Jenkins Pipeline in Openshift

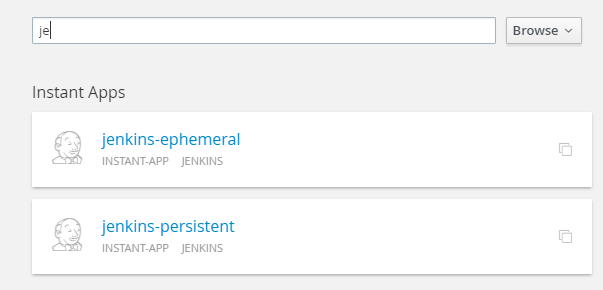
Prerequisites:

1. Openshift Environment(on local, aws ,azure, etc..) version 3 or above
2. Access to openshift console.

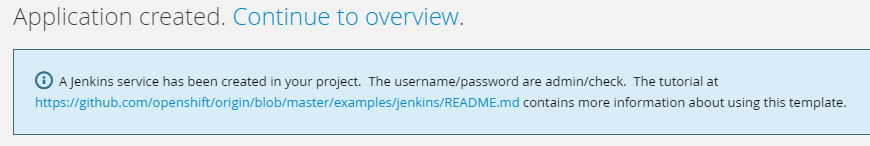
Step 1: Login to the openshift via console.

Step 2: Start Jenkins inside openshift( openshift has Jenkins application within it).

1. Select Jenkins ephemeral and give a password of your choice and click create.

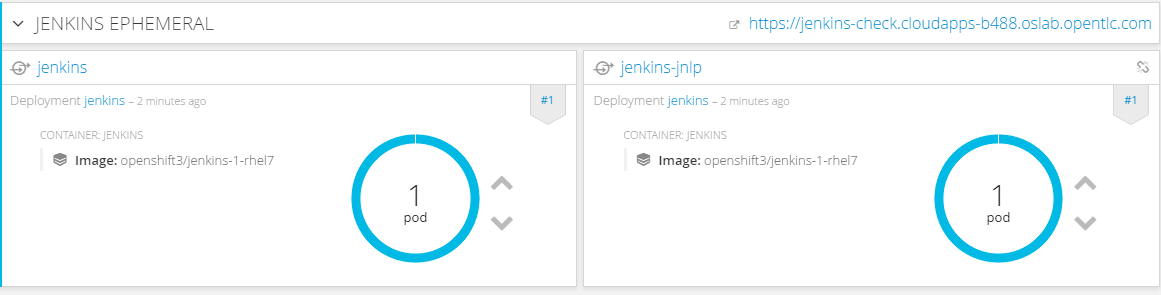
 

Click Create



Continue to Overview

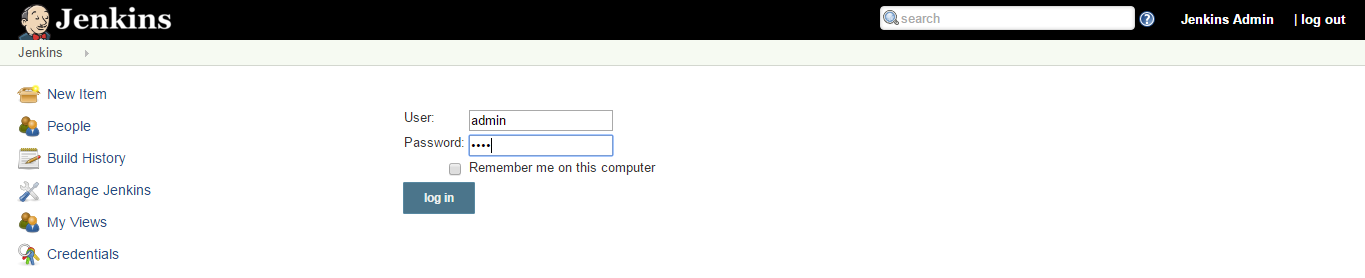
1. You can see Jenkins pod building. Wait for a few minutes for the pod to complete.



