

## Team Members:

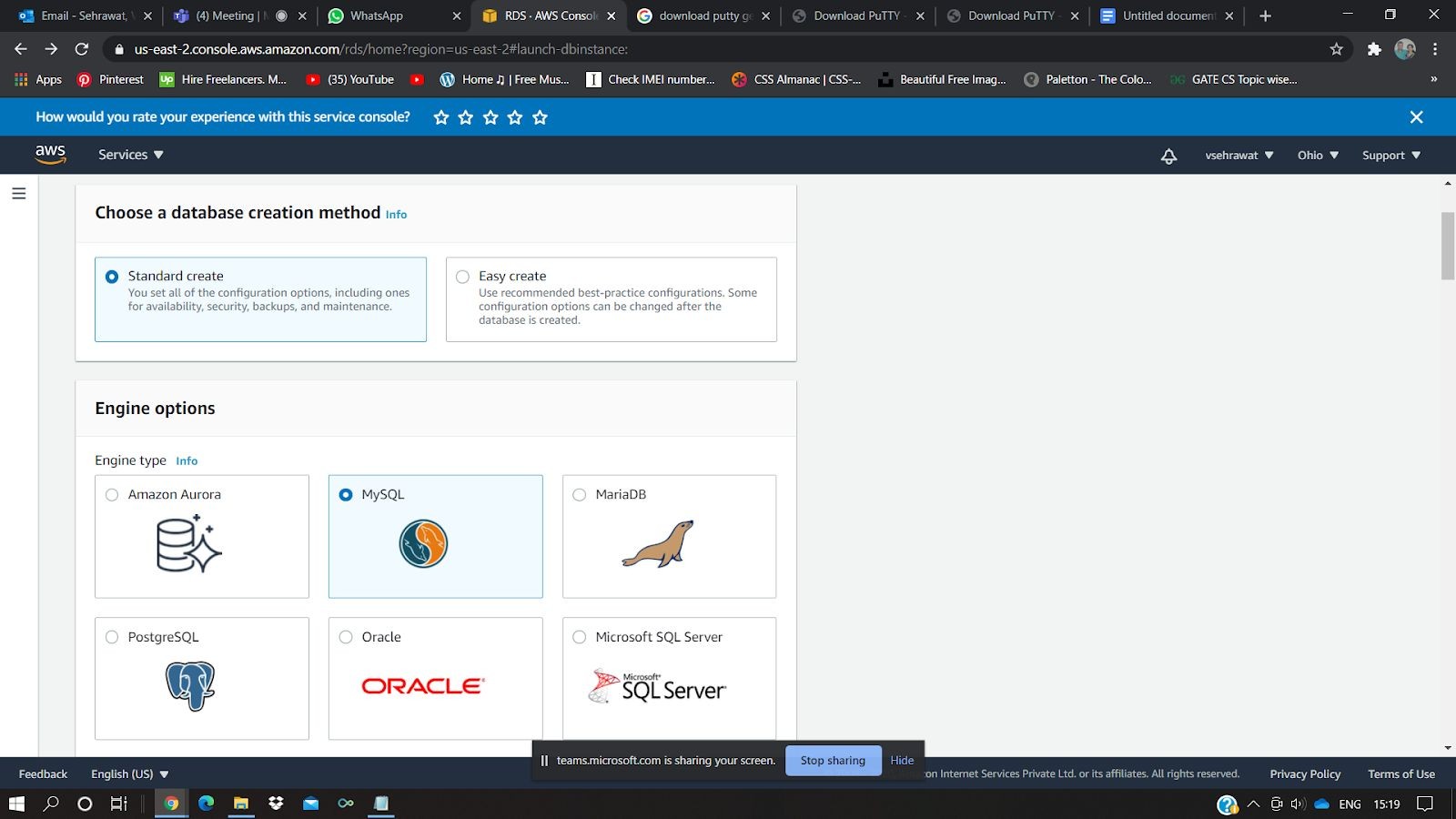
|  |
| --- |
| **Team: J\_FSD\_5** |
| Tirtha Kamble |
| Vaibhav Sehrawat |
| Sohaib Khan |
| Shivam Singh |

