PARKING-BACKEND

Create simple pipeline job in the pipeline script we have to write declarative script as shown below.

pipeline {

agent any

stages {

stage('SCM Checkout') {

steps {

git 'https://github.com/Gkasiraju/parking\_backend.git'

}

}

stage ('Build && SonarQube analysis') {

steps {

withSonarQubeEnv('sonar'){

sh '/opt/maven/bin/mvn clean package sonar:sonar -Dsonar.password=admin123 -Dsonar.login=admin -Dmaven.test.skip=true'

//maven.test.skip property to skip compiling the tests//

}

}

}

stage('Quality Gate'){

steps {

timeout(time: 1, unit: 'HOURS'){

waitForQualityGate abortPipeline: true

}

}

}

stage ('Artifact-Deployment') {

steps {

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

}

}

stage ('Release') {

steps {

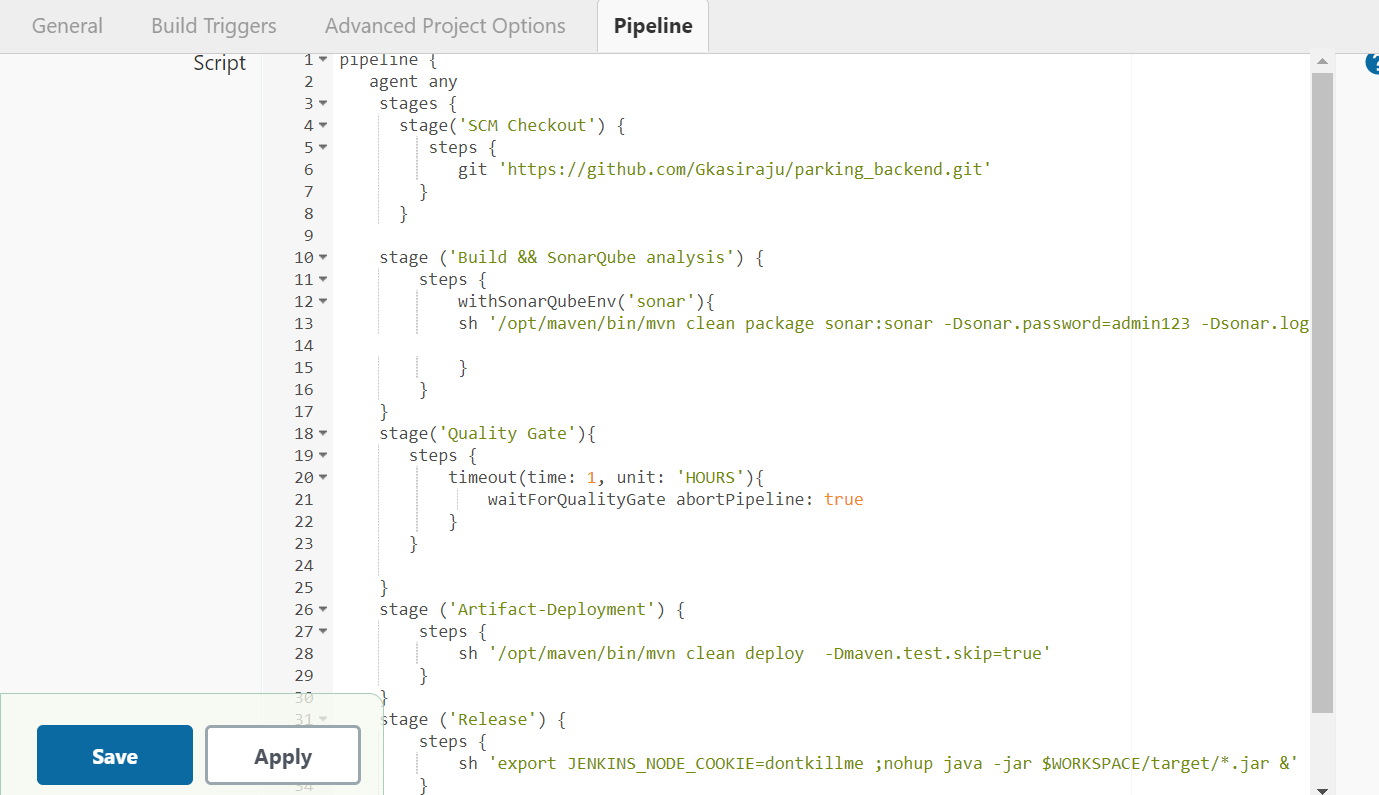
sh 'export JENKINS\_NODE\_COOKIE=dontkillme ;nohup java -jar $WORKSPACE/target/\*.jar &'

}

}

}

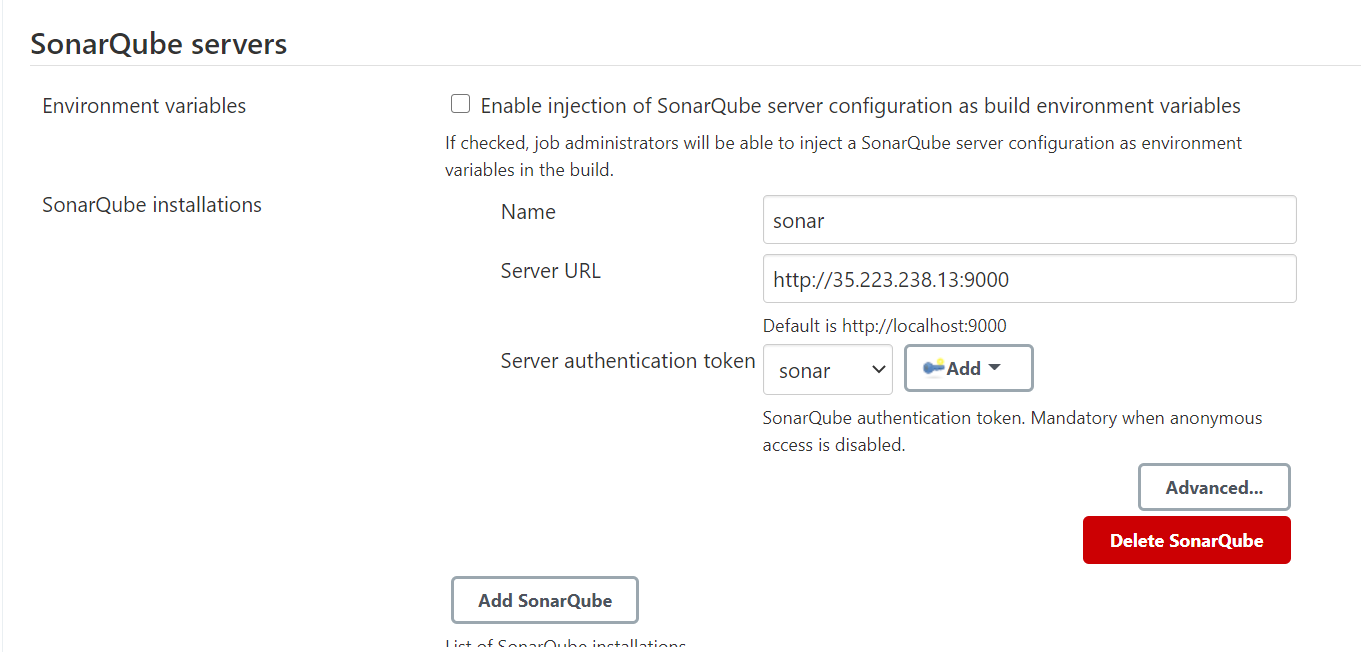
}



*Step 1*:

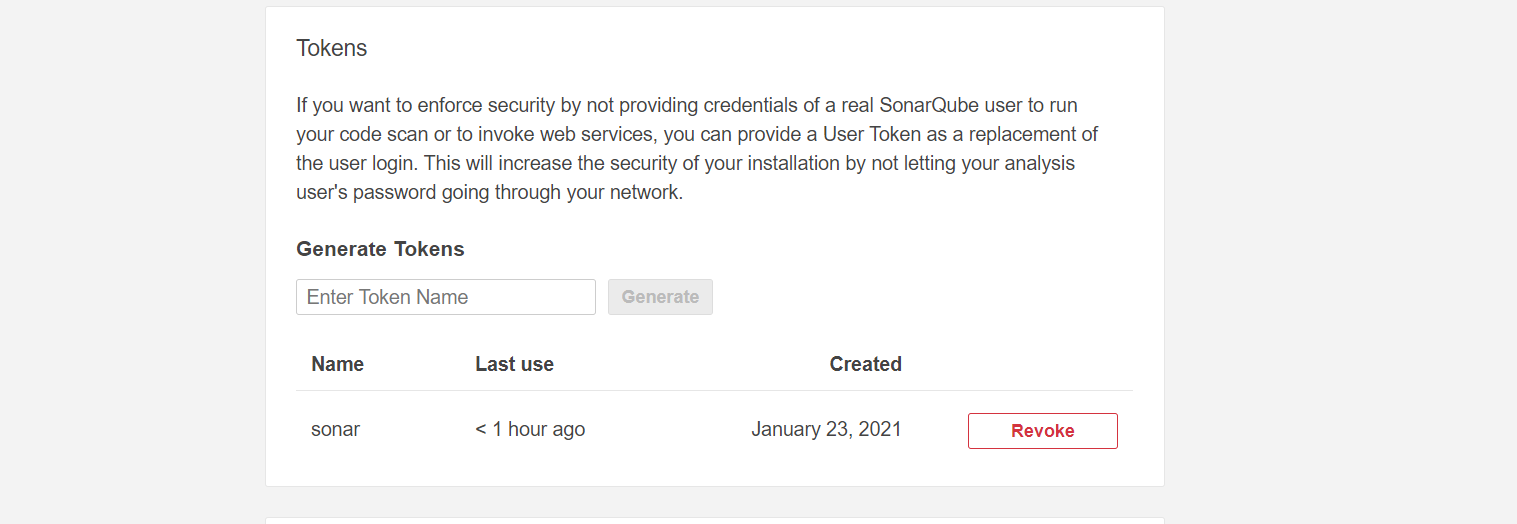
Download SonarQube Scanner plugin and add SonarQube in the path shown below:

Manage plugins->configuration->add SonarQube



Then After we need to generate token in SonarQube in the path :

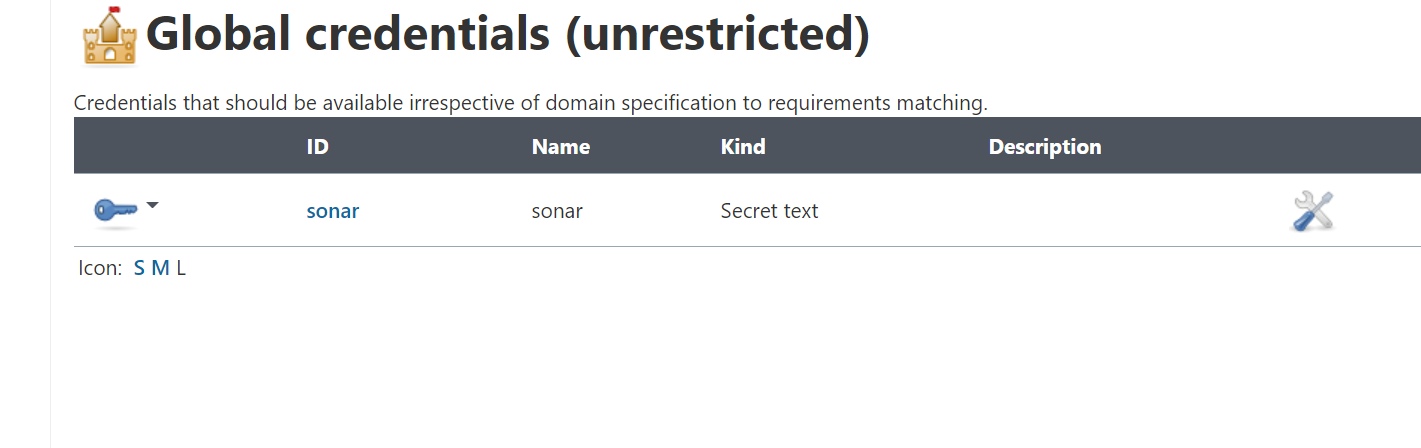
SonarQube->my Account->Generate Token



9863bc0c5eb3e9ef045b0616e26b454adead4c35

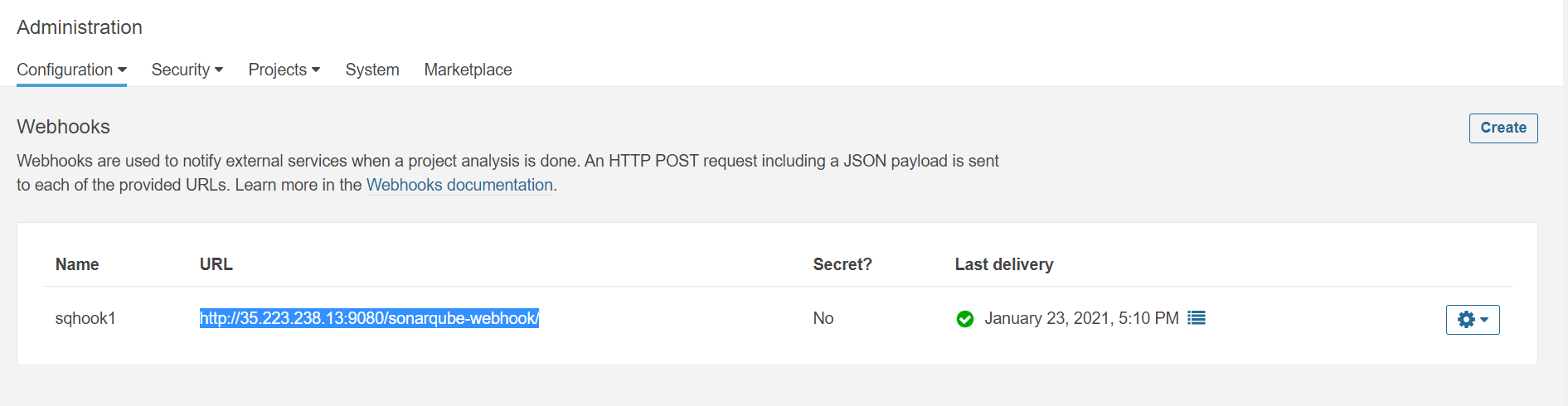
Then go to Jenkins Dashboard we need to add token in the path show below:

Manage Jenkins->Manage Credential->Jenkins->Global credential Here we need to add a token in secret text , After successfully added it will show like this



Then we need to create webhook in Nexus in the path

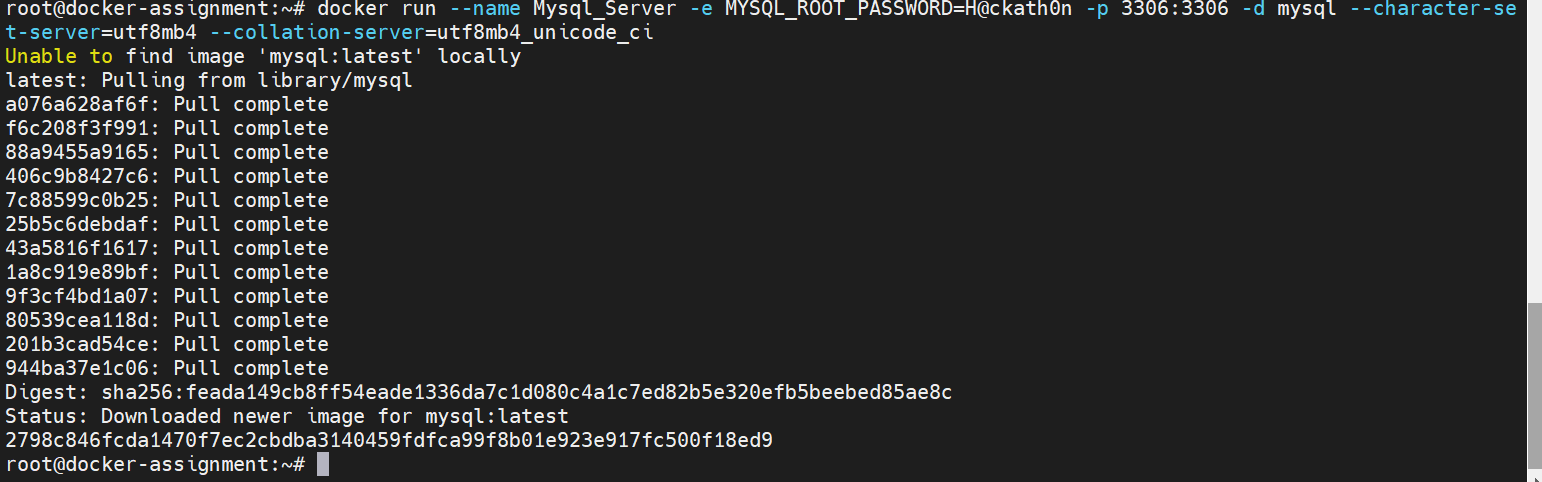
Nexus-> Administaration->configuration->webhook then create webhook as a name of sonar and the url <http://35.223.238.13:9080/sonarqube-webhook/> then create .



Then after we need to setup database as name of Parkingdb.

Goto mobaXterm pull mysql image form hub.docker.com .

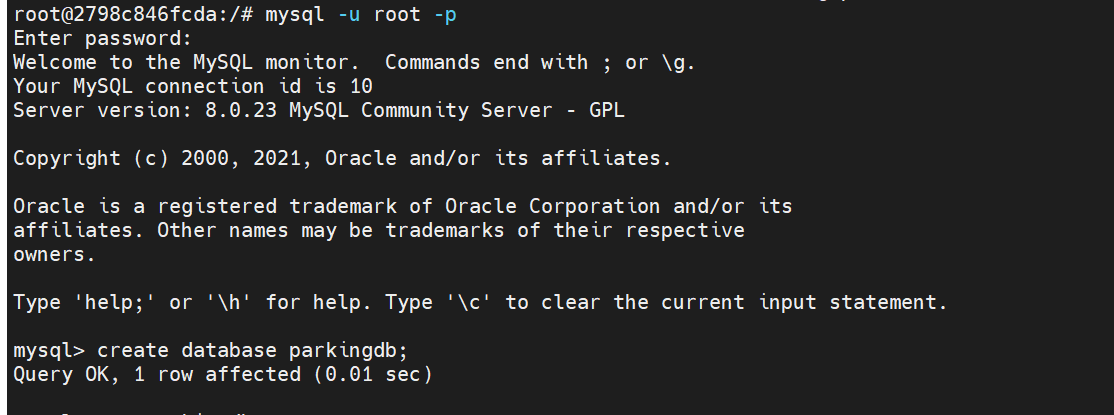
**docker run --name Mysql\_Server -e MYSQL\_ROOT\_PASSWORD=H@ckath0n -p 3306:3306 -d mysql --character-set-server=utf8mb4 --collation-server=utf8mb4\_unicode\_ci**