1. Use the route(http link – top right corner) to access the jenkins dashboard).

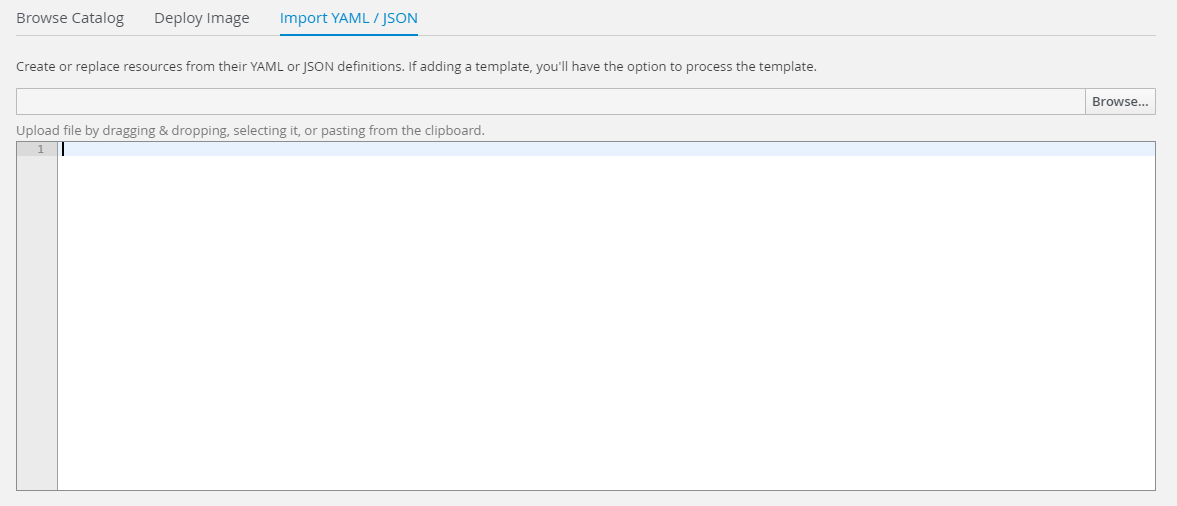
Username: admin

Password: <your apssword>

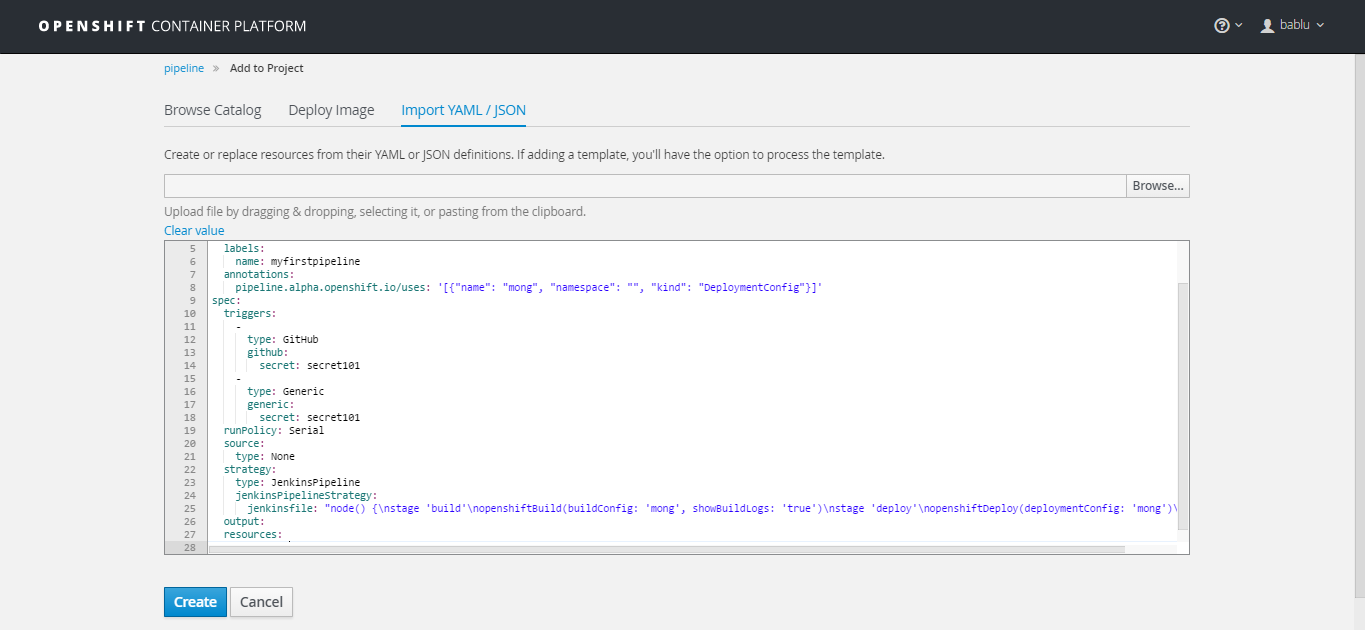


Jenkins by default is linked to Openshift, so we can deal with pipeline via Jenkins and also via Openshift.

1. Create a new pipeline in Openshift.
2. In Openshift console, Go to “Add to Project” section and select “Import JSON/YAML”.



and enter the pipeline yaml text as shown below and click Create.

