |
| mkdir -p /mongodb-config/data/configdb  mkdir -p /mongodb-config/data/logs  touch /mongodb-config/data/logs/configsvr.log |

You can now verify your directory structure using the following command:

|  |
| --- |
| tree /mongodb-config |

Sample output:

Graphical user interface, text

Description automatically generated

Next, create a new configuration file for Config Server using your favorite editor:

|  |
| --- |
| nano /etc/mongodConfig.conf |

Define your storage path, logging path, port, server IP, cluster role, and replica set as shown below (copy and paste below lines):

|  |
| --- |
| storage:  dbPath: /mongodb-config/data/configdb  journal:  enabled: true  systemLog:  destination: file  logAppend: true  path: /mongodb-config/data/logs/configsvr.log  net:  port: 27019  bindIp: (config\_server\_private\_ip)  sharding:  clusterRole: configsvr  replication:  replSetName: configReplSet |

Sample output:Text

Description automatically generated

Next click ctrl+x,y enter

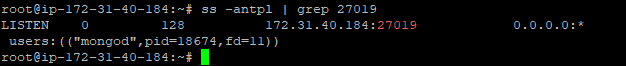
Save and close the file once you are finished. Next, start the config server using the following command:

|  |
| --- |
| mongod --config /etc/mongodConfig.conf & |

Next, check the config server listening port with the following command:

|  |
| --- |
| ss -antpl | grep 27019 |

Sample output:



Next, connect to the config server using the following command:

|  |
| --- |
| mongo (config\_server\_private\_ip):27019 |

Once you are connected to mongo shell, initiate the config server using the following command:

|  |
| --- |
| rs.initiate() |

Sample output:

Text

Description automatically generated

Next, check the status of the server with the following command:

|  |
| --- |
| rs.status() |

Sample output:

Text

Description automatically generated

## **Step-3: Configure Query Router**

Next, log in to the Query Router server and create a directory structure using the following command:

|  |
| --- |
| mkdir -p /mongodb-config/data/logs  touch /mongodb-config/data/logs/mongorouter.log |

Next, create a new configuration file for Query Router with the following command:

|  |
| --- |
| nano /etc/mongoRouter.conf |

Define your log file, IP address, port, and configDB as shown below:

|  |
| --- |
| systemLog:  destination: file  logAppend: true  path: /mongodb-config/data/logs/queryrouter.log  net:  port: 27017  bindIp: (Router\_server\_private\_ip)  sharding:  configDB: configReplSet/Config\_server\_private\_ip):27019 |

Sample output:

Text

Description automatically generated

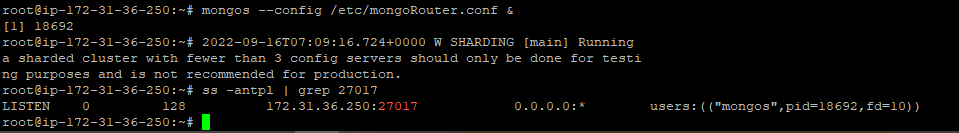
Save and close the file then start the Query Router with the following command:

|  |
| --- |
| mongos --config /etc/mongoRouter.conf & |

At this point, your MongoDB Query Router is started and listening on port 27017. You can check it with the following command:

|  |
| --- |
| ss -antpl | grep 27017 |

Sample output:



Now, connect to the Query Router using the following command:

|  |
| --- |
| mongo (router\_server\_private\_ip):27017 |

Once you are connected, you should get the following output:

Text

Description automatically generated

## **Step-4: Configure Shard-1 and Shard-2**

Next, log in to Shard servers and create a directory structure with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| mkdir -p /mongodb-config/data/sharddb/  mkdir -p /mongodb-config/data/logs  touch /mongodb-config/data/logs/shard.log |

Next, create a new configuration file for Shards with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| nano /etc/mongodShard.conf |

Sample output:

Text

Description automatically generated

Define storage path, log file, port, server IP, cluster role, and replica set as shown below (copy below lines and past in two servers Shard-1 and Shard-2) (shard1- “rs0” , shard2- “rs1” )

|  |
| --- |
| storage:  dbPath: /mongodb-config/data/sharddb  journal:  enabled: true  systemLog:  destination: file  logAppend: true  path: /mongodb-config/data/logs/shard.log  net:  port: 27018  bindIp: (shard\_server\_private\_ip)  sharding:  clusterRole: shardsvr  replication:  replSetName: "rs0" |

Sample output (shard-2):

Text

Description automatically generated

Save and close the file then start the Shard servers with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| mongod --config /etc/mongodShard.conf & |

At this point, the Shard servers is started and listening on port 27018. You can check it with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| ss -antpl | grep 27018 |

Sample output:

Text

Description automatically generated

Next, connect to the Shards with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| mongo (shard\_private\_ip):27018 |

Once you are connected, initiate the Shard servers with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| rs.initiate() |

Sample output:

Text

Description automatically generated

Next, verify the status of the servers with the following command (execute in two servers Shard-1 and Shard-2):

|  |
| --- |
| rs.status() |

Sample output:

Text

Description automatically generated

## **Step-5: Add the Shards to the Cluster**

Next, you will need to add your Shard server to the cluster. First, log in to the Query Router server. Then, connect to the mongo shell with the following command:

|  |
| --- |
| mongo (router\_server\_private\_ip):27017 |

Once you are connected, add your Shard-1 server with the following commands:

|  |
| --- |
| sh.addShard( "rs0/(shard-1\_private\_ip):27018") |

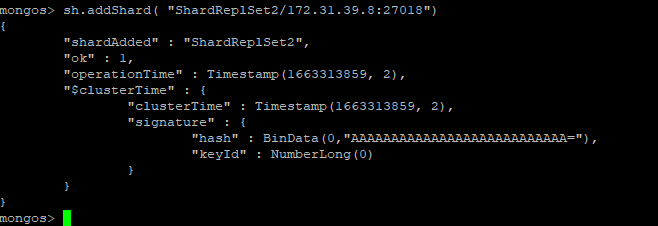
Sample output:Text

Description automatically generated

Once you are connected, add your Shard-2 server with the following commands:

|  |
| --- |
| sh.addShard( "rs1/(shard-2\_private\_ip):27018") |

Sample output:



**Step-6**

## **Enable Sharding for a Database**

Next, you will need to create a new database and enable sharding for the new database. On the Query Router server, connect to the mongo shell and run the following command to create a new database named **peoples**:

|  |
| --- |
| use peoples |

Next, enable the sharding on the peoples database with the following command:

|  |
| --- |
| sh.enableSharding("peoples") |

Next, check the sharding status using the following command:

|  |
| --- |
| sh.status() |

Next, you will need to add a new collection to the database with sharding support. Let’s add a new collection named **collection** to the **peoples** database:

|  |
| --- |
| sh.shardCollection("peoples.collection", {"name":1}) |

Next, insert the documents to the collections with the following command:

|  |
| --- |
| db.collection.save({  "name": "Application List",  "apps": ["Apache", "MariaDB", "Redis", "PHP"],  }) |

**Step-7**

## **Verify Sharding**

Next, you will need to verify if the sharding is working as intended. First, log in to the Shard server and connect to the mongo shell with the following command:

|  |
| --- |
| mongo 216.98.8.110:27018 |

Once you are connected, run the following command to check the database available on the replica set:

|  |
| --- |
| show dbs |

Next, switch database to peoples with the following command:

|  |
| --- |
| use peoples |

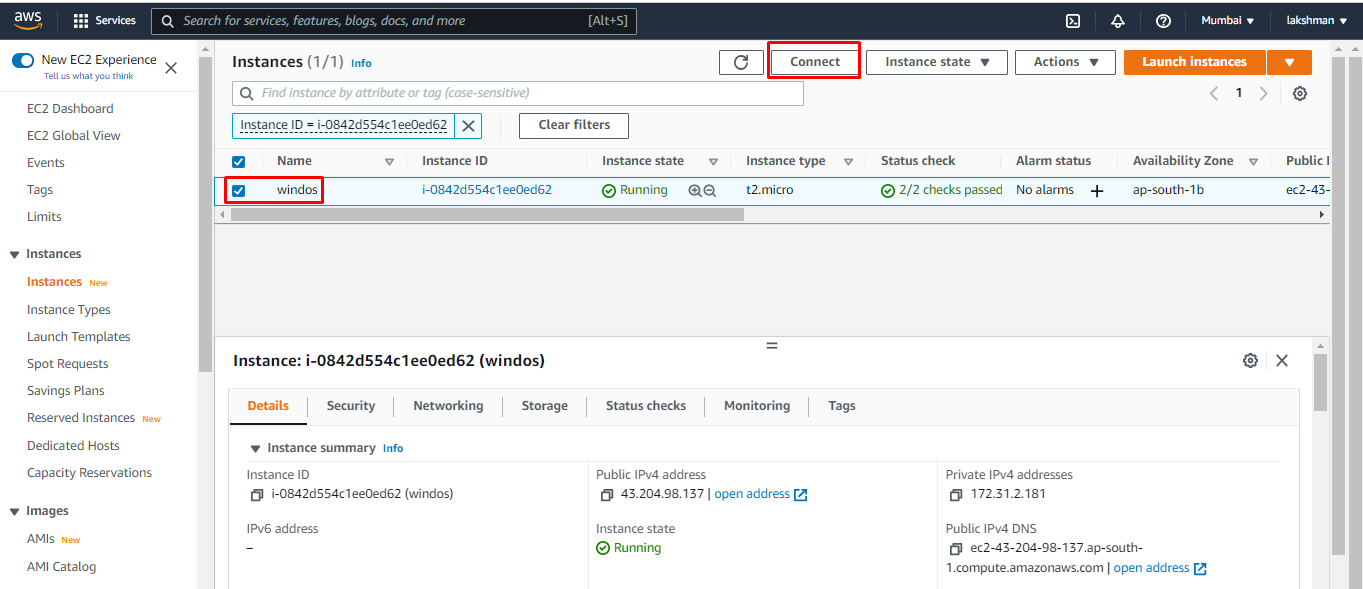
Next, check your collections and documents in the replica set using the following command:

|  |
| --- |
| db.collection.find() |

**Step-7**

**Install mongo dB Compass and connect Database**

Lunch windows instance and login.



Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

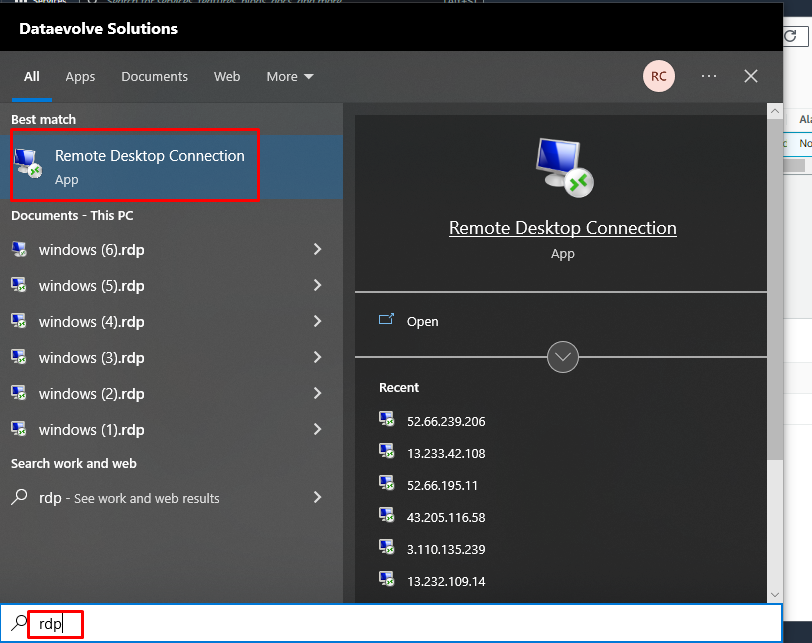
Description automatically generated

Graphical user interface, text, application

Description automatically generated

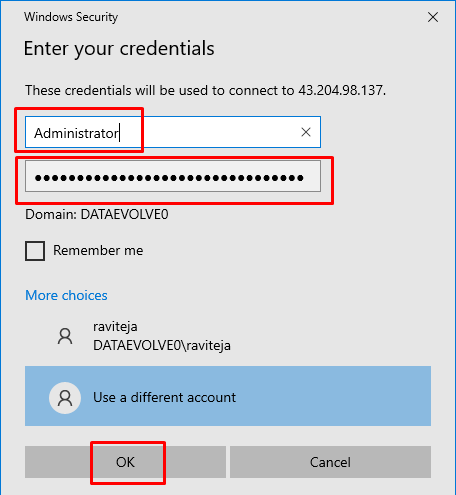
Copy the above password and username

Open RDP.



Graphical user interface, text, application, email

Description automatically generated



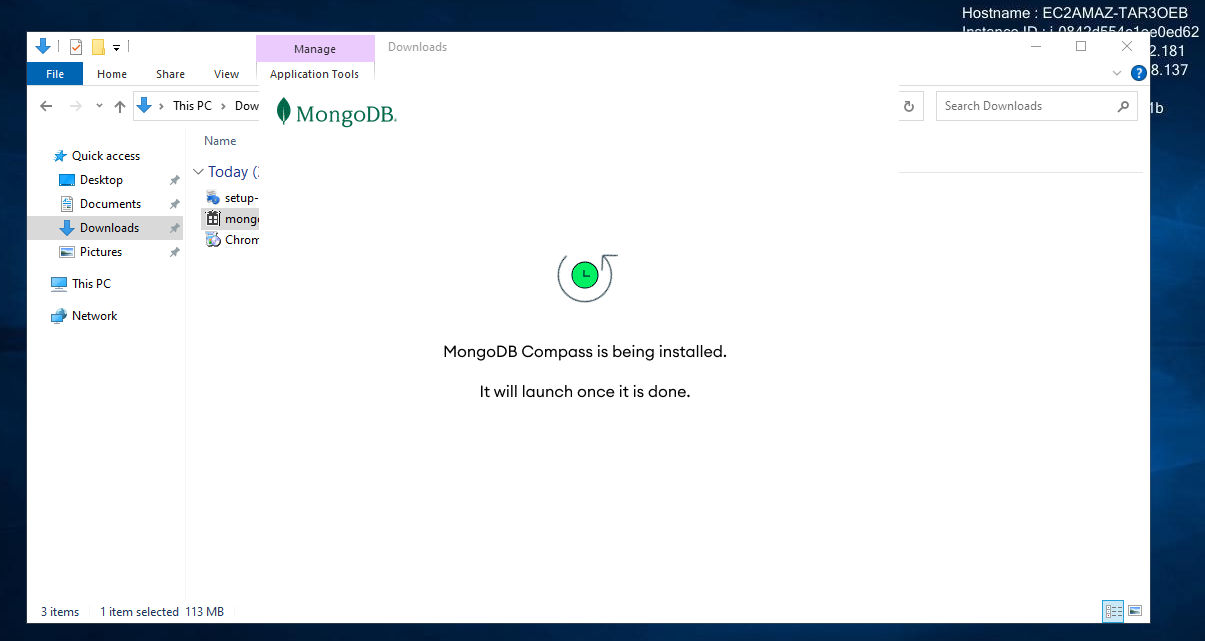
Graphical user interface, text, application, email