**docker exec -it Mysql\_Server /bin/bash**



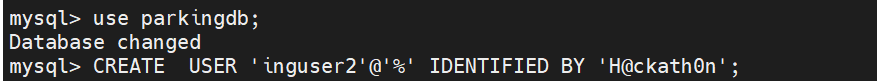
**mysql -u root –p**



**create database parkingdb;**



**use parkingdb;**

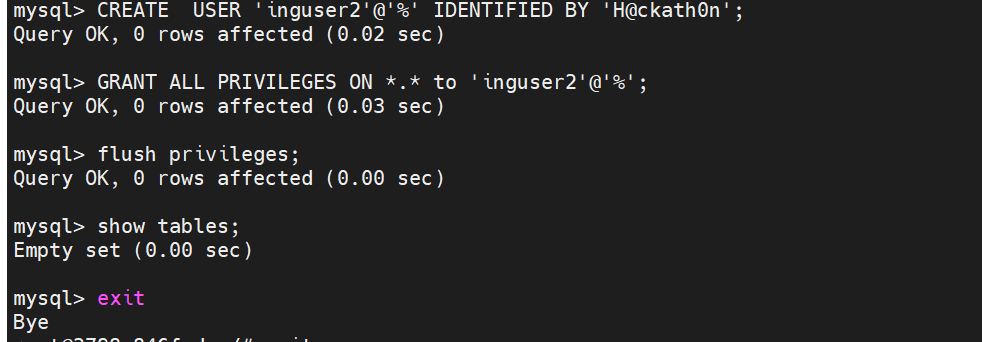


**CREATE  USER 'inguser2'@'%' IDENTIFIED BY 'H@ckath0n';**

**GRANT ALL PRIVILEGES ON \*.\* to 'inguser2'@'%';**

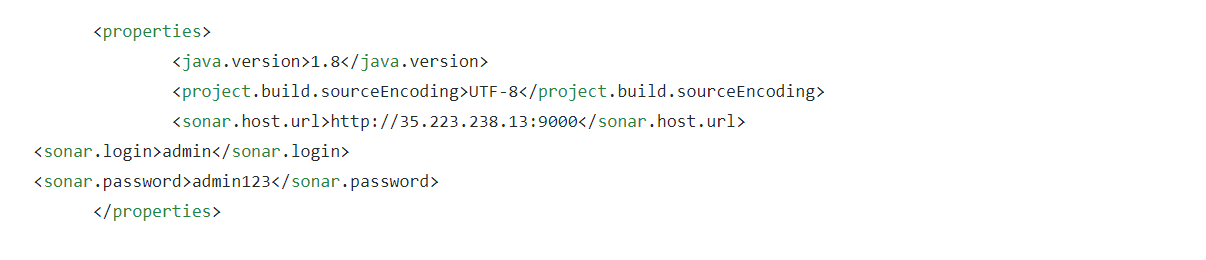
**flush privileges;**

**exit**



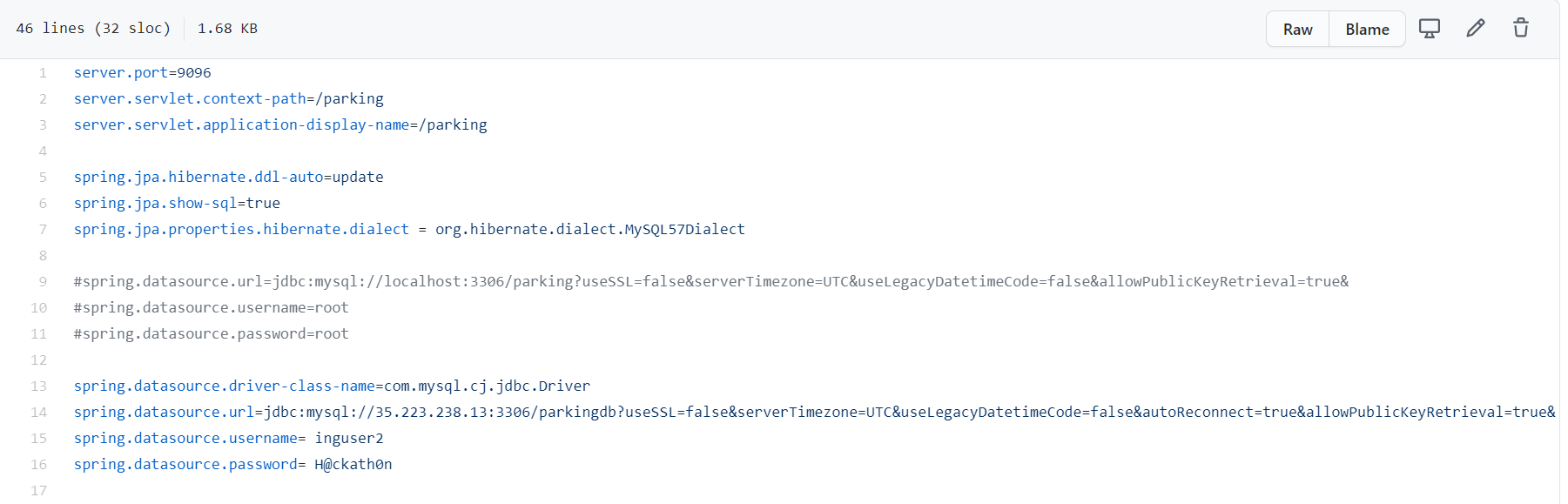
Upto this one I am successfully created Database as a name of Parkingdb

Go to pom.xml add the required snapshot repository as shown below



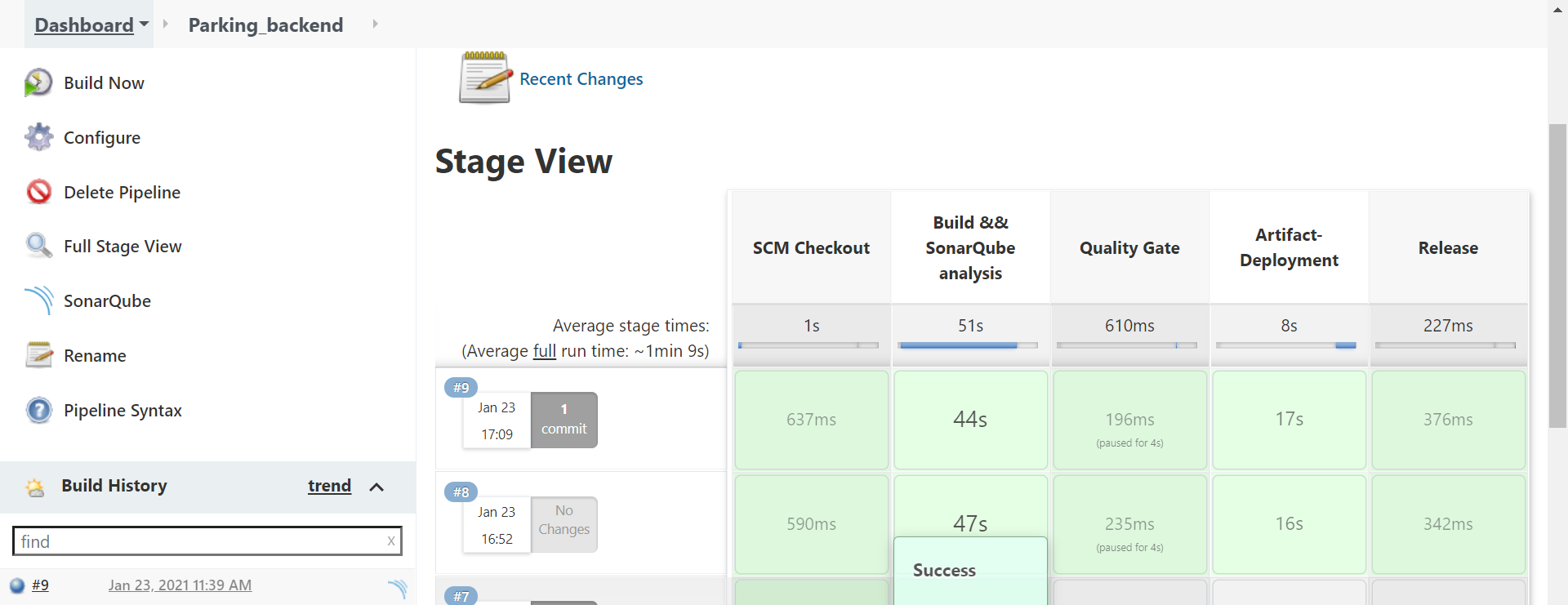


Then I need to change the application. Properties in that I need to add server.port=9096 and add database username as inguser2 as shown below.

