Assignment for Today (UC6)

[https://github.com/kurukundaveera/INGFavBank.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2FINGFavBank.git&data=02%7C01%7Casma.ab%40hcl.com%7Cfa766700b6584783c83808d7696e4fbf%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637093794819838849&sdata=cm%2B5Pud%2BBdcwzhgqHUKZhRCieKsk8e8gtlC%2BCmxypMI%3D&reserved=0)--backend

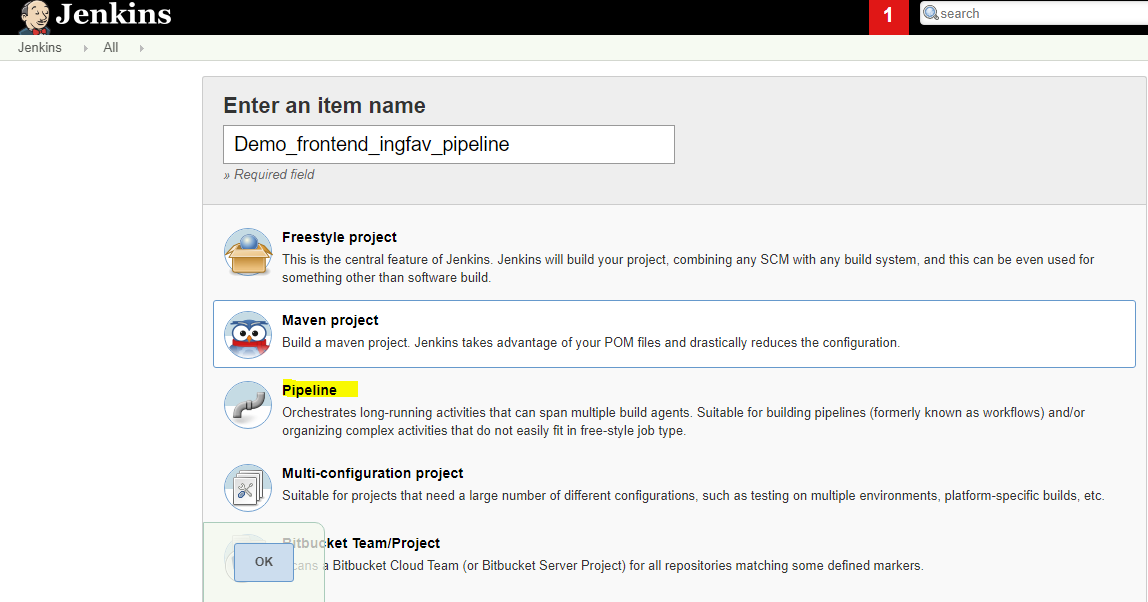
[https://github.com/kurukundaveera/INGFavAccount.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2FINGFavAccount.git&data=02%7C01%7Casma.ab%40hcl.com%7Cfa766700b6584783c83808d7696e4fbf%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637093794819848844&sdata=5Tk7%2FhhOyzfeHSyhwlT1ePjWdM3OJpspUet88bykEEM%3D&reserved=0) --backend

[https://github.com/kurukundaveera/INGFAV\_UI.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2FINGFAV_UI.git&data=02%7C01%7Casma.ab%40hcl.com%7Cfa766700b6584783c83808d7696e4fbf%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637093794819848844&sdata=3yWbjEjPjGxsOI4Om5104vGuwoW0FRG6IukVymdr%2B1M%3D&reserved=0)--frontend

UC6: CI/CD Implementation using Pipeline

1. Frontend of ING FAV bank using scripted pipeline:

* Create a new pipeline job for frontend as –“Demo\_Frontend\_pipeline”.



Fork the repo- [https://github.com/kurukundaveera/INGFAV\_UI.git](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fgithub.com%2Fkurukundaveera%2FINGFAV_UI.git&data=02%7C01%7Casma.ab%40hcl.com%7Cfa766700b6584783c83808d7696e4fbf%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637093794819848844&sdata=3yWbjEjPjGxsOI4Om5104vGuwoW0FRG6IukVymdr%2B1M%3D&reserved=0)

* Now will create pipeline script under ‘Pipeline’ tab-

\*\*\*\*\*\*\*\*\*\*\*\*\*

node('master'){

stage('Git checkout')

{

git 'https://github.com/asma-abdi/INGFAV\_UI'

}

stage('Build Analysis')

{

sh '''

npm install

npm run build

'''

}

stage('Deploy')

{

sh '''

cd /var/lib/jenkins/workspace/Demo\_Frontend\_Ingfav\_pipeline1

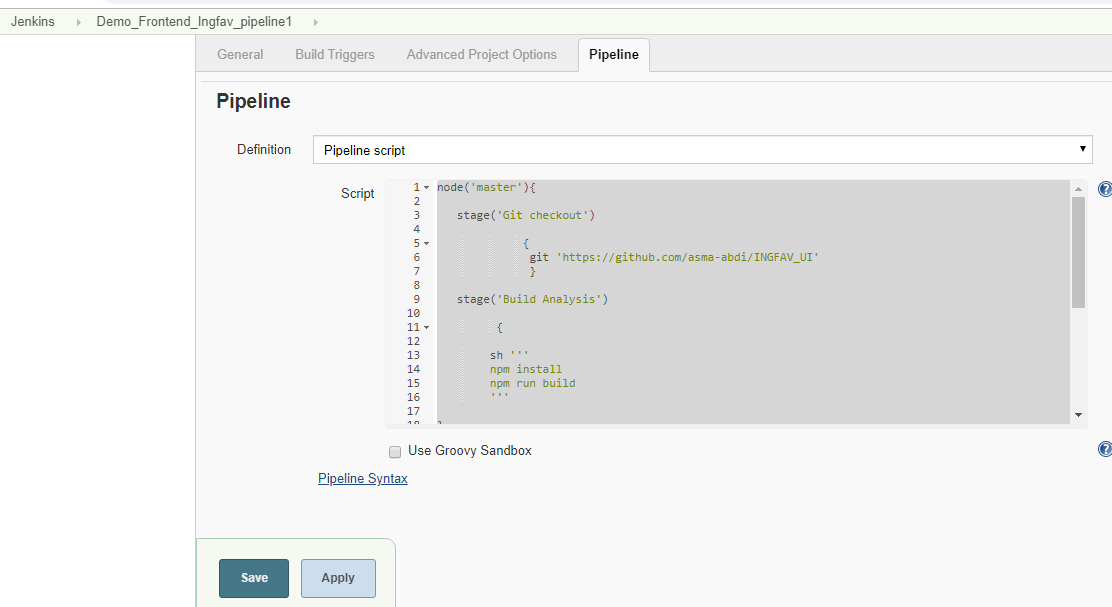
cp -rf build /opt/apache-tomcat-9.0.27/webapps

'''

}

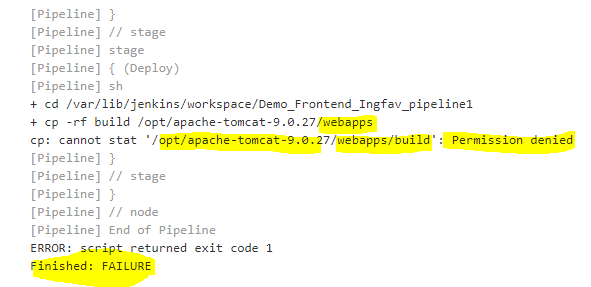
}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

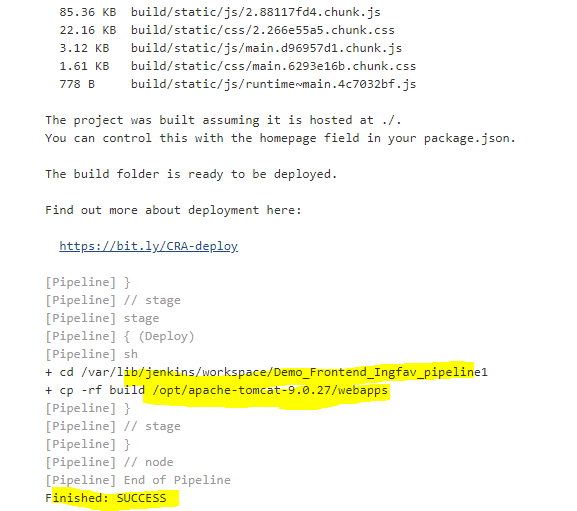


issue: for build success we need to provide permission to ‘webapps’ otherwise will get below error-

