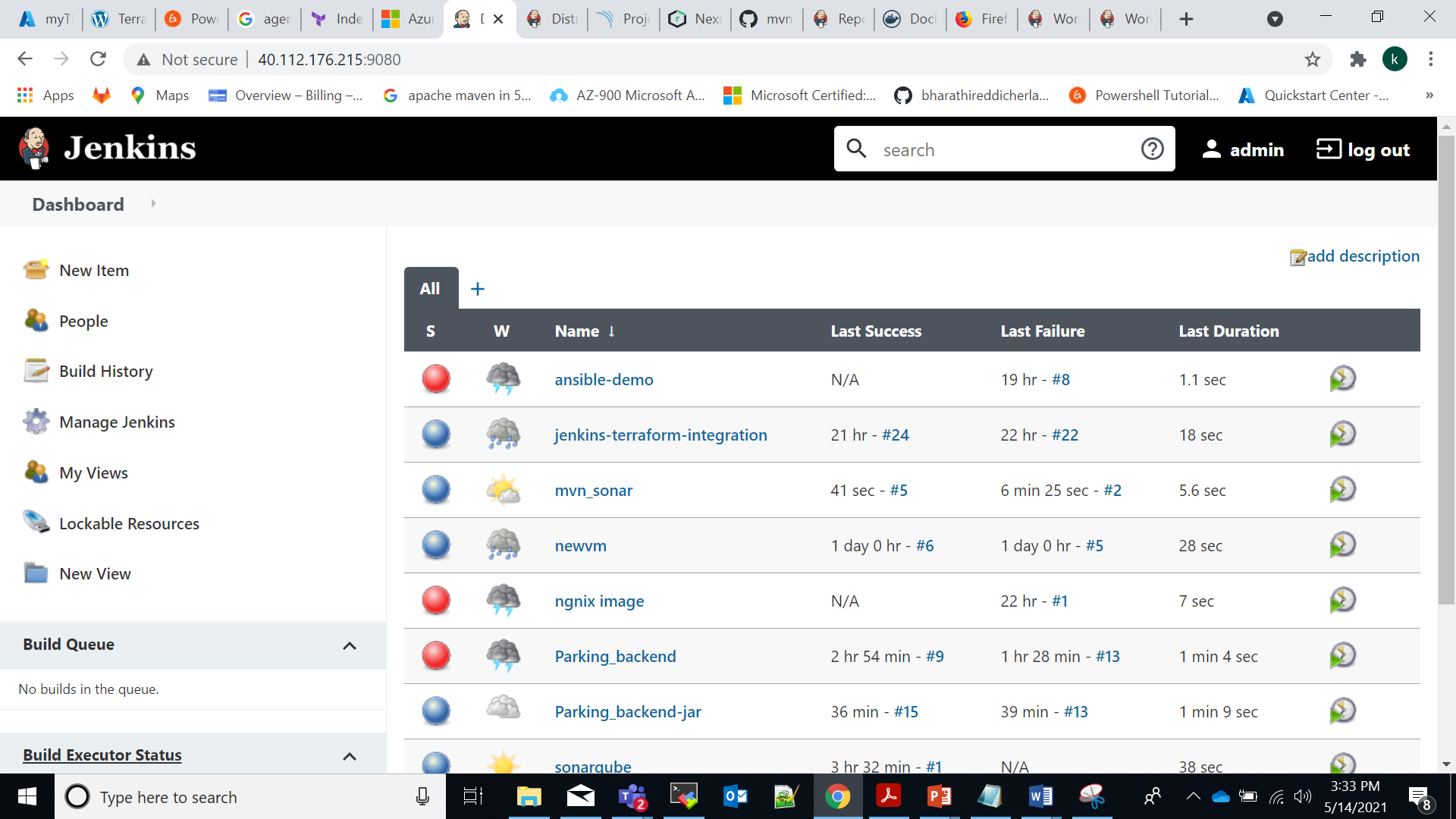
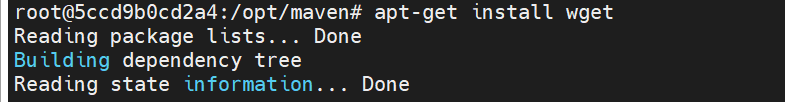
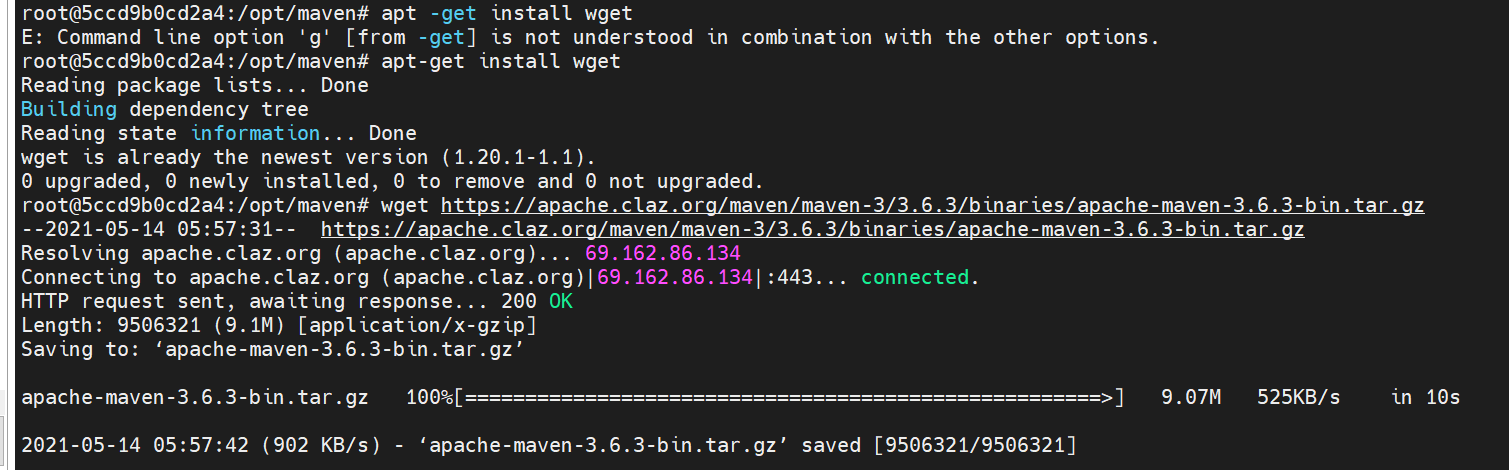
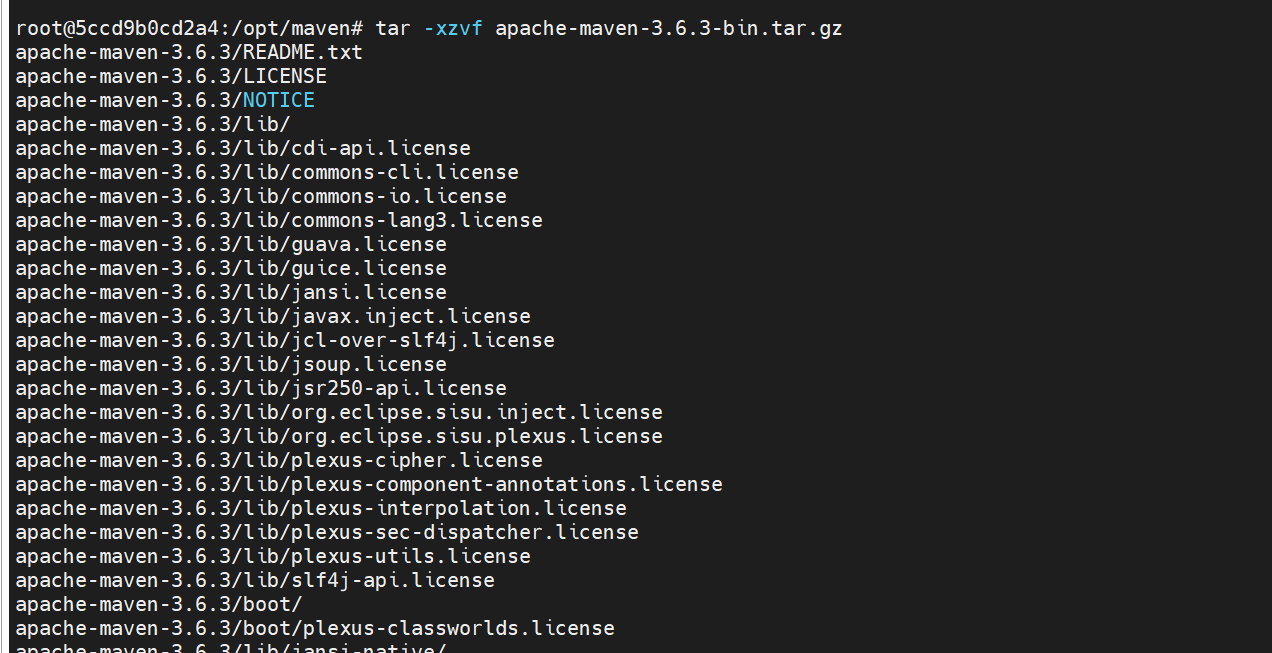
I am running the Jenkins as a docker container as shown below.

