**Fork the below projects.**

[https://github.com/kurukundaveera/parking\_frontend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Fparking_frontend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959136360&sdata=1slkUKyFJ5bvhKgyhAagjEKnzzyRlWi5W1ZKznwXAT0%3D&reserved=0)

[https://github.com/kurukundaveera/parking\_backend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Fparking_backend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959146351&sdata=PcuVEUcPS79xPw1d3NIKQ3Jxf9f37kpl5sqMdR0PN9s%3D&reserved=0)

[https://github.com/kurukundaveera/insurance\_frontend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Finsurance_frontend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959156346&sdata=YEJNOmdruhNf%2BpB3evqQdSOLSYngXrVc30XjOD88Jgg%3D&reserved=0)

[https://github.com/kurukundaveera/insurance\_backend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Finsurance_backend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959156346&sdata=AF5WDhNTqMUDrgENi5v6LM1ui46te%2BMc84OJ4kG6ELI%3D&reserved=0)

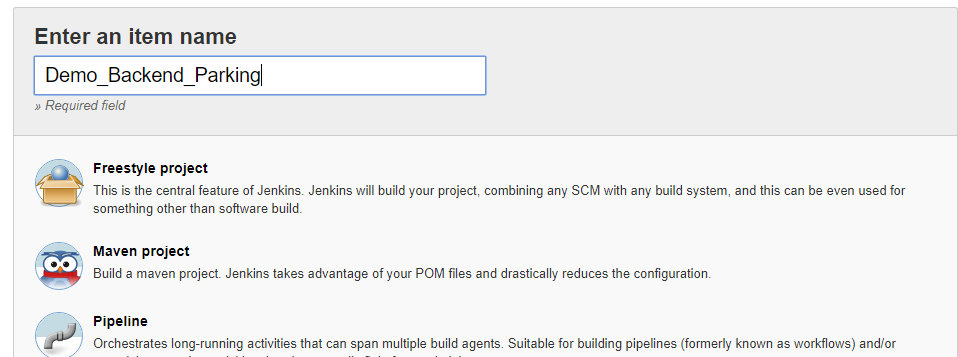
**UC5-PART 1: Parking page UI(REACT) & backend (JAVA)**

1. **Implement using a Freestyle**

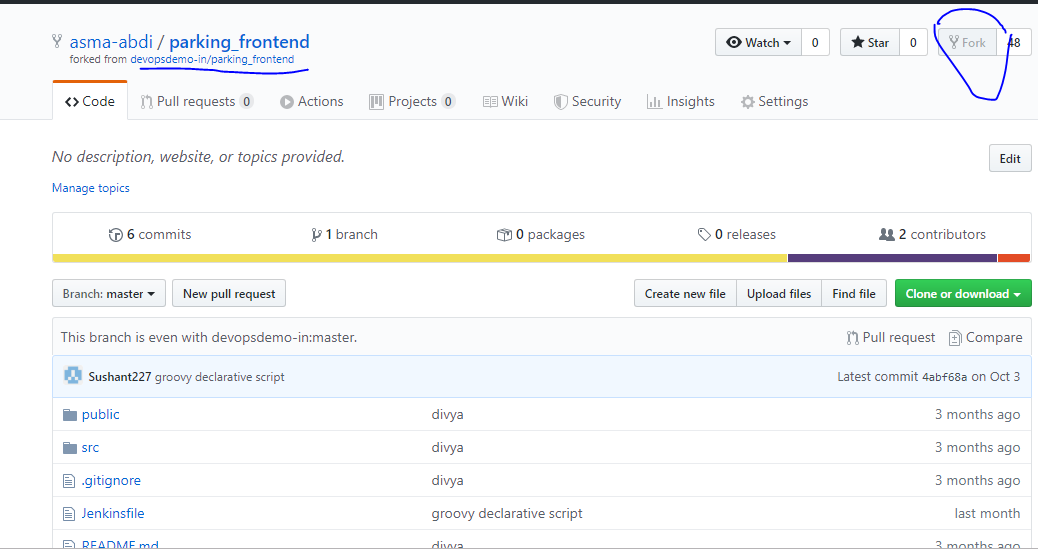
Frontend: [https://github.com/kurukundaveera/parking\_frontend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Fparking_frontend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959136360&sdata=1slkUKyFJ5bvhKgyhAagjEKnzzyRlWi5W1ZKznwXAT0%3D&reserved=0)

Backend: [https://github.com/kurukundaveera/parking\_backend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Fparking_backend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959146351&sdata=PcuVEUcPS79xPw1d3NIKQ3Jxf9f37kpl5sqMdR0PN9s%3D&reserved=0)

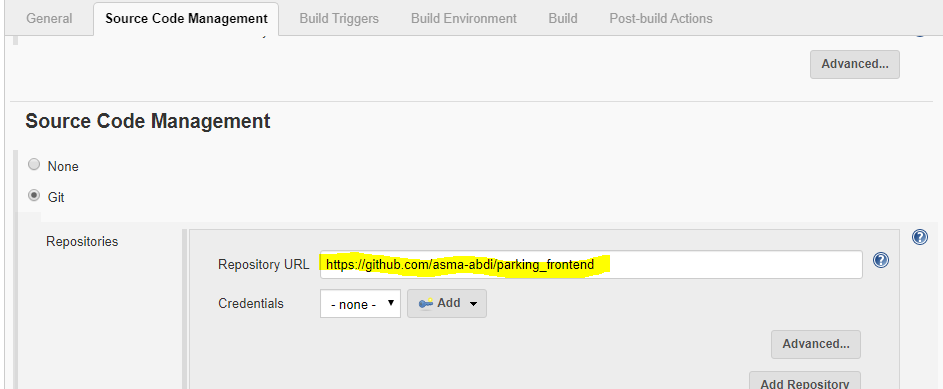
* Go to Jenkins: create a new job—As as Freestyle Project



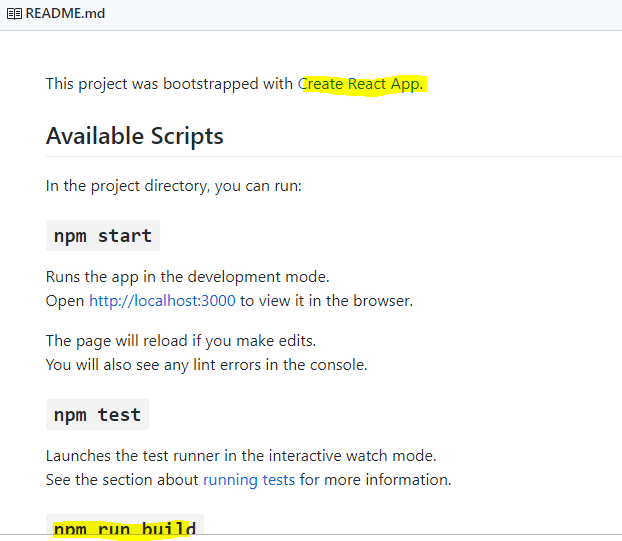
* Now will fork the repository for frontend and will copy the URL. <https://github.com/asma-abdi/parking_frontend>



* In SCM will give the copied frontend repo URL



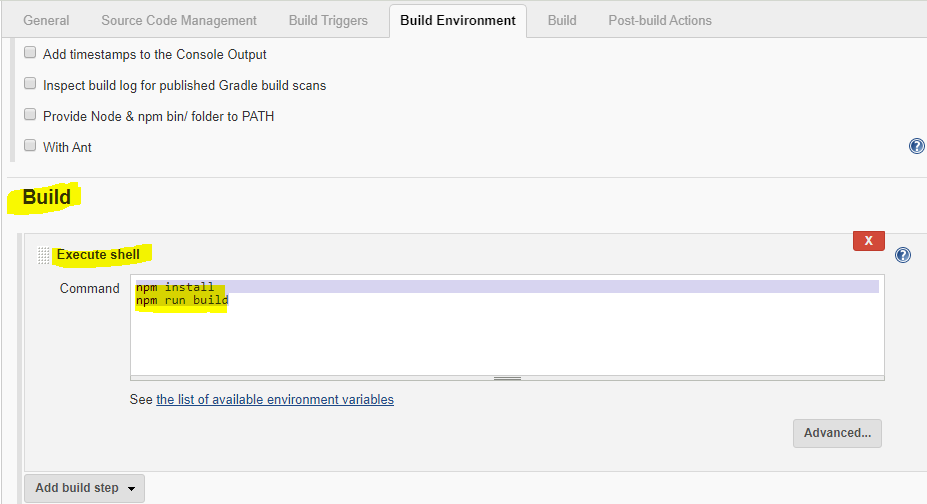
* Now will check whether is Angular/react, for that we can open frontend githubURL can scrolldown will be mentioned that either angular/React



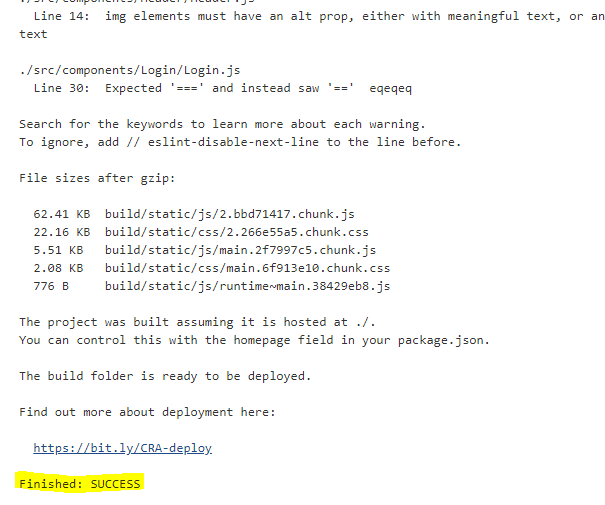
* Now Provide shell command in Build for execute shell:

npm install

npm run build

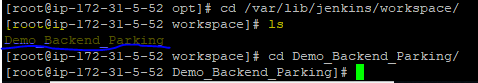


* Now can save it and will build the Run and check the output Console.

