Create a /etc/yum.repos.d/mongodb-org-4.0.repo file so that you can install MongoDB directly using yum:

[root@ip-172-31-89-234 ~]# touch /etc/yum.repos.d/mongodb-org-4.0.repo

[root@ip-172-31-89-234 ~]# vi /etc/yum.repos.d/mongodb-org-4.0.repo

[root@ip-172-31-89-234 ~]# cat /etc/yum.repos.d/mongodb-org-4.0.repo

[mongodb-org-4.0]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/4.0/x86\_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc

**Install the MongoDB packages**

sudo yum install -y mongodb-org

[root@ip-172-31-89-234 ~]# sudo yum install -y mongodb-org

#### Start MongoDB.

sudo service mongod start

vi /etc/mongod.conf

bindIp changed from 127.0.0.1 to 0.0.0.0

and then restart mongod service

service mongod restart

mongo

use databasename

db.collectionname.find()

show collections

<?xml version="1.0" encoding="UTF-8"?>  
  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.companyname</groupId>  
 <artifactId>SparkScalaExamples</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <name>SparkScalaExamples</name>  
 <!-- *FIXME change it to the project's website* -->  
 <url>http://www.example.com</url>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <maven.compiler.source>1.7</maven.compiler.source>  
 <maven.compiler.target>1.7</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.11</version>  
 <scope>test</scope>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.scala-lang/scala-library -->  
 <dependency>  
 <groupId>org.scala-lang</groupId>  
 <artifactId>scala-library</artifactId>  
 <version>2.11.12</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-core -->  
 <dependency>  
 <groupId>org.apache.spark</groupId>  
 <artifactId>spark-core\_2.11</artifactId>  
 <version>2.4.0</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-core -->  
 <dependency>  
 <groupId>org.apache.spark</groupId>  
 <artifactId>spark-sql\_2.11</artifactId>  
 <version>2.4.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.spark</groupId>  
 <artifactId>spark-catalyst\_2.11</artifactId>  
 <version>2.4.0</version>  
 </dependency>  
  
  
 <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>5.1.47</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.hadoop</groupId>  
 <artifactId>hadoop-aws</artifactId>  
 <version>2.7.3</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk -->  
 <dependency>  
 <groupId>com.amazonaws</groupId>  
 <artifactId>aws-java-sdk</artifactId>  
 <version>1.7.4</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk-s3 -->  
 <dependency>  
 <groupId>com.amazonaws</groupId>  
 <artifactId>aws-java-sdk-s3</artifactId>  
 <version>1.11.519</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core -->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-core</artifactId>  
 <version>2.6.5</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core -->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-annotations</artifactId>  
 <version>2.6.5</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core -->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-databind</artifactId>  
 <version>2.6.5</version>  
 </dependency>  
  
  
 <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-sql-kafka-0-10 -->  
 <dependency>  
 <groupId>org.apache.spark</groupId>  
 <artifactId>spark-sql-kafka-0-10\_2.11</artifactId>  
 <version>2.4.0</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/com.crealytics/spark-excel -->  
 <dependency>  
 <groupId>com.crealytics</groupId>  
 <artifactId>spark-excel\_2.11</artifactId>  
 <version>0.11.1</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->  
 <dependency>  
 <groupId>org.postgresql</groupId>  
 <artifactId>postgresql</artifactId>  
 <version>42.2.5</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.elasticsearch/elasticsearch-hadoop -->  
 <dependency>  
 <groupId>org.elasticsearch</groupId>  
 <artifactId>elasticsearch-hadoop</artifactId>  
 <version>6.4.0</version>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.mongodb.spark</groupId>  
 <artifactId>mongo-spark-connector\_2.11</artifactId>  
 <version>2.4.0</version>  
 </dependency>  
  
  
 <!-- https://mvnrepository.com/artifact/com.datastax.spark/spark-cassandra-connector -->  
 <dependency>  
 <groupId>com.datastax.spark</groupId>  
 <artifactId>spark-cassandra-connector\_2.11</artifactId>  
 <version>2.4.0</version>  
 </dependency>  
  