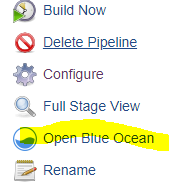


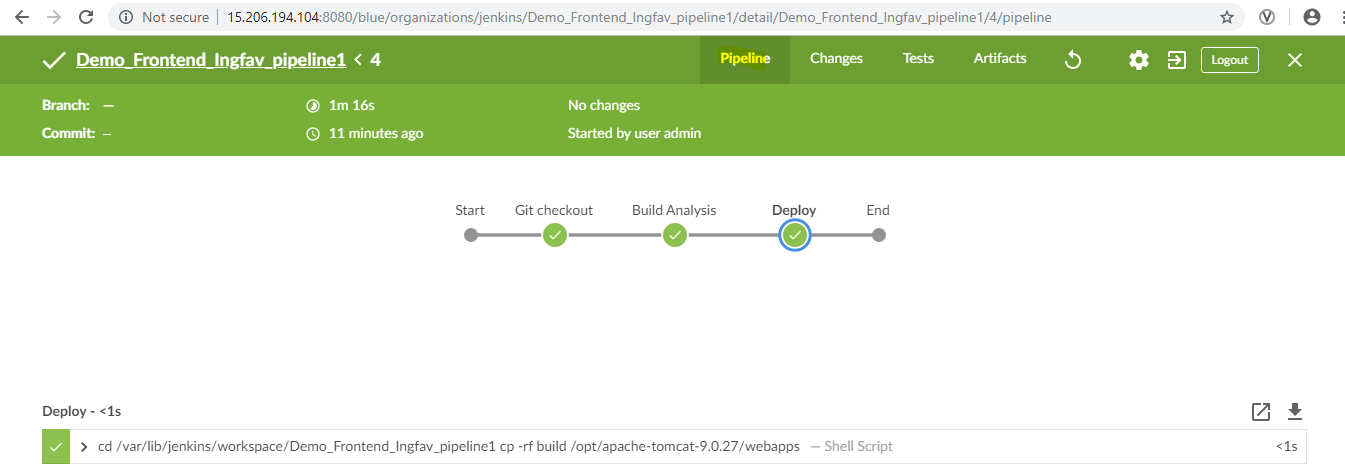


* Now will run the build, so that script will be executed successfully-



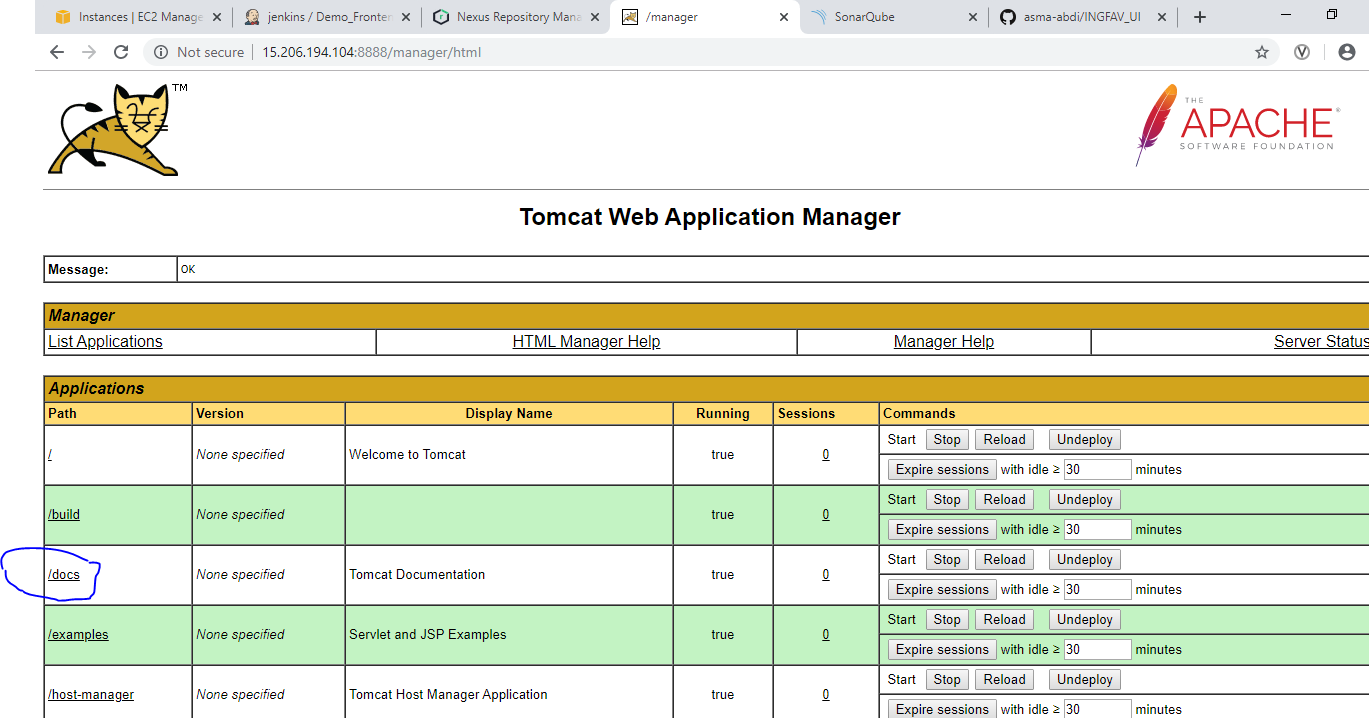
* Now will check the pipeline demo on blue ocean screen-