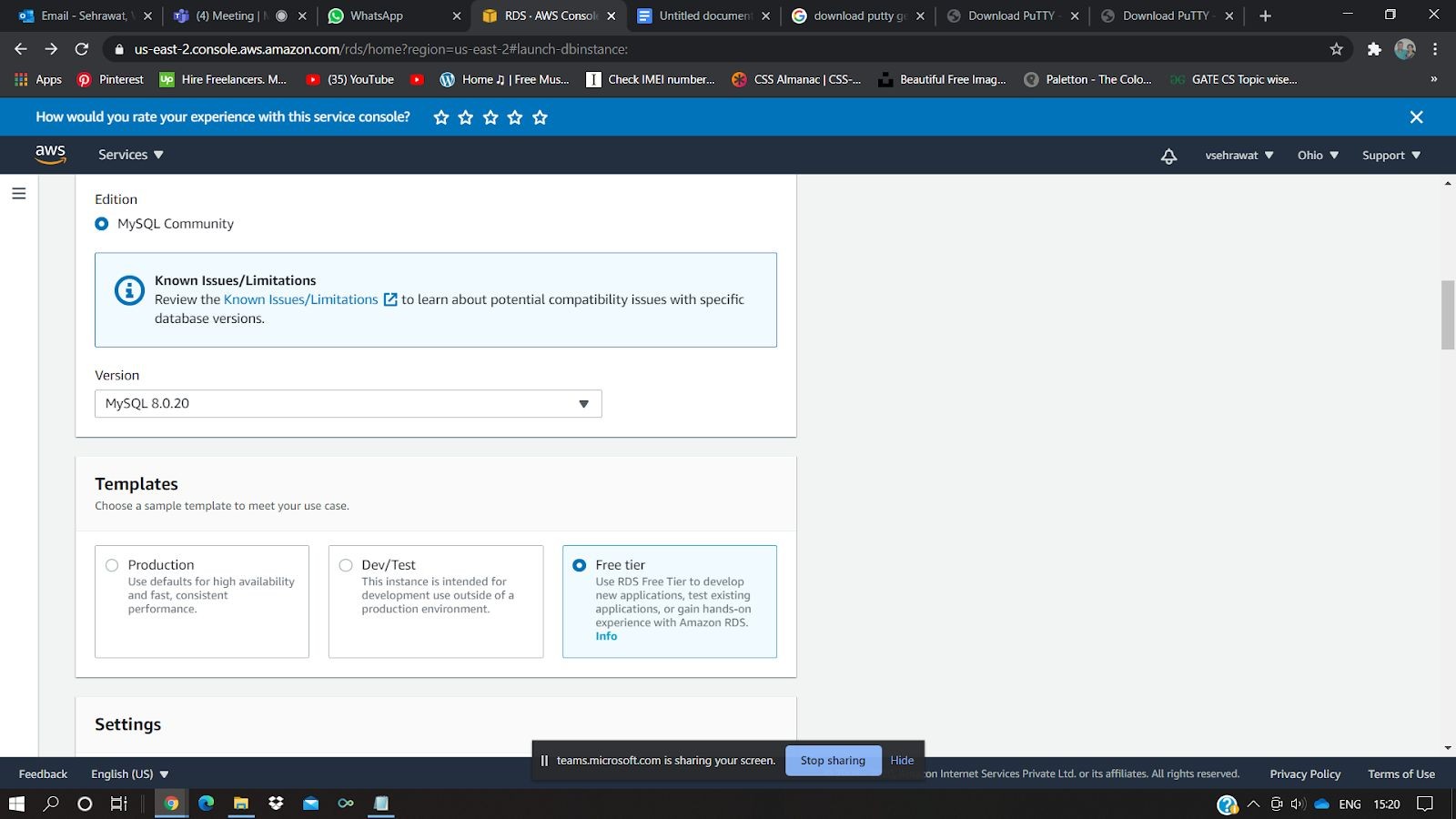
Deployment of Spring Boot microservices and Angular App on AWS with DevOps