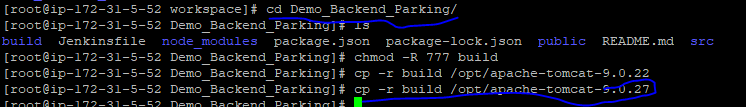


* Once build is successful will deploy the build on Apache Tomcat server:, for that will use below steps:

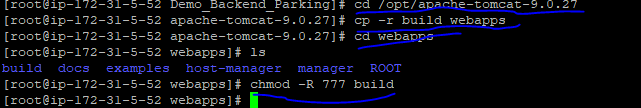
cd /var/lib/jenkins/workspace/-- go to this folder and give the ls command check the UI file:



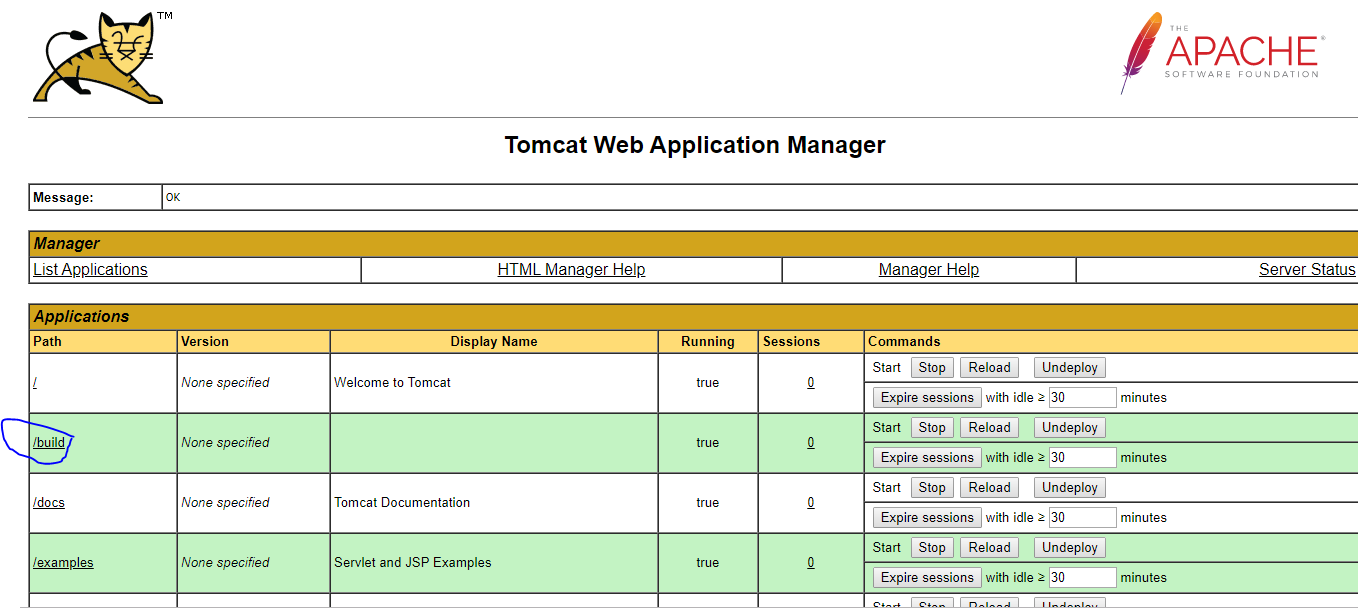
Now will go inside the UI folder and will copy all content to apache server latest version



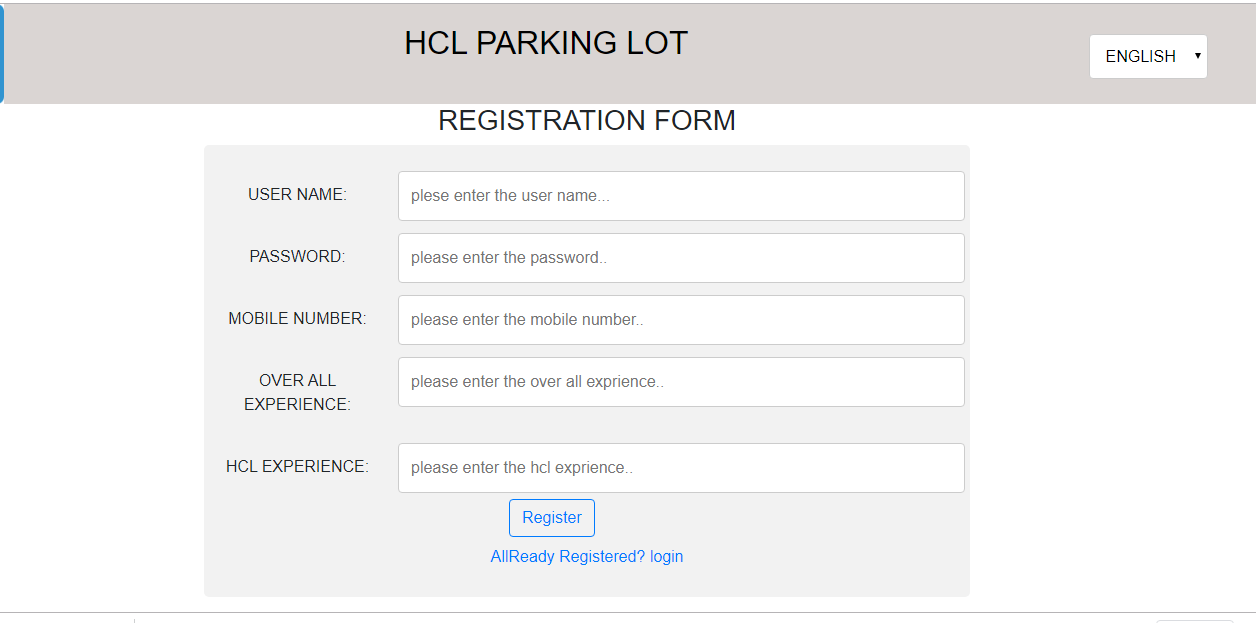
* Now will go inside apache folder and copy everything from build to apache and will give all permission



* Now we will check the UI on tomcat :

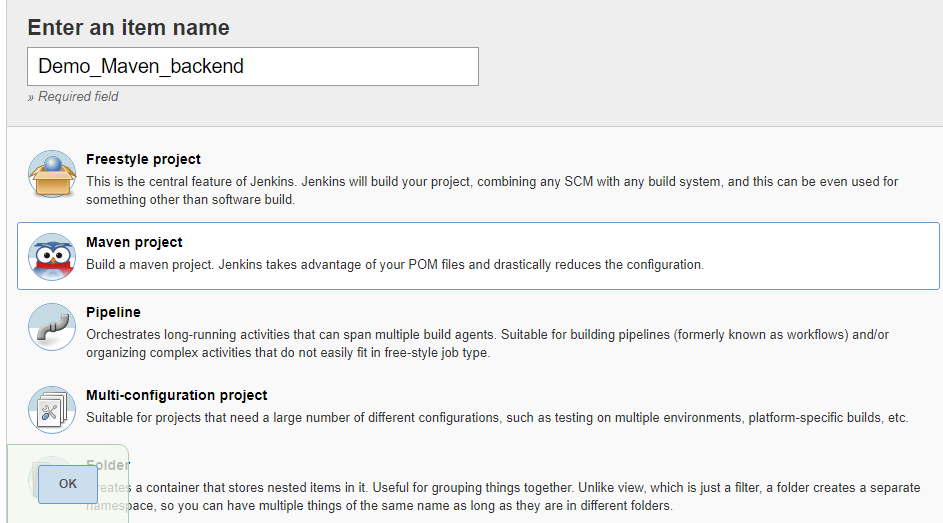


Once will click on build will able to see the frontend page of “Parking”.

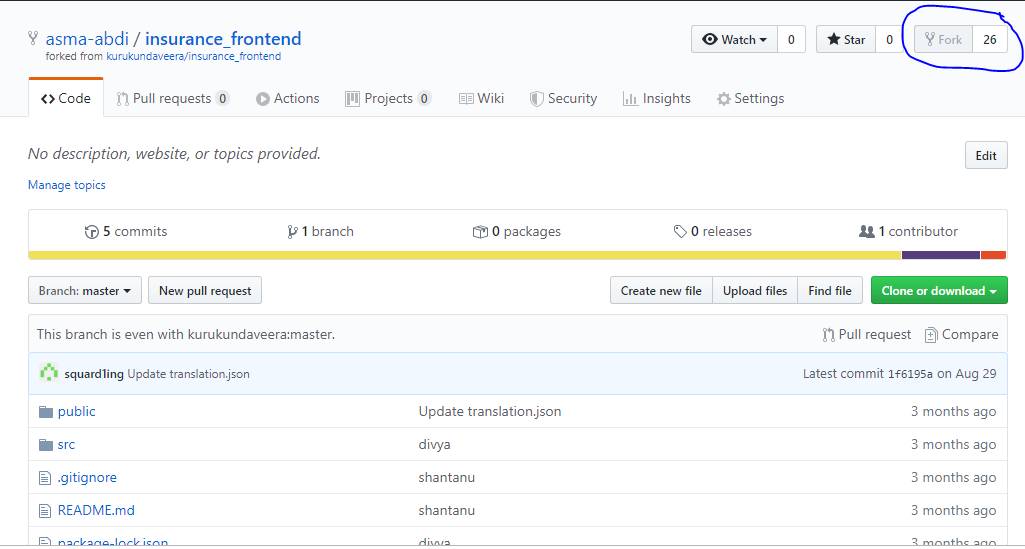


1. **Implement using a Maven Project**

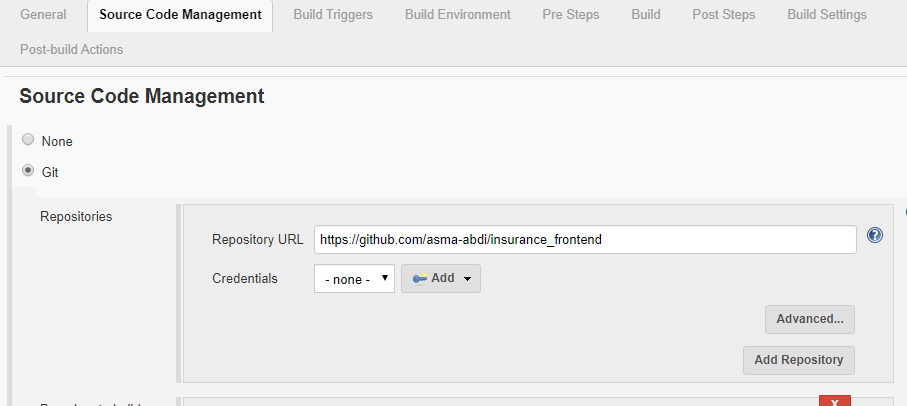
* Now will create one more job for backend. As – Demo\_Maven\_Backend as “maven Project”