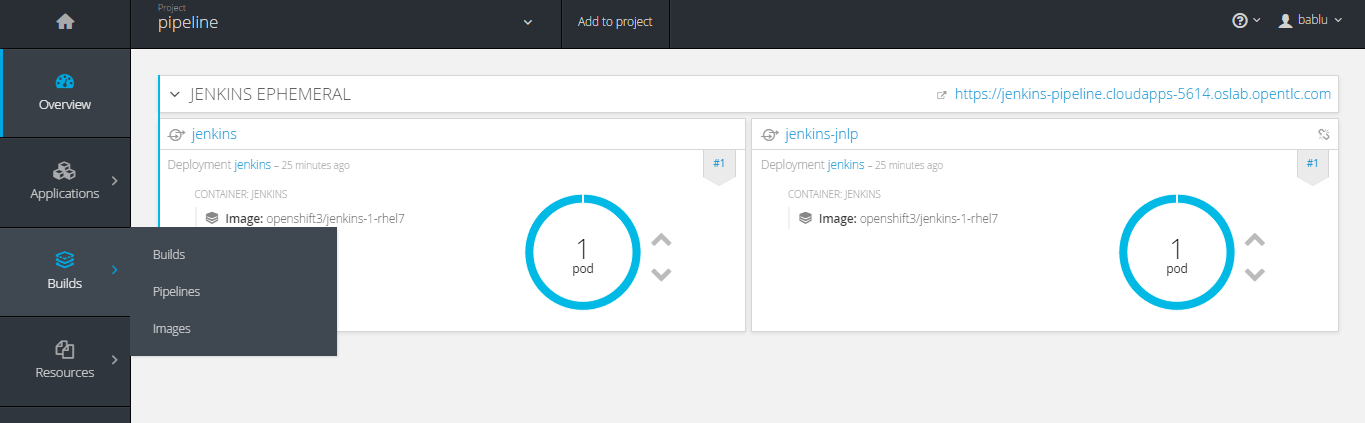


Pipeline yaml

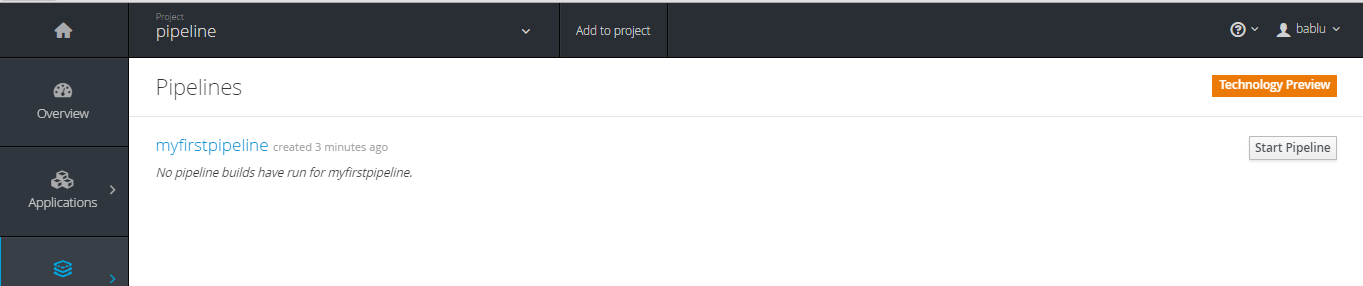
|  |
| --- |
| apiVersion: v1  kind: BuildConfig  metadata:  name: myfirstpipeline  labels:  name: myfirstpipeline  annotations:  pipeline.alpha.openshift.io/uses: '[{"name": "mong", "namespace": "", "kind": "DeploymentConfig"}]'  spec:  triggers:  -  type: GitHub  github:  secret: secret101  -  type: Generic  generic:  secret: secret101  runPolicy: Serial  source:  type: None  strategy:  type: JenkinsPipeline  jenkinsPipelineStrategy:  jenkinsfile: "node() {\nstage 'build'\nopenshiftBuild(buildConfig: 'mong', showBuildLogs: 'true')\nstage 'deploy'\nopenshiftDeploy(deploymentConfig: 'mong')\nopenshiftScale(deploymentConfig: 'mong',replicaCount: '2')\n}"  output:  resources:  postCommit: |



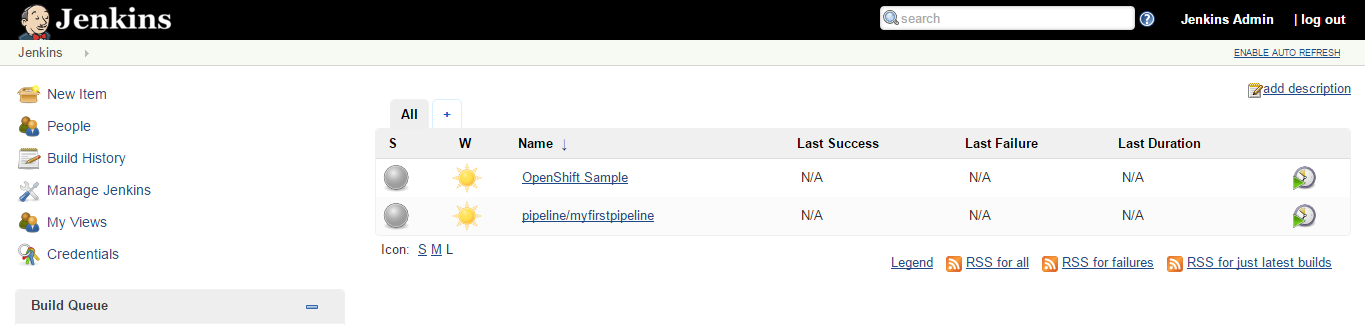
1. Go and verify in the Builds 🡪 Pipelines section in Openshift.



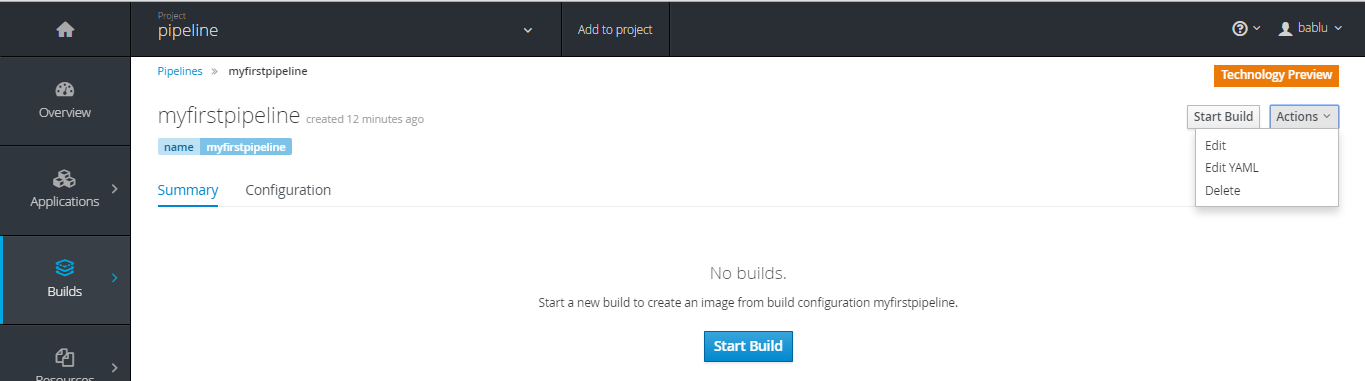
1. You can see a new pipeline has been built.



