

**MADRAS INSTITUTE OF TECHNOLOGY
ANNA UNIVERSITY CHENNAI
CHENNAI - 600 044.**



DEPARTMENT OF INFORMATION TECHNOLOGY

**IT5612 DATA ANALYTICS AND CLOUD COMPUTING
LABORATORY**

RECORD NOTE BOOK

6/8 B. TECH INFORMATION TECHNOLOGY

Name : Mohammed Zuhayr Hussain

Reg. No : 2020506053

Semester : 6/8 IT

Year : 2023

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI

Chennai - 600 044.

BONAFIDE CERTIFICATE

Name : Mohammed Zuhayr Hussain
Reg. No : 2020506053
Subject : IT5612 Data Analytics And Cloud Computing Laboratory
Department : Information Technology

Certified to the bonafide record of practical work done by Mr. Mohammed Zuhayr Hussain in the IT5612 Data Analytics and Cloud Computing Laboratory during the period February 2023 -June 2023.

Date:

STAFF-IN-CHARGE

Submitted for the Practical Examination held on:

Internal Examiner

External Examiner

INDEX

S.No	Date	Name of the Experiment	Page No	Signature
1.	08/02/23	Descriptive Data Analysis		
2.	15/02/23	Univariate Bivariate and Multivariate Analysis		
3.	22/02/23	Linear, Logistic and Multiple Regression		
4.	08/03/23	Naïve Bayesian Classifier and SVM		
5.	15/03/23	Data Visualization		
6a.	29/03/23	MONGO DB installation		
6b.	12/04/23	MONGO DB CRUD		
6c.	19/04/23	MONGO DB Replication		
6d.	19/04/23	MONGO DB Sharding		
7.	26/04/23	Hadoop Installation		
8.	03/05/23	HIVE		

Experiment No: 6a

Date: 29/03/2023

MONGODB INSTALLATION

Aim:

To perform installation of MongoDB.

Installation:

1. Import the public key used by the package management system.

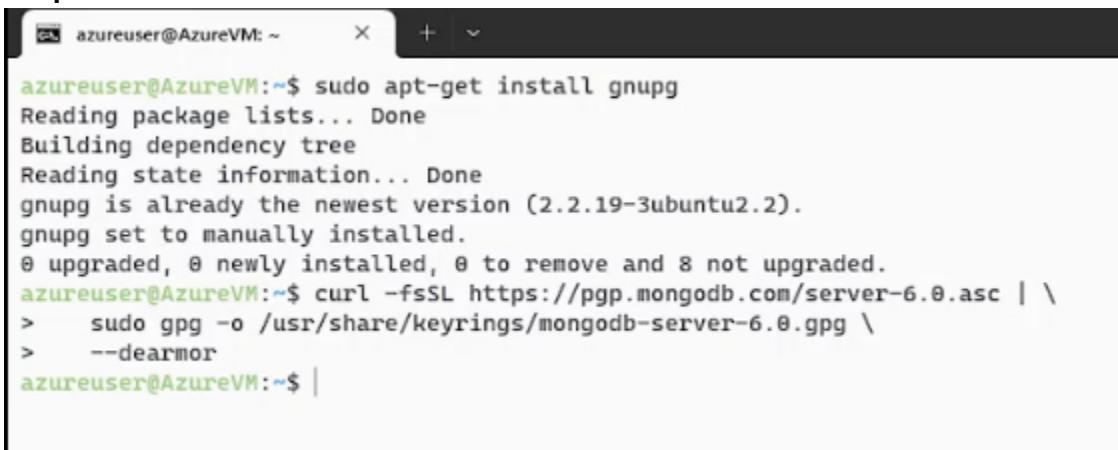
From a terminal, install gnupg if it is not already available:

```
sudo apt-get install gnupg
```

Issue the following command to import the MongoDB public GPG Key from <https://pgp.mongodb.com/server-6.0.asc>

```
curl -fsSL https://pgp.mongodb.com/server-6.0.asc | sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg --dearmor
```

Output:



```
azureuser@AzureVM:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.2).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
azureuser@AzureVM:~$ curl -fsSL https://pgp.mongodb.com/server-6.0.asc | \
>   sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg \
>   --dearmor
azureuser@AzureVM:~$ |
```

2. Create a list file for MongoDB

```
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-6.0.gpg ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
```

Output:

```
azureuser@AzureVM:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.2).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
azureuser@AzureVM:~$ curl -fsSL https://pgp.mongodb.com/server-6.0.asc
>   sudo gpg -o /usr/share/keyrings/mongodb-server-6.0.gpg \
>   --dearmor
azureuser@AzureVM:~$ echo "deb [ arch=amd64,arm64 signed-by=/usr/share/
mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-o
deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-6.0
azureuser@AzureVM:~$ |
```

3. Reload local package database

sudo apt-get update

Output:

```
azureuser@AzureVM:~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu focal-updates
Hit:3 http://azure.archive.ubuntu.com/ubuntu focal-backports
Hit:4 http://azure.archive.ubuntu.com/ubuntu focal-security
Ign:5 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org
Get:6 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org
Get:7 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org
Get:8 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org
Get:9 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org
Fetched 43.1 kB in 1s (45.4 kB/s)
Reading package lists... Done
azureuser@AzureVM:~$ |
```

4. Install the MongoDB packages.

sudo apt-get install -y mongodb-org

Output:

```
azureuser@AzureVM:~$ sudo apt-get install -y mongodb-org
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  mongodb-database-tools mongodb-mongosh mongodb-org-database mongodb-org-database
  mongodb-org-shell mongodb-org-tools
The following NEW packages will be installed:
  mongodb-database-tools mongodb-mongosh mongodb-org mongodb-org-database mongodb-
  mongodb-org-shell mongodb-org-tools
0 upgraded, 9 newly installed, 0 to remove and 8 not upgraded.
Need to get 142 MB of archives.
After this operation, 429 MB of additional disk space will be used.
Get:1 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:2 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:3 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:4 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:5 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:6 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:7 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:8 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Get:9 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64 
Fetched 142 MB in 2s (73.3 MB/s)
```

5. Starting the MongoD service

```
azureuser@AzureVM:~$ mkdir -p data/db
azureuser@AzureVM:~$ nohup mongod --dbpath data/db &
[1] 2868
azureuser@AzureVM:~$ nohup: ignoring input and appending output to 'nohup.out'

azureuser@AzureVM:~$ tail -f nohup.out
{"t":{"$date":"2023-05-23T13:49:24.803+00:00"},"s":"I", "c":"INDEX", "id":20345,
r":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"be3fdxfc5-deca-458a-81b6-1779
t":"index-3--2992632054906932761","collectionIdent":"collection-2--29926320549069327
{"t":{"$date":"2023-05-23T13:49:24.803+00:00"},"s":"I", "c":"REPL", "id":601531
,"attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
{"t":{"$date":"2023-05-23T13:49:24.804+00:00"},"s":"I", "c":"STORAGE", "id":22262,
{"t":{"$date":"2023-05-23T13:49:24.805+00:00"},"s":"I", "c":"STORAGE", "id":20320,
attr":{"namespace":"config.system.sessions","uuidDisposition":"generated","uuid":{u
:{}}}
{"t":{"$date":"2023-05-23T13:49:24.806+00:00"},"s":"I", "c":"CONTROL", "id":20712,
not set up; waiting until next sessions reap interval","attr":{"error":"NamespaceNo
{"t":{"$date":"2023-05-23T13:49:24.806+00:00"},"s":"I", "c":"NETWORK", "id":23015,
/mongodb-27017.sock"}}
 {"t":{"$date":"2023-05-23T13:49:24.806+00:00"},"s":"I", "c":"NETWORK", "id":23015,
0.0.1}}
 {"t":{"$date":"2023-05-23T13:49:24.807+00:00"},"s":"I", "c":"NETWORK", "id":23016,
t":27017,"ssl":"off"}}
 {"t":{"$date":"2023-05-23T13:49:24.853+00:00"},"s":"I", "c":"INDEX", "id":20345,
uilding","attr":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"1384b39f-e6f8-4
ndex":"_id_","ident":"index-5--2992632054906932761","collectionIdent":"collection-4-
 {"t":{"$date":"2023-05-23T13:49:24.853+00:00"},"s":"I", "c":"INDEX", "id":20345,
uilding","attr":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"1384b39f-e6f8-4
ndex":"lsidTTLIndex","ident":"index-6--2992632054906932761","collectionIdent":"cole
```

Result:

MongoDB has been successfully installed on the system.

Experiment No: 6b

Date:12/04/2023

MONGODB CRUD

Aim:

To perform CRUD operations on retail database with MongoDB.