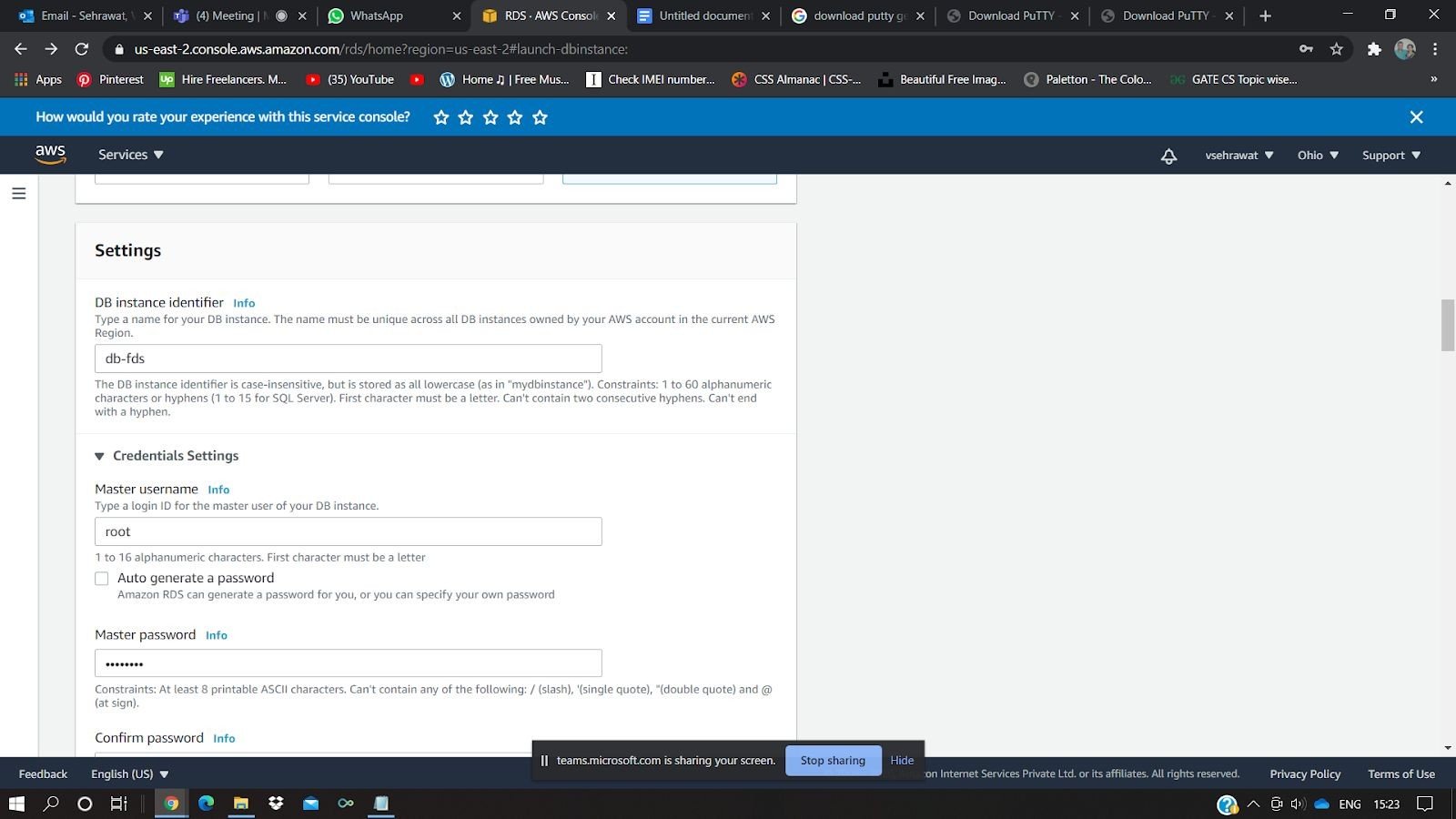
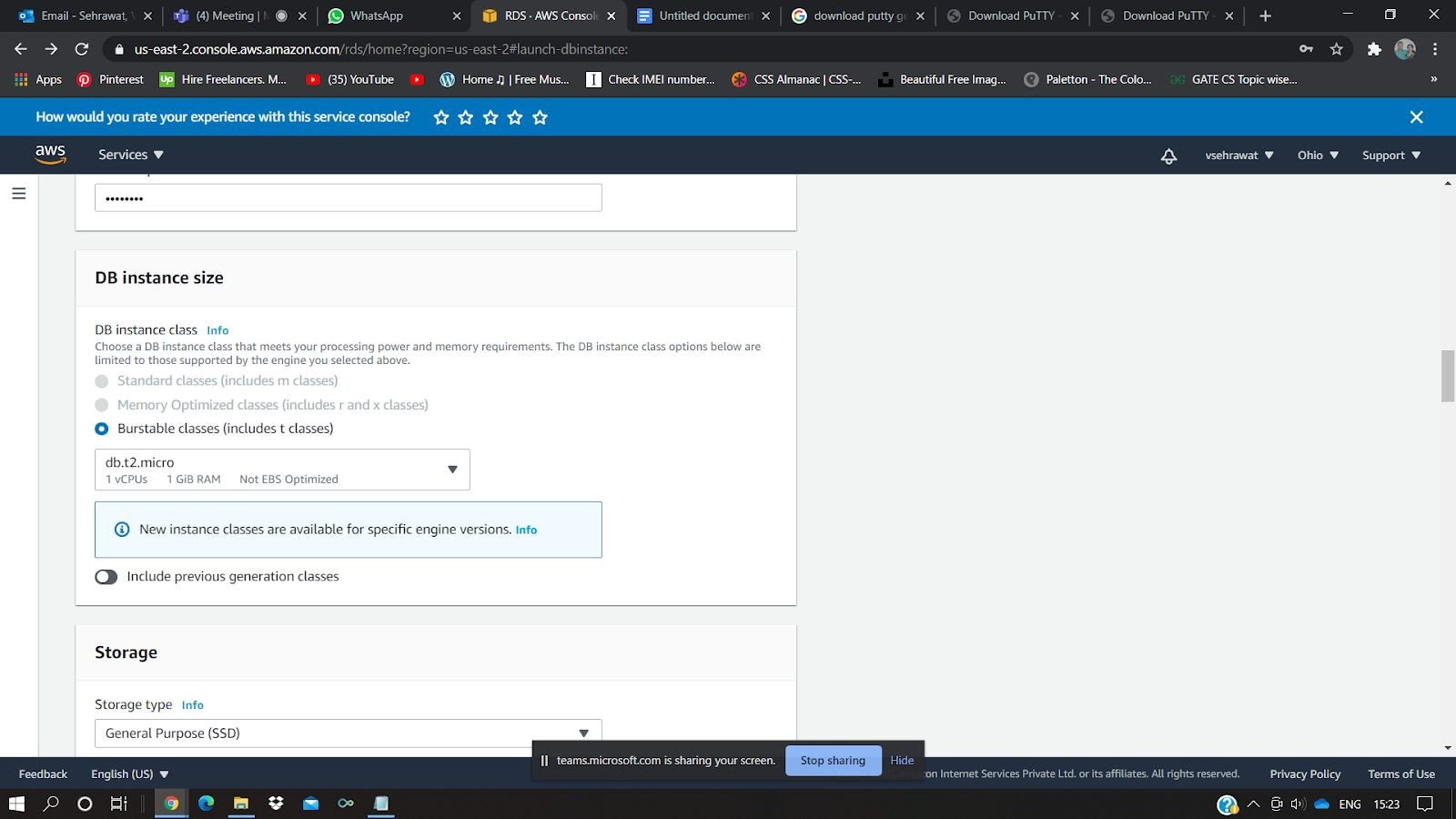
# Whole process in steps:

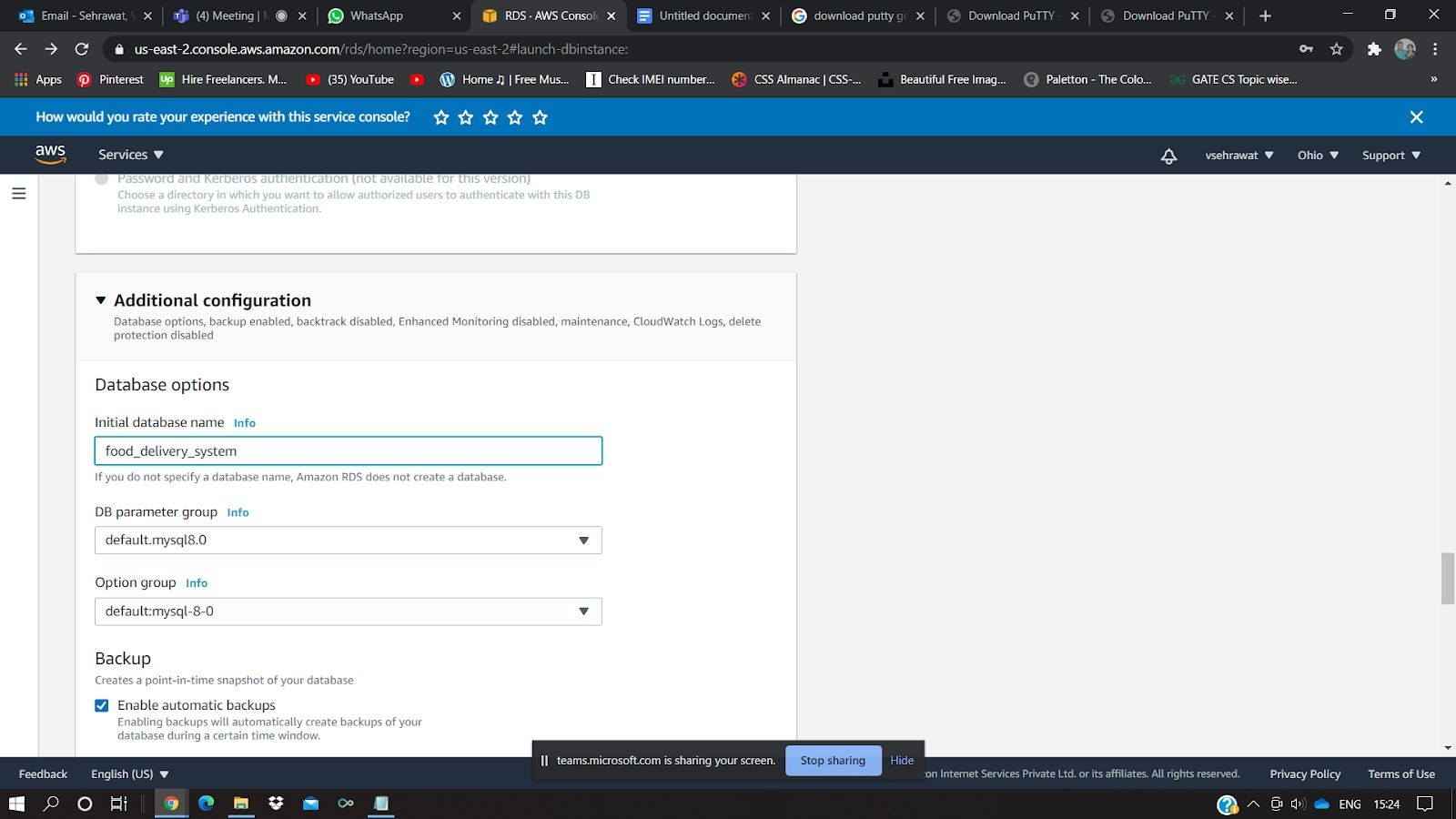
1. Create Spring Boot App all the microservices
2. Create Angular App
3. Create RDS MySql on AWS
4. Connect microservices with newly made RDS instance
5. Create EC2 instance - Install & Setup
   1. Install java1..8
   2. Setup java1.8
   3. Install tomcat9
   4. Setup tomcat9
      1. Edit webapp/\*\*/context.xml (both files)
      2. Edit server.xml (optional)
      3. Edit user\*\*.xml(to login)
6. Install and configure Jenkins on the local system
7. Build freestyle Jenkins project to upload jar/war and deploy it on EC2 tomcat

# Create RDS MySql on AWS:

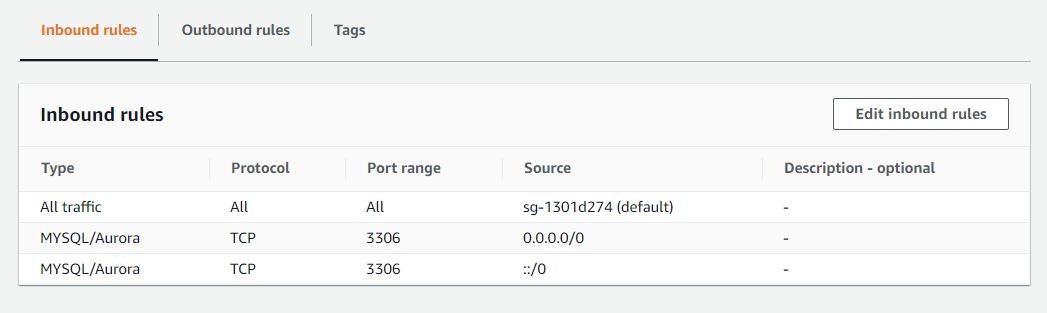
1. Create RDS instance in AWS

2. 

3.



1. Once the instance is ready. Copy endpoint URL of instance and
2. Update security group applied from the instance add Mysql type Inbound role at port 3306 accessible “everywhere”.



1. On cmd **telnet <end-point URL>** if it asking for password, means you can use it now
2. In spring application.properties paste the url at place of “localhost” in datasource url.
3. Run your spring application and check the connection by inserting some value and fetching it.