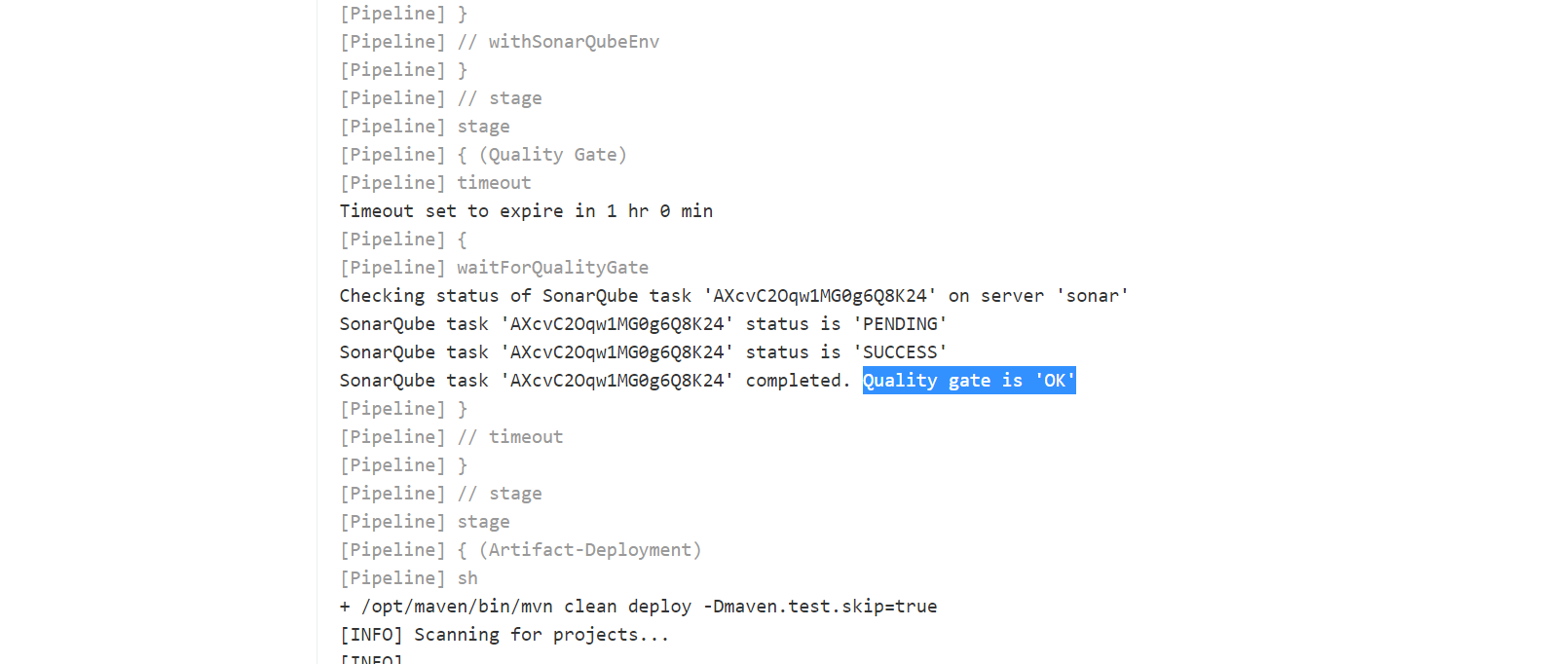


|  |
| --- |
| server.port=9096 |
|  | server.servlet.context-path=/parking |
|  | server.servlet.application-display-name=/parking |
|  |  |
|  | spring.jpa.hibernate.ddl-auto=update |
|  | spring.jpa.show-sql=true |
|  | spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL57Dialect |
|  |  |
|  | #spring.datasource.url=jdbc:mysql://localhost:3306/parking?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&allowPublicKeyRetrieval=true& |
|  | #spring.datasource.username=root |
|  | #spring.datasource.password=root |
|  |  |
|  | spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver |
|  | spring.datasource.url=jdbc:mysql://35.223.238.13:3306/parkingdb?useSSL=false&serverTimezone=UTC&useLegacyDatetimeCode=false&autoReconnect=true&allowPublicKeyRetrieval=true& |
|  | spring.datasource.username= inguser2 |
|  | spring.datasource.password= H@ckath0n |
|  |  |
|  |  |
|  | sonar.host.url=http://35.223.238.13:9000 |
|  |  |
|  |  |

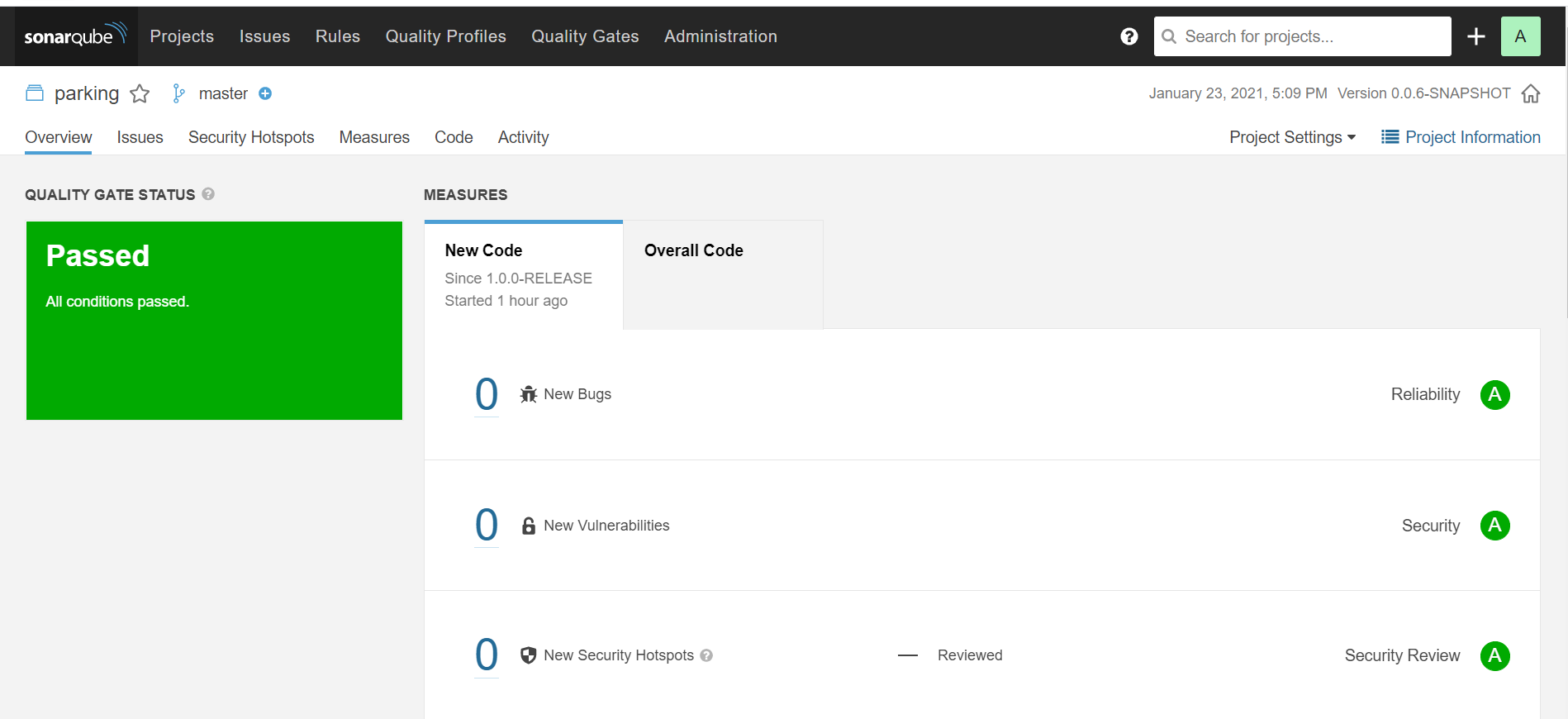
Finally Go to Jenkins click on Build Now