1. This pipeline is also visible in the Jenkins. Switch to Jenkins Dashboard and refresh , you can see the pipeline.

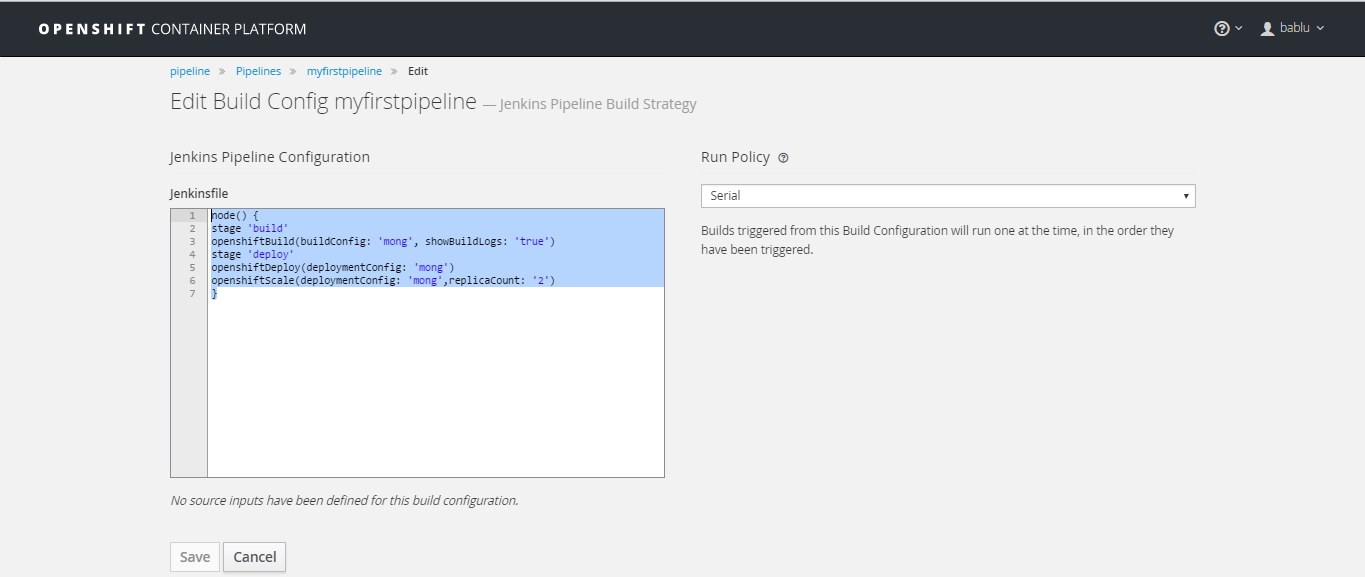


1. In Openshift, the pipeline section only has the build and deploy to Openshift part. Choose the pipeline that you created.
2. We can edit the pipeline using Actions🡪Edit



|  |
| --- |
| node() {  stage 'build'  openshiftBuild(buildConfig: 'mong', showBuildLogs: 'true')  stage 'deploy'  openshiftDeploy(deploymentConfig: 'mong')  openshiftScale(deploymentConfig: 'mong',replicaCount: '2')  } |