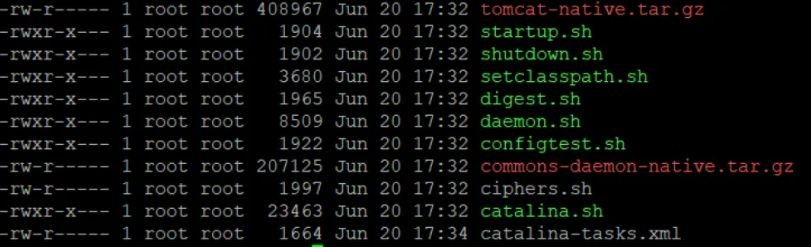
# Create EC2 instance -install and setup

1. Once instance is created in ec2 instance
2. Login to your virtual machine through PuTTY
3. Enter IP of your virtual machine and in auth choose .ppk file
4. Login as ec2-user
5. Now time to install and setup java & tomcat

## sudo yum install java-1.8.0

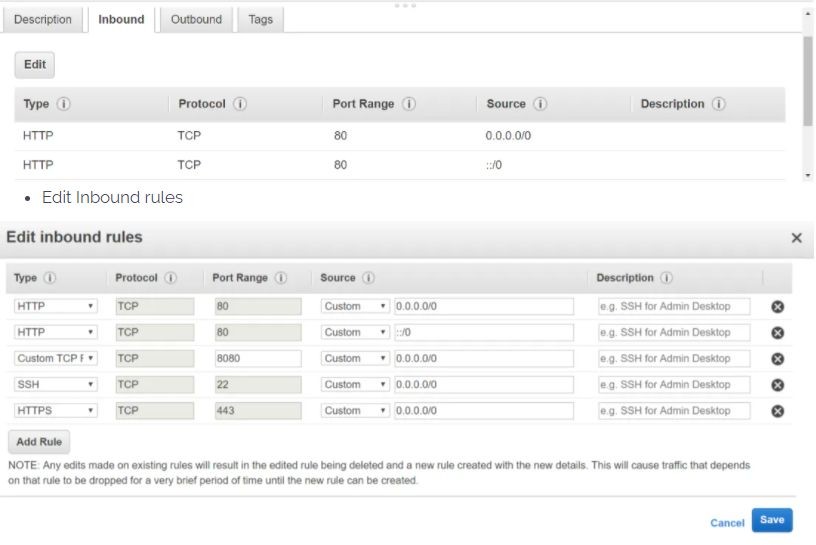
1. Java is installed, now to install tomcat, Got to apache tomcat site
2. Get the link of .tar.gz file of tomcat 9 copy it
3. In EC2 console: wget <tomcat 9-URL>
4. This will help download the file in your console Now to extract:

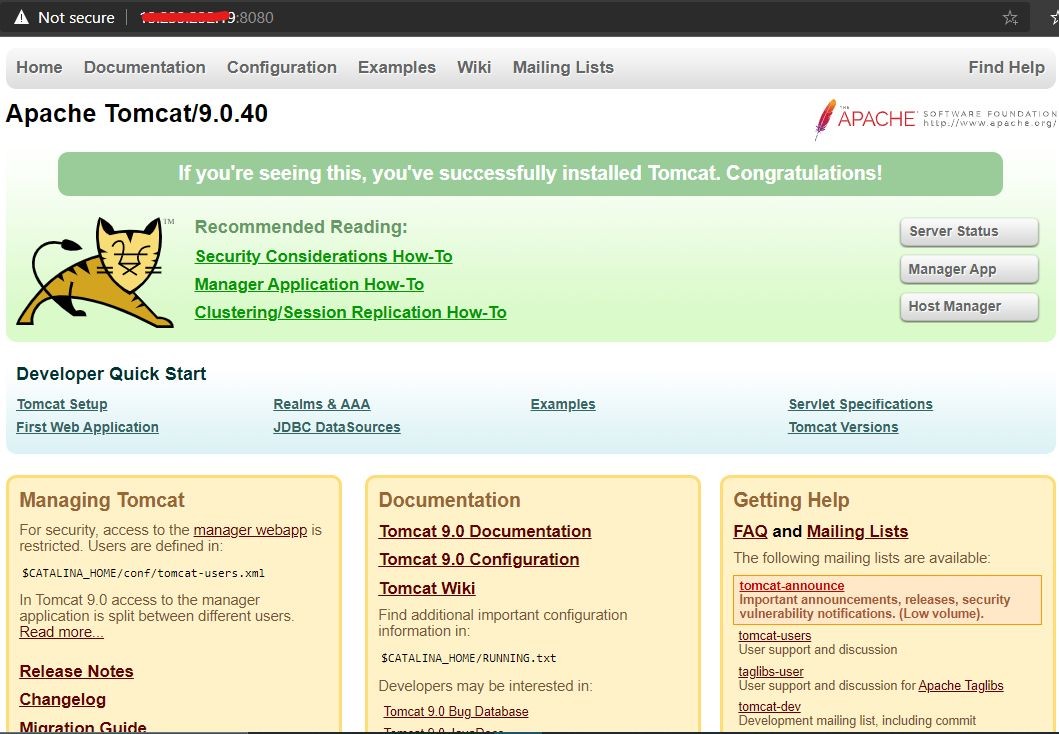
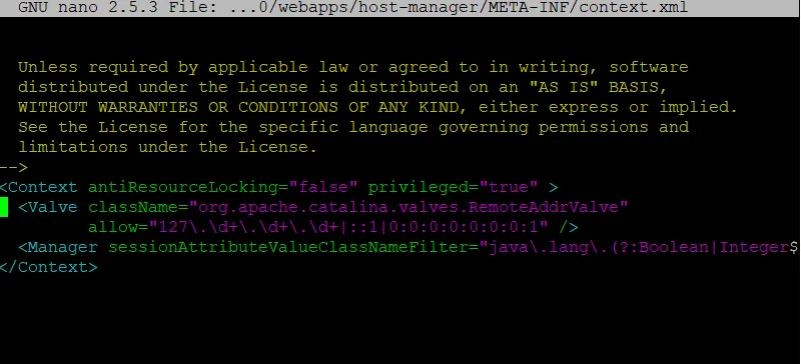
## tar -zvxf apache-tomcat-9.<version>.tar.gz

1. **yum install wget -y**
2. Browse to bin folder
3. ls -ltr
4. 