  
  
 </dependencies>  
  
 <build>  
 <sourceDirectory>src/main/scala</sourceDirectory>  
 <testSourceDirectory>src/test/scala</testSourceDirectory>  
 <plugins>  
 <plugin>  
 <groupId>org.scala-tools</groupId>  
 <artifactId>maven-scala-plugin</artifactId>  
  
 </plugin>  
  
 </plugins>  
  
  
 </build>  
</project>

**import** com.mongodb.spark.\_  
**import** com.mongodb.spark.config.{ReadConfig, WriteConfig}  
**import** org.apache.spark.sql.{DataFrame, SparkSession}  
**import** org.apache.spark.sql.types.{StringType, StructField, StructType}  
**object** SparkScalaWriteToMongoDB {  
 **def** main(args: Array[String]): Unit = {  
 **val** spark = SparkSession.*builder*()  
 .master("local")  
 .appName("MongoSparkConnectorIntro")  
 .config("spark.mongodb.input.uri", "mongodb://3.88.225.247/sparkscalatest.empcollection")  
 .config("spark.mongodb.output.uri", "mongodb://3.88.225.247/sparkscalatest.empcollection")  
 .getOrCreate()  
 **val** employeeJsonSchema=*StructType*(  
 *Seq*(  
 *StructField*("empid", StringType, **true**),  
 *StructField*("empname", StringType, **true**),  
 *StructField*("empsal", StringType, **true**)  
 )  
 )  
  
 **val** employeeDataFrame=spark.read.format("json").schema(employeeJsonSchema).load("src/main/resources/json-resources/employee1.json")  
 // employeeDataFrame.show()  
  
 **val** employeeWriteConfig = *WriteConfig*(*Map*("collection" -> "employee", "writeConcern.w" -> "majority"), *Some*(WriteConfig(spark.sparkContext)))  
 //MongoSpark.save(employeeDataFrame,employeeWriteConfig)  
 **val** readConfig = *ReadConfig*(*Map*("collection" -> "employee", "writeConcern.w" -> "majority"),*Some*(ReadConfig(spark.sparkContext)))  
 **val** employeeDataFrameFromMongoDB = MongoSpark.*load*(spark,readConfig)  
 employeeDataFrameFromMongoDB.show()  
  
 }  
}

**package** com.companyname.demo.mongodbjavaexample;

**import** java.io.File;

**import** java.io.IOException;

**import** org.apache.commons.io.FileUtils;

**import** com.mongodb.BasicDBObject;

**import** com.mongodb.DB;

**import** com.mongodb.DBCollection;

**import** com.mongodb.DBCursor;

**import** com.mongodb.DBObject;

**import** com.mongodb.~~Mongo~~;

**import** com.mongodb.util.~~JSON~~;

/\*\*

\* Hello world!

\*

\*/

**public** **class** App {

**public** **static** **void** main(String[] args) **throws** IOException {

// System.out.println( "Hello World!" );

String jsondata = FileUtils.~~readFileToString~~(**new** File("src/main/resources/employee.json"));

System.***out***.println(jsondata);

DBObject dbObject = (DBObject) ~~JSON~~.~~parse~~(jsondata);

~~Mongo~~ mongo = **new** ~~Mongo~~("3.88.225.247", 27017);

DB db = mongo.~~getDB~~("employeedb");

DBCollection collection = db.getCollection("employee");

//collection.insert(dbObject);

BasicDBObject allQuery = **new** BasicDBObject();

BasicDBObject getempid = **new** BasicDBObject();

getempid.put("empid", "100");

/\*DBCursor cursor = collection.find(allQuery, fields);

while (cursor.hasNext()) {

System.out.println(cursor.next());

}\*/

collection.remove(getempid);

}

}

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.demo</groupId>

<artifactId>mongodbjavaexample</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>mongodbjavaexample</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.6</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver -->

<dependency>

<groupId>org.mongodb</groupId>

<artifactId>mongo-java-driver</artifactId>

<version>3.9.1</version>

</dependency>

</dependencies>

</project>