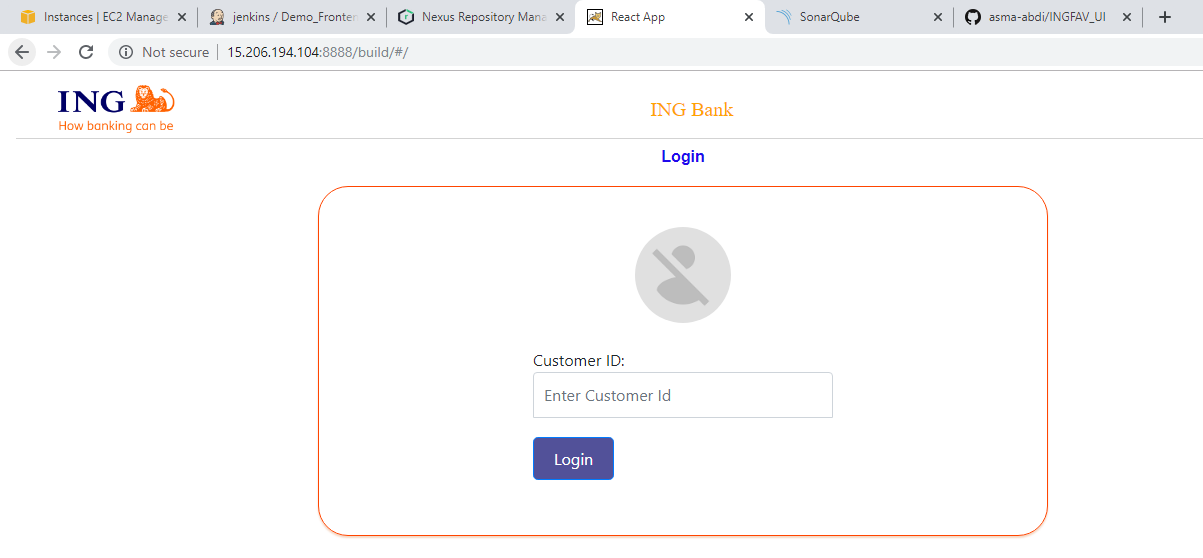


* Now will check the frontend, which has deployed on apache server-

Tomcat login🡪dashboard🡪manager Apps🡪/build , frontend will appear as below,

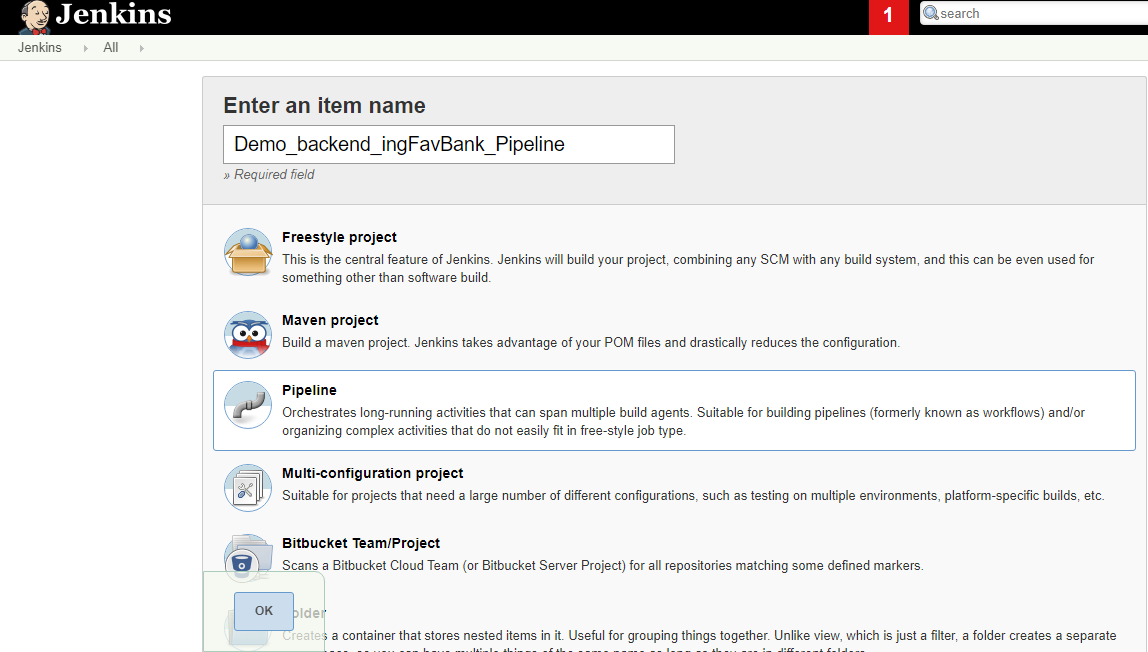


Frontend page(REACT)

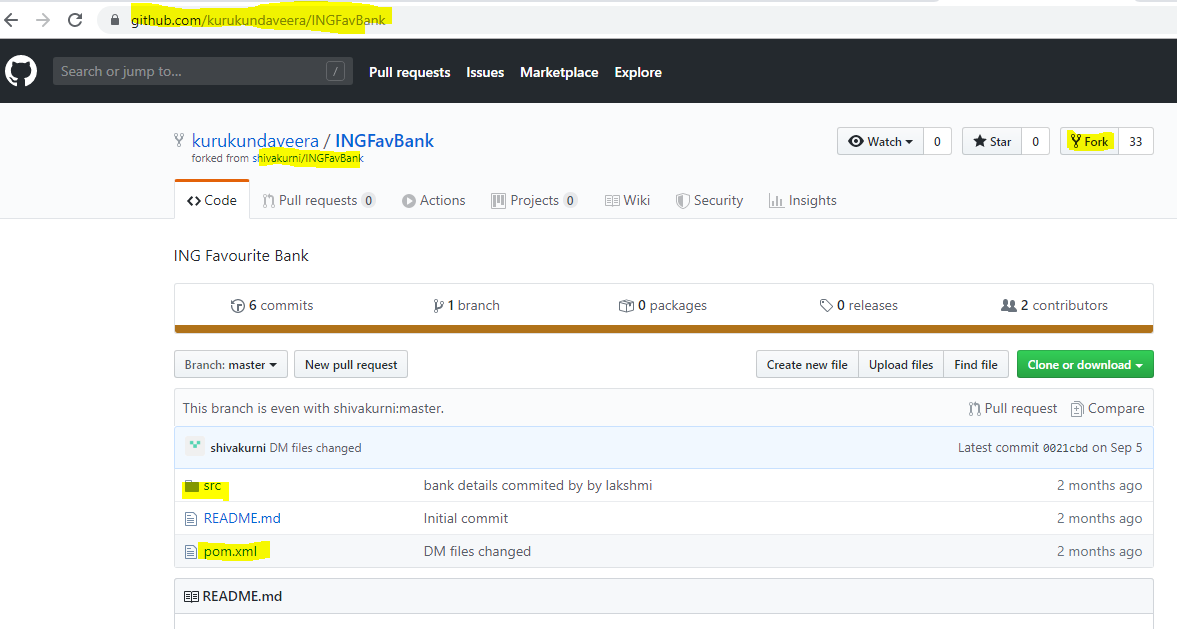


2- Backend using Scripted Pipeline using QualityGatesSonarqube:

* Create a new Jenkins pipeline job name as- ‘Demo\_backend\_ingFavBank\_Pipeline’,

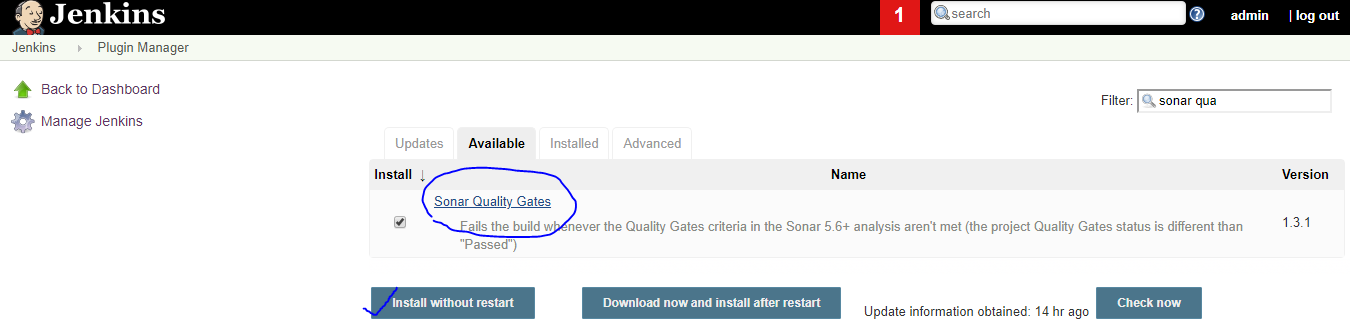


* Fork the repository-
* <https://github.com/kurukundaveera/INGFavBank.git--backend>



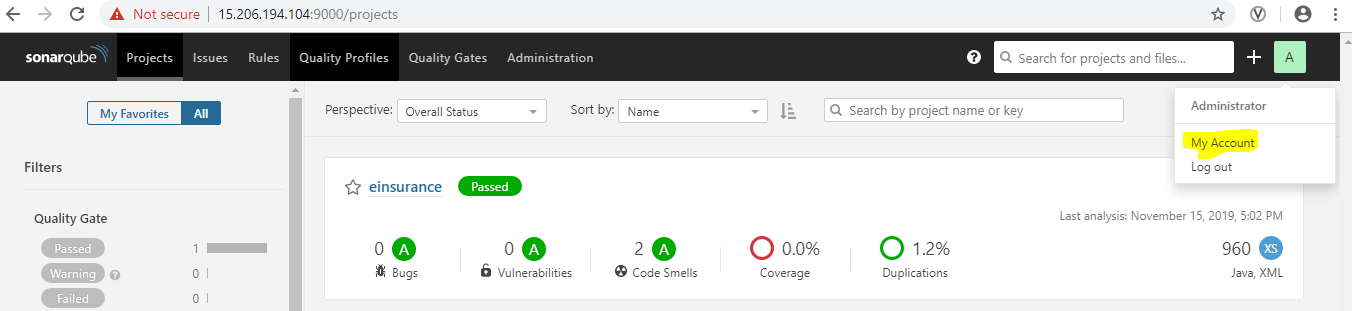
* **CI/CD Pipeline Script:**
* To implement quality gates, needs to install Sonar quality gates plugins in Jenkins-

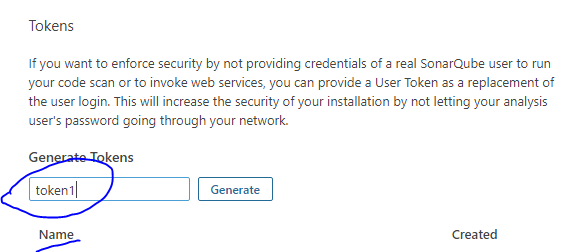
Manage Jenkins🡪Manage Plugins🡪Available🡪Sonar quality gates🡪 install;



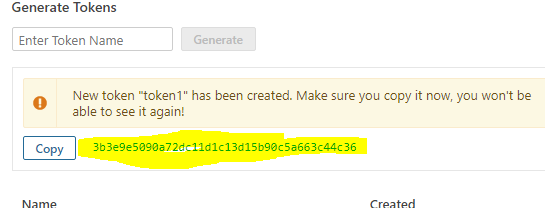
* Now create “token” for quality gate; for that-

Sonarqube dashboard🡪My ACCOUNT🡪security tab🡪page will apeear





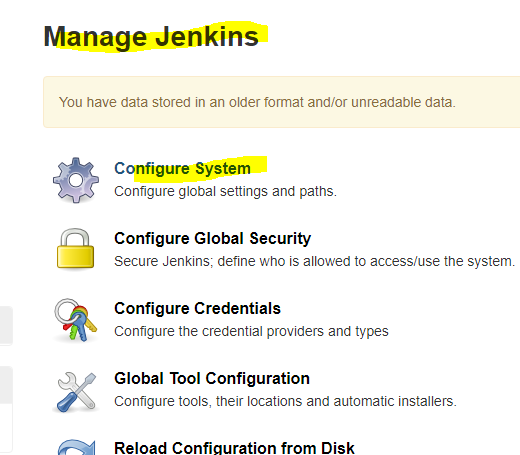
Click on “Generate’, by doing this will get a token that we will use in script.