Creating Database:

```
use retail_database
```

Output:

```
test> use retail_database;
switched to db retail_database
retail_database> █
```

Inserting Data:

Product Collection

```
db.Product.insertMany([ { _id: 1, name: "Product A", price: 10.99 }, { _id: 2, name: "Product B", price: 19.99 }, { _id: 3, name: "Product C", price: 5.99 } ])
```

Output:

```
retail_database> db.Product.insertMany([
...   { _id: 1, name: "Product A", price: 10.99 },
...   { _id: 2, name: "Product B", price: 19.99 },
...   { _id: 3, name: "Product C", price: 5.99 }
... ])
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
retail_database>
```

Customer Collection

```
db.Customer.insertMany([ { _id: 1, name: "John Doe", email: "john@example.com" },
{ _id: 2, name: "Jane Smith", email: "jane@example.com" }, { _id: 3, name: "David Johnson", email: "david@example.com" } ])
```

Output:

```
retail_database> db.Customer.insertMany([
  { _id: 1, name: "John Doe", email: "john@example.com" },
  { _id: 2, name: "Jane Smith", email: "jane@example.com" },
  { _id: 3, name: "David Johnson", email: "david@example.com" }
])
{
  acknowledged: true,
  insertedIds: { '0': 1, '1': 2, '2': 3 }
}
```

Manufacturer Collection

```
db.Manufacturer.insertMany([
  { _id: 1, name: "Manufacturer A", location: "City A" },
  { _id: 2, name: "Manufacturer B", location: "City B" },
  { _id: 3, name: "Manufacturer C", location: "City C" }
])
```

Output:

```
retail_database> db.Manufacturer.insertMany([
  { _id: 1, name: "Manufacturer A", location: "City A" },
  { _id: 2, name: "Manufacturer B", location: "City B" },
  { _id: 3, name: "Manufacturer C", location: "City C" }
])
{
  acknowledged: true,
  insertedIds: { '0': 1, '1': 2, '2': 3 }
}
```

Shipping Collection

```
db.Shipping.insertMany([
  { _id: 1, productId: 1, customerId: 2, shippedDate: ISODate("2023-05-20") },
  { _id: 2, productId: 2, customerId: 1, shippedDate: ISODate("2023-05-21") },
  { _id: 3, productId: 3, customerId: 3, shippedDate: ISODate("2023-05-22") }
])
```

Output:

```
retail_database> db.Shipping.insertMany([
  { _id: 1, productId: 1, customerId: 2, shippedDate: ISODate("2023-05-20") },
  { _id: 2, productId: 2, customerId: 1, shippedDate: ISODate("2023-05-21") },
  { _id: 3, productId: 3, customerId: 3, shippedDate: ISODate("2023-05-22") }
])
{
  acknowledged: true,
  insertedIds: { '0': 1, '1': 2, '2': 3 }
}
retail_database> █
```

Time Collection

```
db.Time.insertMany([
  { _id: 1, year: 2023, month: 5, day: 20 },
  { _id: 2, year: 2023, month: 5, day: 21 },
  { _id: 3, year: 2023, month: 5, day: 22 }
])
```

Output:

```
retail_database> db.Time.insertMany([
...   { _id: 1, year: 2023, month: 5, day: 20 },
...   { _id: 2, year: 2023, month: 5, day: 21 },
...   { _id: 3, year: 2023, month: 5, day: 22 }
... ])
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
```

Reading Data:

Retrieve all products:

```
db.Product.find()
```

Output:

```
retail_database> db.Product.find()
[
  { _id: 1, name: 'Product A', price: 10.99 },
  { _id: 2, name: 'Product B', price: 19.99 },
  { _id: 3, name: 'Product C', price: 5.99 }
]
retail_database>
```

Retrieve a specific product (by _id):

```
db.Product.findOne({ _id: 1 })
```

Output:

```
retail_database> db.Product.findOne({ _id: 1 })
{ _id: 1, name: 'Product A', price: 10.99 }
retail_database>
```

Retrieve all customers:

```
db.Customer.find()
```

Output:

```
retail_database> db.Customer.find()
[
  { _id: 1, name: 'John Doe', email: 'john@example.com' },
  { _id: 2, name: 'Jane Smith', email: 'jane@example.com' },
  { _id: 3, name: 'David Johnson', email: 'david@example.com' }
]
retail_database>
```

Retrieve a specific customer (by _id):

```
db.Customer.findOne({ _id: 1 })
```

Output:

```
retail_database> db.Customer.findOne({ _id: 1 })
{ _id: 1, name: 'John Doe', email: 'john@example.com' }
retail_database>
```

Update Operations:

Update the price of a product:

```
db.Product.updateOne(
  { _id: 1 },
  { $set: { price: 12.99 } }
)
```

Output:

```
retail_database> db.Product.updateOne(
...   { _id: 1 },
...   { $set: { price: 12.99 } }
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
retail_database>
```

Delete Operations:

Delete a product:

```
db.Product.deleteOne({ _id: 3 })
```

Output:

```
retail_database> db.Product.deleteOne({ _id: 3 })
{ acknowledged: true, deletedCount: 1 }
retail_database>
```

Delete all shipping documents:

```
db.Shipping.deleteMany({})
```

Output:

```
retail_database> db.Shipping.deleteMany({})
{ acknowledged: true, deletedCount: 3 }
retail_database> db.Shipping.find()

retail_database> █
```

Result:

MongoDB CRUD operations have been successfully performed on the retail database.

MONGODB REPLICATION

Aim:

To perform CRUD operations on retail database with MongoDB.

Create The Data Directories For The Three Nodes:

```
mkdir -p data/db1  
mkdir -p data/db2  
mkdir -p data/db3
```

Start The Three Nodes In Separate Terminals:

```
mongod --port 27017 --dbpath data/db1 --replSet rs0
```

Output:

```
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo$ mongod --port 27017 --dbpath data/db1 --replSet rs0  
{"t":{"$date":"2023-04-25T12:16:11.833+05:30"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "-", "msg": "spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "incomingInternalClient": {"minWireVersion": 6, "maxWireVersion": 17}, "isInternalClient": true}}  
{"t":{"$date": "2023-04-25T12:16:11.835+05:30"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "-", "msg": "e-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}  
{"t": {"$date": "2023-04-25T12:16:11.835+05:30"}, "s": "I", "c": "NETWORK", "id": 4648601, "ctx": "main", "ms_ TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}  
{"t": {"$date": "2023-04-25T12:16:11.836+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "ms_rvice", "attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}  
{"t": {"$date": "2023-04-25T12:16:11.836+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "ms_rvice", "attr": {"service": "TenantMigrationRecipientService", "namespace": "config.tenantMigrationRecipient"}  
{"t": {"$date": "2023-04-25T12:16:11.836+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "ms_rvice", "attr": {"service": "ShardSplitDonorService", "namespace": "config.tenantSplitDonors"}}  
mongod --port 27018 --dbpath data/db2 --replSet rs0
```

Output:

```
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo  
pragadeshbs@pragadeshbs-pc:~/opt/mongo$ mongod --port 27018 --dbpath data/db2 --replSet rs0  
{"t": {"$date": "2023-04-25T12:16:35.433+05:30"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "main", "ms_ spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "incomingInternalClient": {"minWireVersion": 6, "maxWireVersion": 17}, "isInternalClient": true}}  
{"t": {"$date": "2023-04-25T12:16:35.434+05:30"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "main", "ms_ orce-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}  
{"t": {"$date": "2023-04-25T12:16:35.434+05:30"}, "s": "I", "c": "NETWORK", "id": 4648601, "ctx": "main", "ms_ TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}  
{"t": {"$date": "2023-04-25T12:16:35.435+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "ms_rvice", "attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}  
{"t": {"$date": "2023-04-25T12:16:35.435+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "ms_rvice", "attr": {"service": "TenantMigrationRecipientService", "namespace": "config.tenantMigrationRecipient"}  
{"t": {"$date": "2023-04-25T12:16:35.435+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "ms_rvice", "attr": {"service": "ShardSplitDonorService", "namespace": "config.tenantSplitDonors"}}  
{"t": {"$date": "2023-04-25T12:16:35.435+05:30"}, "s": "I", "c": "CONTROL", "id": 5945603, "ctx": "main", "ms_id": 4918, "port": 27018, "dbPath": "data/db2", "architecture": "64-bit", "host": "pragadeshbs-pc"}}
```

```
mongod --port 27019 --dbpath data/db3 --replSet rs0
```

Output:

```
pragadeshbs@pragadeshbs-pc:~/opt/mongo$ mongod --port 27019 --dbpath data/db3 --replSet rs0
{"t":{"$date":"2023-04-25T12:17:03.914+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main","msg":"forcing enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2023-04-25T12:17:03.915+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"main","msg":"{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17}, "incomingInternalClient":{},"outgoing":{"minWireVersion":6,"maxWireVersion":17}, "isInternalClient":true}}"}
{"t":{"$date":"2023-04-25T12:17:03.915+05:30"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"main","msg":"TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
{"t":{"$date":"2023-04-25T12:17:03.916+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"service","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
 {"t":{"$date":"2023-04-25T12:17:03.916+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"service","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipient"}}
 {"t":{"$date":"2023-04-25T12:17:03.916+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main","msg":"service","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
 {"t":{"$date":"2023-04-25T12:17:03.916+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"main","msg":"{"t":{"$date":"2023-04-25T12:17:03.917+05:30"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandli}}
```

Connect To The Node On Port 27017:

```
mongosh --port 27017
```

```
Initiate the replica set
rs.initiate()
rs.add("localhost:27018")
rs.add("localhost:27019")
```

Output:

```
Activities Terminal Apr 25 12:19
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
pragadeshbs@pragadeshbs-pc:~/o... × pragadeshbs@pragadeshbs-pc:~/o... × pragadeshbs@pragadeshbs-pc:~/o... × mongosh mongodb://127.0.0.1:2701...
test> rs.status()
MongoServerError: no replset config has been received
test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:27017',
  ok: 1
}
rs0 [direct: other] test> rs.add("localhost:27018")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1682405334, i: 1 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1682405334, i: 1 })
}
rs0 [direct: primary] test> rs.add("localhost:27019")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1682405345, i: 1 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1682405345, i: 1 })
}
rs0 [direct: primary] test> rs.status()
{
  set: 'rs0',
  date: ISODate("2023-04-25T06:49:11.770Z"),
  ...
}
```

Check The Status Of The Replica Set:

```
rs.status()
```

Output:

```
Activities Terminal mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000 Apr 25 12:19
pragadeshb@pragadeshbhs-pc:~/o... x pragadeshb@pragadeshbhs-pc:~/o... x pragadeshb@pragadeshbhs-pc:~/o... x mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000 x
rs0 [direct: primary] test> rs.status()
{
  set: 'rs0',
  date: ISODate("2023-04-25T06:49:11.770Z"),
  mystate: 1,
  term: Long("1"),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long("2000"),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1682405347, i: 1 }), t: Long("1") },
    lastCommittedWallTime: ISODate("2023-04-25T06:49:07.951Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1682405347, i: 1 }), t: Long("1") },
    appliedOpTime: { ts: Timestamp({ t: 1682405347, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1682405347, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2023-04-25T06:49:07.951Z"),
    lastDurableWallTime: ISODate("2023-04-25T06:49:07.951Z")
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1682405325, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate("2023-04-25T06:48:45.869Z"),
    electionTerm: Long("1"),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1682405325, i: 1 }), t: Long("-1") },
    lastseenOpTimeAtElection: { ts: Timestamp({ t: 1682405325, i: 1 }), t: Long("-1") },
    numVotesNeeded: 1,
    priorityAtElection: 1,
    electionTimeoutMillis: Long("10000"),
    newTermStartDate: ISODate("2023-04-25T06:48:45.896Z"),
    wMajorityWriteAvailabilityDate: ISODate("2023-04-25T06:48:45.907Z")
  },
  members: [
}
```