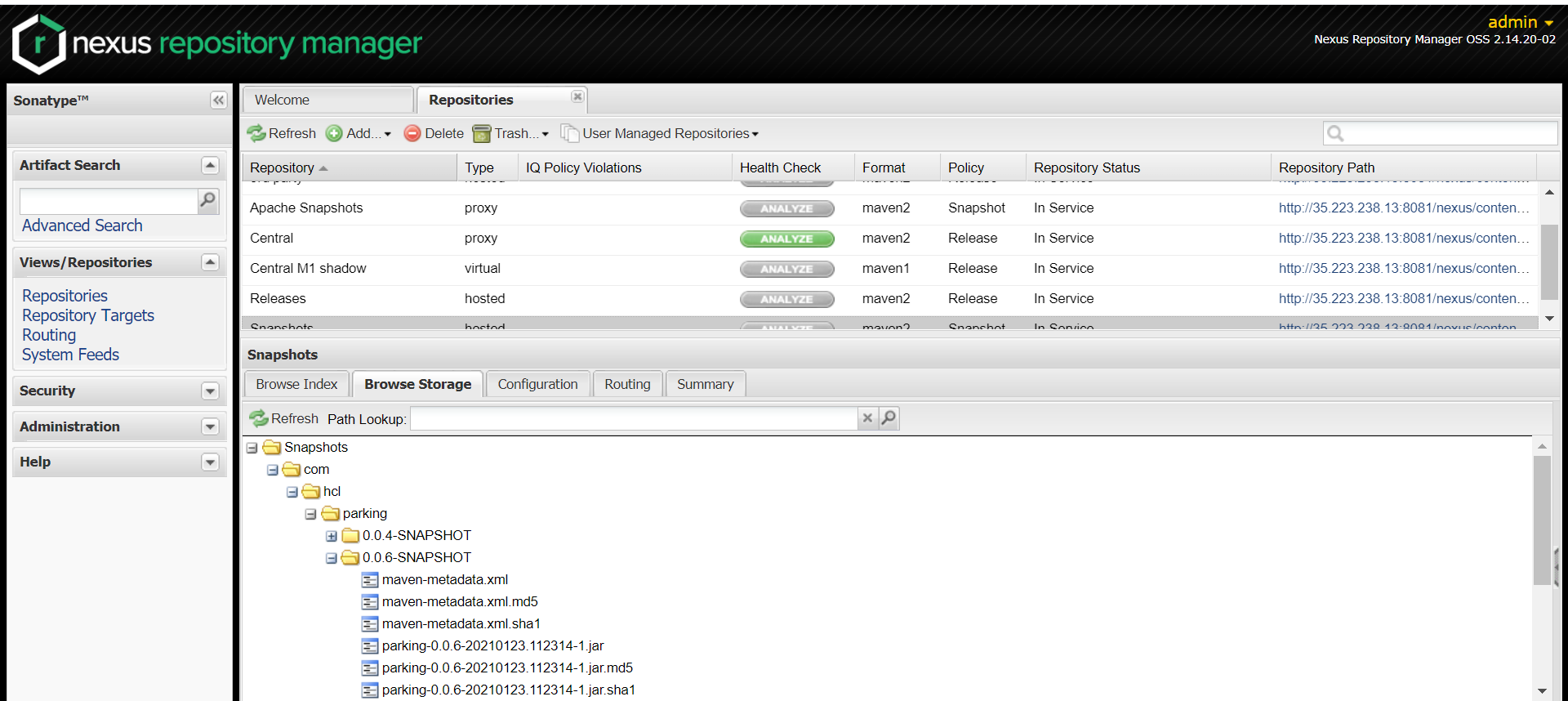
We can also check in console output as shown below



Then go to SonarQube check the code quality and any error all conditions passes and shown below.

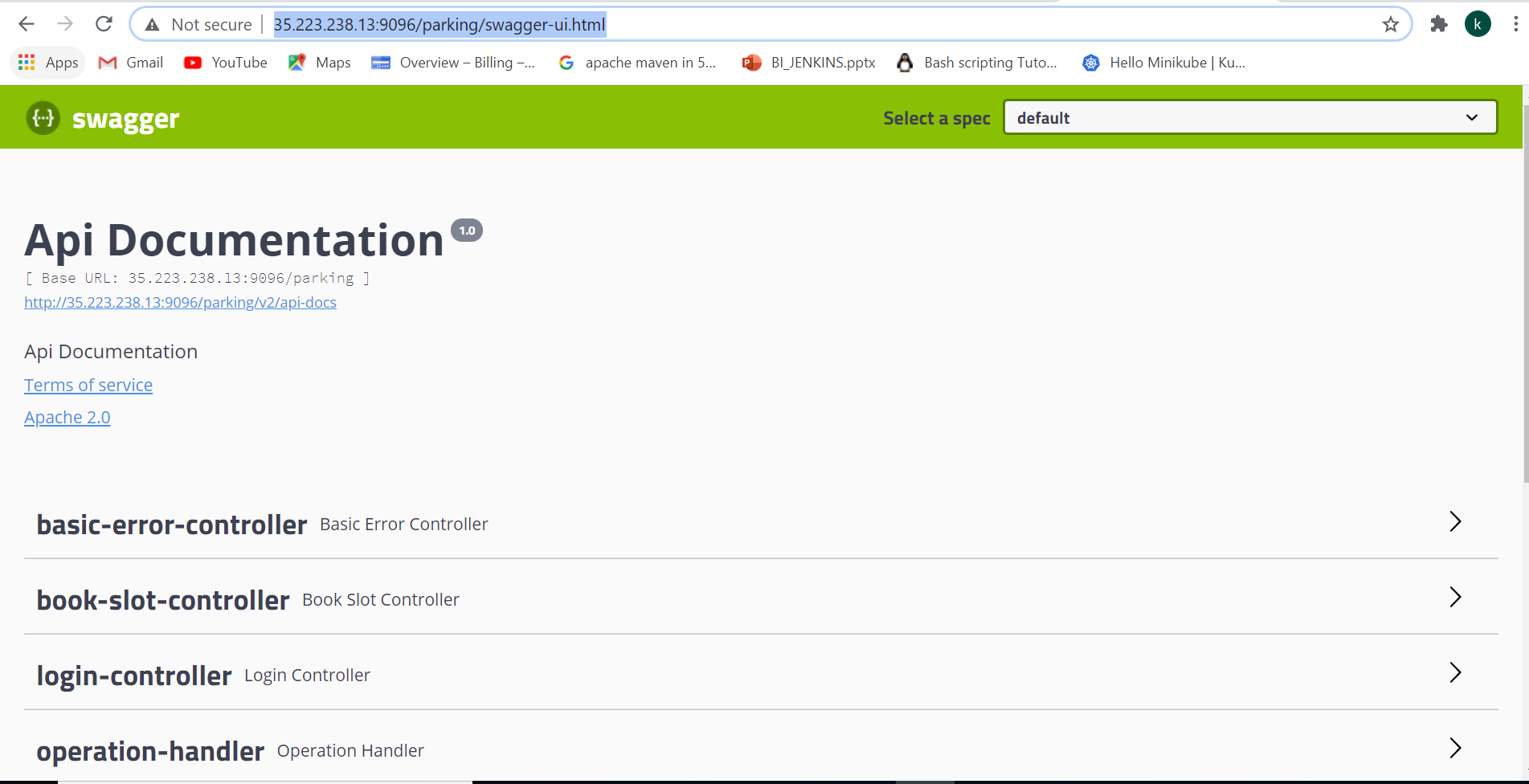


Then we can check the snapshot in nexus in the nexus-repositores.