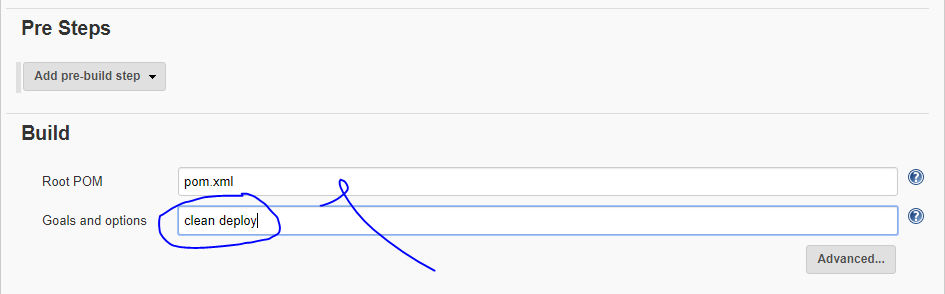
* **Now fork the gituphub repo for backend-** <https://github.com/kurukundaveera/insurance_frontend>. Now “Fork” it and take the URL



* Copy the URL in SCM-



* Pre Steps:

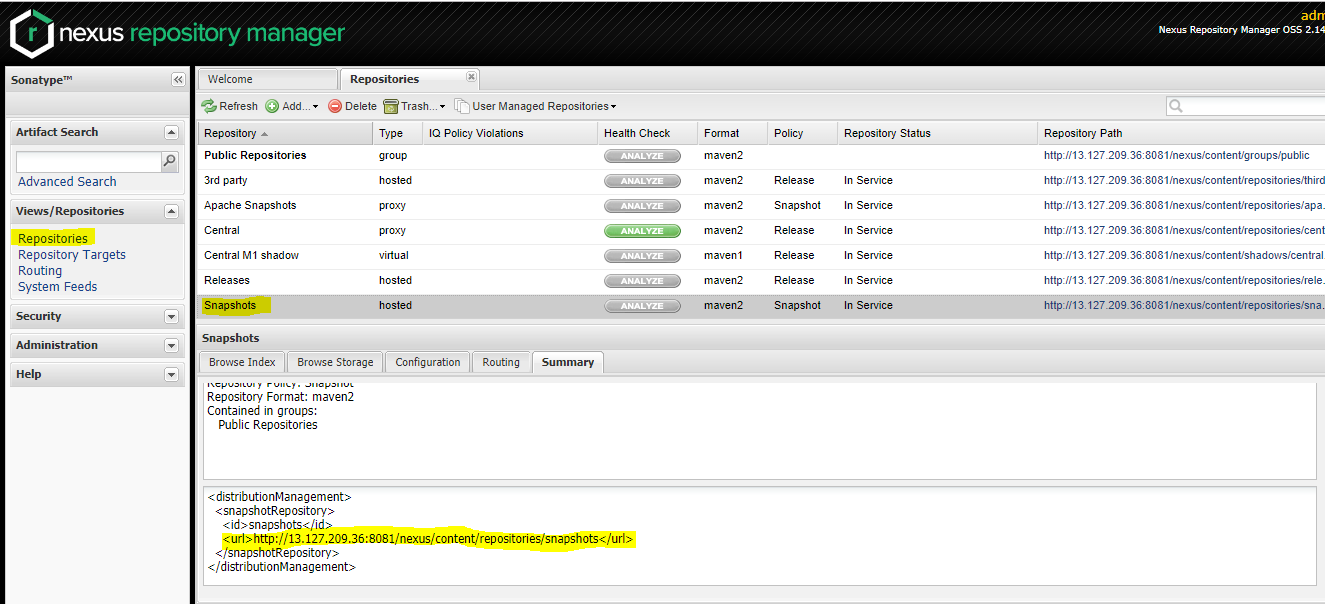


* Post Steps:

**Execute shell🡪mvn sonar:sonar**



* Now will change the “pom.xml”—copy the URL from nexux.



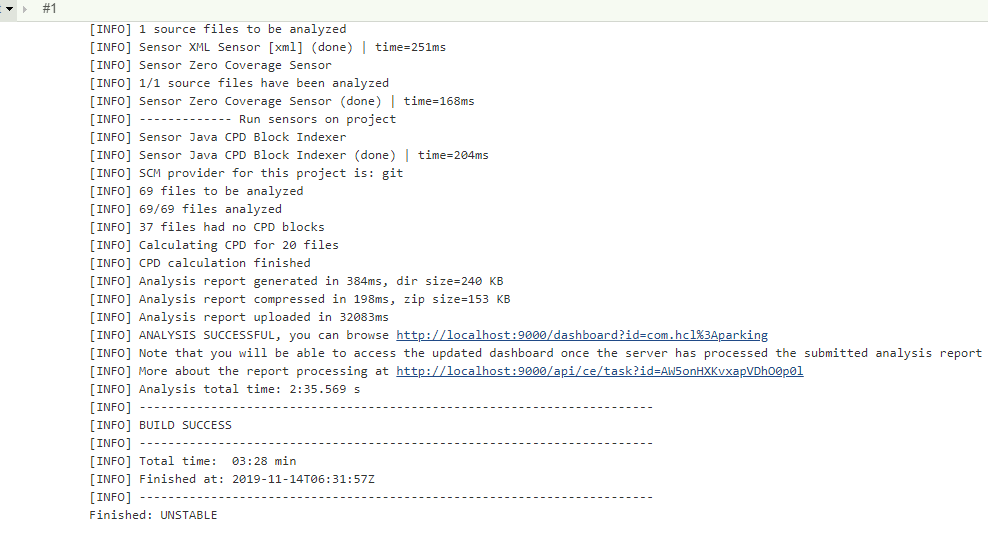
<url>[http://13.127.209.36:8081/nexus/content/repositories/snapshots</url](http://13.127.209.36:8081/nexus/content/repositories/snapshots%3c/url)>

* Go to git hub URL backend and edit the pom.xml as per below-

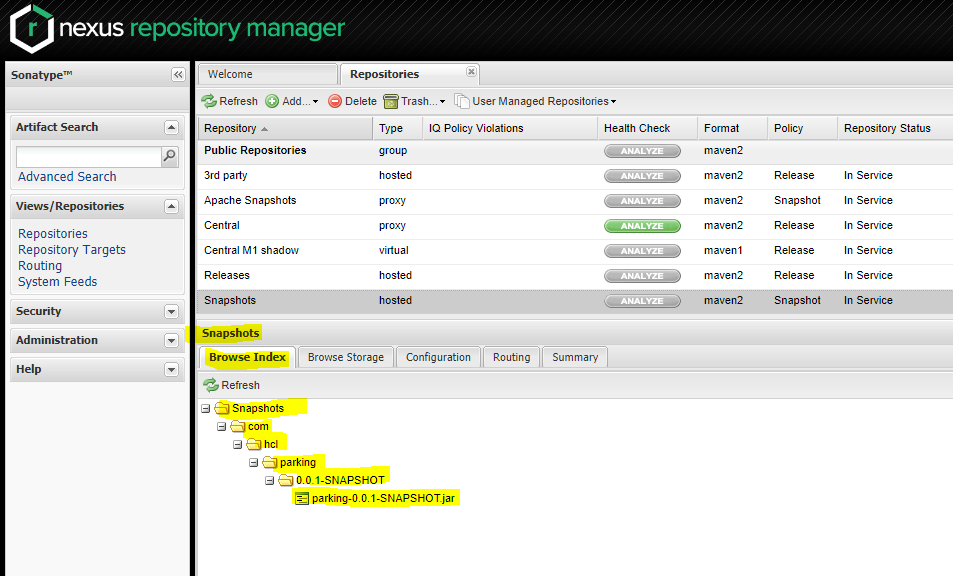
In place of <id> ‘snapshots’ give ‘nexus’ that should same as what we have given in ***settings.xml file***.



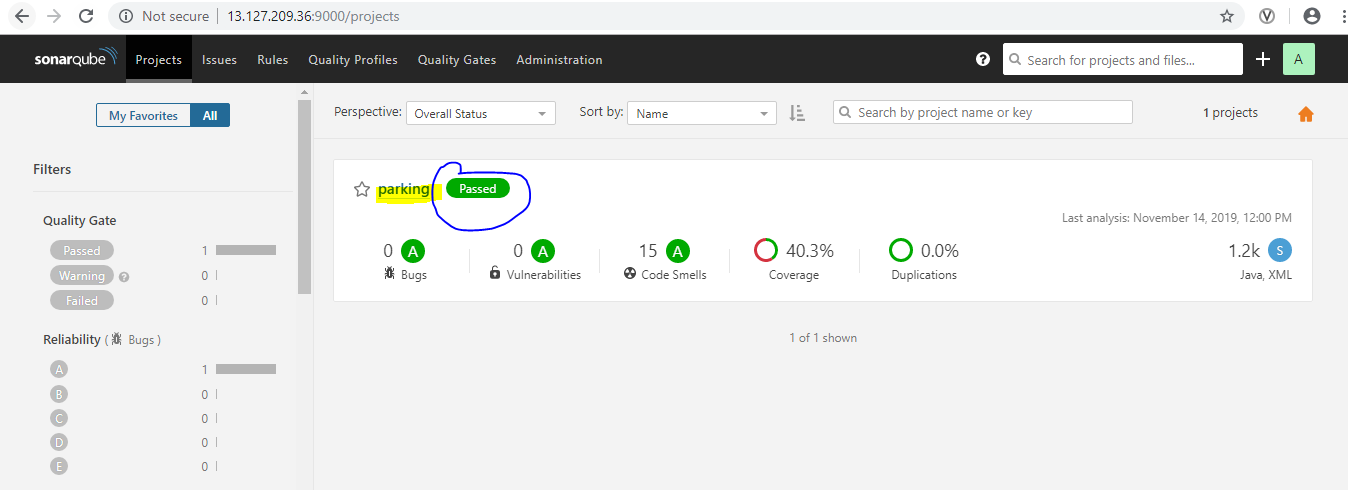
* Now run the build – check for console-