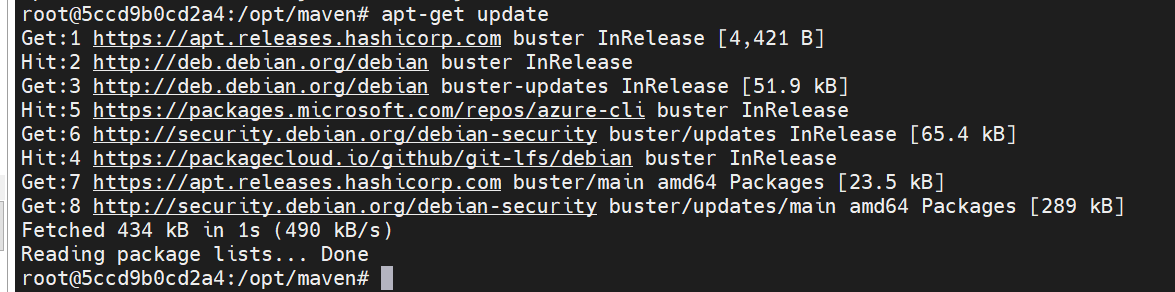


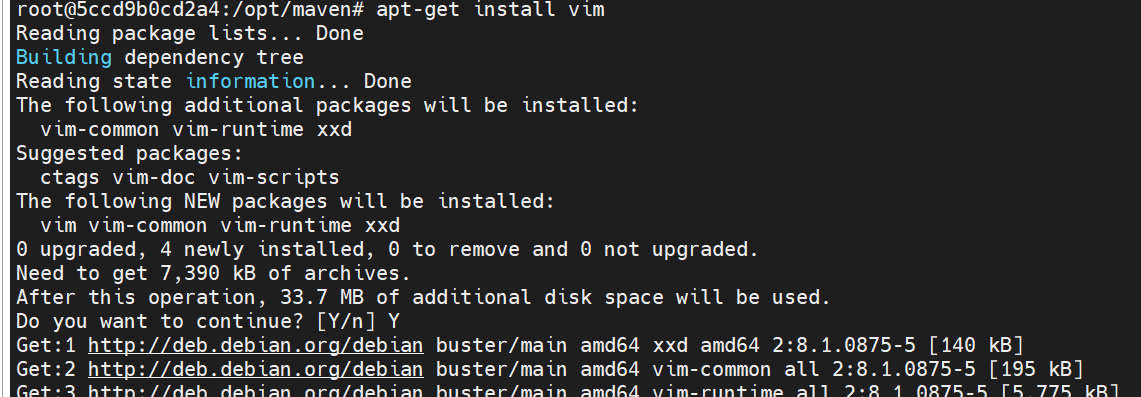
Maven install:

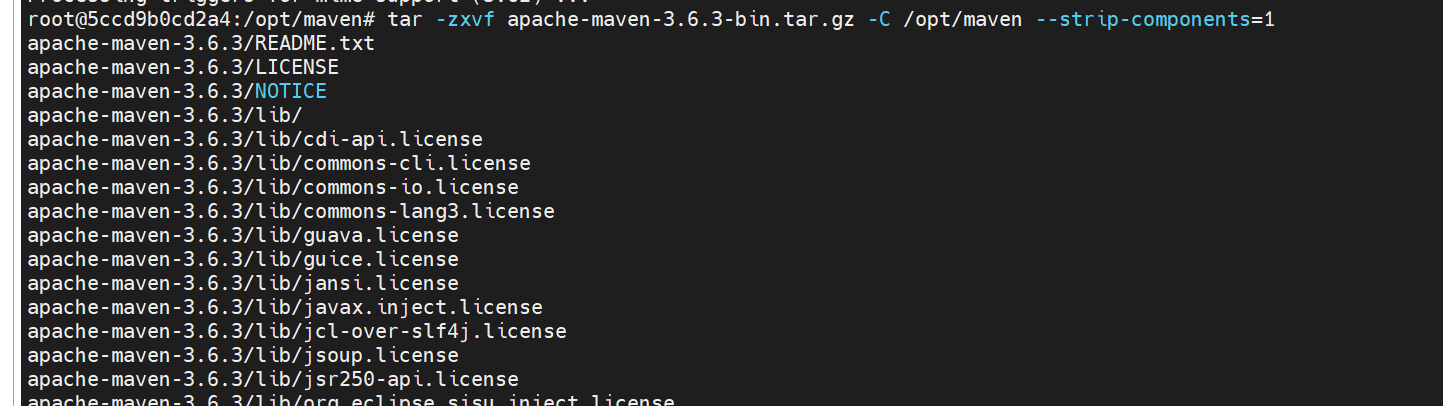


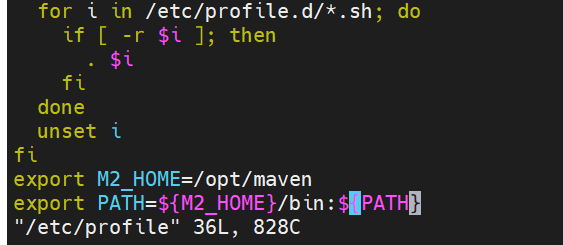


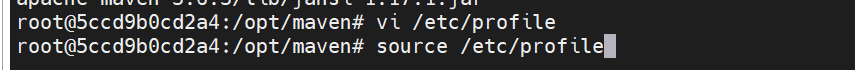


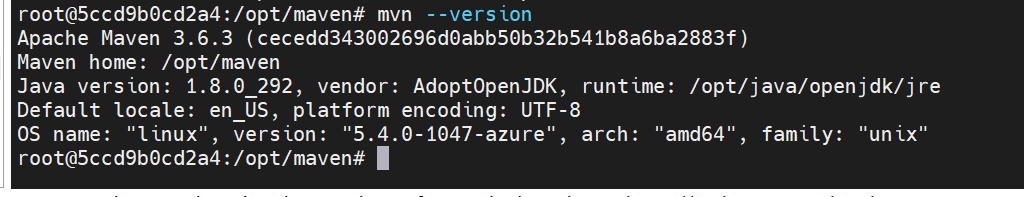




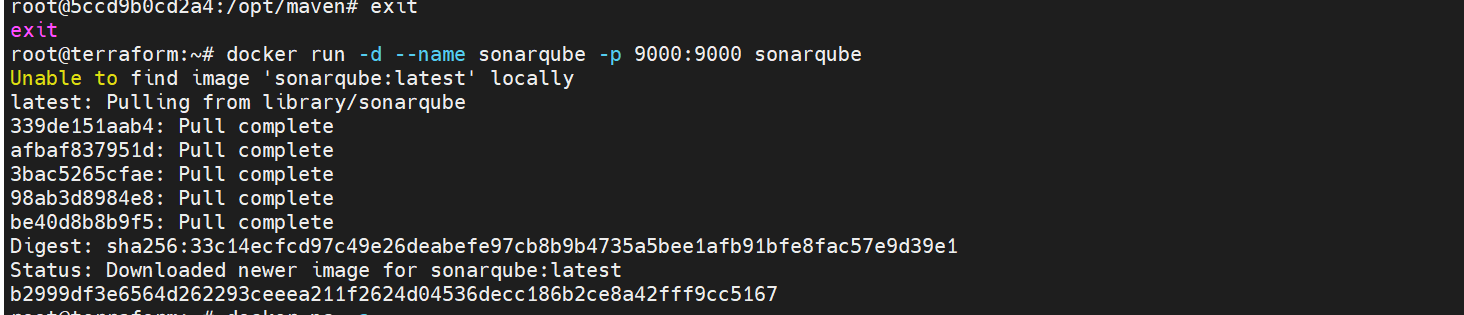


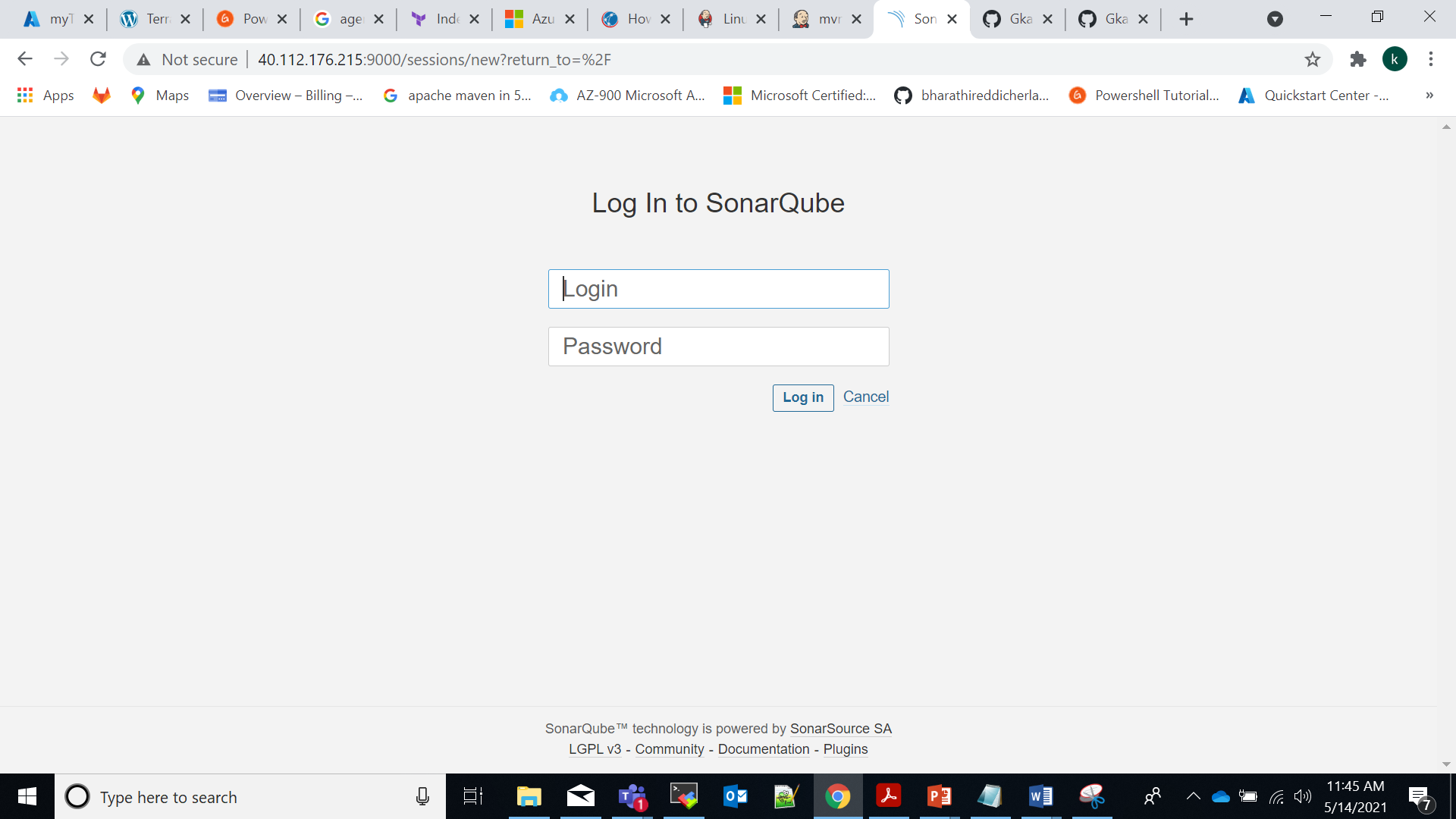




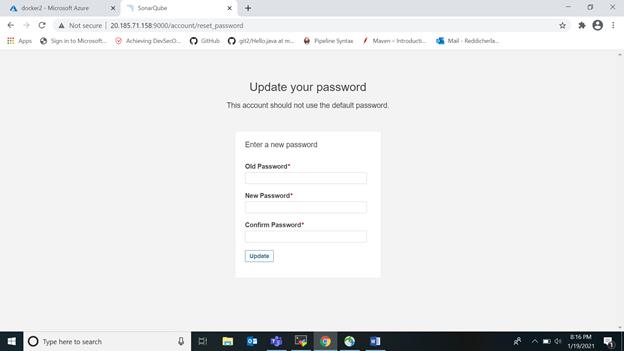


Sonarqube installation:

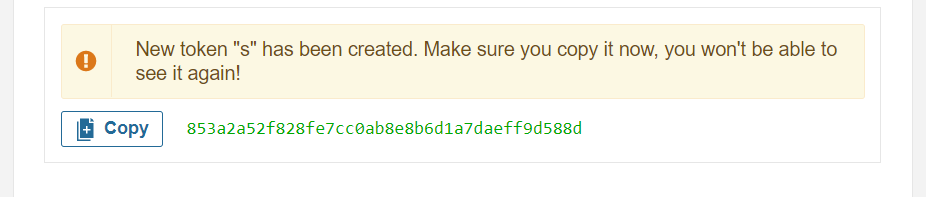


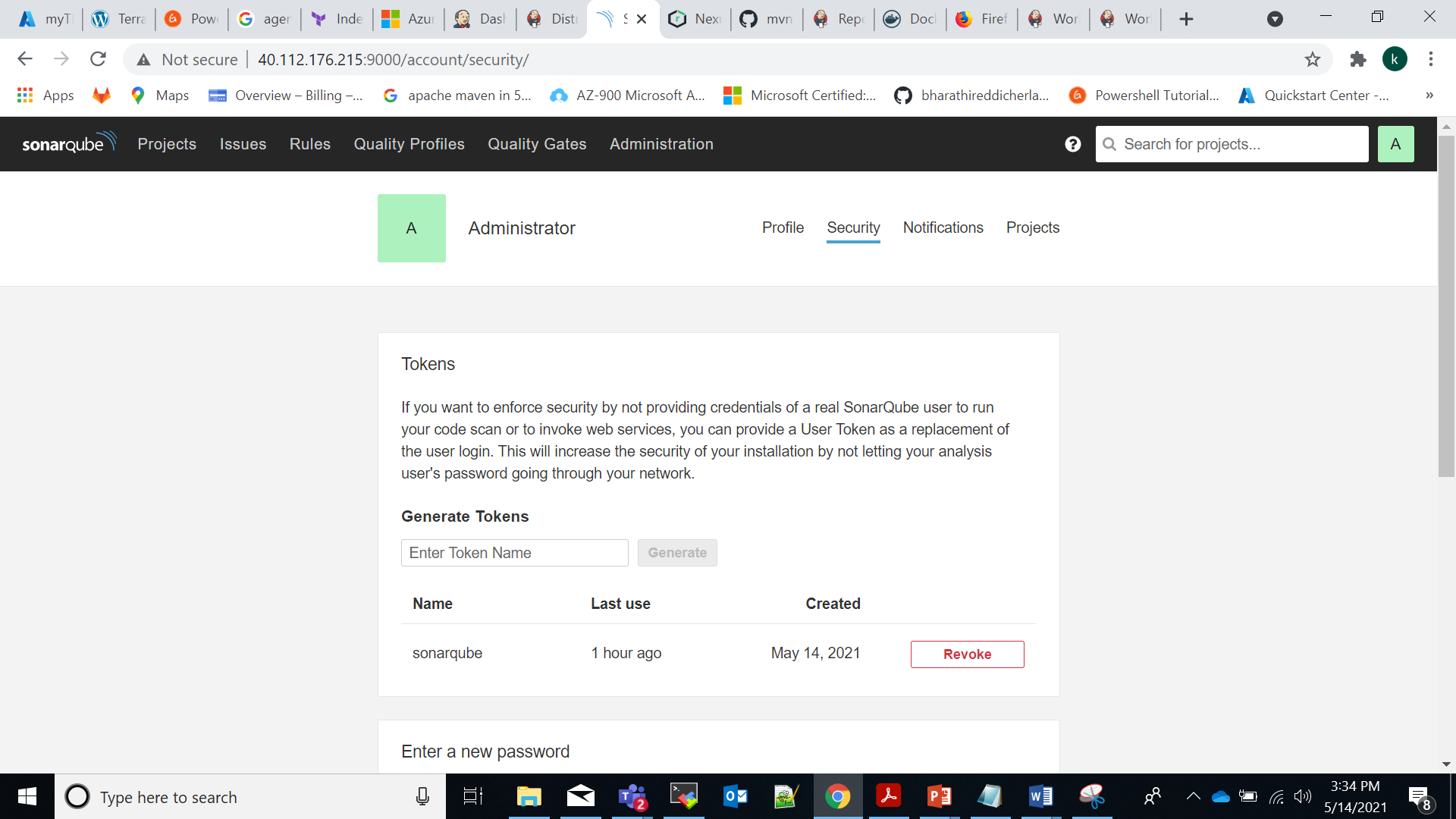


Set the new password admin to admin123

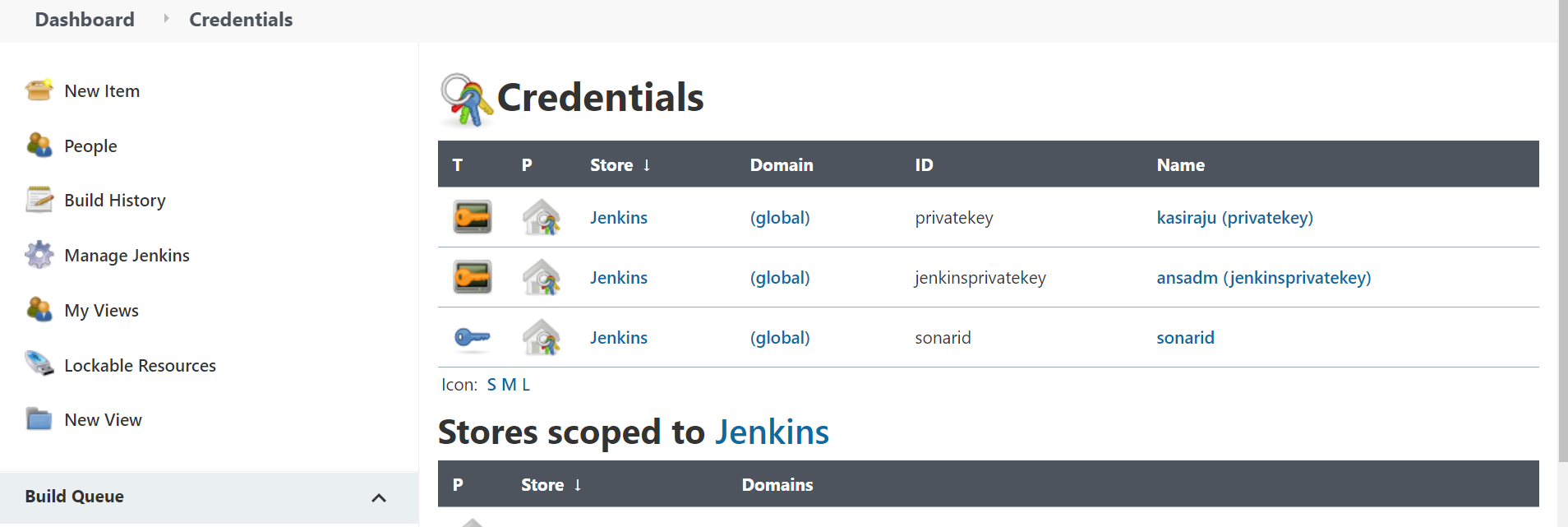
****

Now we need to set the sonarqube token in sonarqube scanner in that token copy the scret text





Now back to Jenkins goto manage Jenkins and open manage credential add the screet text where you copied the token in sonarqube



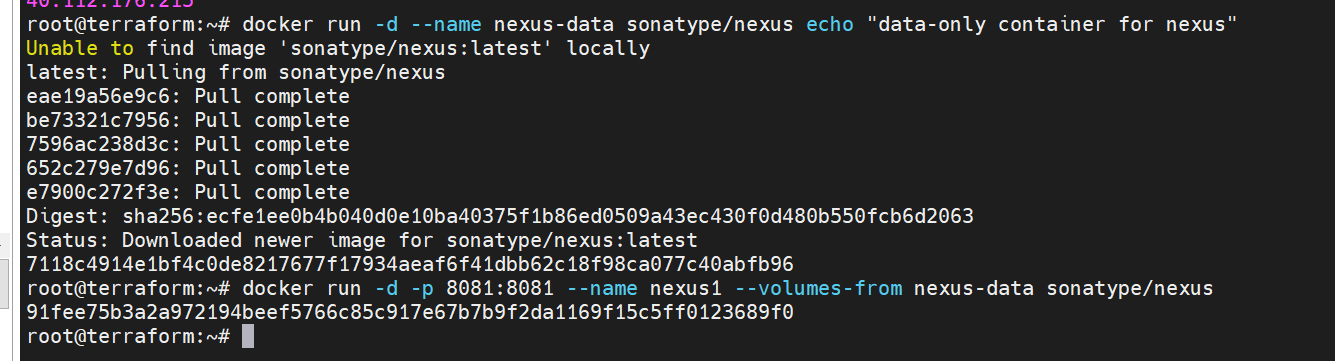
NEXUS installation: here I am pulling the nexus image from my Docker Hub

**Step:1 Create instance run a docker and create a volume as a name of nexus**

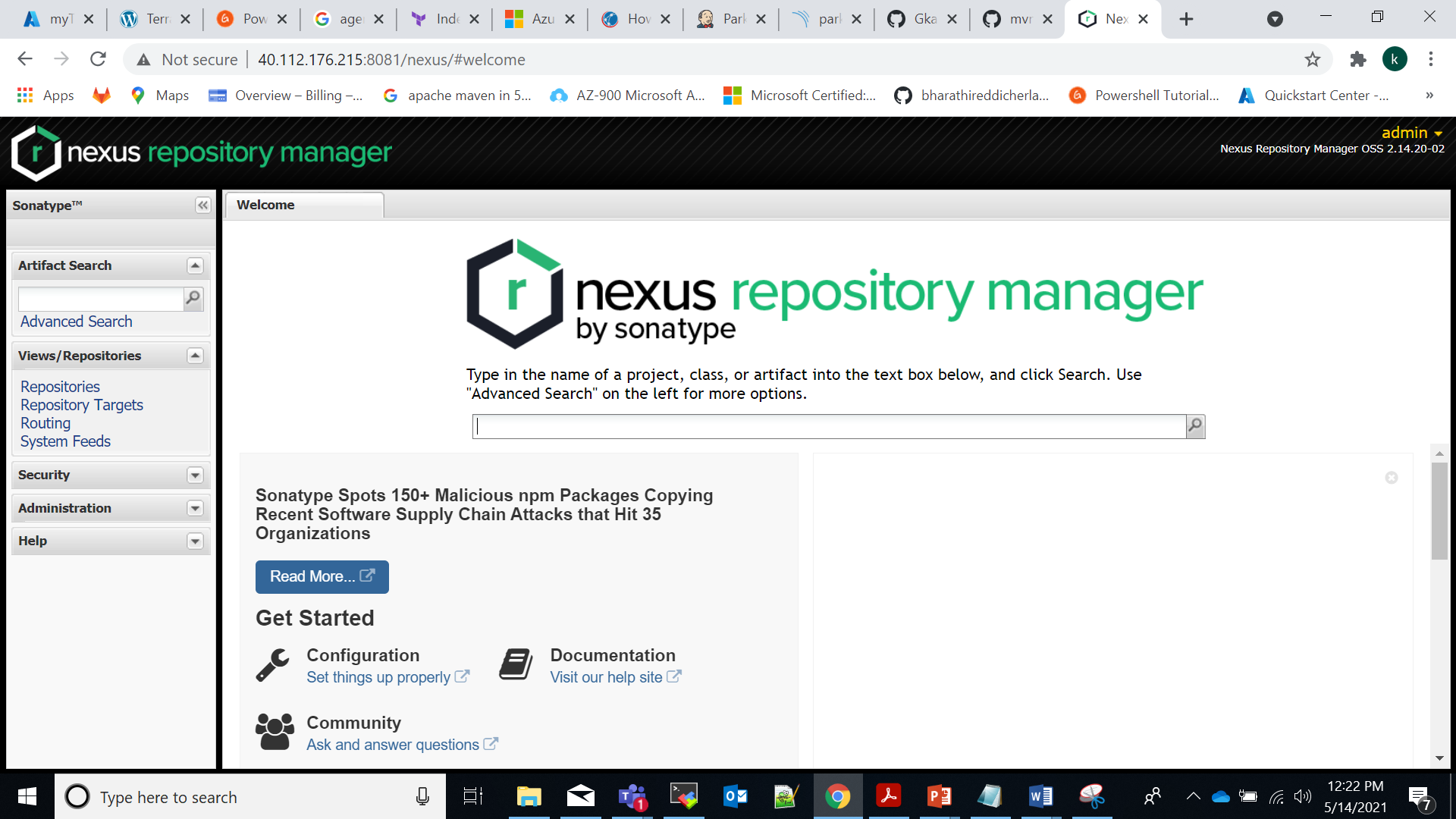
docker run -d --name nexus-data sonatype/nexus echo "data-only container for nexus"

Run a command which give port number 8081 and name of nexcus1:

docker run -d -p 8081:8081 --name nexus1 --volumes-from nexus-data sonatype/nexus