Insert Data On The Primary Node:

```
db.employee.insertOne({
  name: 'John', age: 45, city: 'Berlin'
})
```

Read Data From The Secondary Node On Port 27018

```
db.getMongo().setReadPref('secondary')
db.employee.find()
```

Output:

```
2023-04-25T12:16:35.437+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-04-25T12:16:35.660+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2023-04-25T12:16:35.660+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2023-04-25T12:16:35.660+05:30: Soft rlimits for open file descriptors too low
-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----

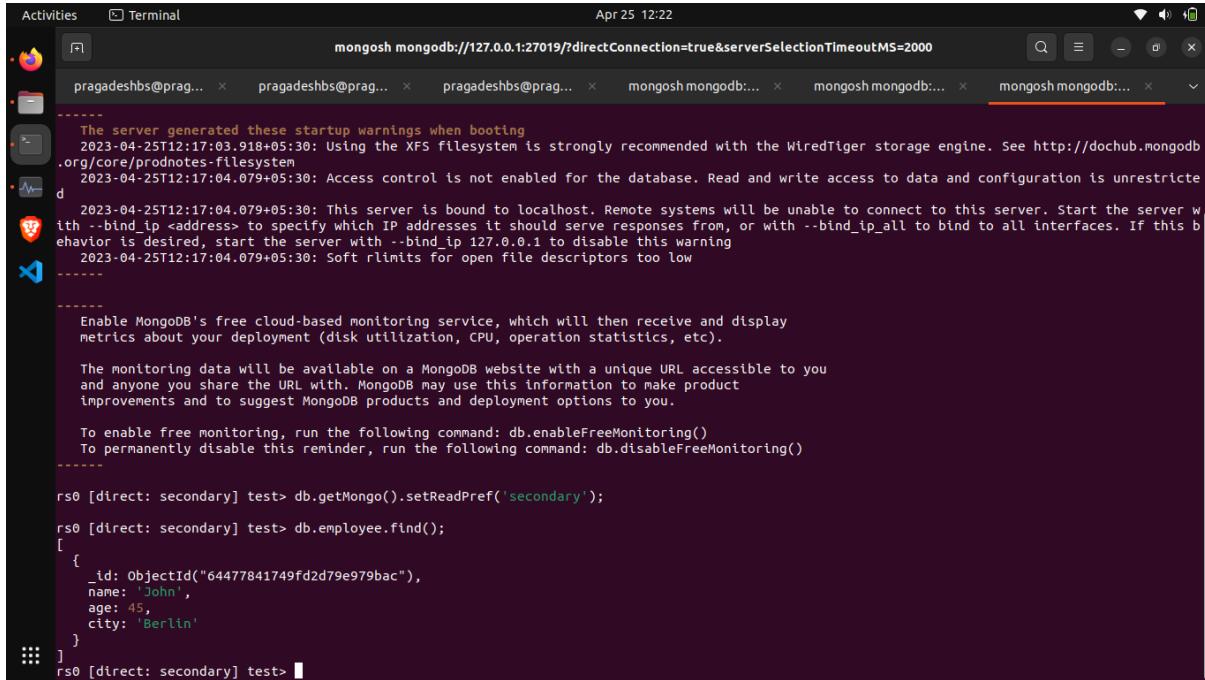
rs0 [direct: secondary] test> db.employee.find();
MongoServerError: not primary and secondaryOk=false - consider using db.getMongo().setReadPref() or readPreference in the connection string
rs0 [direct: secondary] test> db.getMongo().setReadPref('secondary');

rs0 [direct: secondary] test> db.employee.find();
[
  {
    _id: ObjectId("64477841749fd2d79e979bac"),
    name: 'John',
    age: 45,
    city: 'Berlin'
  }
]
rs0 [direct: secondary] test>
```

Read Data From The Secondary Node On Port 27019:

```
db.getMongo().setReadPref('secondary')
db.employee.find()
```

Output:



A screenshot of a terminal window titled "mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000". The window shows startup warnings, configuration options, and the execution of the command db.employee.find().

```
mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000
Apr 25 12:22
pragadeshb@prag... x pragadeshb@prag... x pragadeshb@prag... x mongosh mongodb:... x mongosh mongodb:... x mongosh mongodb:...
-----
The server generated these startup warnings when booting
2023-04-25T12:17:04.918+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-04-25T12:17:04.079+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
d
2023-04-25T12:17:04.079+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2023-04-25T12:17:04.079+05:30: Soft rlimits for open file descriptors too low
-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
rs0 [direct: secondary] test> db.getMongo().setReadPref('secondary');
rs0 [direct: secondary] test> db.employee.find();
[
  {
    _id: ObjectId("64477841749fd2d79e979bac"),
    name: 'John',
    age: 45,
    city: 'Berlin'
  }
]
rs0 [direct: secondary] test>
```

Result:

Replication has been successfully performed on a database with MongoDB.

MONGODB SHARDING

Aim:

To perform sharding on a database with MongoDB.

Create The Data Directories For The Config Server And The Two Shards:

```
mkdir -p data/configdb  
mkdir -p data/db1  
mkdir -p data/db2
```

Start The Config Server:

```
mongod --configsvr --replSet config --dbpath data/configdb
```

Output:

```
pragadeshb... × mongosh m... ×  
pragadeshb$mongod --configsvr --replSet config --db data/configdb  
Error parsing command line: unrecognised option '--db'  
try "mongod --help" for more information  
try "mongod --help" for more information  
pragadeshb$mongod --configsvr --replSet config --dbpath data/configdb  
{"t": {"$date": "2023-04-25T17:54:56.399+05:30"}, "s": "", "c": "CONTROL", "id": 123285, "ctx": "-", "msg": "Automatically disabling TLS 1.0, to force enable TLS 1.3 specify --sslDisabledProtocols 'none'"},  
{"t": {"$date": "2023-04-25T17:54:56.399+05:30"}, "s": "", "c": "NETWORK", "id": 4915701, "ctx": "-", "msg": "Initialized wire specification", "attr": [{"spec": [{"incomingExternalClient": ["M1N1WireVersion": 10, "maxWireVersion": 17], "incomingInternalClient": ["M1N1WireVersion": 10, "maxWireVersion": 17}], "outgoing": [{"M1N1WireVersion": 16, "maxWireVersion": 17}, {"isInternalClient": true}}]}]},  
{"t": {"$date": "2023-04-25T17:54:56.401+05:30"}, "s": "", "c": "NETWORK", "id": 4648561, "ctx": "Main", "msg": "Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."},  
{"t": {"$date": "2023-04-25T17:54:56.403+05:30"}, "s": "", "c": "REPL", "id": 5123800, "ctx": "Main", "msg": "Successfully registered PrimaryOnlyService", "attr": [{"service": "reshardingCoordinatorService", "namespace": "config.reshardingOperations"}]},  
{"t": {"$date": "2023-04-25T17:54:56.403+05:30"}, "s": "", "c": "REPL", "id": 5123801, "ctx": "Main", "msg": "Successfully registered PrimaryOnlyService", "attr": [{"service": "configsvrCoordinatorService", "namespace": "config.sharding_configsvr_coordinators"}]},  
{"t": {"$date": "2023-04-25T17:54:56.403+05:30"}, "s": "", "c": "CONTROL", "id": 1594560, "ctx": "Main", "msg": "Multi threading initialized"},  
{"t": {"$date": "2023-04-25T17:54:56.404+05:30"}, "s": "", "c": "CONTROL", "id": 14615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": [{"pId": 692, "port": 27619, "dbPath": "data/configdb", "architecture": "64-bit", "host": "pragadeshb"}]},  
{"t": {"$date": "2023-04-25T17:54:56.404+05:30"}, "s": "", "c": "CONTROL", "id": 23401, "ctx": "initandlisten", "msg": "Build Info", "attr": [{"buildInfo": "version": "6.0.3", "gitVersion": "c9a99c120371d4dc52ccb15d8c3a036cc8d3b1d", "openSSLVersion": "OpenSSL 3.0.2 15 Mar 2022", "modules": [], "allocator": "tcmalloc", "environment": [{"distro": "ubuntu2204", "distarch": "x86_64", "target_arch": "x86_64"}]}]},  
{"t": {"$date": "2023-04-25T17:54:56.404+05:30"}, "s": "", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": [{"os": {"name": "Ubuntu", "version": "22.04"}}]},  
{"t": {"$date": "2023-04-25T17:54:56.404+05:30"}, "s": "", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line"},  
{"t": {"$date": "2023-04-25T17:54:56.405+05:30"}, "s": "", "c": "STORAGE", "id": 122297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": [{"startUpWarnings": 1}], "attr": [{"storage": "wiredTiger", "options": [{"replicaSet": "config"}, {"sharding": "clusterRole": "configsvr"}, {"storage": {"dbPath": "data/configdb"}]}]}]},  
{"t": {"$date": "2023-04-25T17:54:56.405+05:30"}, "s": "", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": [{"storage": "wiredTiger", "options": [{"create": "cache_size=3354M", "session_max=330000, eviction=(threads_min=4, threads_max=4), config_base=false, statistics=(fast)}, {"log": {"enabled": true, "remove": true, "path": "journal", "compressor": "snappy"}, "builtin_extension_config": {"zstd": {"compression_level": 6}}, "file_manager": {"close_idle_time": 600, "close_scan_interval": 8, "close_handle_minimum": 2000}, "statistics_log": {"wait": 0}, "json_output": {"error": "message", "verbose": 0}, "recovery_progress": 1, "checkpoint_progress": 1, "compact": 0, "evict": 0, "history_store": 0, "recovery": 0, "rts": 0, "salvage": 0, "tiered": 0, "timestamp": 0, "transaction": 0, "verify": 0, "log": 0}], "durationMillis": 146}], "attr": [{"storage": "wiredTiger", "options": [{"replicaSet": "config"}, {"sharding": "clusterRole": "configsvr"}, {"storage": {"dbPath": "data/configdb"}]}]}]},  
{"t": {"$date": "2023-04-25T17:54:56.551+05:30"}, "s": "", "c": "STORAGE", "id": 4795906, "ctx": "initandlisten", "msg": "WiredTiger opened", "attr": [{"storage": "wiredTiger", "options": [{"replicaSet": "config"}, {"sharding": "clusterRole": "configsvr"}, {"storage": {"dbPath": "data/configdb"}]}]}]},  
{"t": {"$date": "2023-04-25T17:54:56.551+05:30"}, "s": "", "c": "RECOVERY", "id": 23987, "ctx": "initandlisten", "msg": "WiredTiger recoveryTimestamp
```