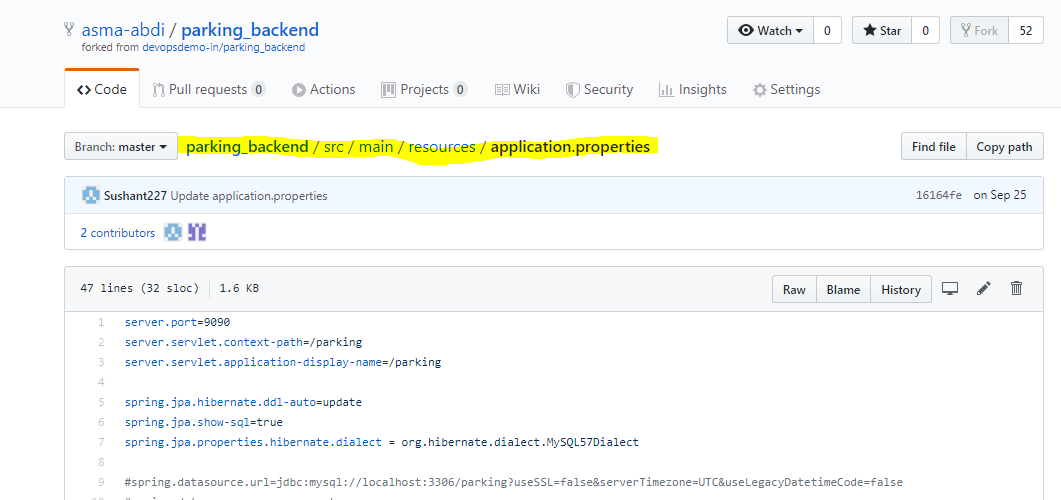
* Now will check the snapshots in nexus repo for the jar files- sanapshots:



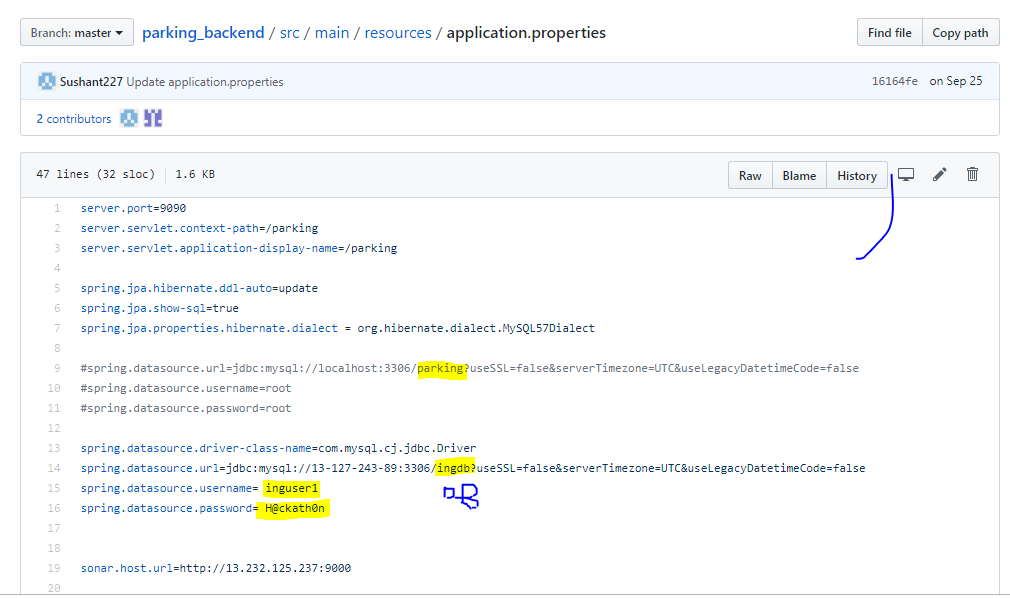
* Can check the buils stability in SONAR:



* For Swagger – first needs to go to backend URL repo- [https://github.com/asma-abdi/parking\_backend-->sc-->main-->resource-->application.properties](https://github.com/asma-abdi/parking_backend--%3esc--%3emain--%3eresource--%3eapplication.properties)



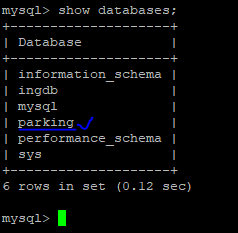
* Now check what is our database(ingdb) name in application file, check for username and password we have created for our db.



* Now we need to create a db with ‘parking’. Even I have created as ingdb.



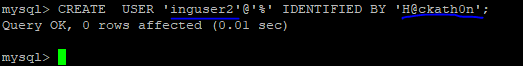


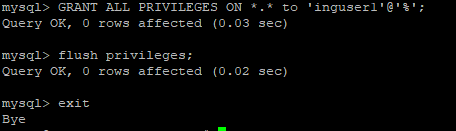


* **Now create the user:**

Inguser2 –it can be any name that we create

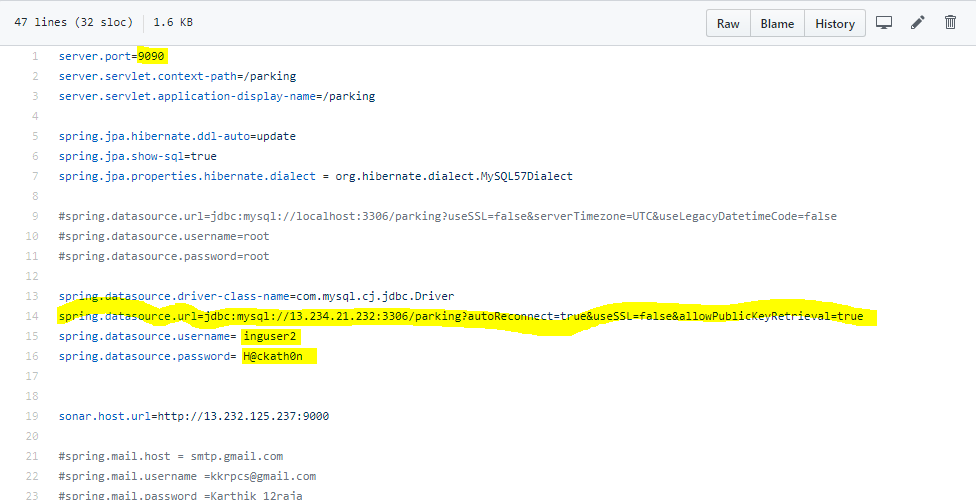
H@ckath0n--> its recommend to give this password, strong.





* Now will change the public ip , username and password,db name

spring.datasource.url=jdbc:mysql://13.127.209.36:3306/parking?autoReconnect=true&useSSL=false&allowPublicKeyRetrieval=true



* Now will run the build again and if it is unstable(due to some test cases fails, etc), can mark the build as success.

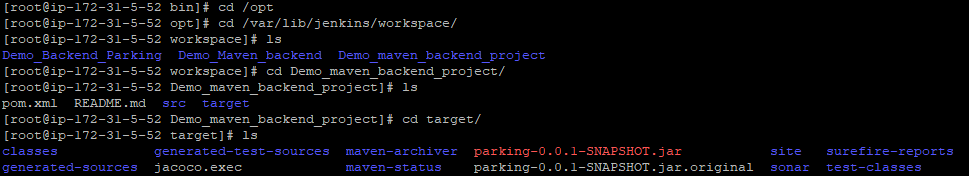


**spring application procedure(To make backend application UP)**

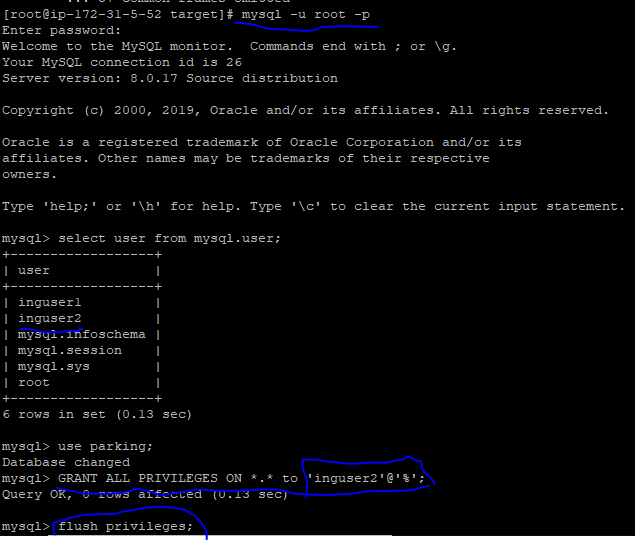
1. First go the path- cd /var/lib/jenkins/workspace/

2. list the dir – ls

3. cd target/

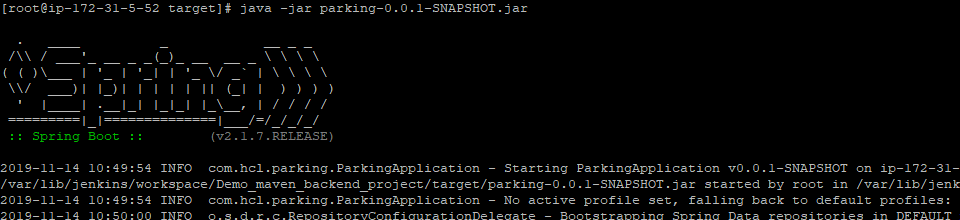


4. check for db and user, id its there will provide the previliges



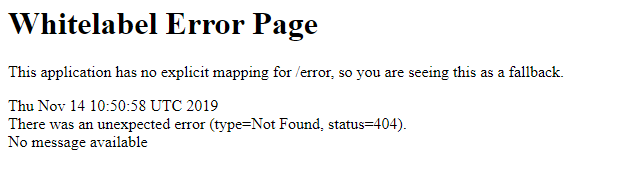
NOTE: its very important to check the db and user while giving privileges.