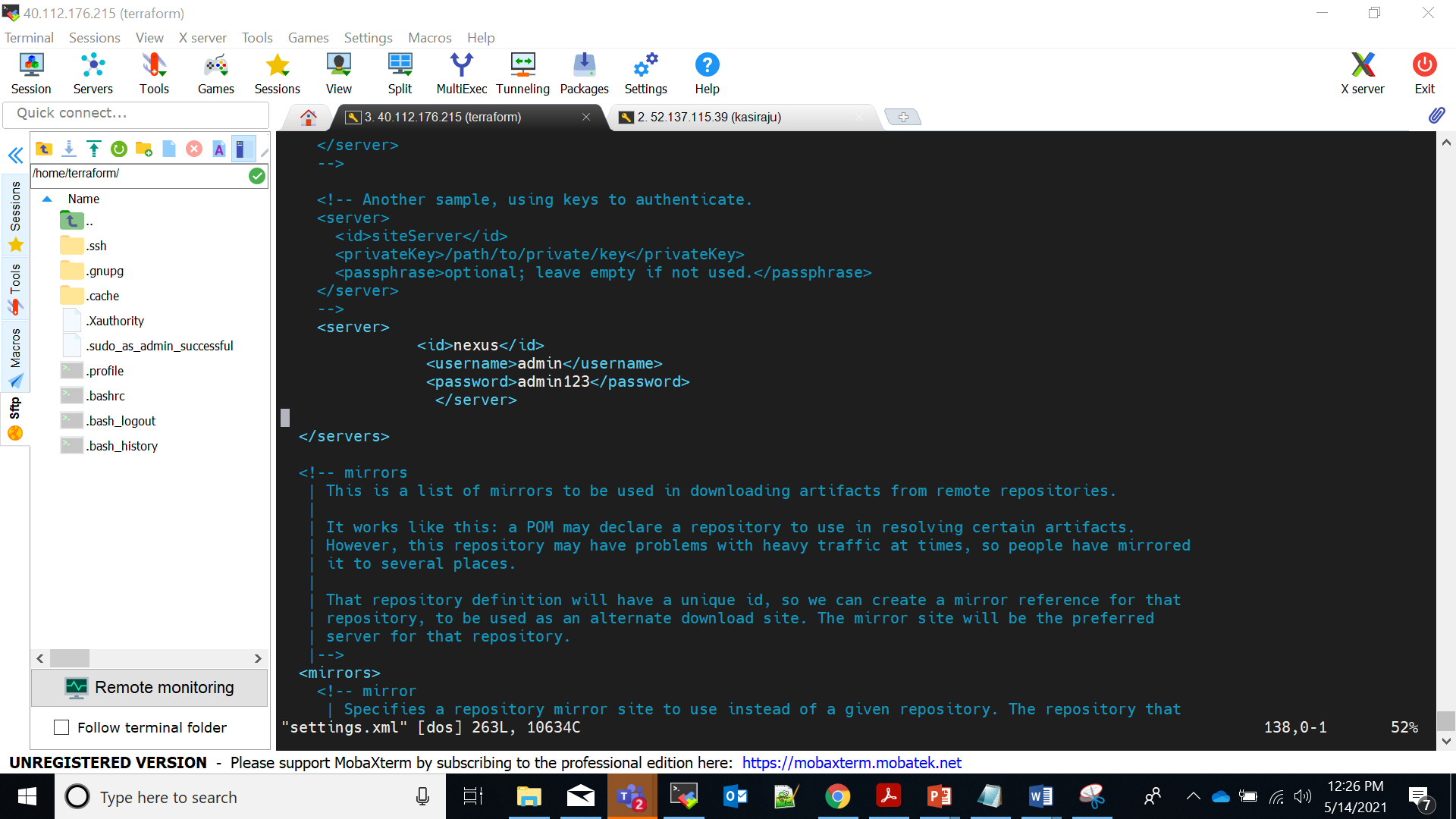


Then go to your browser and hit on http://40.112.176.215:8081/nexus **then you redirect to below page.**