Connect To The Config Server:

```
mongosh --port 27019
```

Output:

```
Activities Terminal mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000 Apr 25 18:19
pragadeshb... x mongosh m... x pragadeshb... x mongosh m... x pragadeshb... x mongosh m... x pragadeshb... x mongosh m...
pragadeshb@pragadeshb-OptiP...:~/opt/mongo$ mongosh
Current Mongosh Log ID: 6447c6aaed1272353af295ec
Connecting to: mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8.0
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27019
pragadeshb@pragadeshb-OptiP...:~/opt/mongo$ mongosh --port 27019
Current Mongosh Log ID: 6447c6b59c13944e1ccb160
Connecting to: mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8.0
Using MongoDB: 6.0.5
Using Mongosh: 1.8.0
For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2023-04-25T17:54:56.405+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-04-25T17:54:56.562+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2023-04-25T17:54:56.562+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2023-04-25T17:54:56.563+05:30: Soft rlimits for open file descriptors too low
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
```

Initiate The Config Server Replica Set:

```
rs.initiate()
```

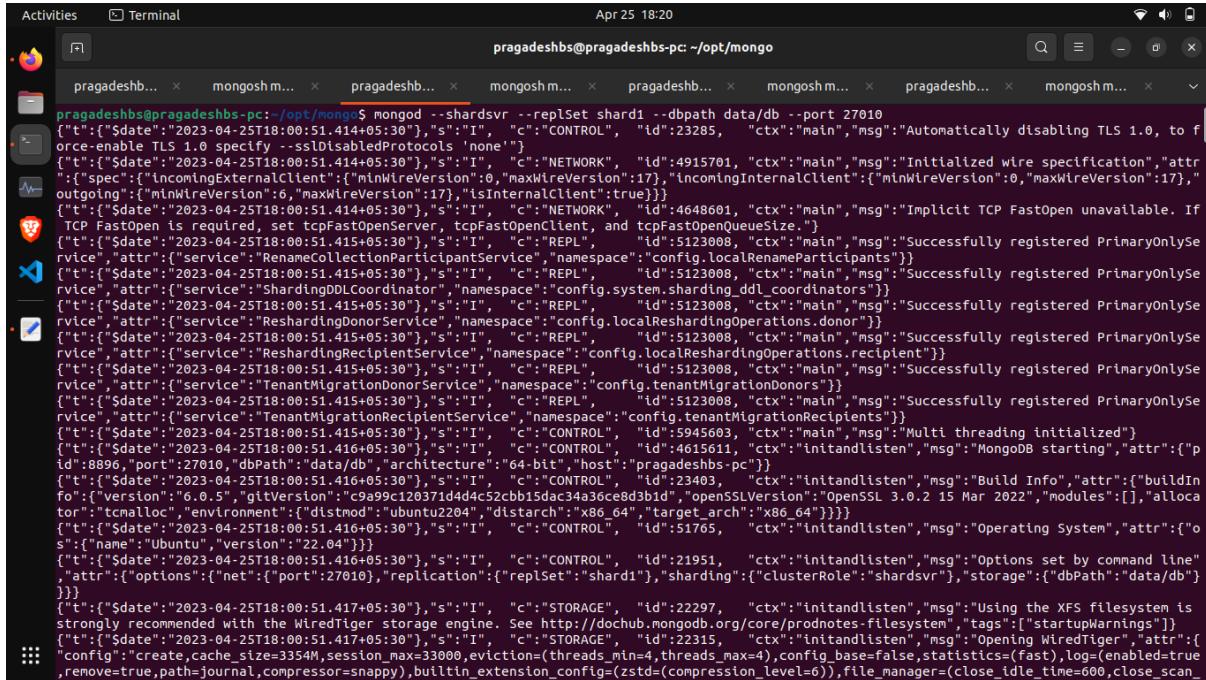
Output:

```
Activities Terminal mongosh mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000 Apr 25 18:19
pragadeshb... x mongosh m... x pragadeshb... x mongosh m... x pragadeshb... x mongosh m... x pragadeshb... x mongosh m...
test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:27019',
  ok: 1,
  lastCommittedOpTime: Timestamp({ t: 1682425546, i: 1 })
}
config [direct: other] test>
config [direct: primary] test> rs.status();
{
  set: 'config',
  date: ISODate("2023-04-25T12:27:30.914Z"),
  myState: 1,
  term: Long("1"),
  syncSourceHost: '',
  syncSourceId: -1,
  configsvr: true,
  heartbeatIntervalMillis: Long("2000"),
  majorityVoteCount: 1,
  writeMajorityCount: 1,
  votingMembersCount: 1,
  writableVotingMembersCount: 1,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    lastCommittedWallTime: ISODate("2023-04-25T12:27:30.530Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    appliedOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2023-04-25T12:27:30.530Z"),
    lastDurableWalltime: ISODate("2023-04-25T12:27:30.530Z")
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1682425605, i: 1 }),
  electionCandidateMetrics: [
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate("2023-04-25T12:25:46.349Z"),
    lastElectionTerm: 1
  ],
  opTime: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    lastCommittedWallTime: ISODate("2023-04-25T12:27:30.530Z"),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    appliedOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    durableOpTime: { ts: Timestamp({ t: 1682425650, i: 1 }), t: Long("1") },
    lastAppliedWallTime: ISODate("2023-04-25T12:27:30.530Z"),
    lastDurableWalltime: ISODate("2023-04-25T12:27:30.530Z")
  }
}
```

Start The First Shard:

```
mongod --shardsvr --replicaSet shard1 --dbpath data/db1 --port 27010
```

Output:

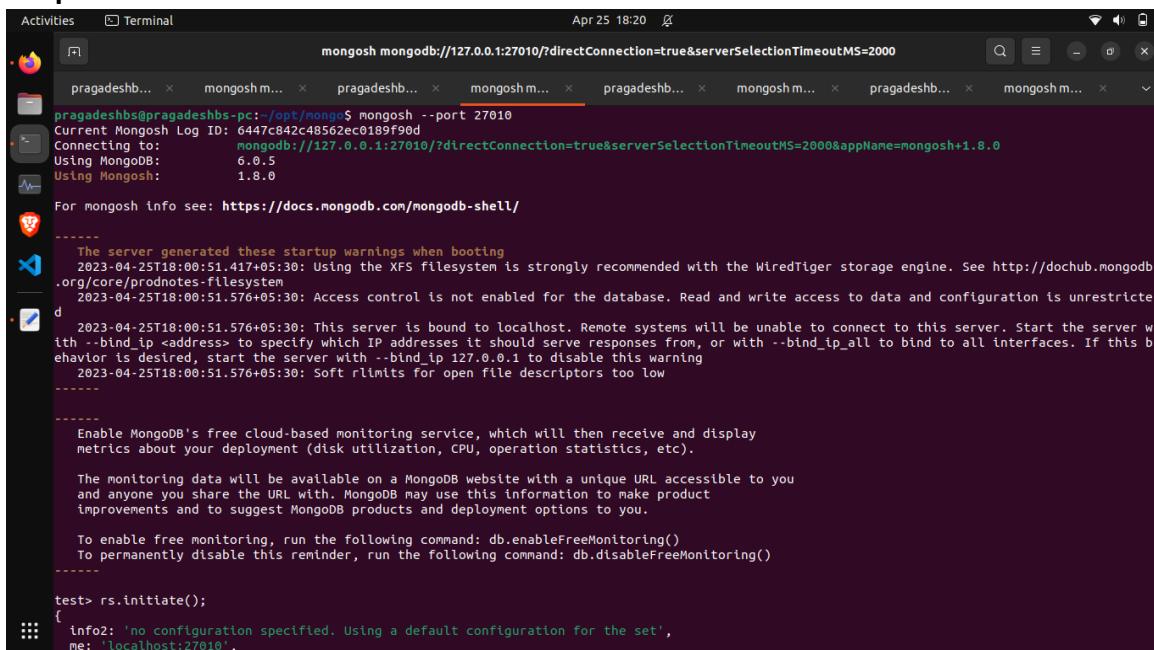


```
Activities Terminal Apr 25 18:20
pragadeshb... mongosh m... pragadeshb... mongosh m... pragadeshb... mongosh m... pragadeshb... mongosh m...
pragadeshb@pragadeshb-PC:/opt/mongo$ mongod --shardsvr --replicaSet shard1 --dbpath data/db --port 27010
{"t": {"$date": "2023-04-25T18:00:51.414+05:30"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
[{"t": {"$date": "2023-04-25T18:00:51.414+05:30"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "main", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "outgoing": {"minWireVersion": 6, "maxWireVersion": 17}, "isInternalClient": true}}}
[{"t": {"$date": "2023-04-25T18:00:51.414+05:30"}, "s": "I", "c": "NETWORK", "id": 4648601, "ctx": "main", "msg": "Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "RenameCollectionParticipantService", "namespace": "config.localRenameParticipants"}}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ShardingDDLCoordinator", "namespace": "config.system.sharding_ddl_coordinators"}}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ReshardingDonorService", "namespace": "config.localReshardingOperations.donor"}}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ReshardingRecipientService", "namespace": "config.localReshardingOperations.recipient"}}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationRecipientService", "namespace": "config.tenantMigrationRecipients"}}
[{"t": {"$date": "2023-04-25T18:00:51.415+05:30"}, "s": "I", "c": "CONTROL", "id": 5945603, "ctx": "main", "msg": "Multi threading initialized"}
[{"t": {"$date": "2023-04-25T18:00:51.416+05:30"}, "s": "I", "c": "CONTROL", "id": 4615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": 8896, "port": 27010, "dbPath": "data/db", "architecture": "64-bit", "host": "pragadeshb-PC"}}
[{"t": {"$date": "2023-04-25T18:00:51.416+05:30"}, "s": "I", "c": "CONTROL", "id": 23403, "ctx": "initandlisten", "msg": "Build Info", "attr": {"buildInfo": {"version": "6.0.5", "gitVersion": "c9a99c120371d4d4c52ccb15dac34a36ce8d3b1d", "openSSLVersion": "OpenSSL 3.0.2 15 Mar 2022", "modules": [], "allocator": "tcmalloc", "environment": {"distmod": "ubuntu2204", "distarch": "x86_64", "target_arch": "x86_64"}}, "options": {"net": {"port": 27010}, "replication": {"replicaSet": "shard1"}, "sharding": {"clusterRole": "shardsvr"}, "storage": {"dbPath": "data/db"}]}
[{"t": {"$date": "2023-04-25T18:00:51.417+05:30"}, "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": ["startupWarnings"]}
[{"t": {"$date": "2023-04-25T18:00:51.417+05:30"}, "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": {"create_cache_size": 3354M, "session_max": 33000, "eviction": "threads_min=4,threads_max=4", "config_base": false, "statistics": "fast", "log": {"enabled": true, "remove": true, "path": "journal", "compressor": "snappy"}, "builtin_extension_config": {"zstd": {"compression_level": 6}}, "file_manager": {"close_idle_time": 600, "close_scan_time": 1000}}}
[{"t": {"$date": "2023-04-25T18:00:51.416+05:30"}, "s": "I", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"name": "Ubuntu", "version": "22.04"}}, "tags": ["startupWarnings"]}
[{"t": {"$date": "2023-04-25T18:00:51.416+05:30"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"port": 27010}, "replication": {"replicaSet": "shard1"}, "sharding": {"clusterRole": "shardsvr"}, "storage": {"dbPath": "data/db"}}, "tags": ["startupWarnings"]}]
[{"t": {"$date": "2023-04-25T18:00:51.417+05:30"}, "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": ["startupWarnings"]}
[{"t": {"$date": "2023-04-25T18:00:51.417+05:30"}, "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": {"create_cache_size": 3354M, "session_max": 33000, "eviction": "threads_min=4,threads_max=4", "config_base": false, "statistics": "fast", "log": {"enabled": true, "remove": true, "path": "journal", "compressor": "snappy"}, "builtin_extension_config": {"zstd": {"compression_level": 6}}, "file_manager": {"close_idle_time": 600, "close_scan_time": 1000}}}
[{"t": {"$date": "2023-04-25T18:00:51.417+05:30"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"port": 27010}, "replication": {"replicaSet": "shard1"}, "sharding": {"clusterRole": "shardsvr"}, "storage": {"dbPath": "data/db"}}, "tags": ["startupWarnings"]}]
```

Connect To The First Shard:

```
mongosh --port 27010
```

Output:



```
Activities Terminal Apr 25 18:20
pragadeshb... mongosh m... pragadeshb... mongosh m... pragadeshb... mongosh m... pragadeshb... mongosh m...
pragadeshb@pragadeshb-PC:/opt/mongo$ mongosh --port 27010
Current Mongosh Log ID: 6447c842c48562ec089f90d
Connecting to: mongod://127.0.0.1:27010/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8.0
Using MongoDB: 6.0.5
Using Mongosh: 1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2023-04-25T18:00:51.417+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-04-25T18:00:51.576+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
d
2023-04-25T18:00:51.576+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2023-04-25T18:00:51.576+05:30: Soft rlimits for open file descriptors too low
-----

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

-----
test> rs.initiate();
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:27010',
```

Initiate The First Shard Replica Set:

```
rs.initiate()
```

Output:

```
pragadeshb@pragadeshb-OptiPlex-5090: ~ /opt/mongo
[{"t": {"$date": "2023-04-25T18:01:05.803+05:30"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}, {"t": {"$date": "2023-04-25T18:01:05.803+05:30"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "main", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 17}}}, {"t": {"$date": "2023-04-25T18:01:05.805+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "RenameCollectionParticipantsService", "namespace": "config.localRenameParticipants"}}, {"t": {"$date": "2023-04-25T18:01:05.806+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ShardingDDLCoordinator", "namespace": "config.system.sharding_ddl_coordinators"}}, {"t": {"$date": "2023-04-25T18:01:05.806+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ReshardingDonorService", "namespace": "config.localReshardingOperations.donor"}}, {"t": {"$date": "2023-04-25T18:01:05.806+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "ReshardingRecipientsService", "namespace": "config.localReshardingOperations.recipient"}}, {"t": {"$date": "2023-04-25T18:01:05.806+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}, {"t": {"$date": "2023-04-25T18:01:05.806+05:30"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationRecipientsService", "namespace": "config.tenantMigrationRecipients"}}, {"t": {"$date": "2023-04-25T18:01:05.806+05:30"}, "s": "I", "c": "CONTROL", "id": 5945603, "ctx": "main", "msg": "Multi threading initialized"}, {"t": {"$date": "2023-04-25T18:01:05.807+05:30"}, "s": "I", "c": "CONTROL", "id": 4615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": 8970, "port": 27011, "dbPath": "/data/db2", "architecture": "64-bit", "host": "pragadeshb-OptiPlex-5090"}}, {"t": {"$date": "2023-04-25T18:01:05.807+05:30"}, "s": "I", "c": "CONTROL", "id": 23403, "ctx": "initandlisten", "msg": "Build Info", "attr": {"buildInfo": {"version": "6.0.5", "gitVersion": "c9a99c120371d4d4c52ccb15ac3a36ce8d3bid", "openSSLVersion": "OpenSSL 3.0.2 15 Mar 2022", "modules": [], "allocator": "tcmalloc", "environment": {"distmod": "ubuntu2204", "distarch": "x86_64", "target_arch": "x86_64"}}, {"t": {"$date": "2023-04-25T18:01:05.807+05:30"}, "s": "I", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"name": "Ubuntu", "version": "22.04"}}, {"t": {"$date": "2023-04-25T18:01:05.807+05:30"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {"net": {"port": 27011}, "replication": {"replset": "shard2"}, "sharding": {"clusterRole": "shardsvr"}, "storage": {"dbPath": "/data/db2"}}, {"t": {"$date": "2023-04-25T18:01:05.808+05:30"}, "s": "I", "c": "STORAGE", "id": 22297, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem", "tags": ["startupWarnings"]}, {"t": {"$date": "2023-04-25T18:01:05.808+05:30"}, "s": "I", "c": "STORAGE", "id": 22315, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"config": {"create,cache_size=3354M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,remove=true,path=journal,compressor=snappy),builtin_extension_config=(zstd=(compression_level=6)),ftle_manager=(close_idle_time=600,close_scan_
```

Start The Second Shard:

```
mongod --shardsvr --replSet shard2 --dbpath data/db2 --port 27011
```

Connect To The Second Shard:

```
mongosh --port 27011
```

Initiate The Second Shard Replica Set:

```
rs.initiate()
```

Start The Mongos Router:

```
mongos --configdb config/127.0.0.1:27019
```

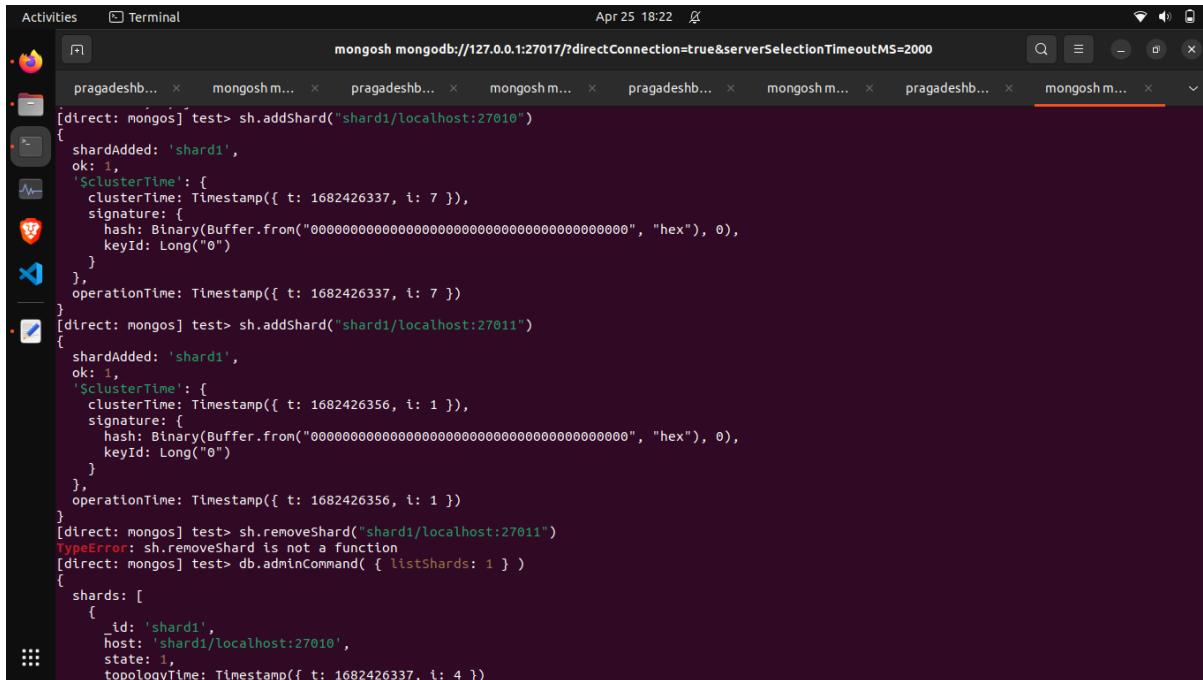
Output:

```
pragadeshb@pragadeshb-OptiPlex-5090: ~ /opt/mongo
[{"t": {"$date": "2023-04-25T18:05:49.927Z"}, "s": "I", "c": "SHARDING", "id": 24132, "ctx": "-", "msg": "Running on sharding servers should only be done for testing purposes and is not recommended for production."}, {"t": {"$date": "2023-04-25T18:05:49.931+05:30"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "-", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}, {"t": {"$date": "2023-04-25T18:05:49.931+05:30"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "main", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 17}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 17}}}, {"t": {"$date": "2023-04-25T18:05:49.933+05:30"}, "s": "I", "c": "REPL", "id": 4648601, "ctx": "main", "msg": "Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize.", {"t": {"$date": "2023-04-25T18:05:49.933+05:30"}, "s": "I", "c": "HEALTH", "id": 5935593, "ctx": "main", "msg": "Database Read and write access to data and configuration is unrestricted", "tags": ["startupWarnings"]}
```