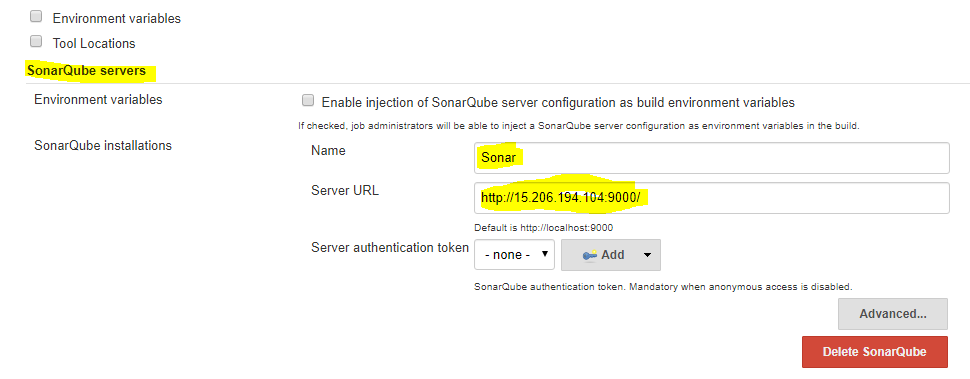
Token: 3b3e9e5090a72dc11d1c13d15b90c5a663c44c36

* Now Configure Sonarqube services in Jenkins:

Jenkins Dashboard🡪Manage Jenkins🡪Configure system

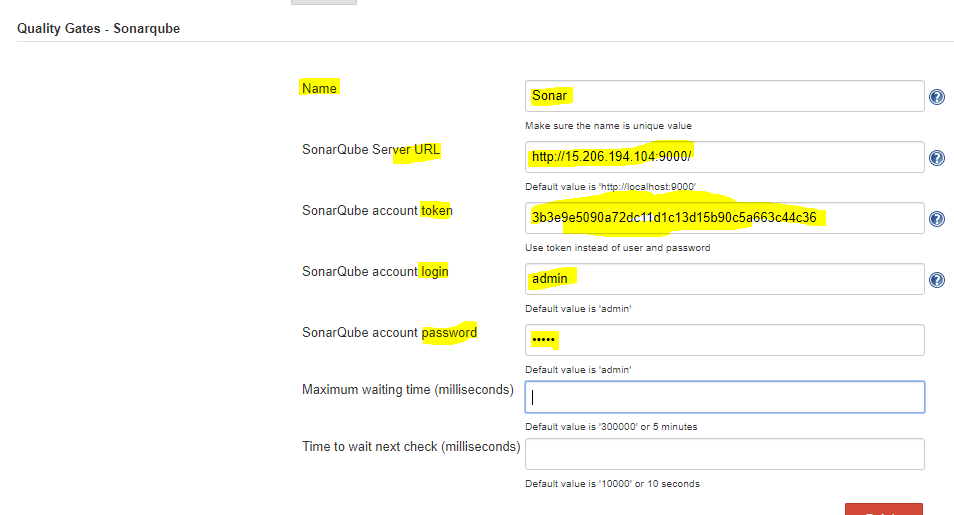


Sonarqube server:



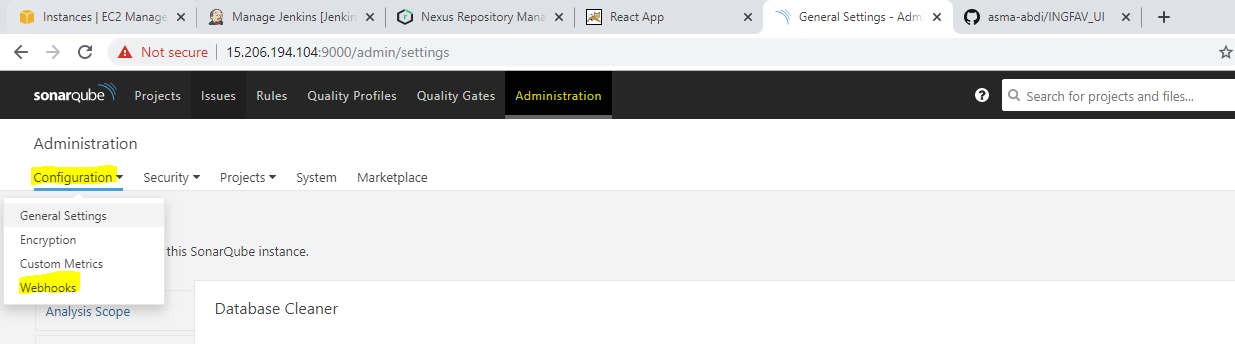
**Name should be same as that defined for SonarScanner for global configuration(case sensitive). Server**[**URL:http:publicip:port/**](URL:http:publicip:port/)

* Sonar Quality gate settings:



* Setup Sonar Webhooks:

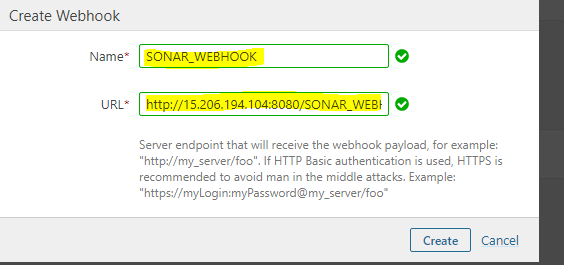
SonarQube dashboard🡪Administration tab🡪dropdown ‘Configuration’🡪webhooks🡪Create



Webhook Creation

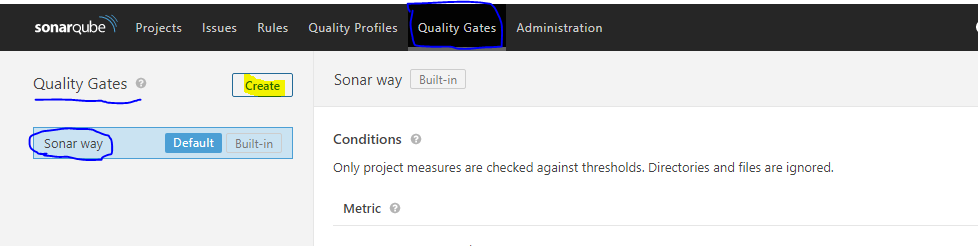
Name:any valid name convention as “SONAR\_WEBHOOK’

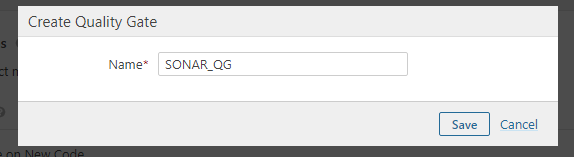
URL:Jenkins URL/webhook name as- <http://15.206.194.104:8080/SONAR_WEBHOOK>S/



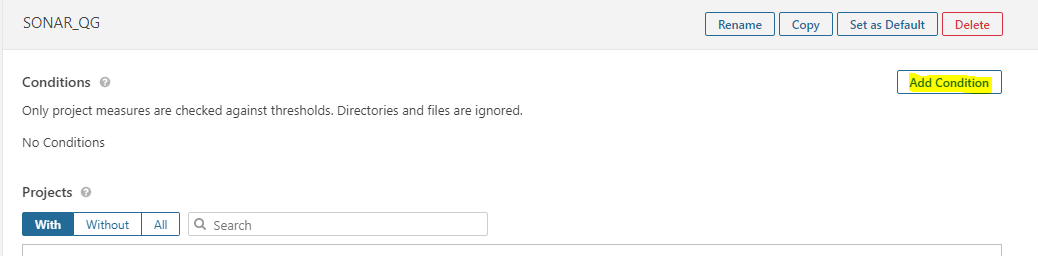
* Quality gate Setup:

Sonar Dashboard🡪Quality Gates Tab🡪create

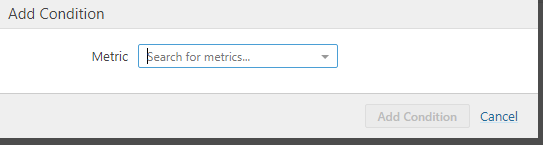


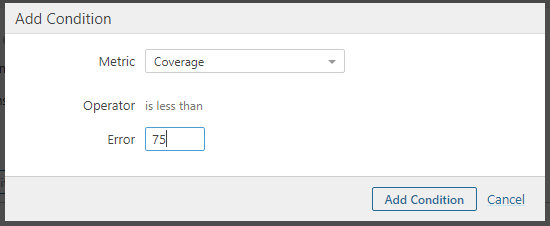


* Now Add Condition of Quality gates that needs to satisfied at deployment of code coverage-



Under metric select “coverage”, and can define the condition less than eg. 75.





NOTE: after build successful, if coverage is less than 75 it will be fail.

* CI/CDPipeline Script using Sonarqube quality gate:

node('master'){

stage('Git checkout')

{

git 'https://github.com/asma-abdi/INGFavBank'

}

stage('Build Analysis')

{

withSonarQubeEnv('sonar')

{

sh '/opt/maven/bin/mvn clean install sonar:sonar -Dsonar.password=admin -Dsonar.login=admin -Dmaven.test.skip=true'

}

}

stage('Quality Gate Check')

{

timeout(time:5, unit: 'MINUTES')

{

defqg = waitForQualityGate()

if (qg.status != 'OK')

{

error "Pipeline aborted due to quality gate failure: ${qg.status}"

}

}

}

stage('Deploy')

{

sh '/opt/maven/bin/mvn clean deploy '

}

stage('Execute')

{

sh 'export JENKINS\_NODE\_COOKIE=dontKillMe ;nohup java -jar $WORKSPACE/target/\*.jar &'

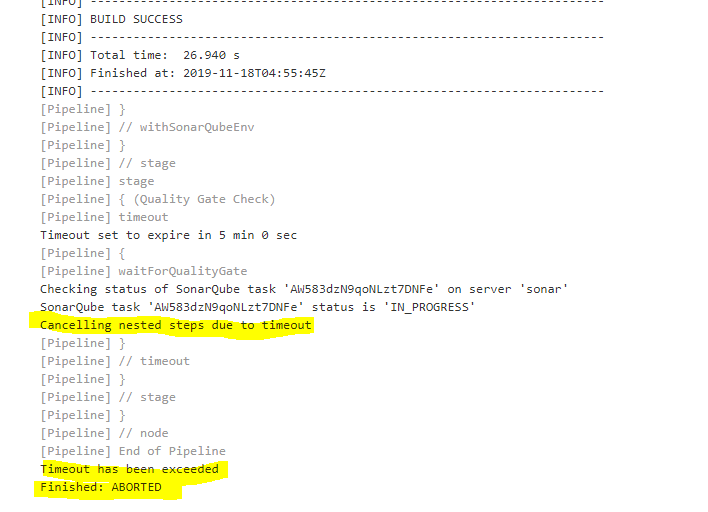
}

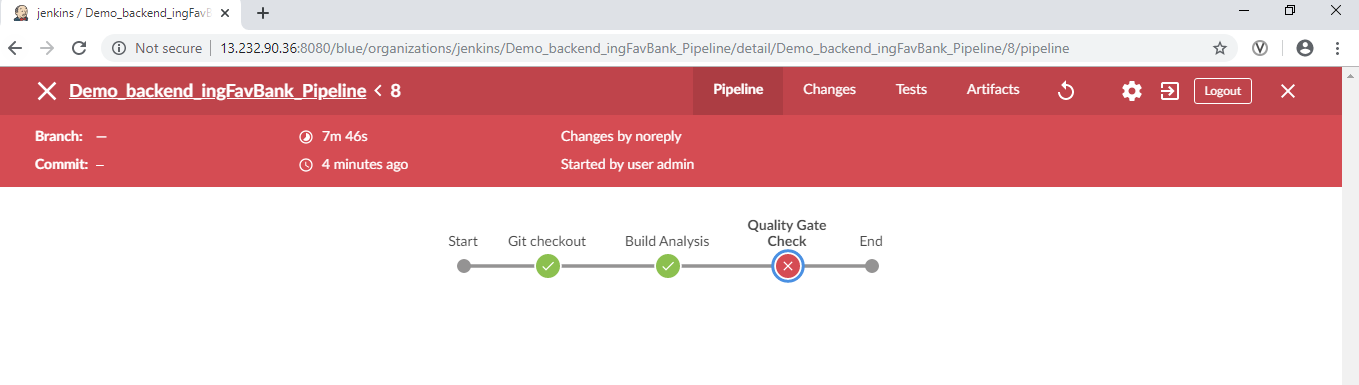
}

NOTE:

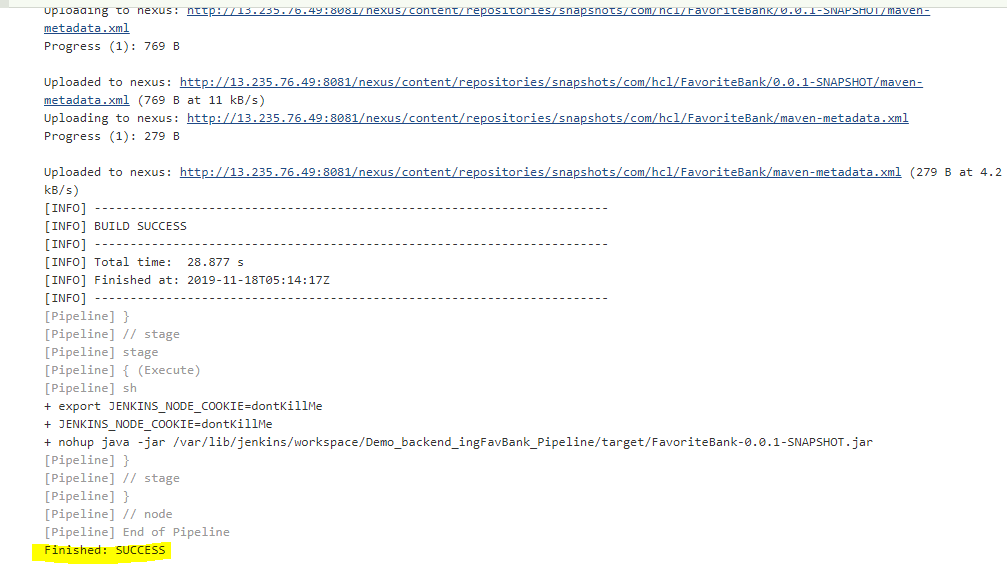
QG fail Scenario1:-If the build execution time is more than 5 minutes , in that case pipeline will be aborted reason being QG will wait for response till 5 minutes unit. After that it will abort the pipeline, so that build will also get failure.