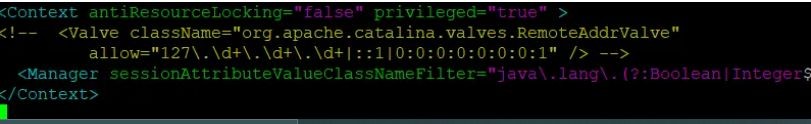
## chmod +x startup.sh

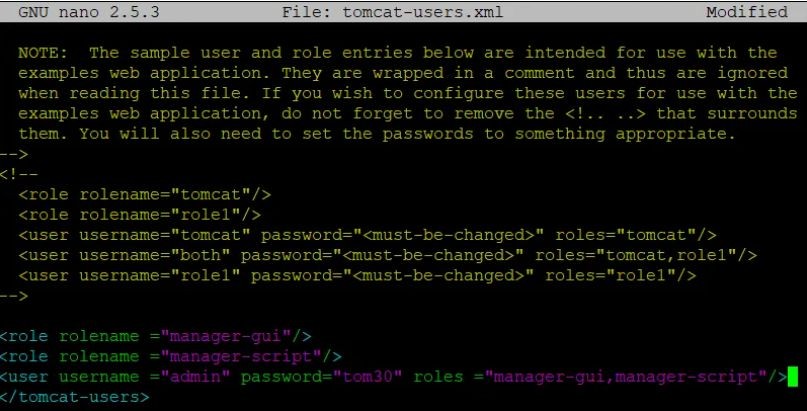
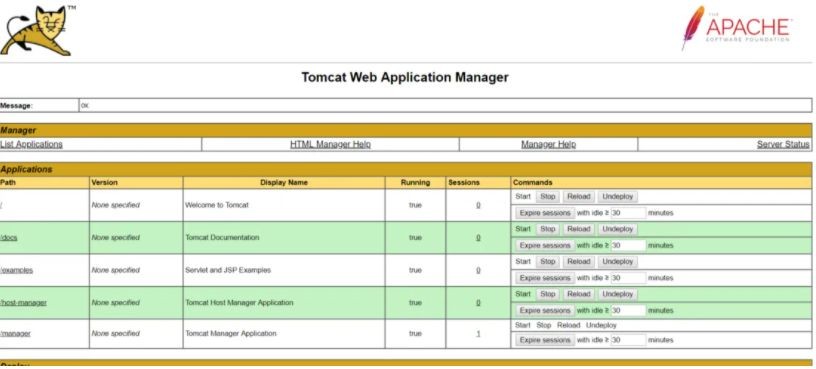
1. **chmod +x shutdown.sh**
2. **./startup.sh** this will startup the tomcat server at <your public IP>:8080
3. Now time to change inbound rules as follows in security group associated with your ec2 instanc



1. Now access in browser: <publicc IP>:8080
2. 21.

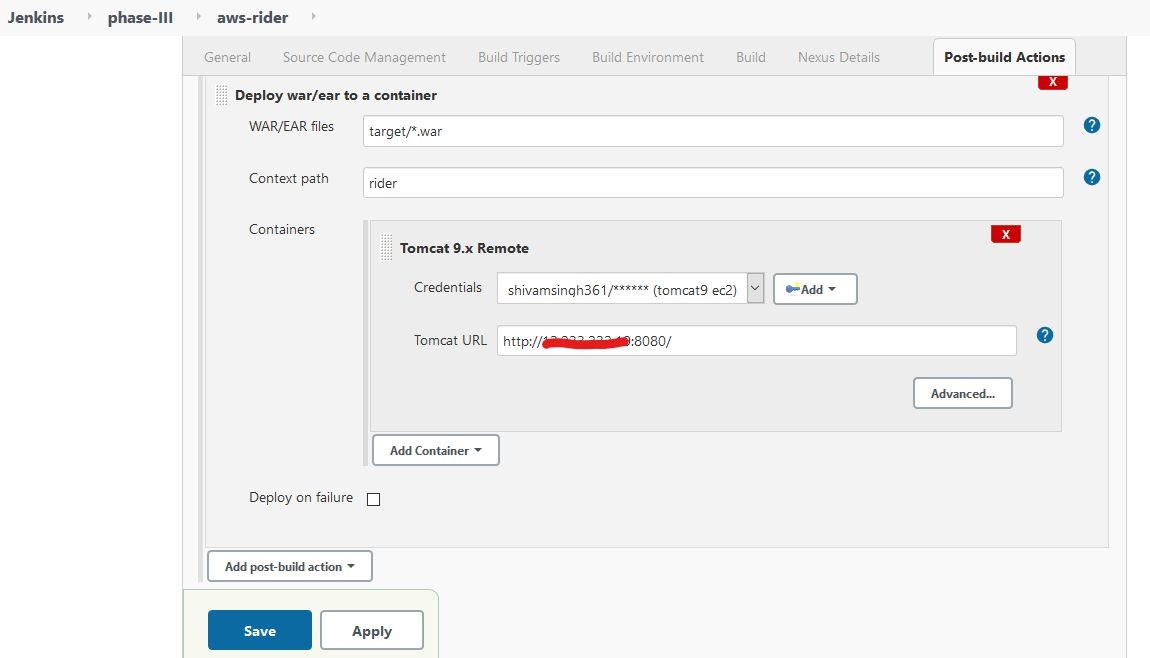
change context.xml file in webapp/manager/META-INF and