**Open mobaXterm then go to maven config folder**

**Cd /opt/maven/conf/ vi settings.xml**



**In the servers tage write the server details of nexus shown below**

**<server>**

**<id>nexus</id>**

**<username>admin</username>**

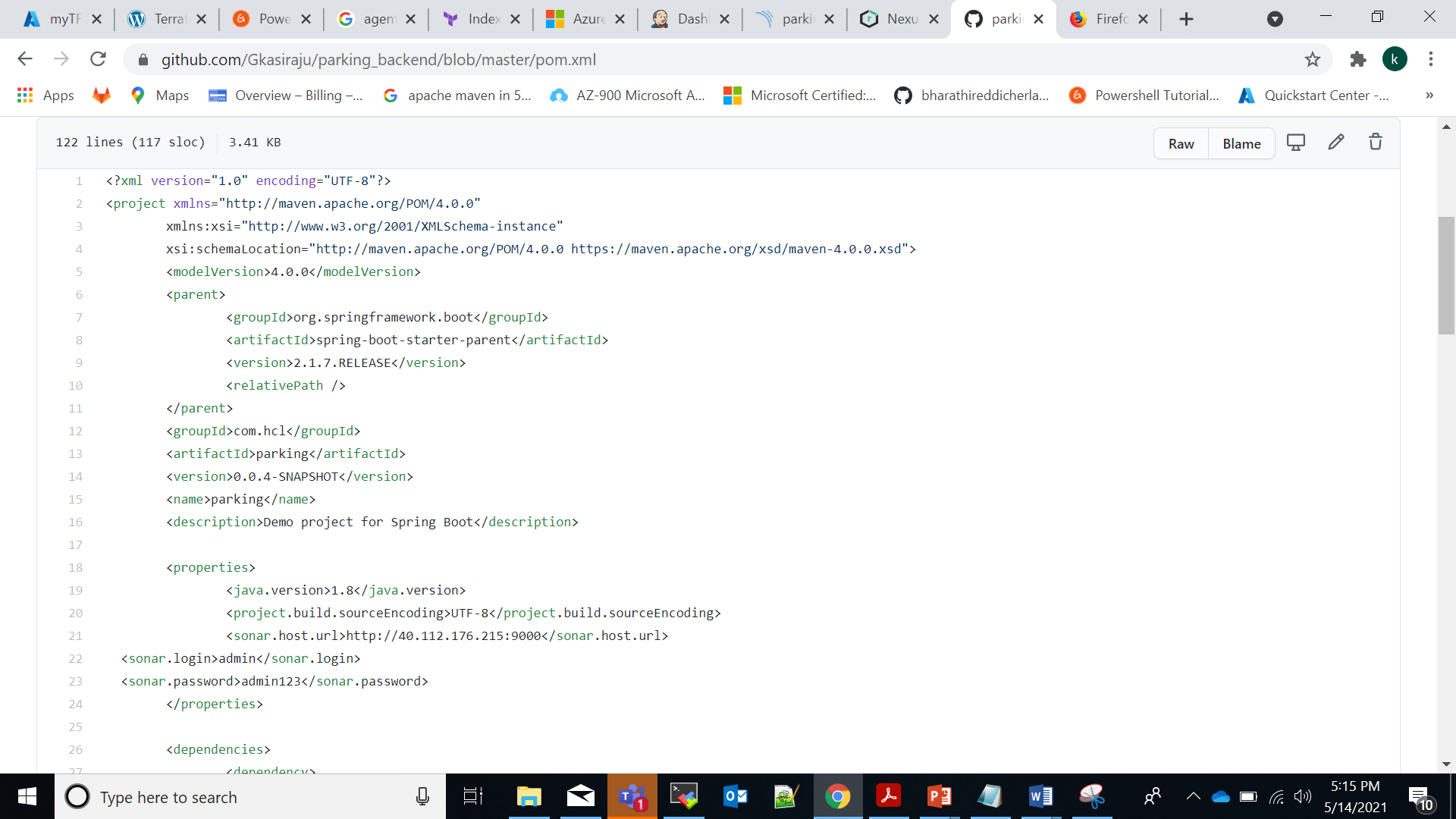
**<password>admin123</password>**

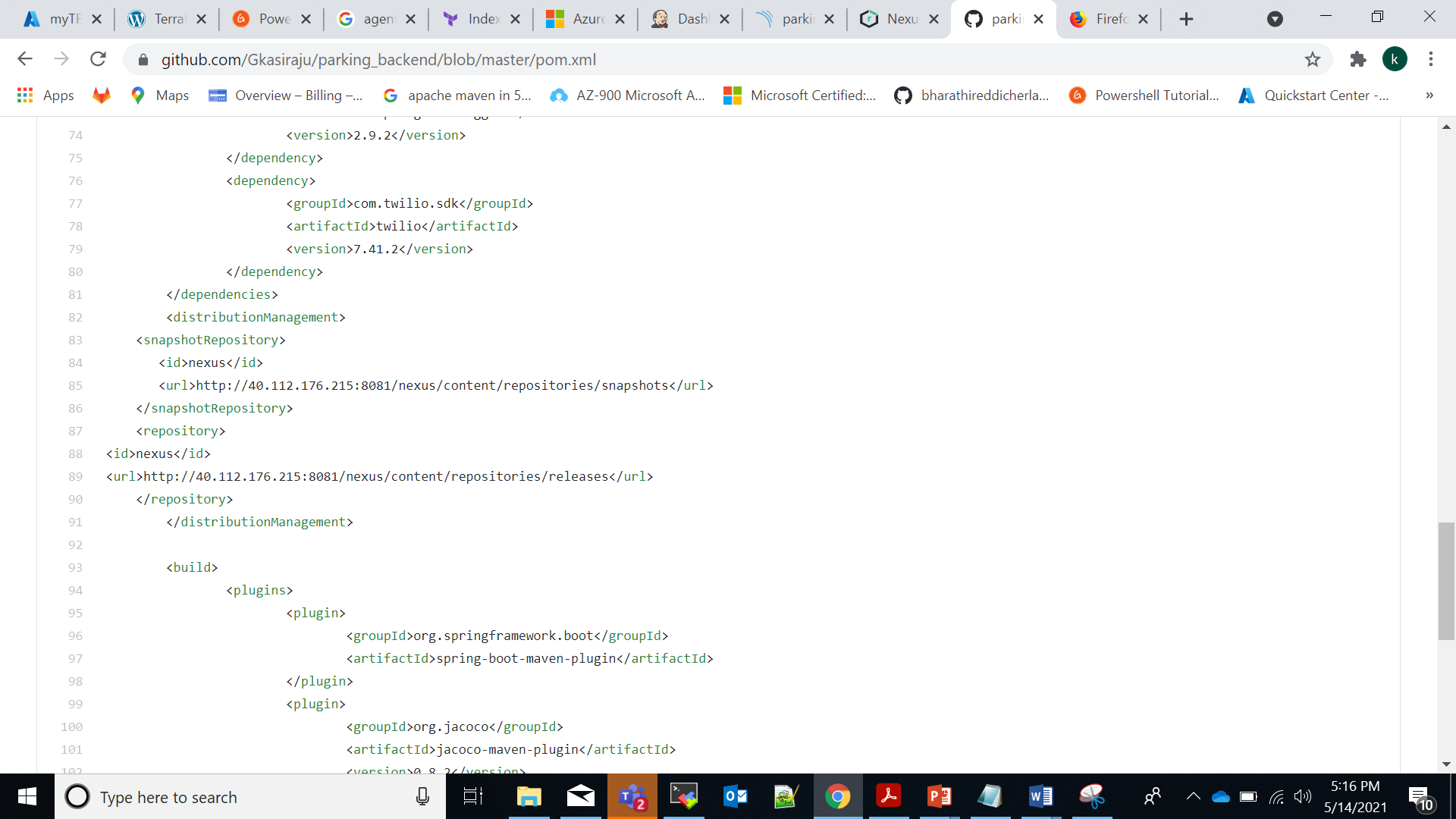
**</server>**

**</servers>**

Now I am creating pipeline script which is clone the source from the github.com and it goto automatic and code quality analysis and quality gate status and deploying the artifact in repository is called nexus

Open the pom.xml and add the required sonarqube credential as shown below





pipeline {

agent any

stages {

stage('SCM Checkout') {

steps {

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

}

}

stage ('Publish Unit Test Report')

{ steps {

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

junit allowEmptyResults: true, testResults: '\*\*/target/surefire-reports/\*.xml'

}

}

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'

}

}

}

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 &'

}

}

stage('jar download'){

steps{

sh 'wget "http://40.112.176.215:8081/nexus/service/local/artifact/maven/content?r=snapshots&g=com.hcl&a=parking&v=LATEST&p=jar" --content-disposition'

// sh 'wget "http://40.112.176.215:8081/service/local/artifact/maven/redirect?g=com.hcl&a=parking&v=LATEST"'

//--content-disposition

sh 'wget "http://40.112.176.215:8081/nexus/service/local/artifact/maven/content?r=releases&g=com.hcl&a=parking&v=LATEST&p=jar" --content-disposition'