5. now will run the jar file-- [root@ip-172-31-5-52 target]# java -jar parking-0.0.1-SNAPSHOT.jar

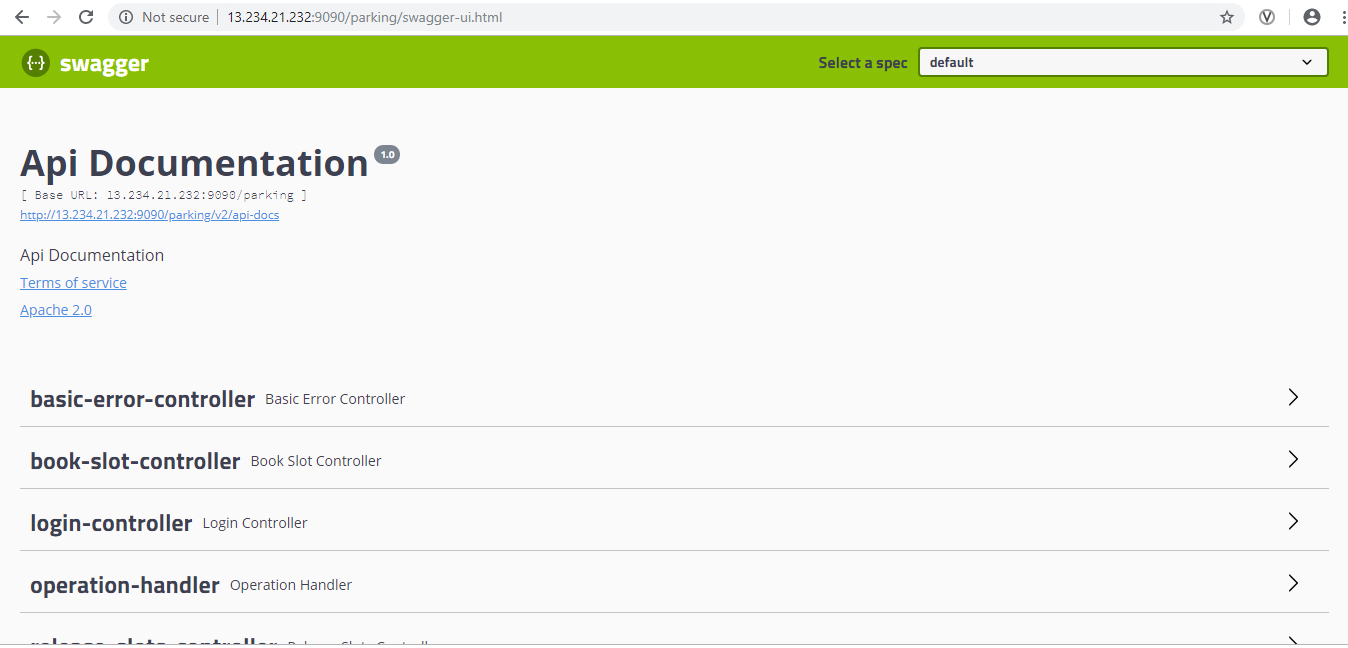


6. Now will check the backend on Swagger: publicip:defaultport/db/swagger-ui.html🡪<http://13.234.21.232:9090/parking/swagger-ui.html>

NOTE: swagger-ui if we try to give then below error will come up-



Swagger page will look like –



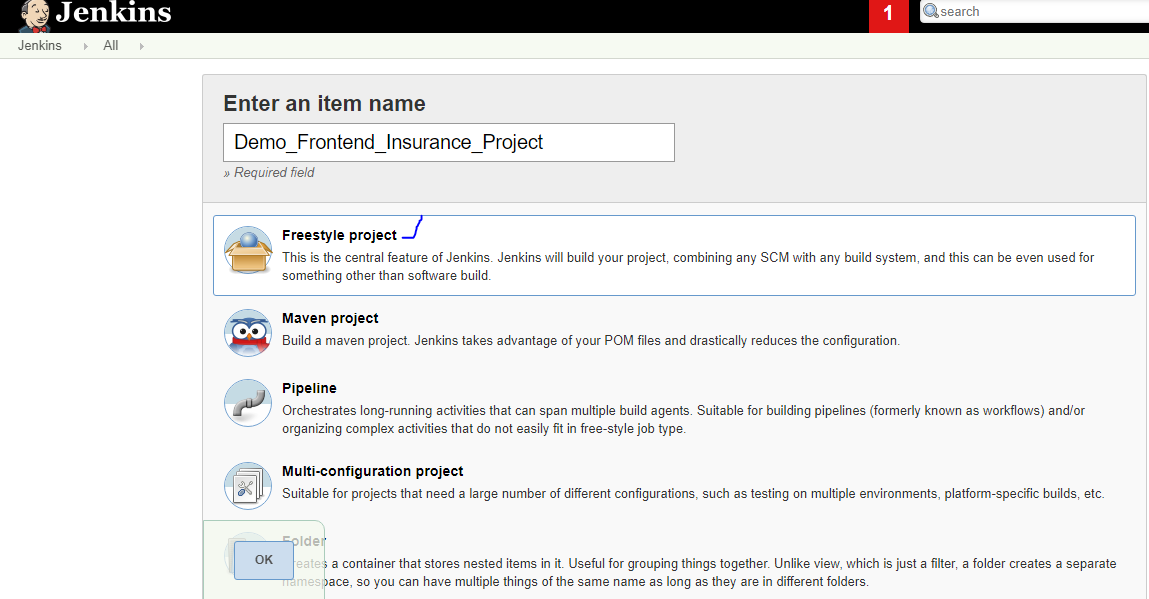
UC5-PART-2: Insurance Frountend(REACT) and Backend(Java)

**Frontend:** [https://github.com/kurukundaveera/insurance\_frontend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Finsurance_frontend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959156346&sdata=YEJNOmdruhNf%2BpB3evqQdSOLSYngXrVc30XjOD88Jgg%3D&reserved=0)

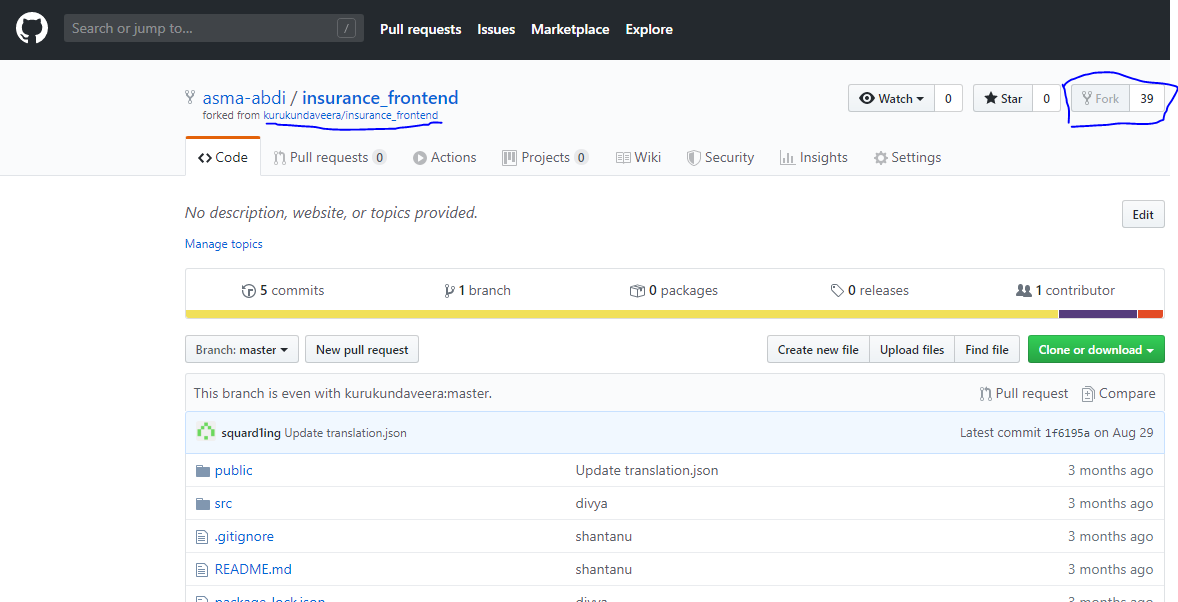
**Backend:** [https://github.com/kurukundaveera/insurance\_backend.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2Finsurance_backend.git&data=02%7C01%7Casma.ab%40hcl.com%7C925574ef82614f92d4ec08d768a43f38%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637092926959156346&sdata=AF5WDhNTqMUDrgENi5v6LM1ui46te%2BMc84OJ4kG6ELI%3D&reserved=0)

1. **Implement using a Freestyle**

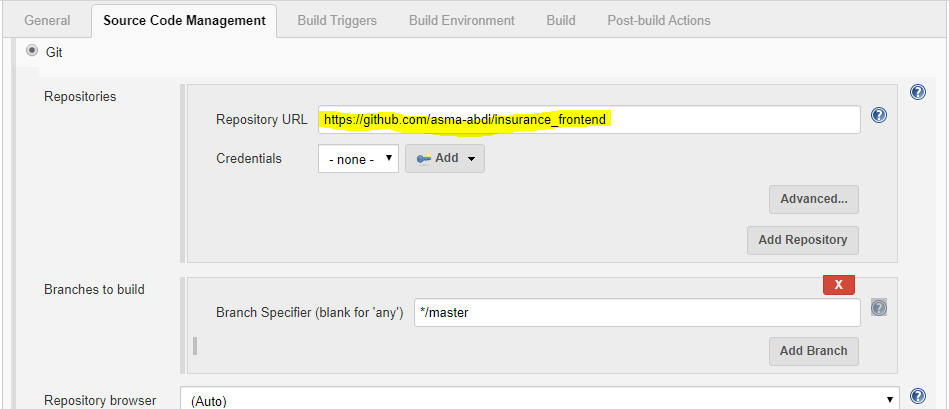
* Create a demo project as a freestyle project🡪