webapp/host-manager/META-INF both file like following

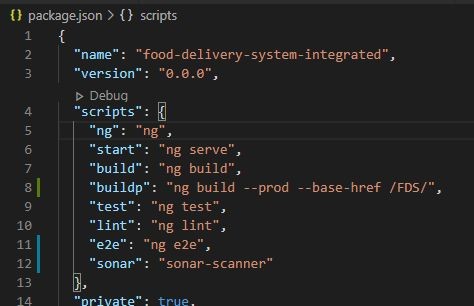
1. Comment the value tag in both context.xml files
2. 
3. 
4. 25.

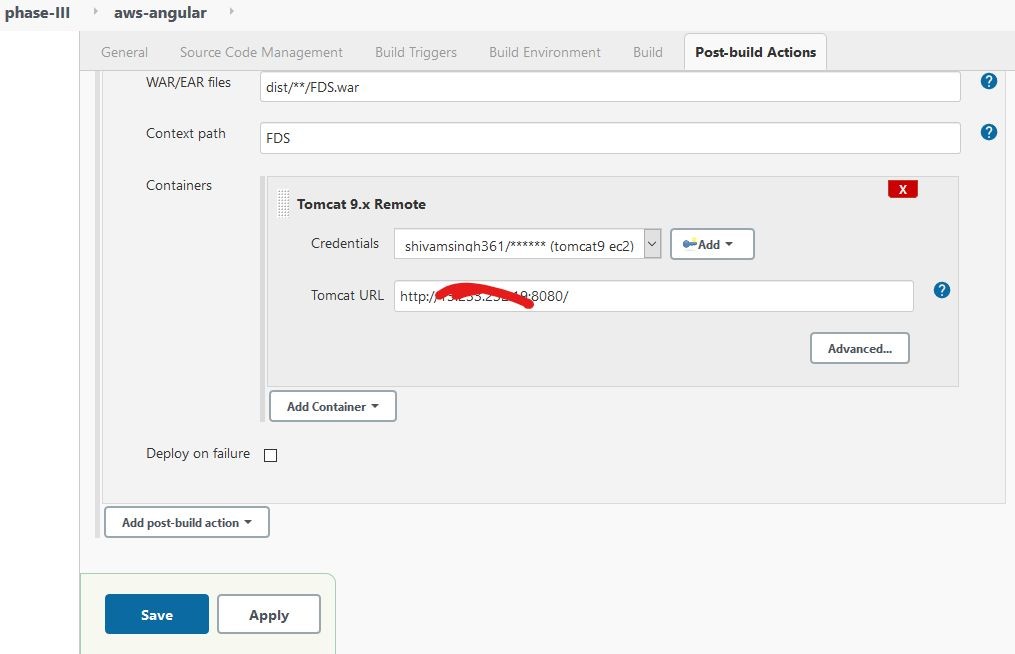
# Deploy spring boot / Angular to remote tomcat server:

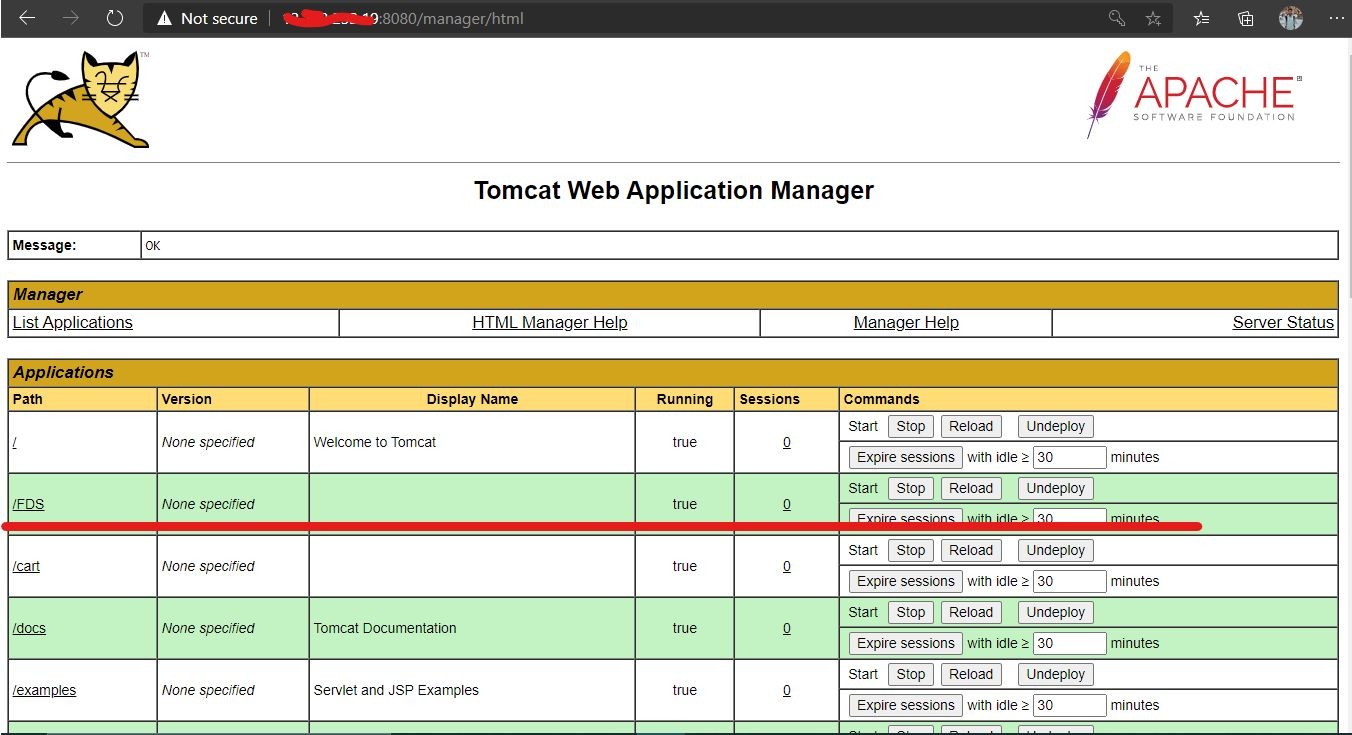
1. Open Jenkins
2. Create free style project
3. Choose a custom workspace and enter the Base Dir URL of the file system.
4. In post build Add credentials and