Build Failure at Console Output Screen-

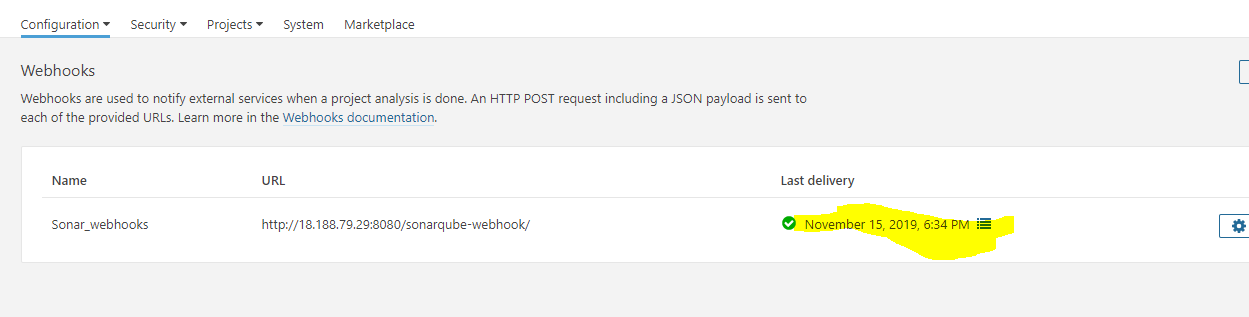




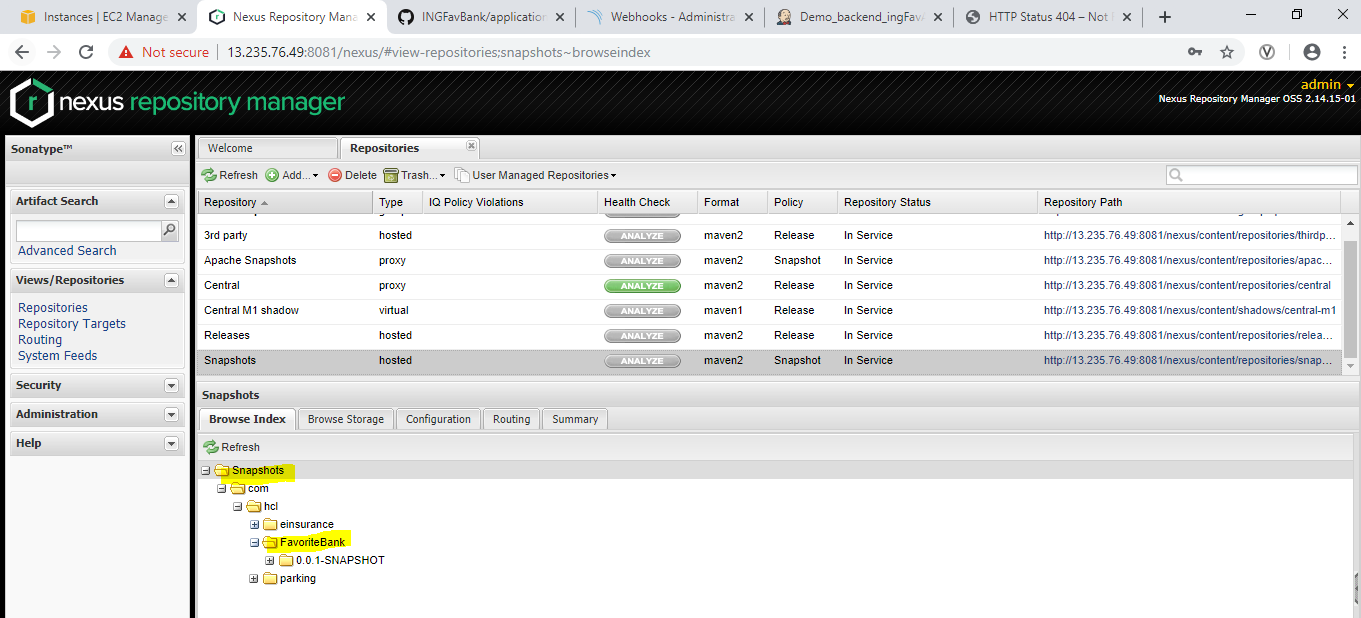
Scenario: Successsful build



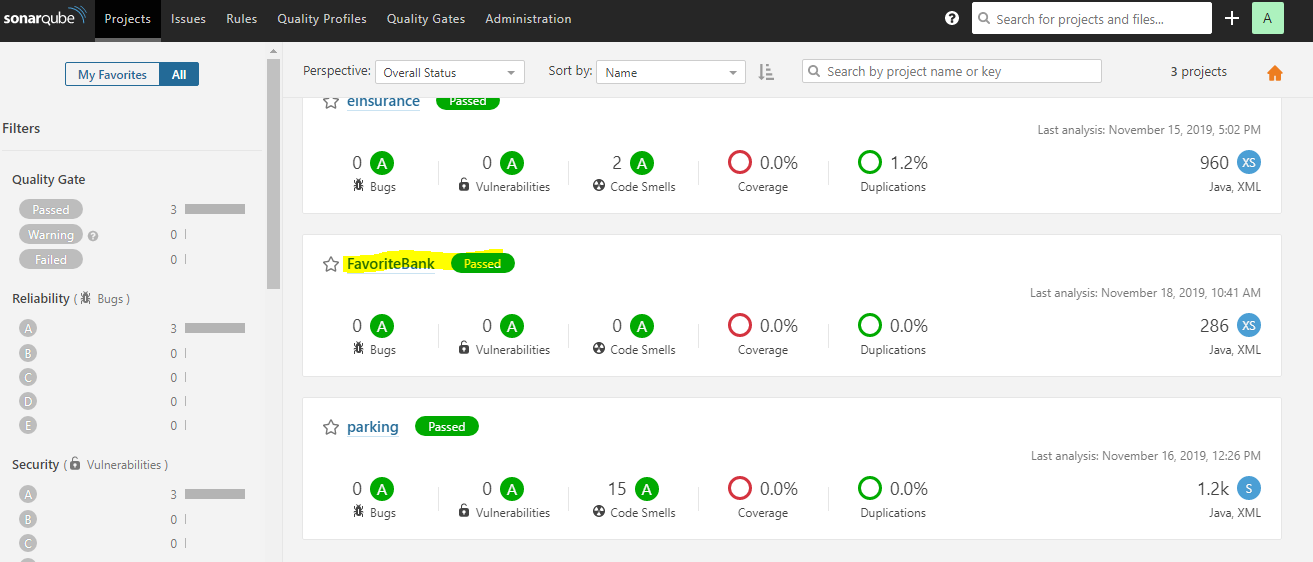
* Now my build is success now I can go and check in sonar webhooks it should gve a check mark.