Connect To The Mongos Router And Add Shards:

```
mongosh
sh.addShard("shard1/localhost:27010")
sh.addShard("shard2/localhost:27011")
```

Output:

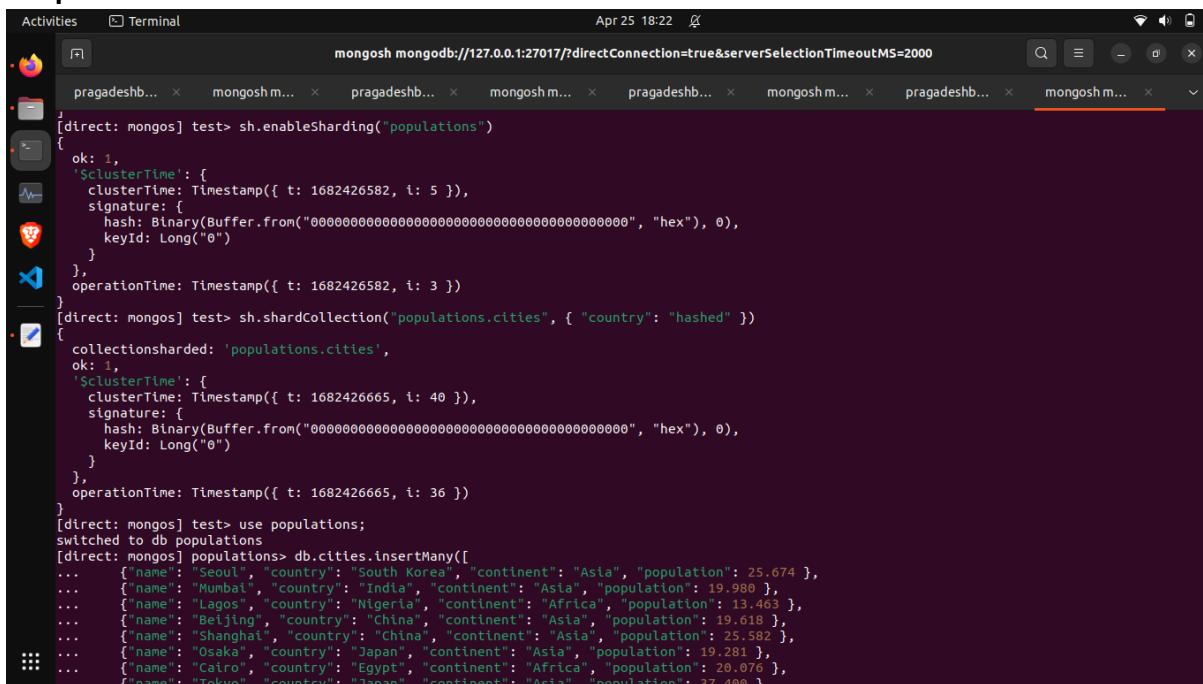


```
Activities Terminal mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
[direct: mongos] test> sh.addShard("shard1/localhost:27010")
{
  shardAdded: 'shard1',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1682426337, i: 7 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1682426337, i: 7 })
}
[direct: mongos] test> sh.addShard("shard1/localhost:27011")
{
  shardAdded: 'shard1',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1682426356, i: 1 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1682426356, i: 1 })
}
[direct: mongos] test> sh.removeShard("shard1/localhost:27011")
TypeError: sh.removeShard is not a function
[direct: mongos] test> db.adminCommand( { listShards: 1 } )
{
  shards: [
    {
      _id: 'shard1',
      host: 'shard1/localhost:27010',
      state: 1,
      topologyTime: Timestamp({ t: 1682426337, i: 4 })
    }
  ]
}
```

Enable Sharding On The Database:

```
sh.enableSharding("populations")
sh.shardCollection("populations.cities", { "country": "hashed" })
```

Output:



```
Activities Terminal mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
pragadeshb... mongosh.m... pragadeshb... mongosh.m... pragadeshb... mongosh.m... pragadeshb... mongosh.m...
[direct: mongos] test> sh.enableSharding("populations")
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1682426582, i: 5 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1682426582, i: 3 })
}
[direct: mongos] test> sh.shardCollection("populations.cities", { "country": "hashed" })
{
  collectionsharded: "populations.cities",
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1682426605, i: 40 }),
    signature: {
      hash: Binary(Buffer.from("00000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1682426605, i: 36 })
}
[direct: mongos] test> use populations;
switched to db populations
[direct: mongos] populations> db.cities.insertMany([
...   { "name": "Seoul", "country": "South Korea", "continent": "Asia", "population": 25.674 },
...   { "name": "Mumbai", "country": "India", "continent": "Asia", "population": 19.980 },
...   { "name": "Lagos", "country": "Nigeria", "continent": "Africa", "population": 13.463 },
...   { "name": "Beijing", "country": "China", "continent": "Asia", "population": 19.618 },
...   { "name": "Shanghai", "country": "China", "continent": "Asia", "population": 25.582 },
...   { "name": "Osaka", "country": "Japan", "continent": "Asia", "population": 19.281 },
...   { "name": "Cairo", "country": "Egypt", "continent": "Africa", "population": 20.076 },
...   { "name": "Tokyo", "country": "Japan", "continent": "Asia", "population": 37.498 },
... ])
```

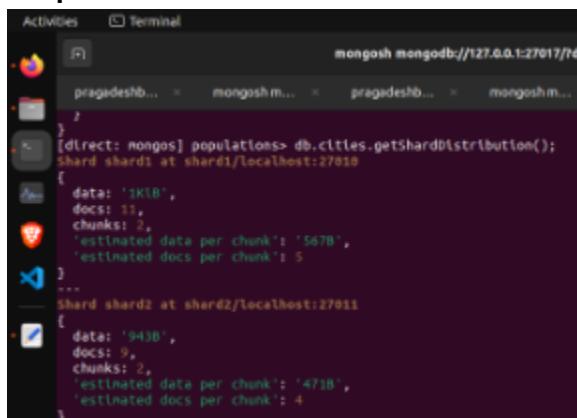
Insert Some Data:

```
db.cities.insertMany([
  {"name": "Seoul", "country": "South Korea", "continent": "Asia", "population": 25.674 },
  {"name": "Mumbai", "country": "India", "continent": "Asia", "population": 19.980 },
  {"name": "Lagos", "country": "Nigeria", "continent": "Africa", "population": 13.463 },
  {"name": "Beijing", "country": "China", "continent": "Asia", "population": 19.618 },
  {"name": "Shanghai", "country": "China", "continent": "Asia", "population": 25.582 },
  {"name": "Osaka", "country": "Japan", "continent": "Asia", "population": 19.281 },
  {"name": "Cairo", "country": "Egypt", "continent": "Africa", "population": 20.076 },
  {"name": "Tokyo", "country": "Japan", "continent": "Asia", "population": 37.400 },
  {"name": "Karachi", "country": "Pakistan", "continent": "Asia", "population": 15.400 },
  {"name": "Dhaka", "country": "Bangladesh", "continent": "Asia", "population": 19.578 },
  {"name": "Rio de Janeiro", "country": "Brazil", "continent": "South America", "population": 13.293 },
  {"name": "S\u00e3o Paulo", "country": "Brazil", "continent": "South America", "population": 21.650 },
  {"name": "Mexico City", "country": "Mexico", "continent": "North America", "population": 21.581 },
  {"name": "Delhi", "country": "India", "continent": "Asia", "population": 28.514 },
  {"name": "Buenos Aires", "country": "Argentina", "continent": "South America", "population": 14.967 },
  {"name": "Kolkata", "country": "India", "continent": "Asia", "population": 14.681 },
  {"name": "New York", "country": "United States", "continent": "North America", "population": 18.819 },
  {"name": "Manila", "country": "Philippines", "continent": "Asia", "population": 13.482 },
  {"name": "Chongqing", "country": "China", "continent": "Asia", "population": 14.838 },
  {"name": "Istanbul", "country": "Turkey", "continent": "Europe", "population": 14.751 }
])
```

Check The Sharding Status:

```
db.cities.getShardDistribution()
```

Output:



```
Activities Terminal
mongosh mongodb://127.0.0.1:27017/?direct=oplog
pragadeshb... x mongosh m... x pragadeshb... x mongosh m...
[direct: mongos] populations> db.cities.getShardDistribution();
Shard shard1 at shard1/localhost:27018
{
  data: '1MEB',
  docs: 11,
  chunks: 2,
  'estimated data per chunk': '567B',
  'estimated docs per chunk': 5
}
...
Shard shard2 at shard2/localhost:27019
{
  data: '943B',
  docs: 9,
  chunks: 2,
  'estimated data per chunk': '473B',
  'estimated docs per chunk': 4
}
```

Result:

Sharding has been successfully performed on a database with MongoDB.