* Now need to fork the backend repository in github <https://github.com/kurukundaveera/insurance_frontend>, so that we can use it in our project-



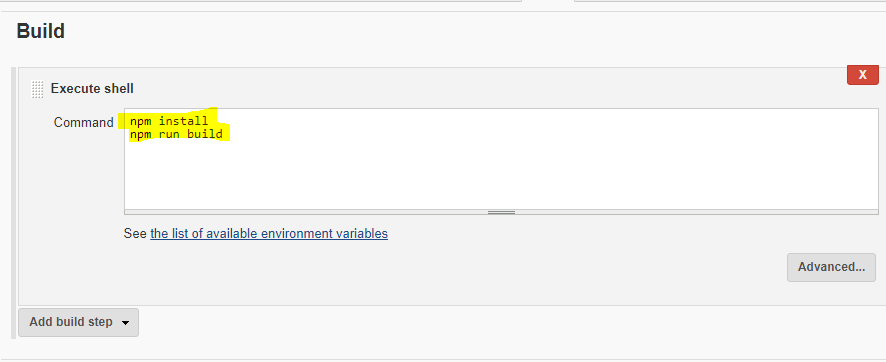
* Now we can copy the git URL and will use as source for the project:
* SCM



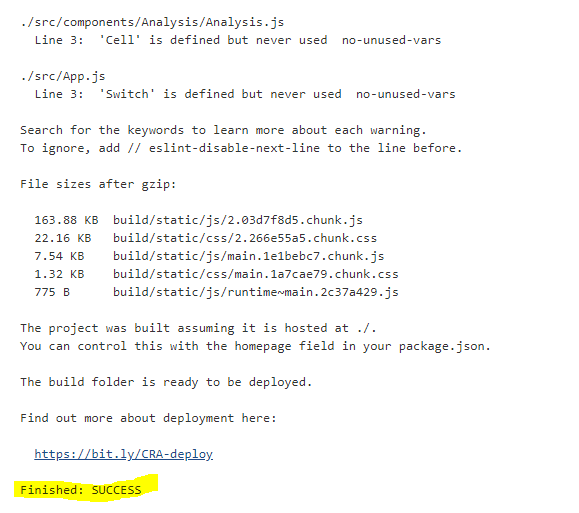
* At Build step 🡪execute shell-

Npm install

Npm run build



* Now can apply🡪save, will run the build for success and can also check on output console.



* **To deploy the war to Tomacat Webapps**

[root@ip-172-31-35-155 opt]# cd /var/lib/jenkins/workspace/

[root@ip-172-31-35-155 workspace]# ls

[root@ip-172-31-35-155 workspace]# cd SAMPLE\_UI/

[root@ip-172-31-35-155 SAMPLE\_UI]# ls

[root@ip-172-31-35-155 SAMPLE\_UI]# cp -r build /opt/apache-tomcat-9.0.27

[root@ip-172-31-35-155 SAMPLE\_UI]# cd /opt/apache-tomcat-9.0.27

[root@ip-172-31-35-155 apache-tomcat-9.0.22]# ls

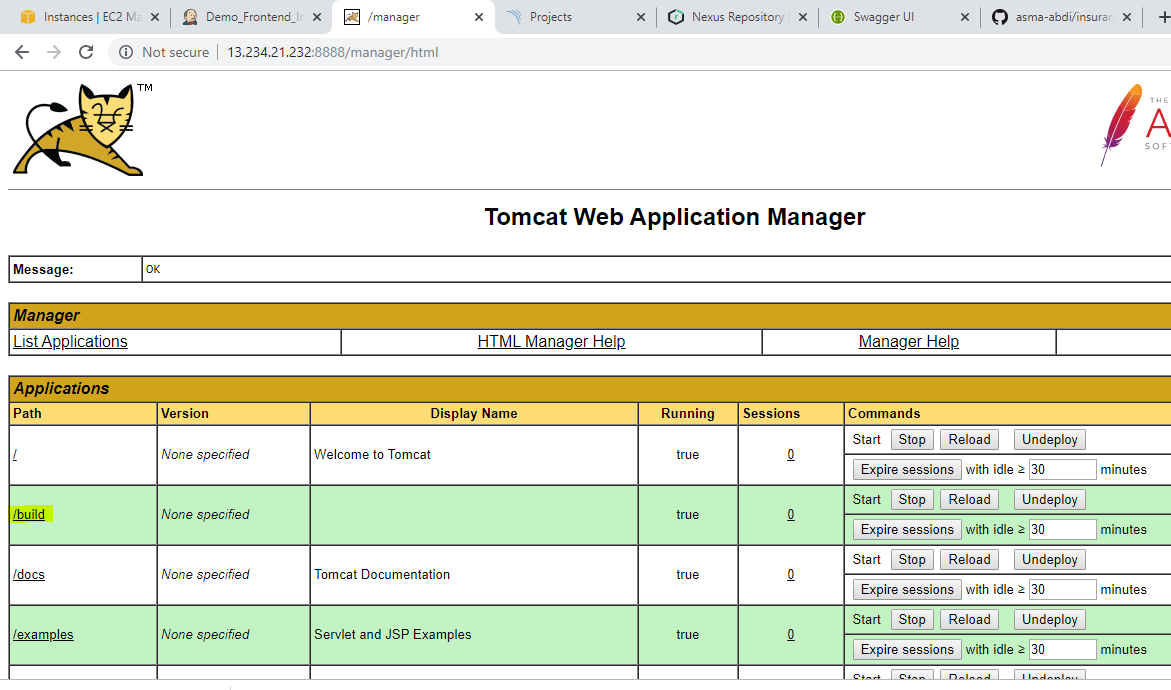
[root@ip-172-31-35-155 apache-tomcat-9.0.22]# cp -r build webapps

[root@ip-172-31-35-155 apache-tomcat-9.0.22]# cd webapps

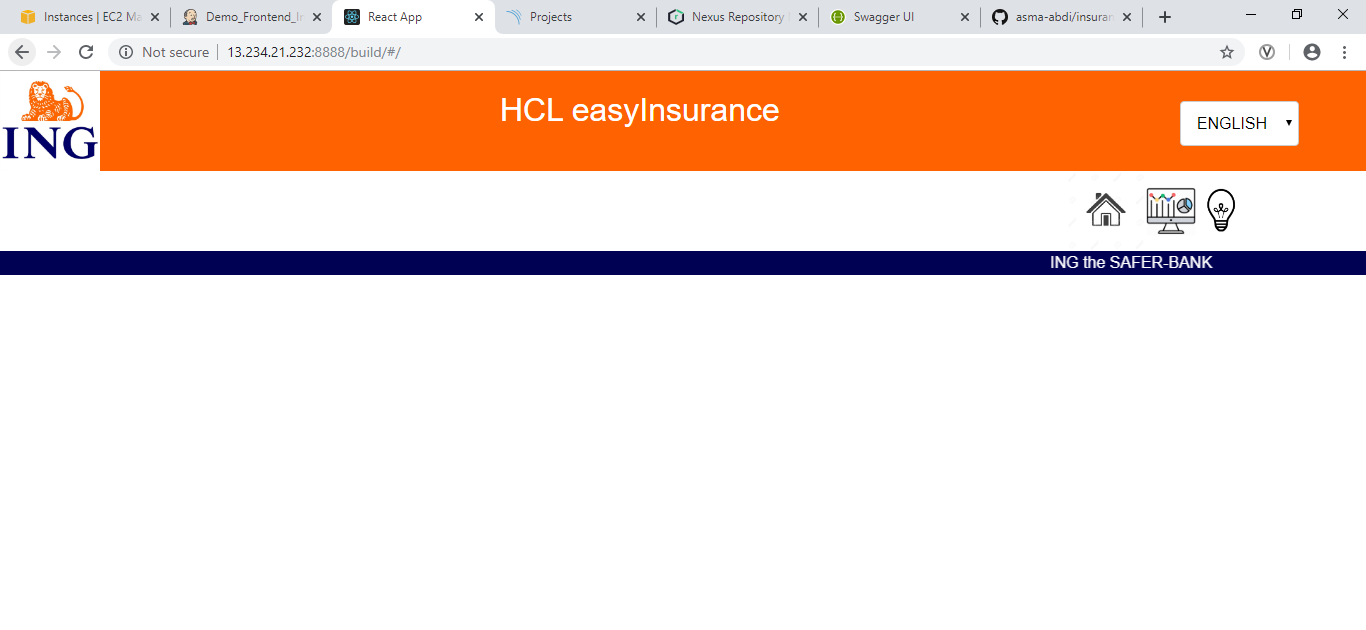
[root@ip-172-31-35-155 webapps]# ls

[root@ip-172-31-35-155 webapps]# chmod -R 777 build

* After successfully deploy of war on Apache Server, will login in on to Apache-Tomcat server🡪manager App🡪cleck on ‘Build’- now we will able to use the frontend page of HCL insurance.