}

}

}

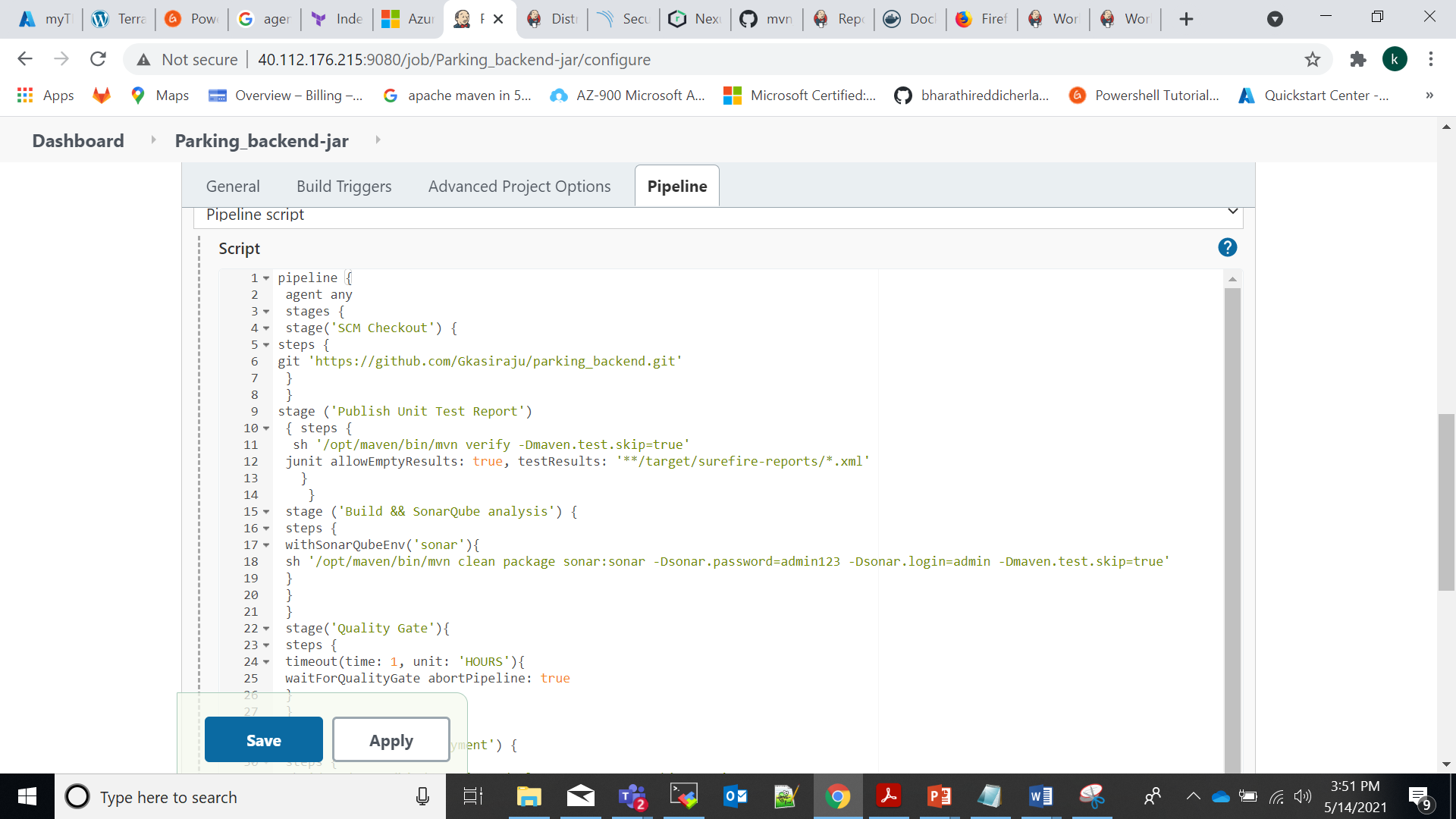
}

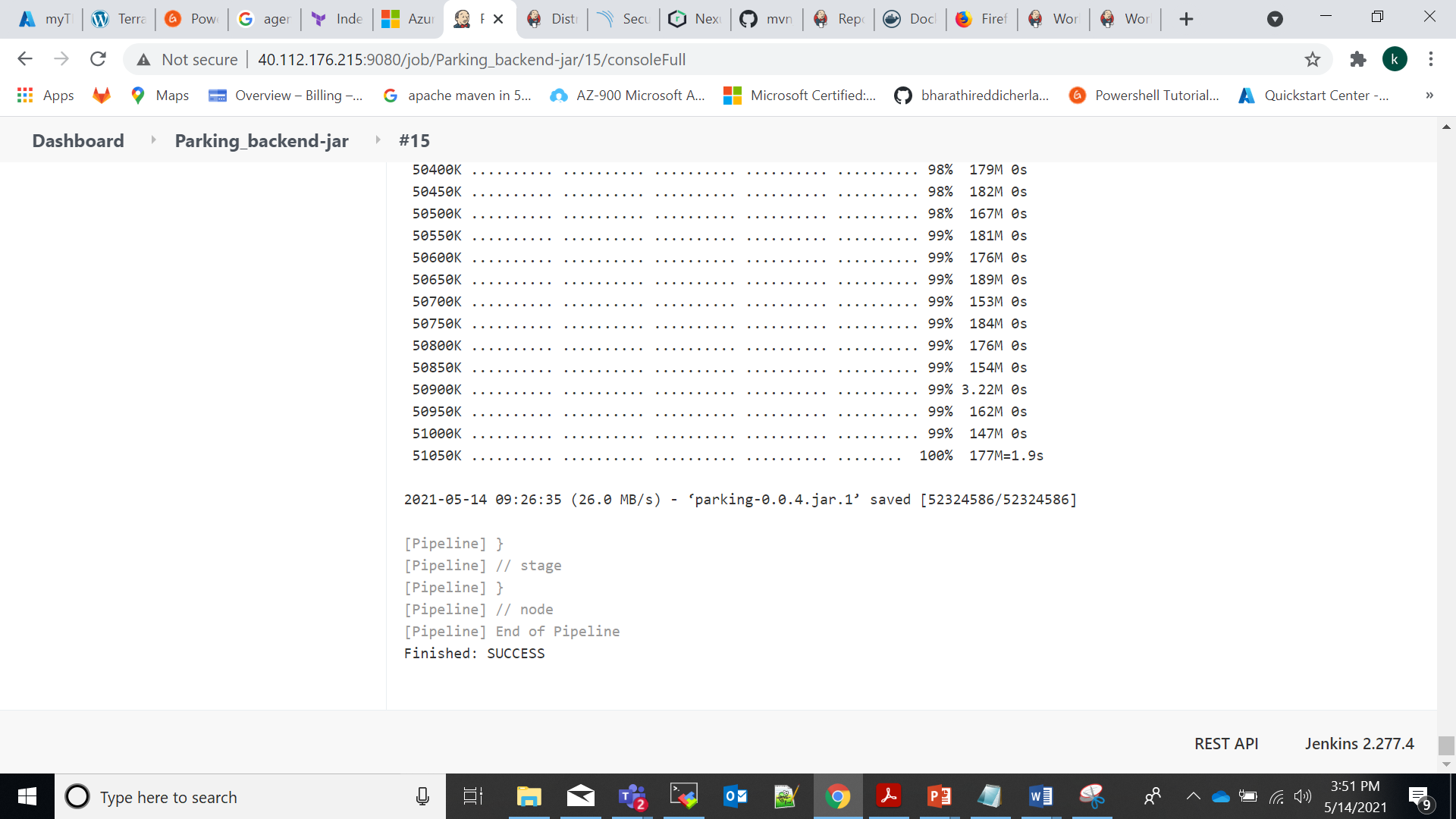
Add sonarqube server

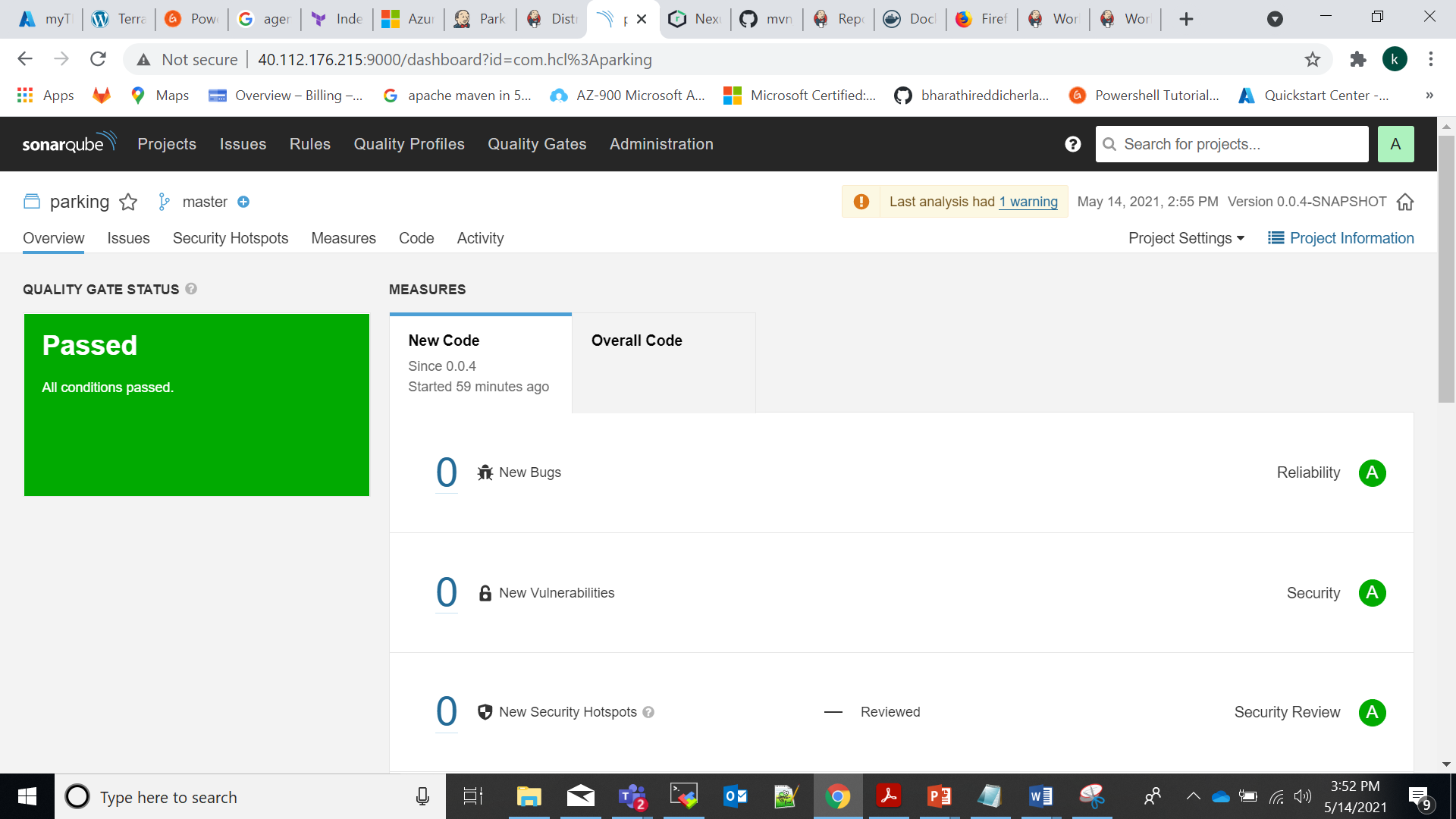
|  |
| --- |
| <properties> |
|  | <java.version>1.8</java.version> |
|  | <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> |
|  | <sonar.host.url>http://40.112.176.215:9000</sonar.host.url> |
|  | <sonar.login>admin</sonar.login> |
|  | <sonar.password>admin123</sonar.password> |
|  | </properties> |