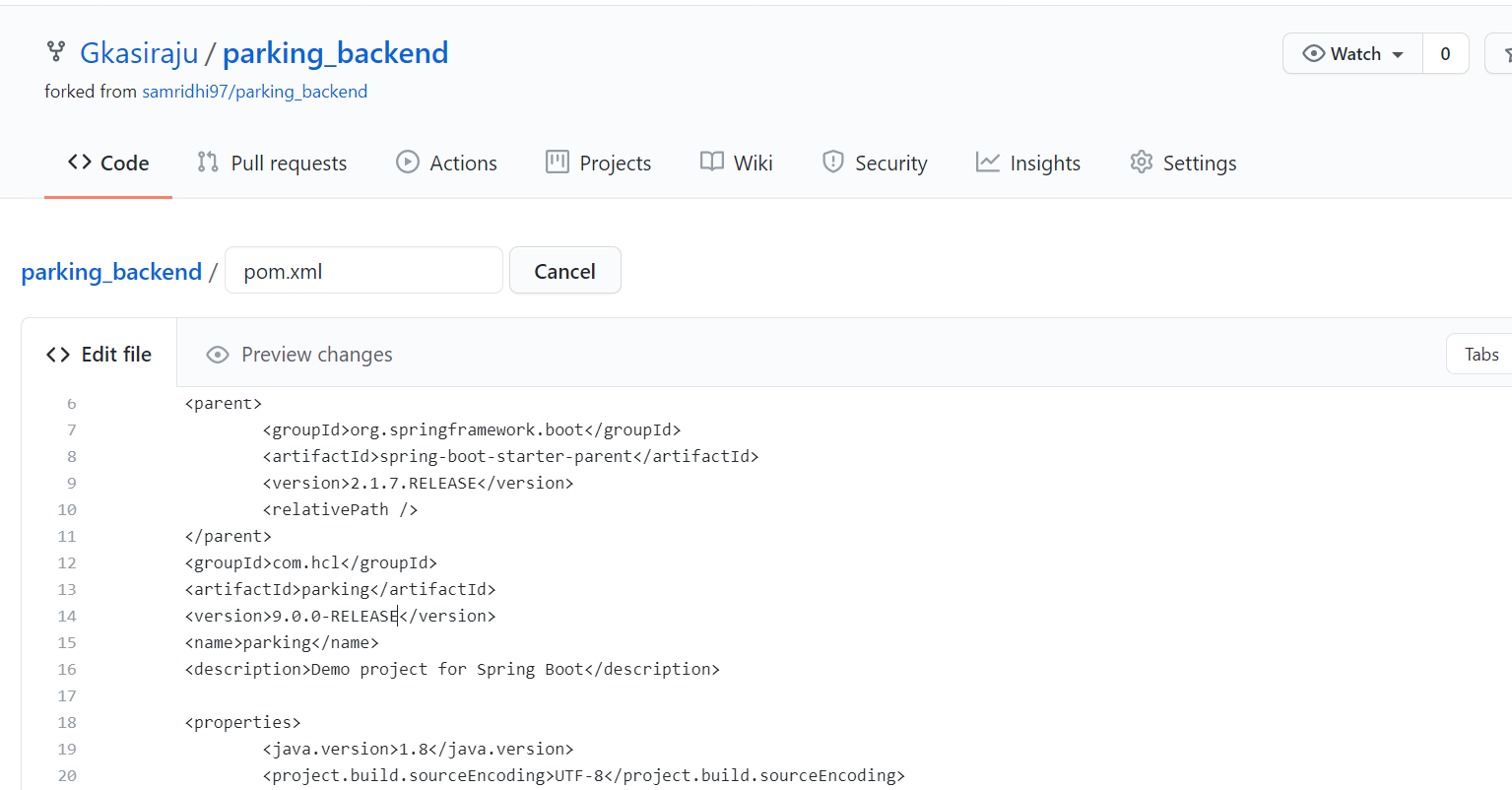
Then finally we can check in [**http://35.223.238.13:9096/parking/swagger-ui.html**](http://35.223.238.13:9096/parking/swagger-ui.html)

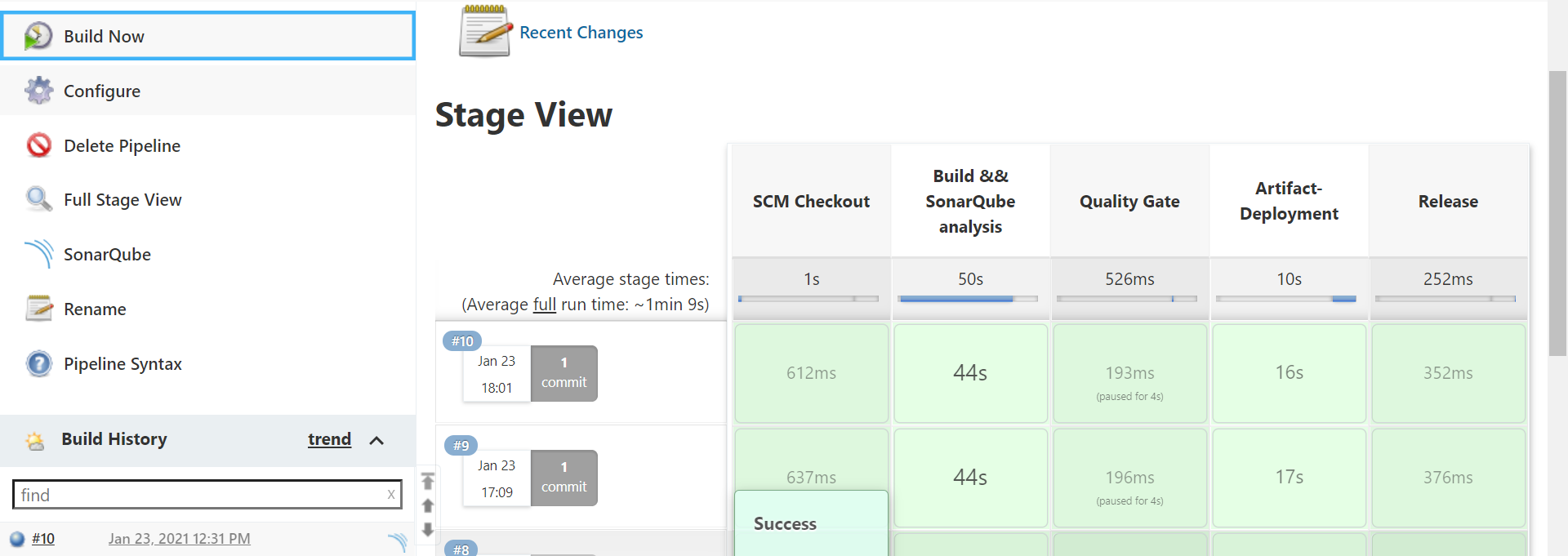
**here we can check it as shown below.**