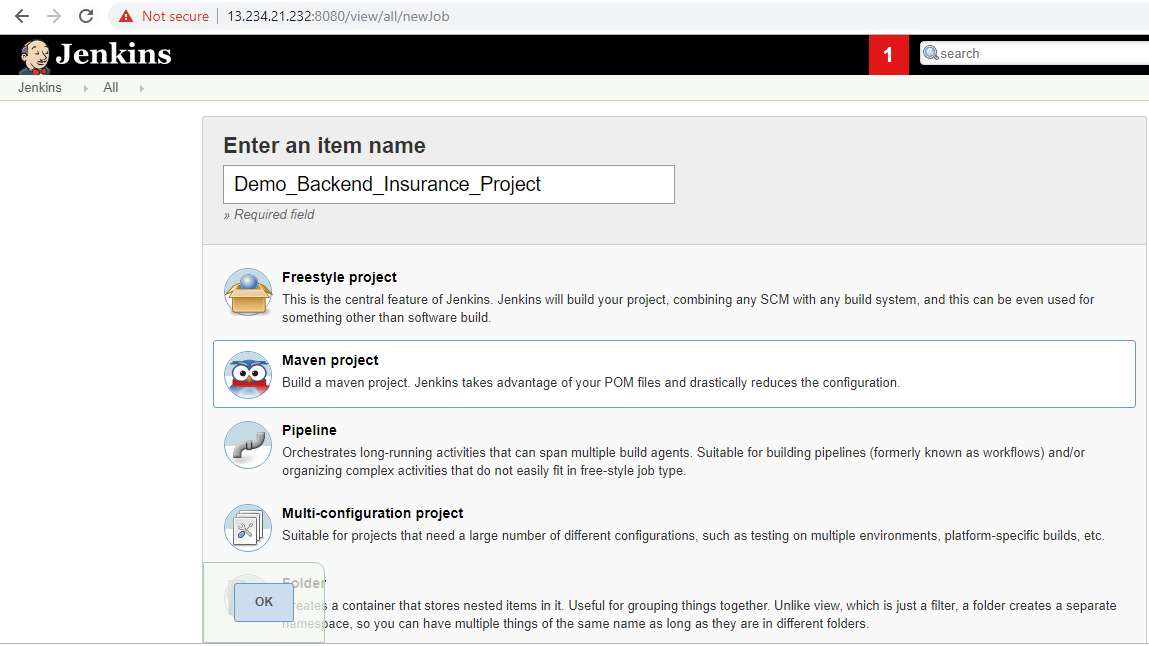


* Click on /build- page will open-

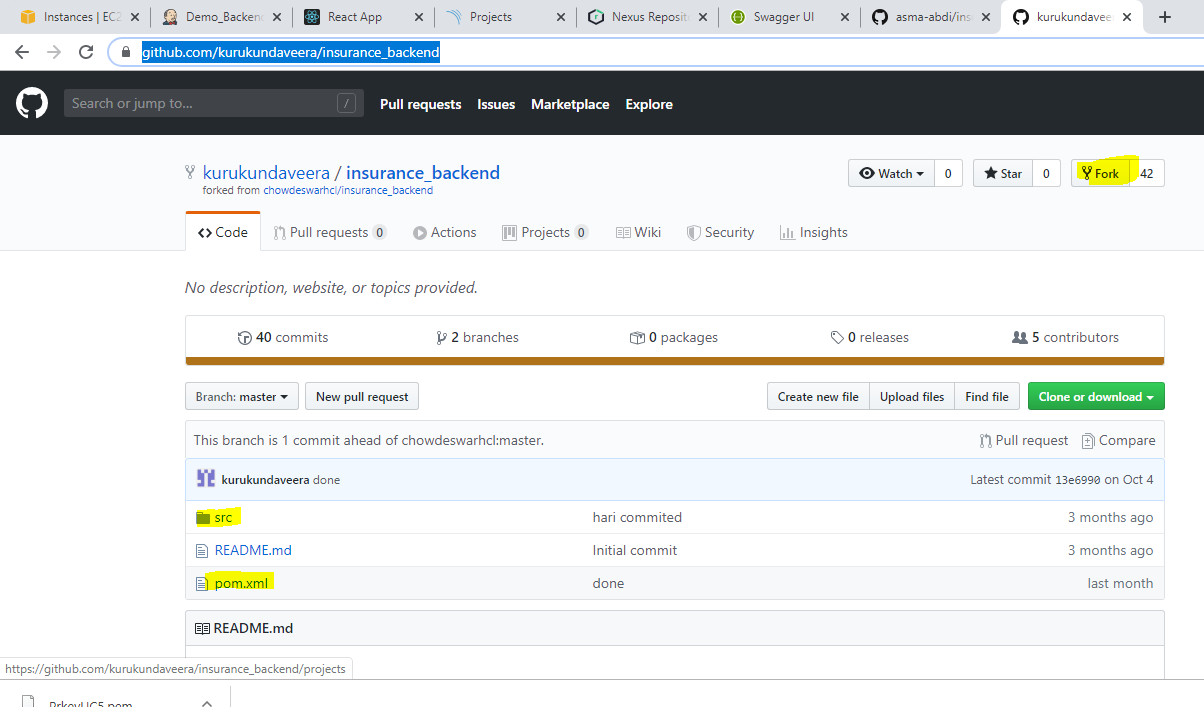


1. **Implement using maven Backend Project-**

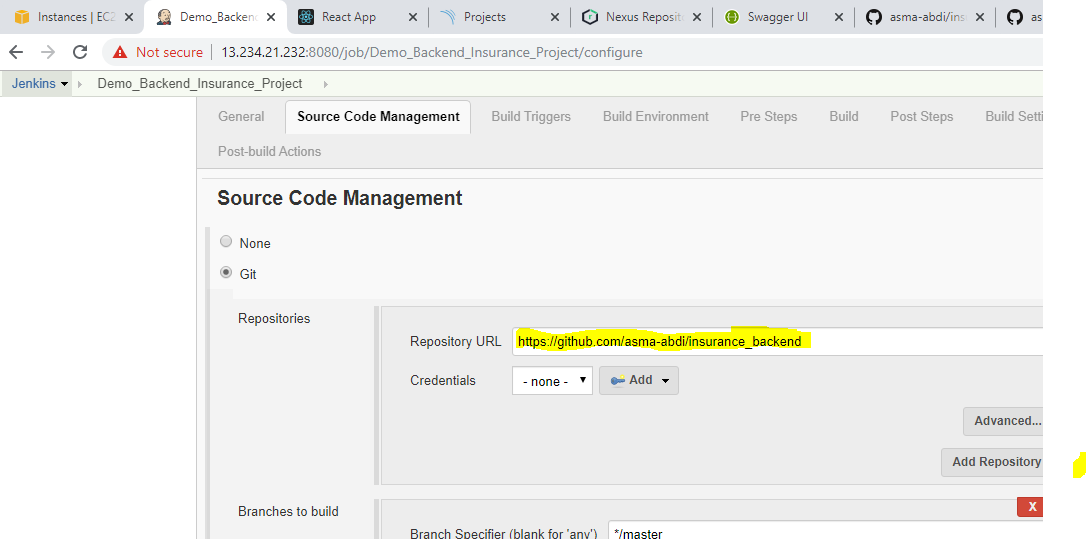
* Create a new maven project in Jenkins , New item-



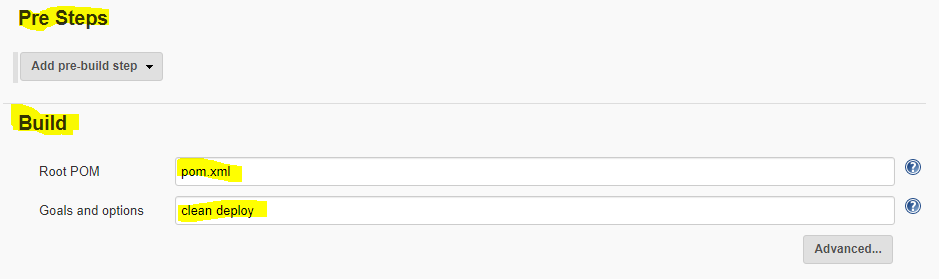
* Now will fork the backend repo- <https://github.com/kurukundaveera/insurance_backend>, that be having maven pom.xml file and src for the backend-



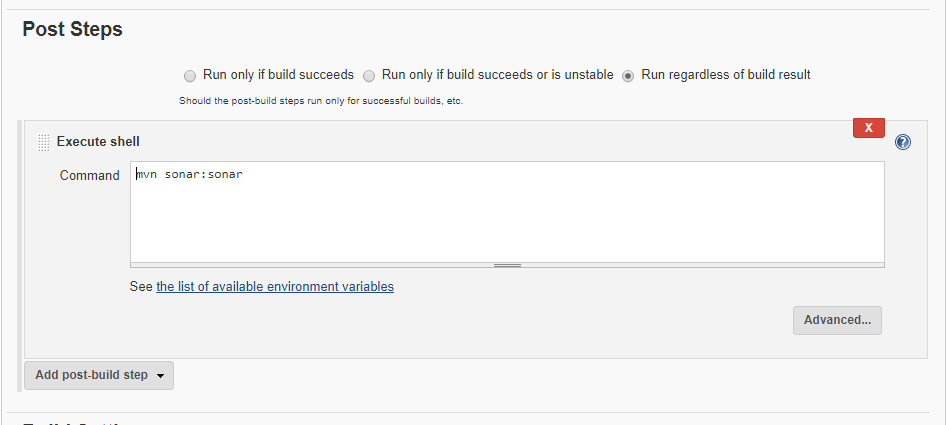
* Now we take the URL of forked repo and use as source for our backend Jenkins job.
* Jenkins🡪SCM- <https://github.com/asma-abdi/insurance_backend>



* Now will give ‘Build’ details🡪clean deploy the pom.xml

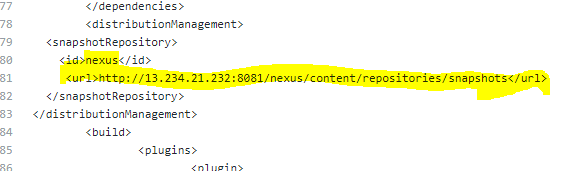


* Post Step: mvn sonar:sonar



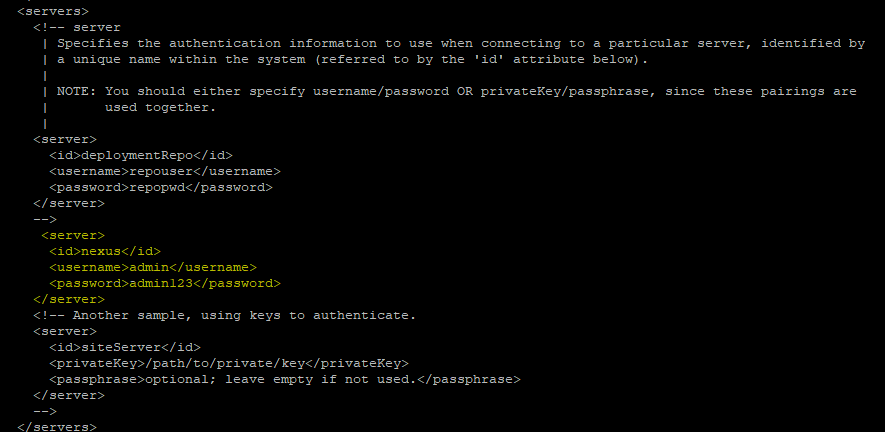
* Now, will edit the pom.xml—copy the URL from nexus and edit in gitrepo.