HADOOP INSTALLATION

Aim:

To study about the architecture of Hadoop and its components.

Theory:

Hadoop is a framework written in Java that utilizes a large cluster of commodity hardware to maintain and store big size data. Hadoop works on MapReduce Programming. Algorithm that was introduced by Google.

The Hadoop Architecture Mainly consists of 4 components:

- i. MapReduce
- ii. HDFS(Hadoop distributed File System)
- iii. YARN(Yet Another Resource Framework)
- iv. Common Utilities or Hadoop Common

Components Of Hadoop Architecture

1. MapReduce

MapReduce is an Algorithm or a data structure that is based on the YARN framework. The major feature of MapReduce is to perform the distributed processing in parallel in a Hadoop cluster which Makes Hadoop fast. MapReduce has 2 main tasks which are divided phase-wise. In first phase, Map is utilized and in next phase Reduce is utilized. The Input is provided to the Map() function and then its output is used as an input to the Reduce() function to receive the final output. The Input is a set of Data. The Map() function breaks data blocks into tuples that are key-value pairs. These key-value pairs are now sent as input to the Reduce(). The Reduce() function then combines the broken key-value pairs based on its key value and form set of tuples. Some operation like sorting, summation, etc. Is performed which is then sent to the final Output Node which generates the final output.

2. HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system for Hadoop. It contains a master/slave architecture. This architecture consist of a single NameNode performs the role of master, and multiple DataNodes performs the role of a slave. Both NameNode and DataNode are capable enough to run on commodity machines. The Java language is used to develop HDFS. So any machine that supports Java language can easily run the NameNode and DataNode software. The NameNode manages the file system namespace by executing an operation like the opening, renaming and closing the files. The HDFS cluster contains several DataNodes each

of which contains multiple data blocks that store data. The DataNodes read and write requests from the file system's clients.

3. YARN

YARN is a Framework on which MapReduce works. YARN performs 2 operations that are Job scheduling and Resource Management. The Purpose of Job scheduler is to divide a big task into small jobs so that each job can be assigned to various slaves in a Hadoop cluster and processing can be maximized. Job Scheduler also keeps track of which job is important, which job has more priority, dependencies between the jobs and all the other information like job timing, etc. And the use of resource manager is to manage all the resources that are made available for running a Hadoop cluster.

4. Hadoop Common

Hadoop common or Common utilities are nothing but Java library and Java files or the java scripts that are needed for all the other components present in a Hadoop cluster. These utilities are used by HDFS, YARN, and MapReduce for running the cluster. Hadoop Common verifies that Hardware failure in a Hadoop cluster is common so it needs to be solved automatically in software by Hadoop Framework.

Installation:

Download Hadoop And Java:

```
curl https://dlcdn.apache.org/hadoop/common/hadoop-3.3.5/hadoop-3.3.5.tar.gz > hadoop  
tar -xzvf hadoop
```

Output:

```
$curl https://dlcdn.apache.org/hadoop/common/hadoop-3.3.5/hadoop-3.3.5.tar.gz > hadoop  
% Total    % Received % Xferd  Average Speed   Time     Time      Current  
                                         Dload  Upload   Total Spent  Left Speed  
100  673M  100  673M    0     0  5471k      0  0:02:06  0:02:06 --:-- 4259k  
$ls  
apache-cassandra-4.1.1  hadoop  jdk1.8.0_202  
$tar -xzf hadoop  
$rm hadoop  
$ls  
apache-cassandra-4.1.1  hadoop-3.3.5  jdk1.8.0_202
```

Set The Path In .Bashrc File:

```
export JAVA_HOME=/user/pragadeshb/jdk1.8.0_202  
export PATH=$HOME/bin:$JAVA_HOME/bin:$PATH
```

Output:

```
export JAVA_HOME=~/opt/jdk1.8.0_202
export HADOOP_HOME=~/opt/hadoop-3.3.5
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$PATH
"~/.bashrc" 121L, 3896B
```

Modify Hadoop Configuration Files:

```
etc/hadoop/core-site.xml
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:50000</value>
</property>

etc/hadoop/yarn-site.xml
<property>
<name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<description>The hostname of the RM.</description>
<name>yarn.resourcemanager.hostname</name>
<value>localhost</value>
</property>
<property>
<description>The address of the applications manager interface in the
RM.</description>
<name>yarn.resourcemanager.address</name>
<value>localhost:8032</value>
</property>
etc/hadoop/hdfs-site.xml
<property>
<name>dfs.namenode.name.dir</name>
<value>/home/pragadeshb/hadoop-dir/namenode-dir</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/pragadeshb/hadoop-dir/datanode-dir</value>
</property>
etc/hadoop/mapred-site.xml
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value></property>
```

Output:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
 Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

 http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>dfs.namenode.name.dir</name>
<value>~/hadoop-dir/namenode-dir</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>~/hadoop-dir/datanode-dir</value>
</property>
</configuration>
~
```

Enter Java Path In The Following Files:

etc/hadoop/hadoop-env.sh, etc/hadoop/mapred-env.sh, etc/hadoop/yarn-env.sh
export JAVA_HOME=/user/pragadeshbs/jdk1.8.0_202

etc/hadoop/slaves
Localhost

Format Namenode:

```
cd hadoop-2.9.1
bin/hadoop namenode -format
```

Output:

```
00 millis
2023-05-23 17:56:42,561 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2023-05-23 17:56:42,562 INFO util.GSet: VM type          = 64-bit
2023-05-23 17:56:42,563 INFO util.GSet: 0.029999999329447746% max memory 848 MB = 260.5
2023-05-23 17:56:42,563 INFO util.GSet: capacity        = 2^15 = 32768 entries
2023-05-23 17:56:42,630 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1544367798
2023-05-23 17:56:42,661 INFO common.Storage: Storage directory /home/pragadeshbs/hadoop
ted.
2023-05-23 17:56:42,721 INFO namenode.FSImageFormatProtobuf: Saving image file /home/pr
age.ckpt_00000000000000000000 using no compression
2023-05-23 17:56:43,054 INFO namenode.FSImageFormatProtobuf: Image file /home/pragade
sh.t_0000000000000000 of size 403 bytes saved in 0 seconds .
2023-05-23 17:56:43,088 INFO namenode.NNStorageRetentionManager: Going to retain 1 imag
2023-05-23 17:56:43,136 INFO namenode.FSNamesystem: Stopping services started for activ
2023-05-23 17:56:43,137 INFO namenode.FSNamesystem: Stopping services started for stand
2023-05-23 17:56:43,148 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 wh
2023-05-23 17:56:43,149 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at PragadeshPC./127.0.1.1*****
$
```

Start All Hadoop Related Services:

sbin/start-all.sh

Output:

```
$sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as pragadeshbs in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [PragadeSHPC]
Starting resourcemanager
Starting nodemanagers
$
```

WEBUI:

Output:

The screenshot shows the Hadoop Web UI interface. On the left, there's a sidebar with a yellow elephant icon and the word "hadoop". The main area has a title "About the Cluster". A sub-header "Cluster Metrics" is followed by a table with various cluster statistics. Below the table, several status messages are listed. The right side of the screen shows a "Cluster overview" section.

	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	1	0	0 B	8 GB	0 B	1	0	0	0	0	0

Cluster overview

Cluster ID: 1396885203337
ResourceManager state: STARTED
ResourceManager HA state: active
ResourceManager started on: 7-Apr-2014 21:10:03
ResourceManager version: 2.3.0 from 1567123 by jenkins source checksum c7fdde3f13261050c7dc3b9de345ce on 2014-02-11T13:52Z
Hadoop version: 2.3.0 from 1567123 by jenkins source checksum dfe46336fb6a044bc124392ec06b85 on 2014-02-11T13:40Z

Result:

Hadoop has been successfully installed on the system.

HIVE

Aim:

To install hive and perform CRUD operations.

Installation:**Modify The Hadoop File To Enable Map Reduce:**

```
/etc/hadoop/mapred-site.xml
<property>
<name>yarn.app.mapreduce.am.env</name>
<value>HADOOP_MAPRED_HOME=${full path of your hadoop distribution
directory}</value>
</property>
<property>
<name>mapreduce.map.env</name>
<value>HADOOP_MAPRED_HOME=${full path of your hadoop distribution
directory}</value>
</property>
<property>
<name>mapreduce.reduce.env</name>
<value>HADOOP_MAPRED_HOME=${full path of your hadoop distribution
directory}</value>
</property>
```

Download Mysql For Metastore:

```
sudo apt install mysql-server
```

Download Hive:

```
curl https://dlcdn.apache.org/hive/hive-2.3.9/apache-hive-2.3.9-bin.tar.gz >
apache-hive-2.3.9-bin.tar.gz
tar -xzf apache-hive-2.3.9-bin.tar.gz
```

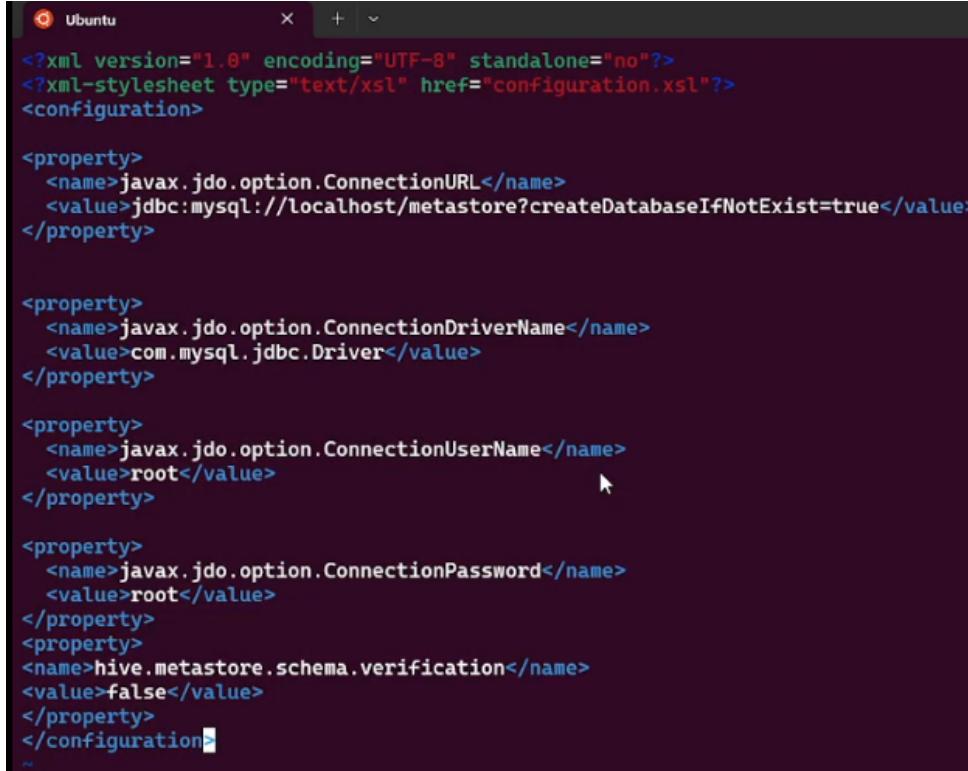
Output:

```
$curl https://dlcdn.apache.org/hive/hive-3.1.3/apache-hive-3.1.3-bin.tar.gz > apache-hive-3.1.3-bin.tar.gz
% Total    % Received % Xferd  Average Speed   Time   Time   Current
          Dload Upload   Total Spent   Left Speed
100 311M 100 311M    0     0  7295k      0  0:00:43  0:00:43 --:--:-- 6754k
$tar -xzf apache-hive-3.1.3-bin.tar.gz
$ls
apache-cassandra-4.1.1  apache-hive-3.1.3-bin  apache-hive-3.1.3-bin.tar.gz  hadoop-3.3.5  jdk1.8.0_202
$rm apache-hive-3.1.3-bin.tar.gz
```

Create The Hive-Site.Xml File And Add The Following:

```
touch conf/hive-site.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true</value>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>com.mysql.jdbc.Driver</value>
</property>
<property>
<name>javax.jdo.option.ConnectionUserName</name>
<value>root</value>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>root</value>
</property>
<property>
<name>hive.metastore.schema.verification</name>
<value>false</value>
</property>
</configuration>
```

Output:



A screenshot of a terminal window titled "Ubuntu". The window displays the XML configuration file "hive-site.xml" with syntax highlighting. The XML code is identical to the one provided in the previous code block, detailing the connection settings for MySQL metastore.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true</value>
</property>
<property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>com.mysql.jdbc.Driver</value>
</property>
<property>
<name>javax.jdo.option.ConnectionUserName</name>
<value>root</value>
</property>
<property>
<name>javax.jdo.option.ConnectionPassword</name>
<value>root</value>
</property>
<property>
<name>hive.metastore.schema.verification</name>
<value>false</value>
</property>
</configuration>
```

Create The Metastore Database In Mysql:

```
bin/schematool -dbType mysql -initSchema
```

Output:

```
$bin/schematool -dbType mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/pragadeshbs/opt/apache-hive-2.3.9-bin/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/pragadeshbs/opt/hadoop-3.3.5/share/hadoop/common/lib/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:      jdbc:mysql://localhost/metastore?createDatabaseIfNotExist=true
Metastore Connection Driver :   com.mysql.jdbc.Driver
Metastore connection User:     root
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is automatically registered via the SPI and manual loading of the driver class is generally not required.
Starting metastore schema initialization to 2.3.0
Initialization script hive-schema-2.3.0.mysql.sql
Initialization script completed
schemaTool completed
$
```

View Metadata In Mysql:

```
mysql -u root -p
show databases;
use metastore;
show tables;
```

Output:

```
+-----+
| SKEWED_STRING_LIST
| SKEWED_STRING_LIST_VALUES
| SKEWED_VALUES
| SORT_COLS
| TABLE_PARAMS
| TAB_COL_STATS
| TBLS
| TBL_COL_PRIVS
| TBL_PRIVS
| TXNS
| TXN_COMPONENTS
| TYPES
| TYPE_FIELDS
| VERSION
| WRITE_SET
+-----+
57 rows in set (0.00 sec)
```

```
mysql> |
```

Start Hive:

```
bin/hive
```

Output:

```
$bin/hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/pragadeshbs/opt/apache-hive-2.3.1-bin/lib/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/pragadeshbs/opt/hadoop-common-project/hadoop-common/2.7.1/lib/slf4j-log4j12.jar!/org/apache/logging/slf4j/Log4jLoggerFactory.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for further details.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/pragadeshbs/opt/apache-hive-2.3.1-bin/conf/hive-site.properties
2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future ve
line (i.e. spark, tez) or using Hive 1.X releases.

hive> █
```

Create A Table:

```
CREATE TABLE products (
    product_id INT,
    product_name STRING,
    price DOUBLE,
    quantity INT
);
```

Output:

```
hive> CREATE TABLE products (
    >     product_id INT,
    >     product_name STRING,
    >     price DOUBLE,
    >     quantity INT
    > );
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. It is recommended to use the new class instead.
OK
Time taken: 9.75 seconds
hive> INSERT INTO TABLE products
    > VALUES
    >     (1, 'Shirt', 29.99, 100),
    >     (2, 'Jeans', 49.99, 50),
    >     (3, 'Shoes', 79.99, 20),
    >     (4, 'Hat', 14.99, 75),
    >     (5, 'Socks', 4.99, 200);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future ve
line (i.e. spark, tez) or using Hive 1.X releases.
Query ID = pragadeshbh_20230523205445_4650a187-5d74-489b-90e4-e499540d0a9b
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1684855340156_0001, Tracking URL = http://PragadeSHPC:8088/proxy/jobs/1684855340156_0001
```

Insert Data Into The Hive Table:

```
INSERT INTO TABLE products
VALUES
(1, 'Shirt', 29.99, 100),
(2, 'Jeans', 49.99, 50),
(3, 'Shoes', 79.99, 20),
(4, 'Hat', 14.99, 75),
(5, 'Socks', 4.99, 200);
```

Output:

```
hive> INSERT INTO TABLE products
> VALUES
> (1, 'Shirt', 29.99, 100),
> (2, 'Jeans', 49.99, 50),
> (3, 'Shoes', 79.99, 20),
> (4, 'Hat', 14.99, 75),
> (5, 'Socks', 4.99, 200);
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future version (i.e. spark, tez) or using Hive 1.X releases.
Query ID = pragadeshbh_20230523205445_4650a187-5d74-489b-90e4-e499540d0a9b
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1684855340156_0001, Tracking URL = http://PragadeSHPC.:8088/proxy/
Kill Command = /home/pragadeSHPC/.ant/hadoop-2.2.5/bin/hadoop job -kill job_1684855340156_0001
```

Reading Data From Table:

```
SELECT * FROM products;
```

Output:

```
hive> SELECT * FROM products;
OK
1      Shirt    29.99   100
2      Jeans    49.99   50
3      Shoes    79.99   20
4      Hat      14.99   75
5      Socks    4.99    200
Time taken: 0.524 seconds, Fetched: 5 row(s)
hive> SELECT * FROM products WHERE price > 40;;
OK
2      Jeans    49.99   50
3      Shoes    79.99   20
Time taken: 1.037 seconds, Fetched: 2 row(s)
hive>
```

Updating Data In The Table (Using Insert Overwrite):

```
INSERT OVERWRITE TABLE products
SELECT
  CASE WHEN product_id = 1 THEN product_id ELSE product_id END,
  CASE WHEN product_id = 1 THEN 'New Shirt' ELSE product_name END,
  CASE WHEN product_id = 1 THEN 39.99 ELSE price END,
  CASE WHEN product_id = 1 THEN quantity ELSE quantity END
FROM products;
```

Output:

```
hive> INSERT OVERWRITE TABLE products
> SELECT
>   CASE WHEN product_id = 1 THEN product_id ELSE product_id END,
>   CASE WHEN product_id = 1 THEN 'New Shirt' ELSE product_name END,
>   CASE WHEN product_id = 1 THEN 39.99 ELSE price END,
>   CASE WHEN product_id = 1 THEN quantity ELSE quantity END
> FROM products;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in th
ine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = pragadeshbs_20230523210205_07af5314-ca12-4dfd-bb41-33e5040ff543
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1684855340156_0003, Tracking URL = http://PragadeshPC.:
Kill Command = /home/pragadeshbs/opt/hadoop-3.3.5/bin/hadoop job -kill jo
```

Deleting Data From The Table (Using Insert Overwrite):

```
INSERT OVERWRITE TABLE products
SELECT * FROM products
WHERE product_id <> 5;
```

Output:

```
hive> INSERT OVERWRITE TABLE products
> SELECT * FROM products
> WHERE product_id <> 5;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions
ine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = pragadeshbs_20230523210236_ac8587c5-57ee-47de-aaee-031f71a145dd
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1684855340156_0004, Tracking URL = http://PragadeshPC.:8088/proxy/applic
Kill Command = /home/pragadeshbs/opt/hadoop-3.3.5/bin/hadoop job -kill job_1684855340156_0
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-05-23 21:02:51,002 Stage-1 map = 0%,  reduce = 0%
2023-05-23 21:02:59,403 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.81 sec
MapReduce Total cumulative CPU time: 5 seconds 810 msec
Ended Job = job_1684855340156_0004
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:50000/user/hive/warehouse/products/.hive-staging_
363464-1/-ext-10000
Loading data to table default.products
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Cumulative CPU: 5.81 sec  HDFS Read: 4593 HDFS Write: 143 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 810 msec
OK
Time taken: 24.721 seconds
```

Result:

Hive has been successfully installed on the system and CRUD operations are performed.