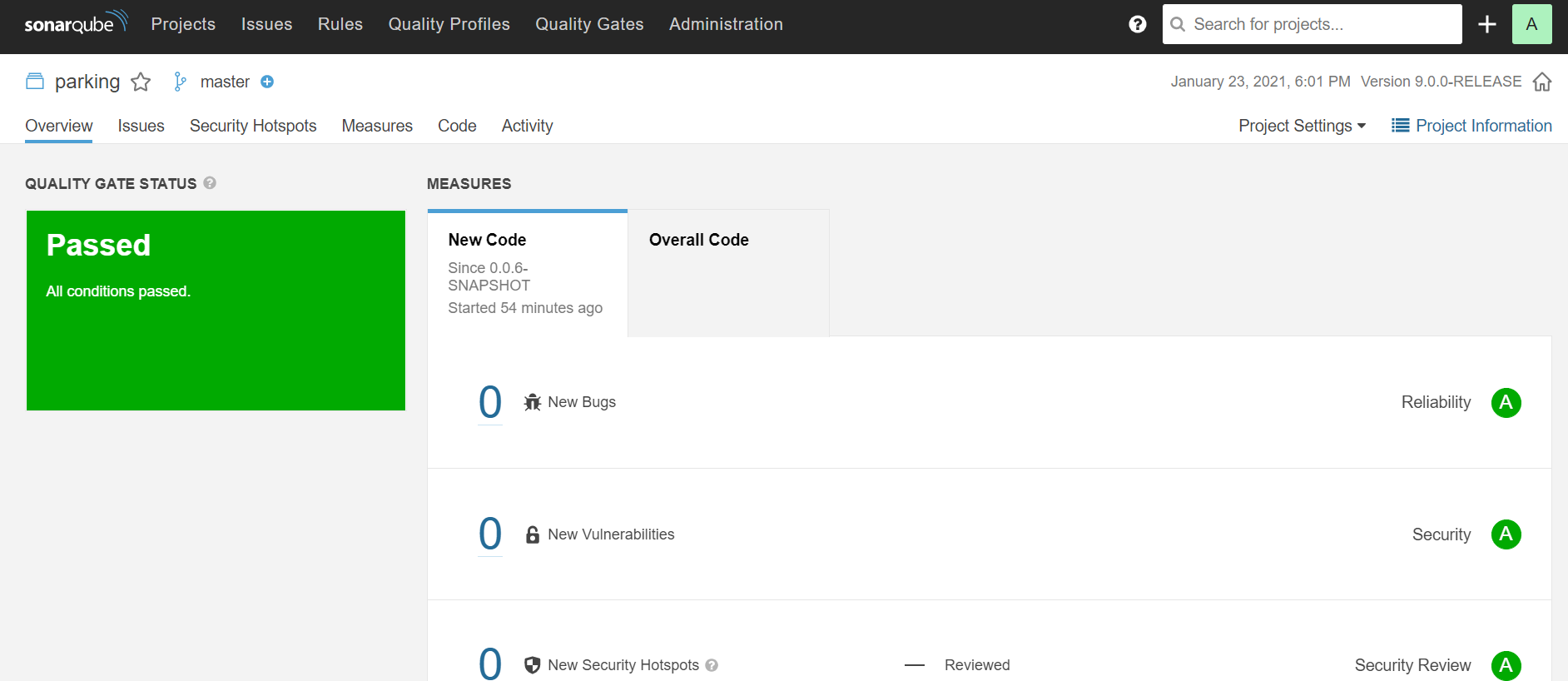


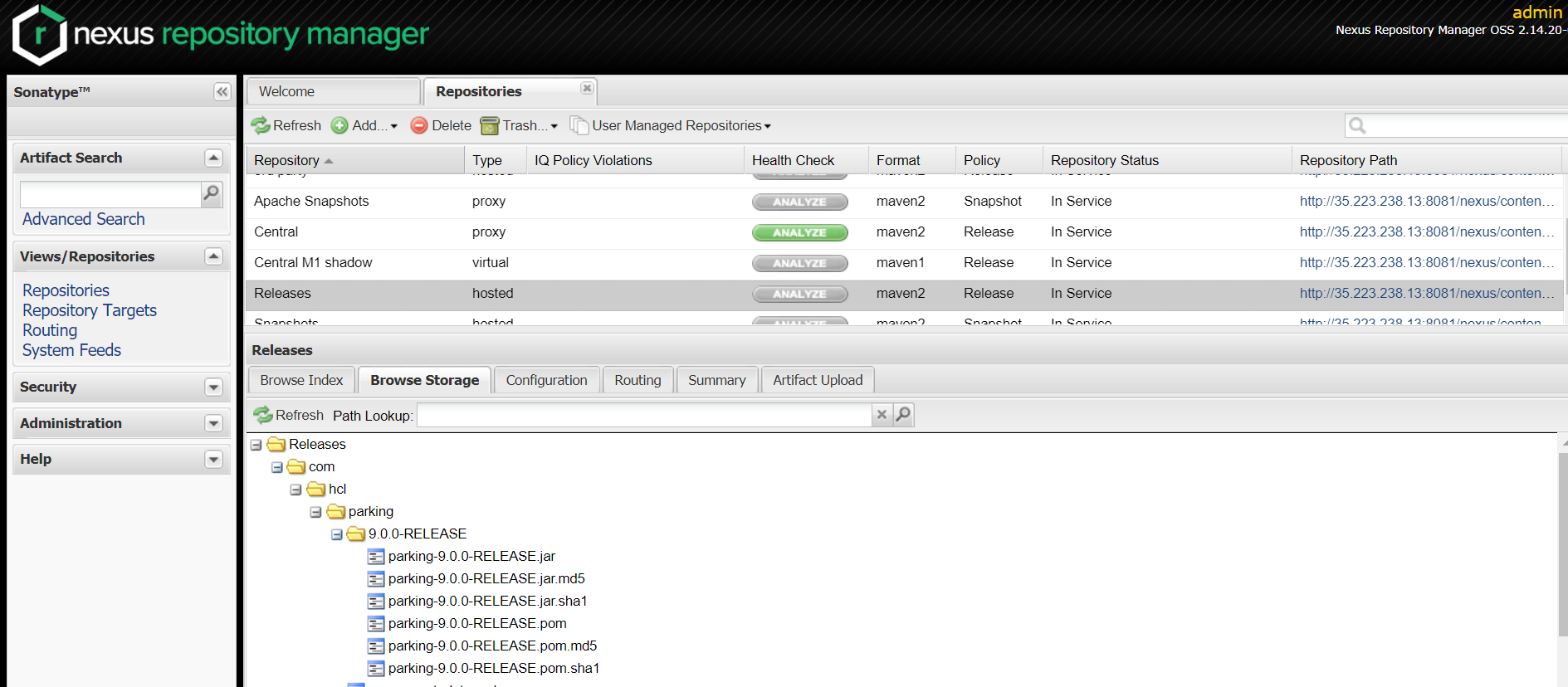
We can also release the artifacts in nexus.

We can change the github snapshot to release as shown below









|  |
| --- |
| <build> |
|  |
|  | <plugins> |
|  | <plugin> |
|  | <groupId>org.springframework.boot</groupId> |
|  | <artifactId>spring-boot-maven-plugin</artifactId> |
|  | </plugin> |
|  | </plugins> |
|  | </build> |