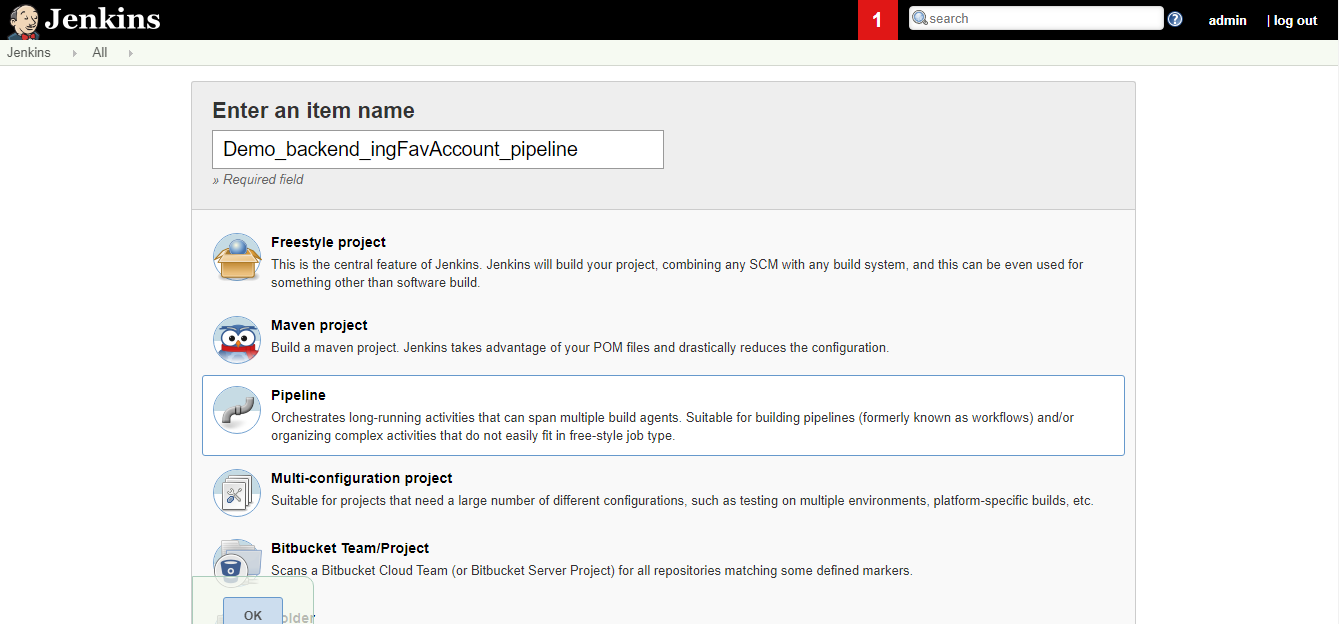
* Nexus

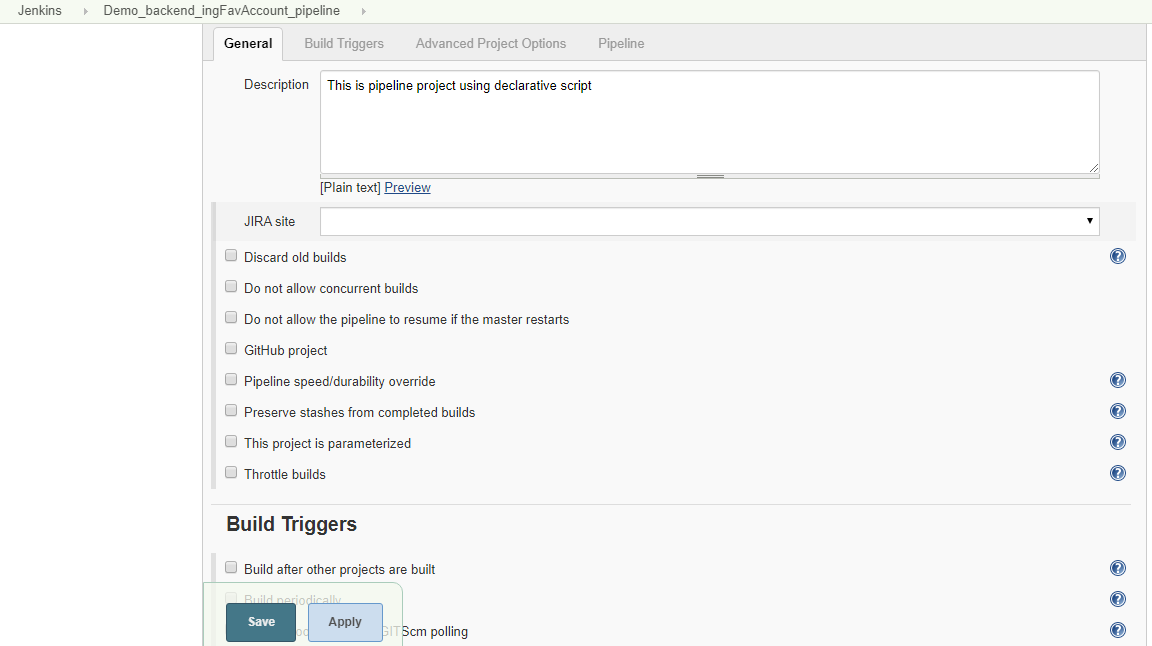


* Sonar



3- IngFavAccount-declarative





* Declarative script:

\*\*\*\*\*\*\*\*\*\*\*\*

pipeline {

agent any

stages {

stage('SCM Checkout') {

steps {

git url:'https://github.com/asma-abdi/INGFavAccount.git'

}

}

stage('Build') {

steps {

sh"/opt/maven/bin/mvn clean package -Dmaven.test.skip=true"

}

}

stage('Build Analysis') {

steps {

withSonarQubeEnv('sonar')

{

sh "/opt/maven/bin/mvn clean verify sonar:sonar -Dmaven.test.skip=true"

}

}

}

stage("Quality Gate") {

steps {

timeout(time: 50, unit: 'MINUTES') {

waitForQualityGateabortPipeline: true

}

}

}

stage('Deploy') {

steps {

sh "/opt/maven/bin/mvn clean deploy -Dmaven.test.skip=true"

}

}

stage('Release') {

steps {

sh"export JENKINS\_NODE\_COOKIE=dontKillMe; nohup java -jar $WORKSPACE/target/\*.jar &"

}

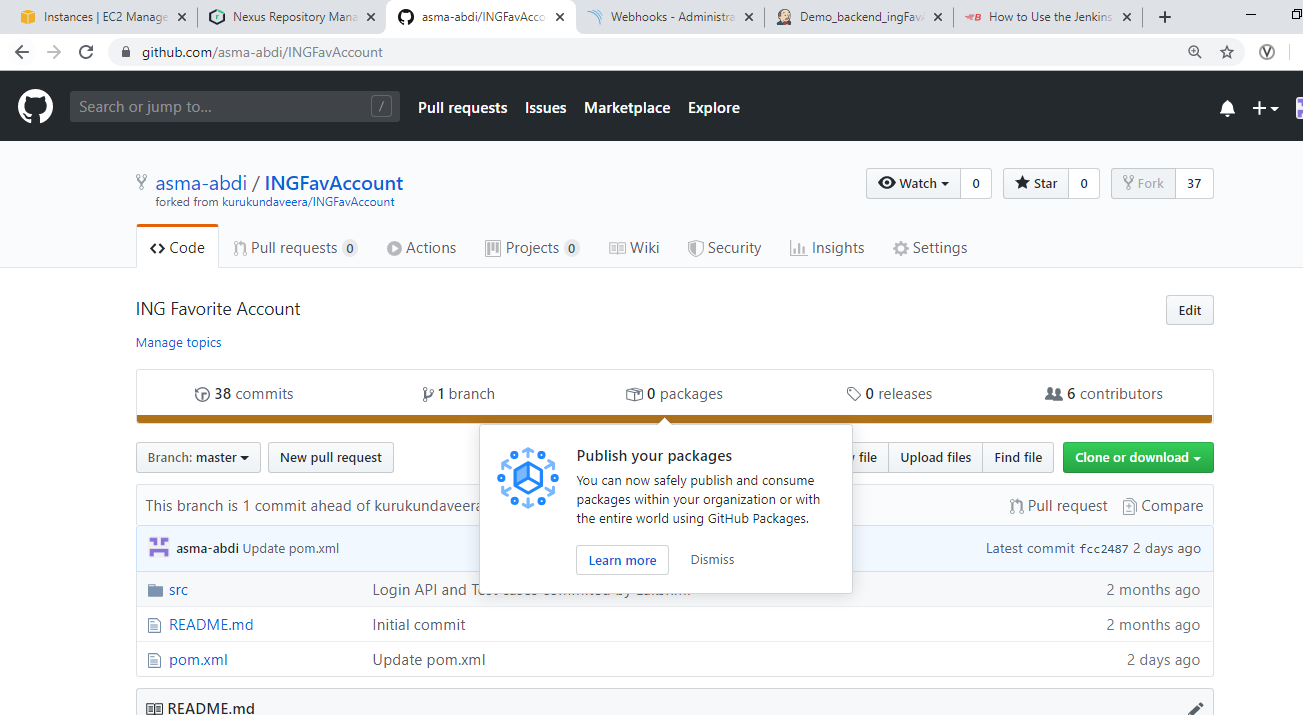
}

}

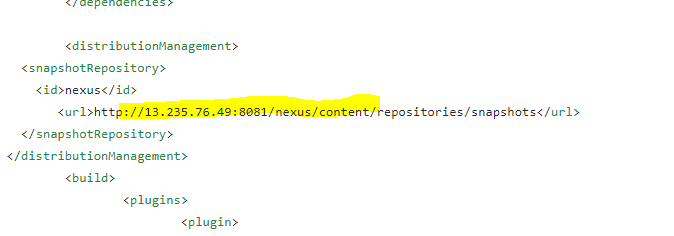
}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

