**Server Ip address : 156.67.220.67**

**server.address=156.67.220.67 -- add to application.properties file**

**nohup java -jar test.jar & -- To run the java app permanently**

1. **sudo apt-get install openjdk-8-jre**
2. **sudo update-alternatives --config java (Select your version )**

**Nginx server install :-**

**sudo apt-get install nginx**

**Restart server :-**

**sudo service nginx restart­­­­**

**How to install mysql server :-**

sudo apt-get install mysql-server

Follow these links : <https://linuxize.com/post/how-to-install-mysql-on-ubuntu-18-04/>

And <https://linuxize.com/post/how-to-reset-a-mysql-root-password/>

INSERT INTO mysql.user (Host, User, Password) VALUES ('%', 'root', password('YOURPASSWORD'));

GRANT ALL ON \*.\* TO 'root'@'%' WITH GRANT OPTION;

**How to un-install mysql server :-**

1. sudo apt-get remove --purge mysql\*
2. sudo apt-get purge mysql\*
3. sudo apt-get autoremove
4. sudo apt-get autoclean
5. sudo apt-get remove dbconfig-mysql
6. sudo apt-get dist-upgrade

**View userList, Creating User, Setting Password and Host and Granting PRIVILEGES** :-

Select user, host from mysql.user;

create user predecode;

UPDATE mysql.user SET Host='localhost:3306' WHERE Host='localhost' AND User='predecode';

UPDATE mysql.db SET Host='localhost' WHERE Host='%' AND User='predecode';

UPDATE mysql.user SET authentication\_string = PASSWORD('Sumit@135') WHERE User = 'predecode' AND Host = 'localhost';

GRANT ALL PRIVILEGES ON database\_name.\* TO 'predecode'@'localhost';

FLUSH PRIVILEGES;

GRANT USAGE ON \*.\* TO 'predecode'@'localhost'

CREATE USER 'new\_user'@'localhost' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON \* . \* TO 'new\_user'@'localhost';

FLUSH PRIVILEGES;

GRANT CREATE, SELECT ON \* . \* TO 'user\_name'@'localhost';

REVOKE ALL PRIVILEGES ON \* . \* FROM 'user\_name'@'localhost';

Creating Schema :-

CREATE SCHEMA `predecode` ;

To See all schema :-

Show databases;

========================================================================

**How to install java on the server :-**

1. Open putty :-
2. Connect to the server with user name and password :-
3. Hit the following command
4. apt-get update && apt-get upgrade
5. apt-get install default-jre
6. apt-get install default-jdk

=========================================================================

**How To Install Git on the server :-**

1. Sudo apt update
2. Sudo apt install git

========================================================================

**How To Install NodeJs on the server :-**

1. Sudo apt install nodejs
2. Sudo npm install -g pm2 -- This command is used to run the server permanently.
3. Sudo apt install npm

========================================================================

netstat -ntpl :- to see which application is running on which port.

=======================================================================

PM2 :- Used for permanent run of a serve

pm2 stop

pm2 restart

pm2 delete

pm2 start “ng serve --host 0.0.0.0 --disableHostCheck” --name angular\_app\_name;

**pm2 monit** : To monitor logs, custom metrics, application information.

## **Import the public key used by the package management system**

wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -

## **Add Sources**

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list

## **Reload the local package database.**

sudo apt update

## **Install the MongoDB packages.**

sudo apt install -y mongodb-org

## **Start and verify the service**

sudo systemctl start mongodsudo systemctl status mongod

## **Enable the service start on every reboot**

sudo systemctl enable mongod

sudo rm /var/lib/mongodb/mongod.lock

mongod --repair

sudo service mongodb start

## **Start and verify the service**

sudo systemctl start mongodsudo systemctl status mongod

## **Enable the service start on every reboot**

sudo systemctl enable mongod

sudo service mongod stop

sudo apt-get purge mongodb-org\*

sudo rm -r /var/log/mongodb

sudo apt-get autoclean

sudo apt-get autoremove

sudo apt-get purge mongodb-org\*

sudo rm /lib/systemd/system/mongod.service

chown -R mongodb:mongodb /var/lib/mongodb

chown mongodb:mongodb /tmp/mongodb-27017.sock

1 — Go to the TMP directory: **cd /tmp**  
2 — Check if you have the mongodb sock file: **ls \*.sock**

3 — Change the user:group permission:

**chown mongodb:mongodb <YOUR\_SOCK>**  
4 — Start MongoDB: **sudo service mongod start**  
5 — Check the MongoDB status: **sudo service mongod status**

[Sudeep Timalsina](https://medium.com/u/9615abf9151a?source=post_page-----417ffc78cb11--------------------------------)

 added a important comment, during the step **3** if you get Operation Not Permited, execute it with **sudo chown mongodb:mongodb <YOUR\_SOCK>**

As the result, you should see something like this:

db.createUser(

{

user: "dbAdmin",

pwd: “DB@Admin”,*// or cleartext password*

roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]

}

)

mongo --port 27017 -u "myUserAdmin" -p "DB@Admin" --authenticationDatabase "admin"

db.grantRolesToUser("myUserAdmin",

[{ role: "read", db: "krishnasweets"}] );

Wget <https://springboot-java.s3.ap-south-1.amazonaws.com/KrishnaSweets-0.0.1-SNAPSHOT.jar> to get the jar from s3 into ec2 server

Nohup java –jar filename.jar to run java on ec2 server