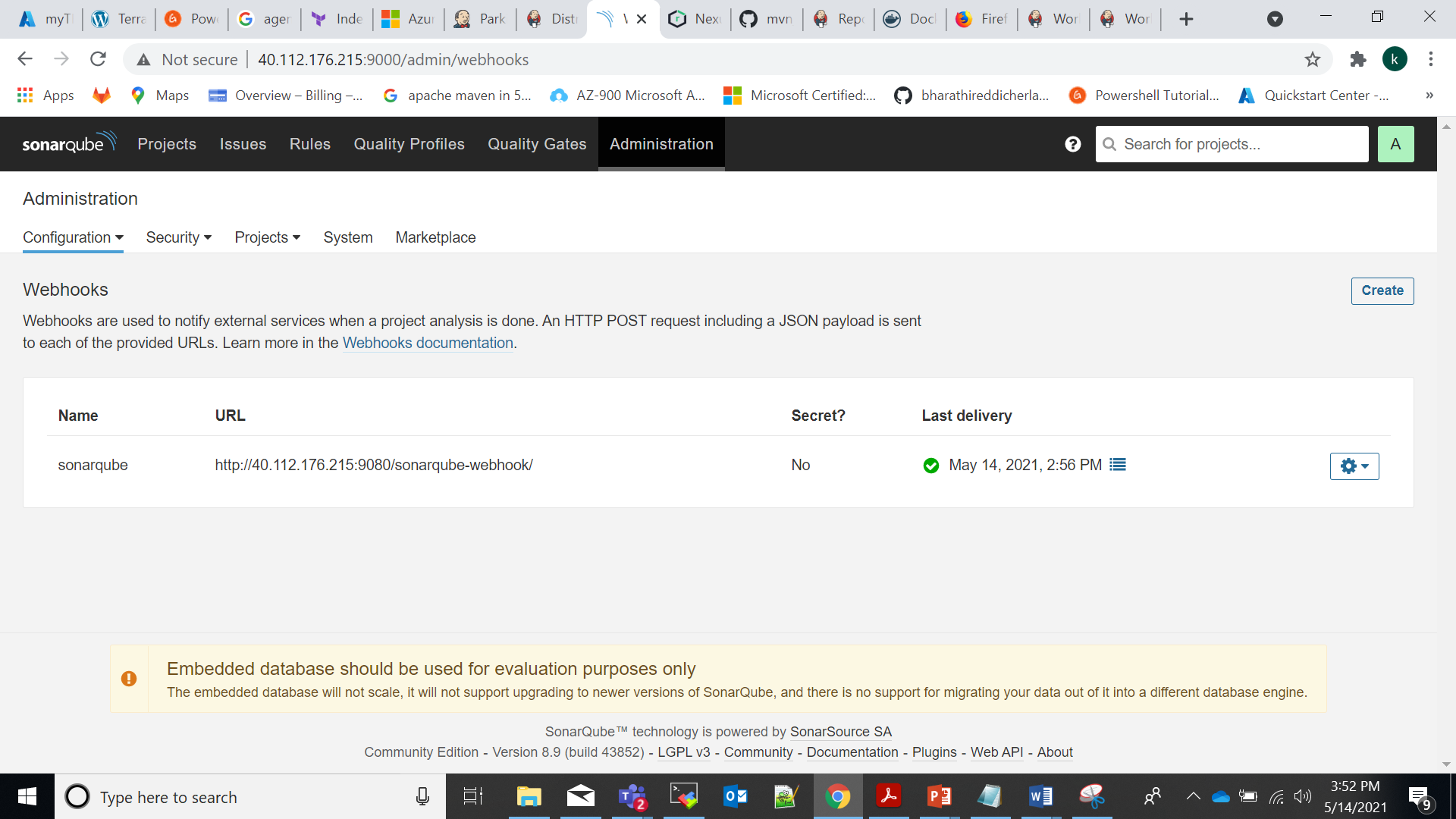
In pom.xml add this one

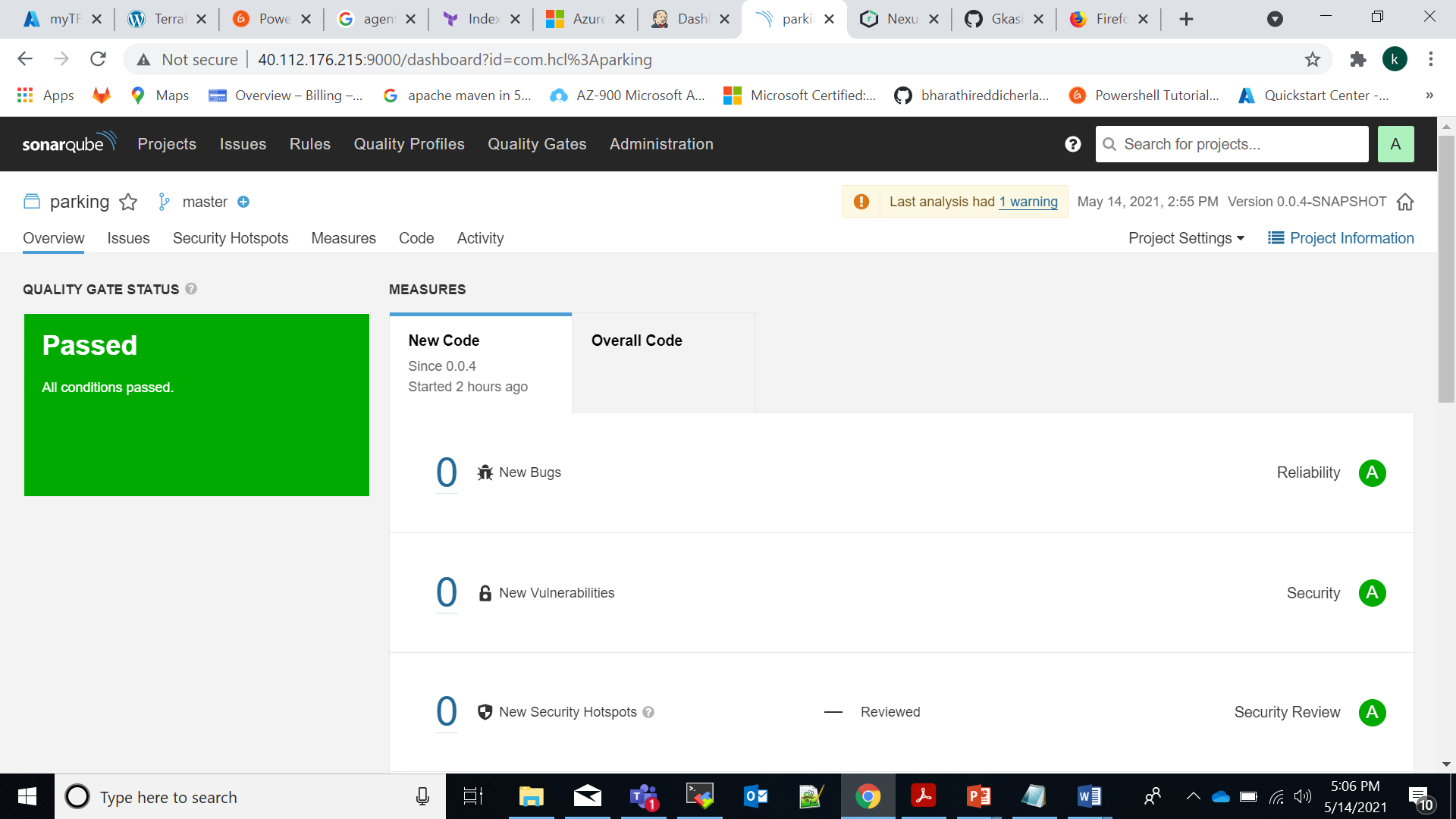
|  |
| --- |
| <distributionManagement> |
|  | <snapshotRepository> |
|  | <id>nexus</id> |
|  | <url>http://40.112.176.215:8081/nexus/content/repositories/snapshots</url> |
|  | </snapshotRepository> |
|  | <repository> |
|  | <id>nexus</id> |
|  | <url>http://40.112.176.215:8081/nexus/content/repositories/releases</url> |
|  | </repository> |
|  | </distributionManagement> |

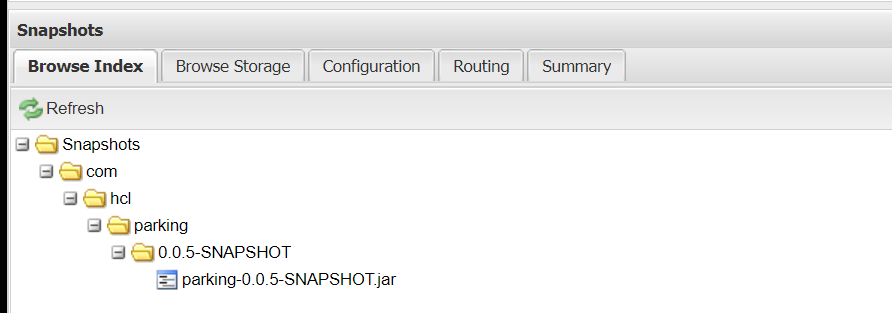




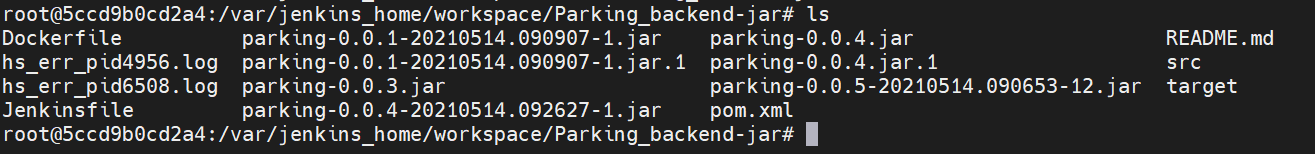








This is shows the when you successfully build snapshot is release then I write extra snip that is downloaded in to my workspace if we rebuild it will take lastest version of snapshot



Now its time to release do some changes in pom.xml changes the version snapshot to release