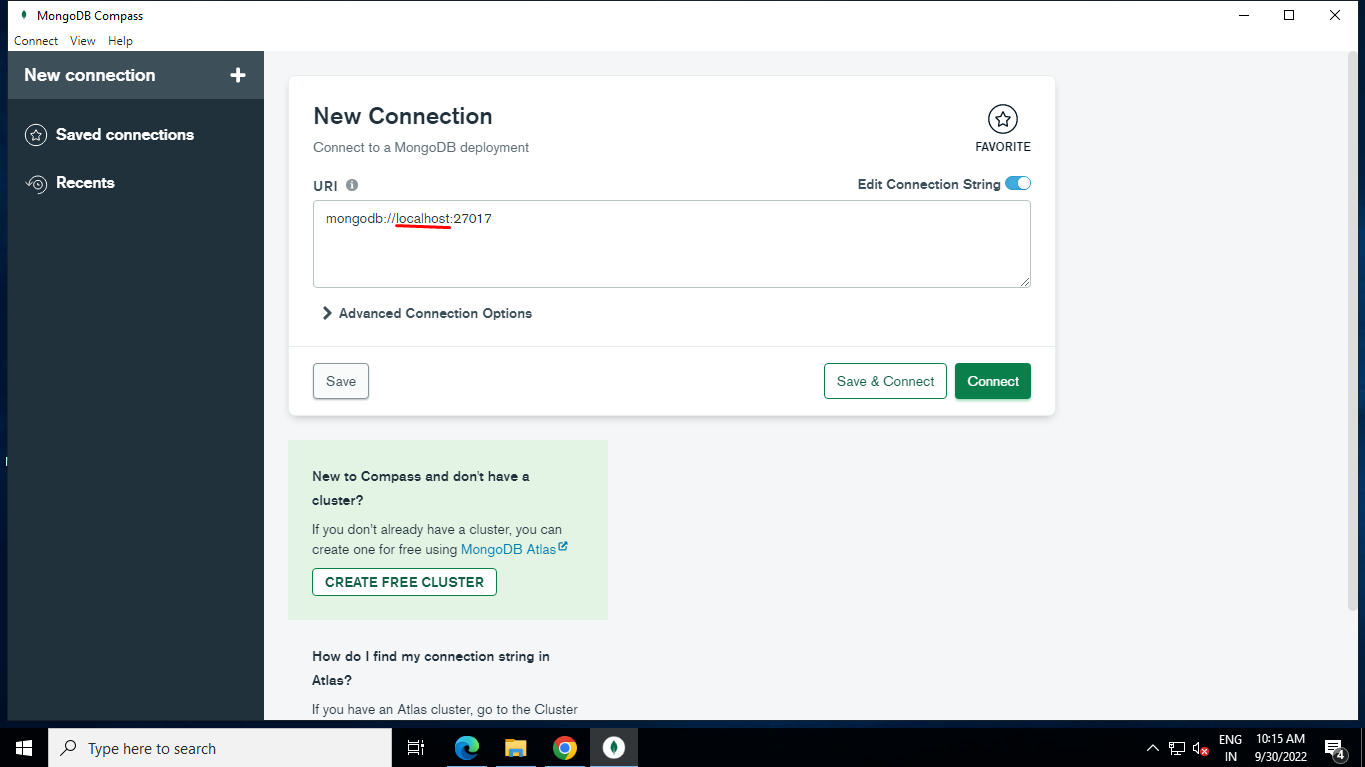
Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Next install mongodb compass in windows instance.





Give the router server private ip in the place of local host.

Graphical user interface, application

Description automatically generated

After login

A screenshot of a computer

Description automatically generated

Reference websites:

<https://cloudinfrastructureservices.co.uk/mongodb-sharding/>

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-mongodb-replica-set-on-ubuntu-20-04>

<https://www.techsupportpk.com/2020/03/fault-tolerant-mongodb-sharded-cluster-using-shared-storage-ubuntu-debian.html>