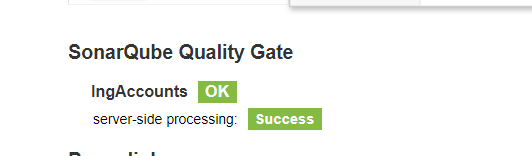
* Fork the repo



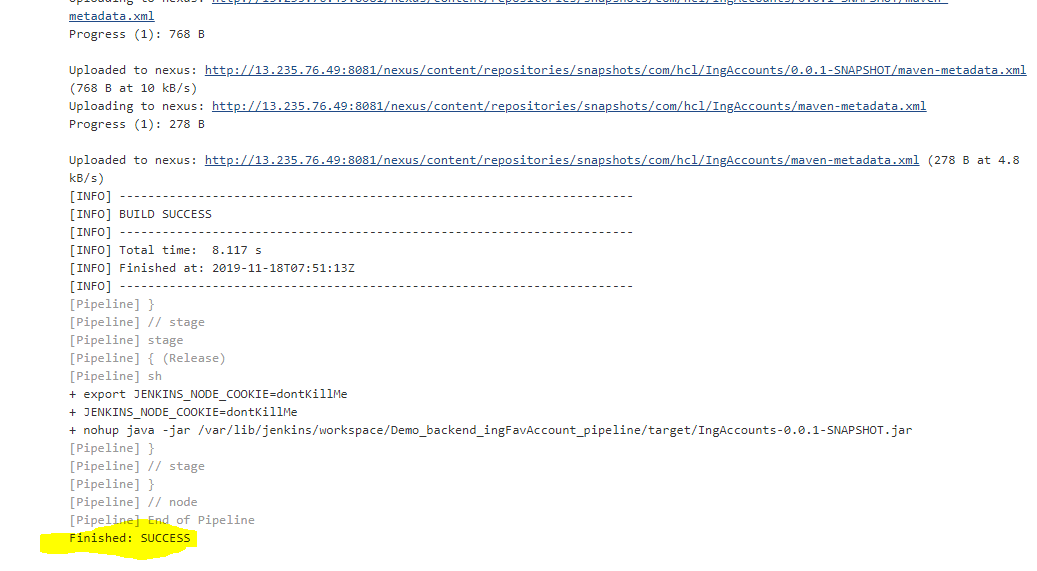
* Pom.xml



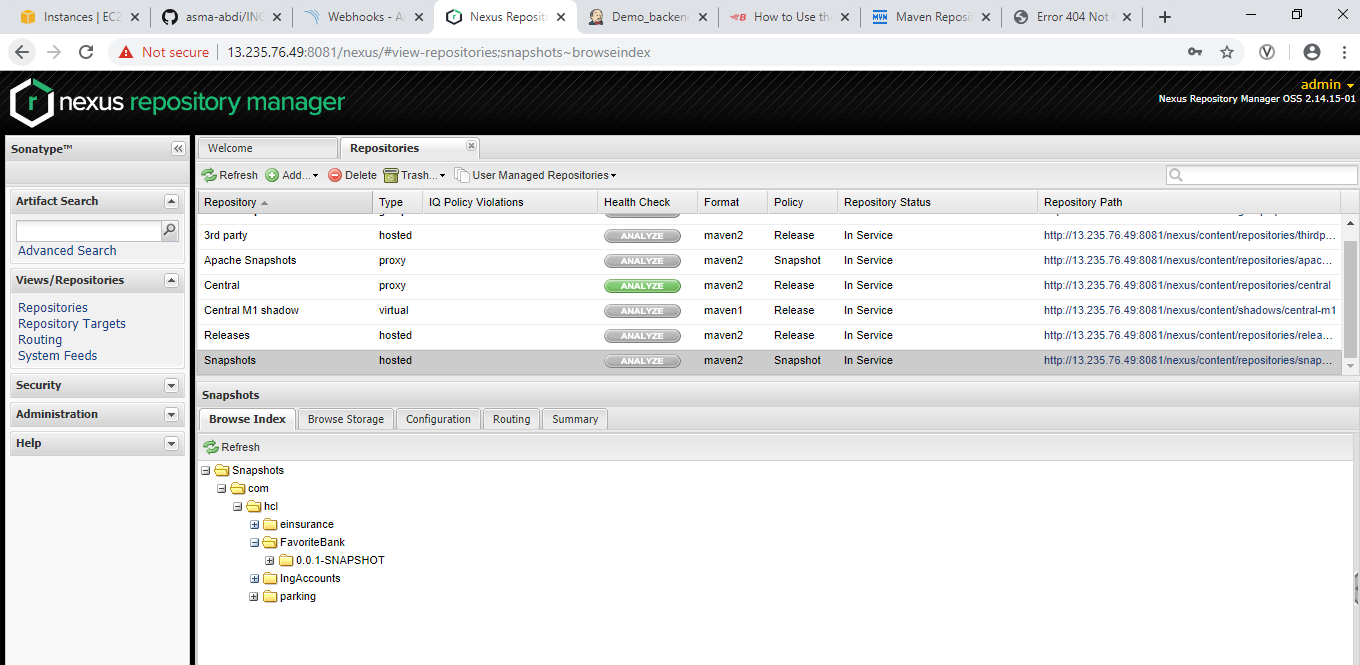
* Application.properties
* Quality gates



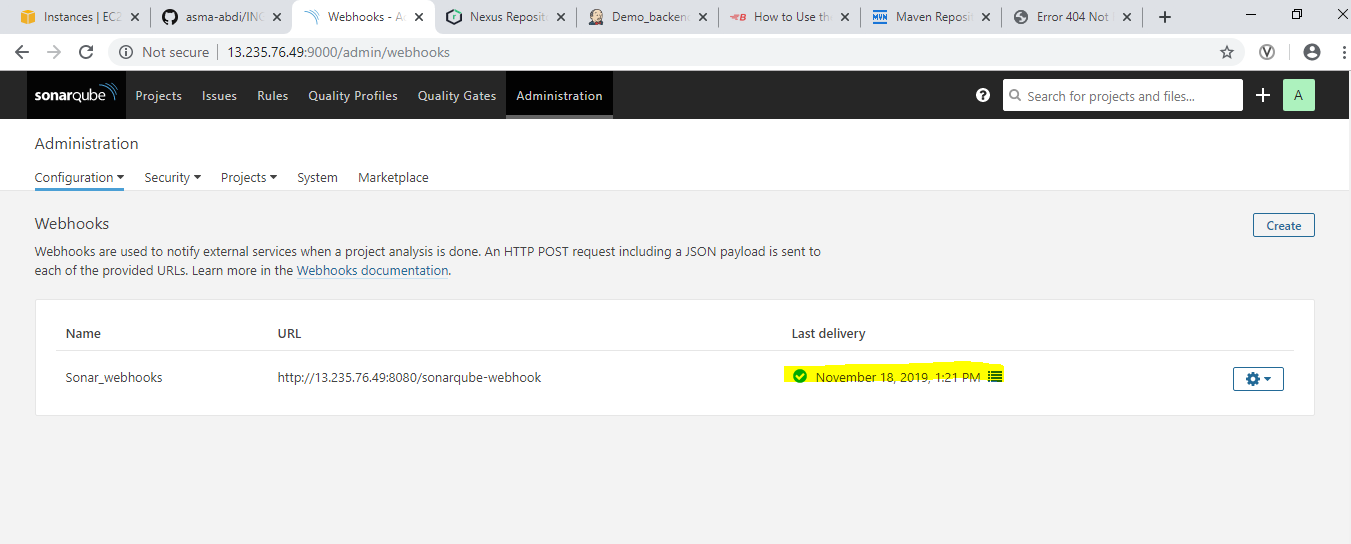
* Build success

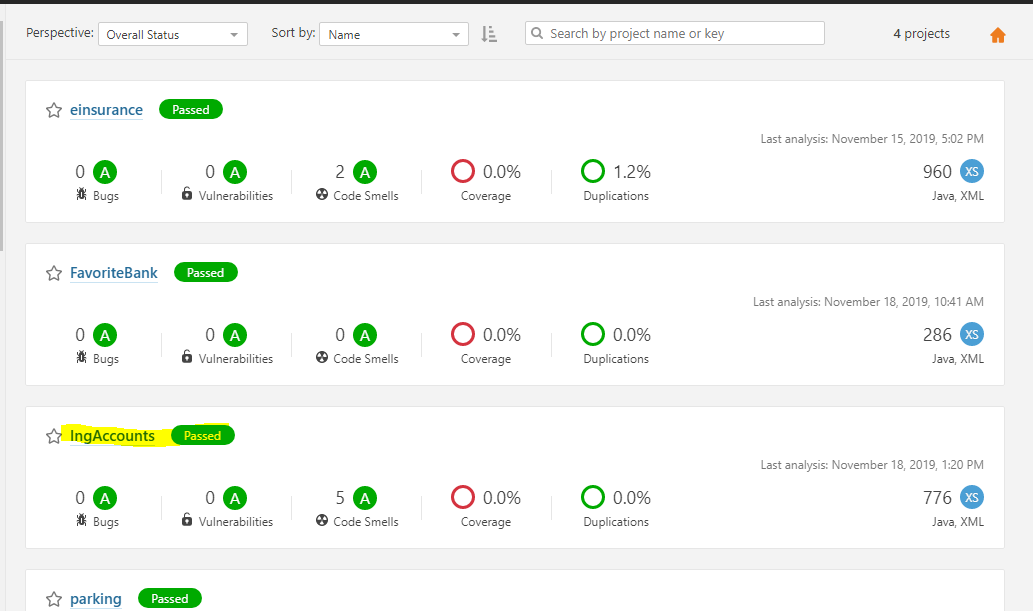


* Nexus:



* Sonar: webhooks





* Blue ocean-