5. 

1. Save and build the war file of your spring boot can be deployed on the server.
2. Similarly in case of angular deploy the .war file similarly as above. Note context path should be the same as base href (here FDS). used in the build method.



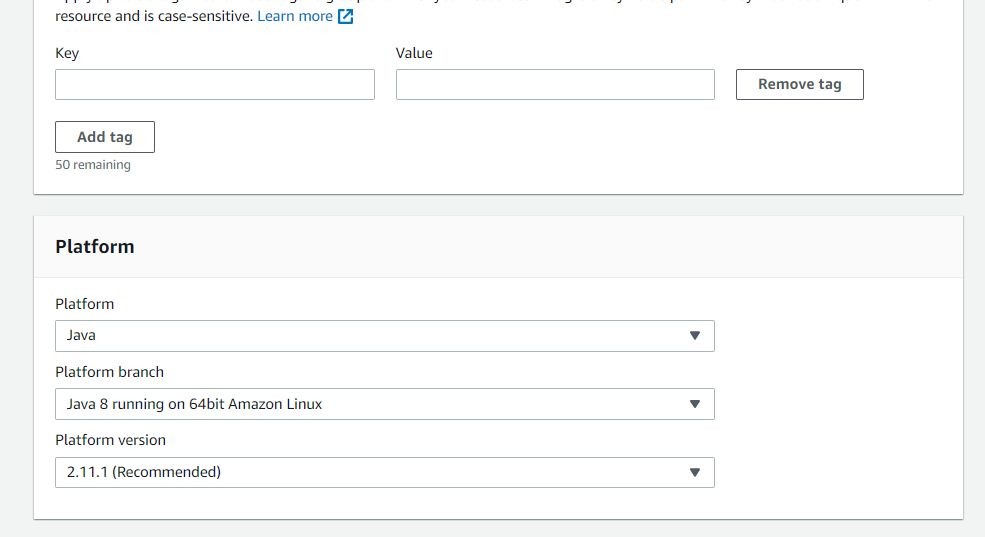
8. 

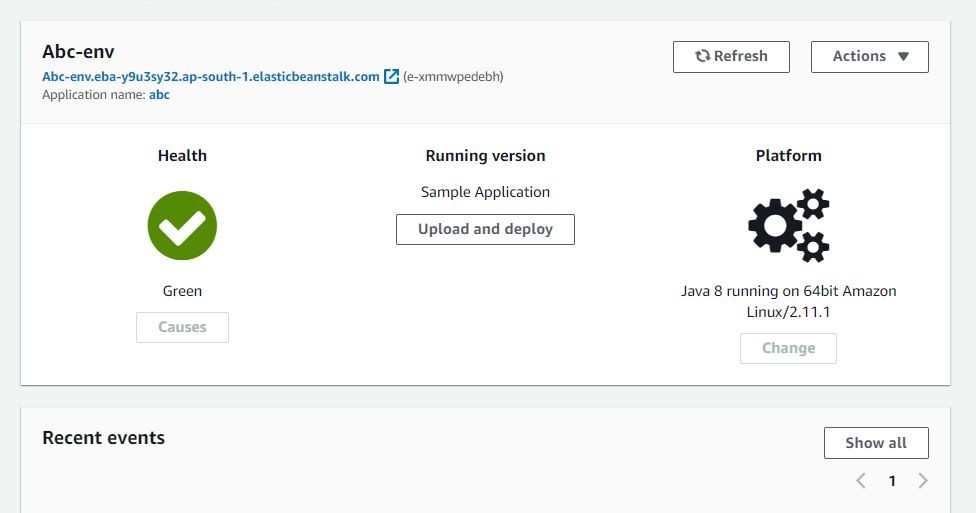
9.

10. Done.

# Deploying spring boot over Elastic Beanstalk (Extra):

1. Create Application -> write ap name
2. Choose java ->java8 running on Linux

3. 

1. Choose Sample application
2. Create application
3. You can click to check the sample app on the blue link given above.
4. Go to configuration -> databases->edit
5. Copy the RDS db endpoint URL, if not given add one and copy.
6. In the Springboot app paste the same in .properties file in place of localhost.
7. Also set server.port as 5000
8. Now On env Home click upload and deploy
9. Upload .jar file of updated spring boot app.
10. Once Health is Green. All done