<url>[http://13.234.21.232:8081/nexus/content/repositories/snapshots</url](http://13.234.21.232:8081/nexus/content/repositories/snapshots%3c/url)>

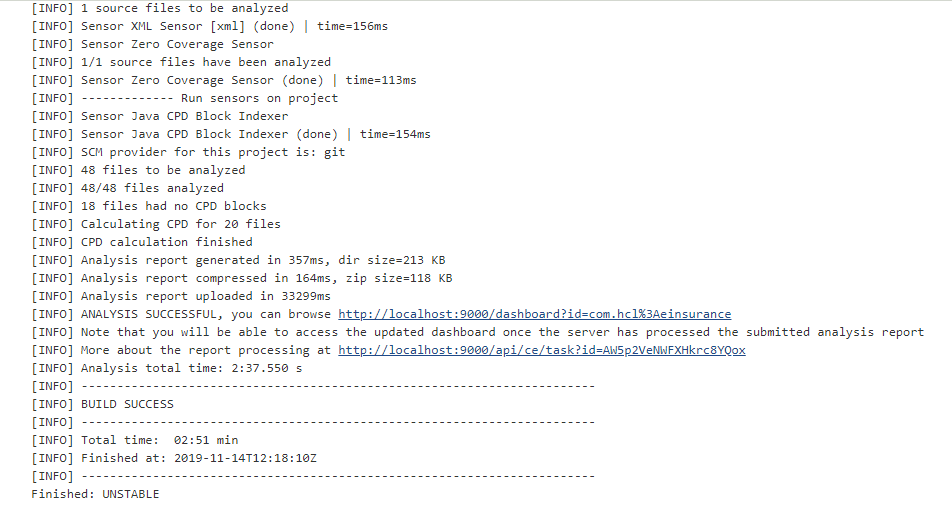


* Now commit the changes

NOTE: Settings.xml also should be corrected like that, i.e. very essential to be in place needs to be taken care,.

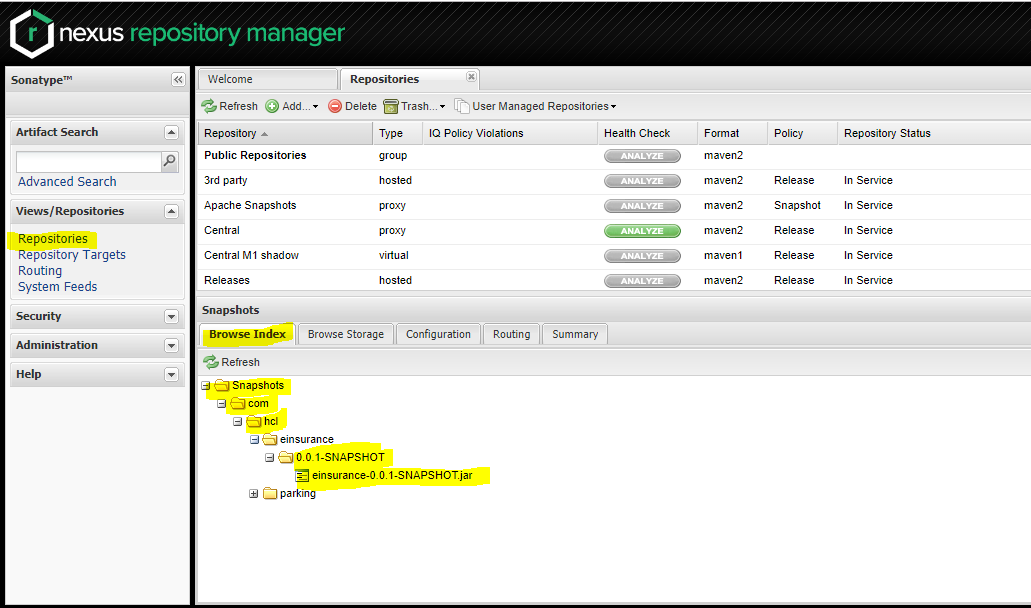


* After changes done in pom.xml, has to execute the build until it gets successful(Unstable also fine)

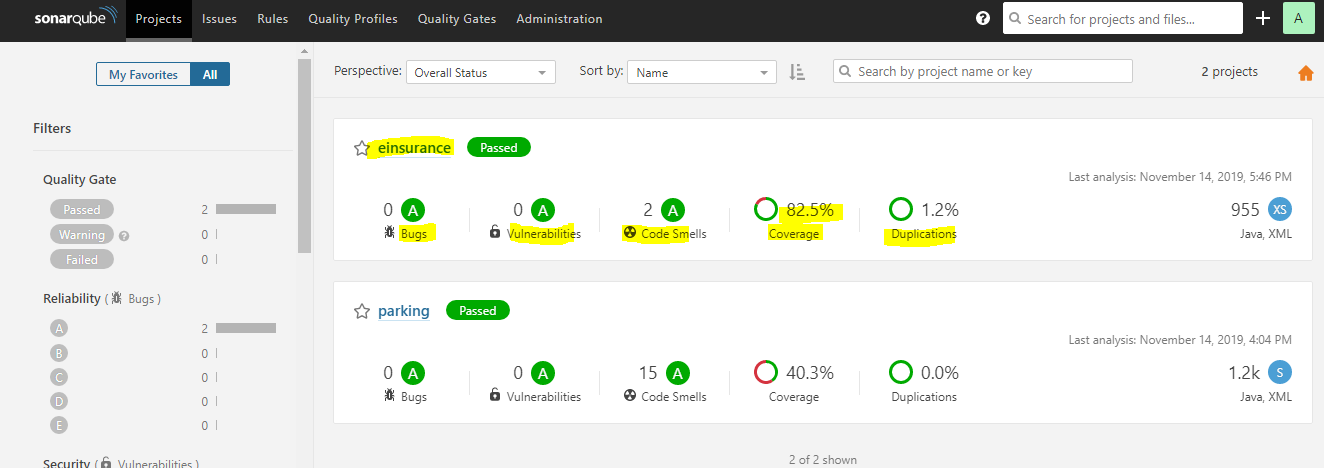


* Now will check nexus for successful transfer for artifacts-

NOTE: sometimes needs to ‘refresh’ if changes are not showing.

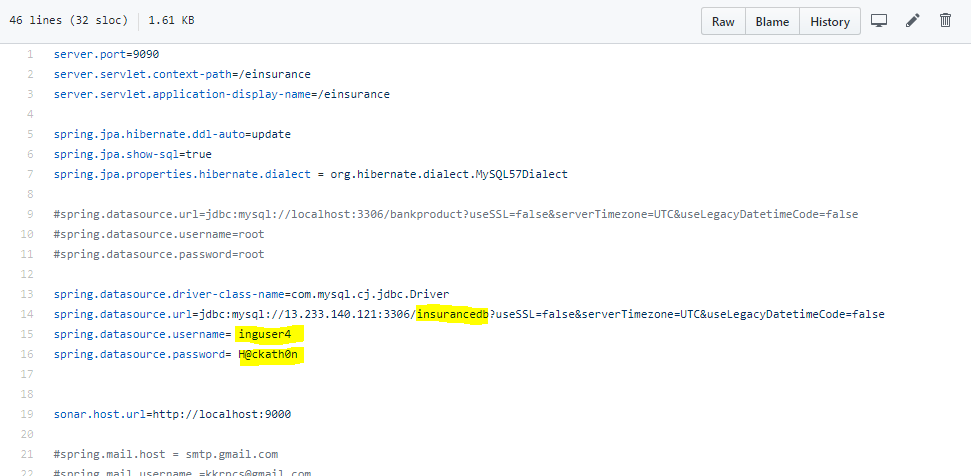


* Also will verify the code should passed in SONARQUBE, other matrices- bugs,code smells, coverage etc-

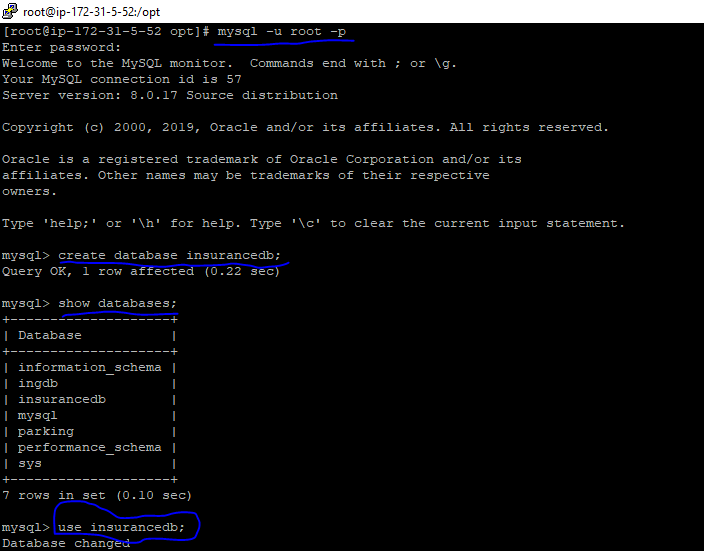


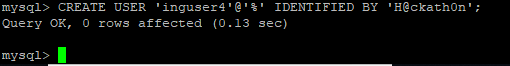
* Now will can refer Application.properties file-

Will create a db,user name(or can create other name as well)as name give in that file-

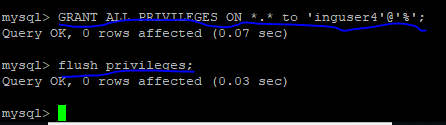


* Create db and user :





Grant the privileges and flush for the user:



* Now will change the port:that haven’t use so far though I had given as ‘9999’:

🡪edit the line as-



* Now will run the build again for successful, and will check the swagger-