**Pipeline section with Build and deploy to openshift**

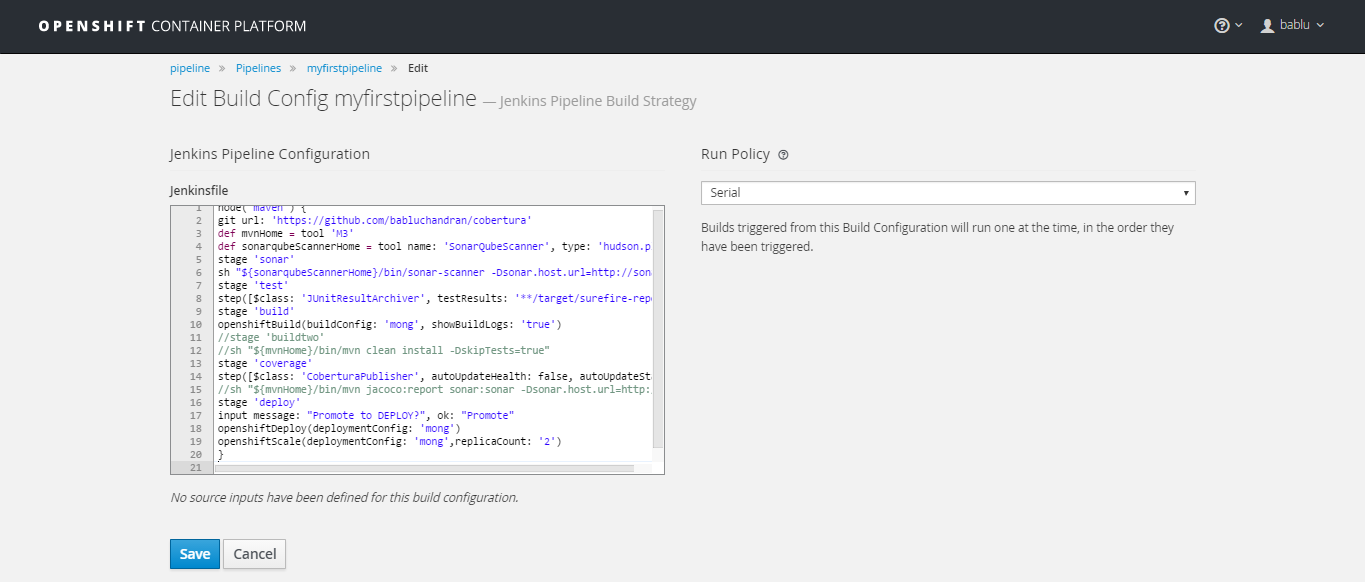


1. In the above code “mong” refers to the name of the application. So , Jenkins will deploy and build application with name “mong” inside openshift.
2. We need to add other pipeline tasks like Junit testing, sonarqube analysis and Cobertura coverage. Just add these using the pipeline code as below.

Replace the existing Jenkins file with the following code.

|  |
| --- |
| node('maven') {  git url: 'https://github.com/babluchandran/cobertura'  def mvnHome = tool 'M3'  def sonarqubeScannerHome = tool name: 'SonarQubeScanner', type: 'hudson.plugins.sonar.SonarRunnerInstallation'  stage 'sonar'  sh "${sonarqubeScannerHome}/bin/sonar-scanner -Dsonar.host.url=http://sonarqube-stage.cloudapps-b7a2.oslab.opentlc.com/ -Dsonar.login= -Dsonar.projectName=mong -Dsonar.projectVersion=1 -Dsonar.projectKey=xxx -Dsonar.sources=. -DskipTests=true"  stage 'test'  step([$class: 'JUnitResultArchiver', testResults: '\*\*/target/surefire-reports/TEST-\*.xml'])  stage 'build'  openshiftBuild(buildConfig: 'mong', showBuildLogs: 'true')  //stage 'buildtwo'  //sh "${mvnHome}/bin/mvn clean install -DskipTests=true"  stage 'coverage'  step([$class: 'CoberturaPublisher', autoUpdateHealth: false, autoUpdateStability: false, coberturaReportFile: '\*\*/target/site/cobertura/coverage.xml', failUnhealthy: false, failUnstable: false, maxNumberOfBuilds: 0, onlyStable: false, sourceEncoding: 'ASCII', zoomCoverageChart: false])  //sh "${mvnHome}/bin/mvn jacoco:report sonar:sonar -Dsonar.host.url=http://sonarqube-stage.cloudapps-b7a2.oslab.opentlc.com -DskipTests=true"  stage 'deploy'  input message: "Promote to DEPLOY?", ok: "Promote"  openshiftDeploy(deploymentConfig: 'mong')  openshiftScale(deploymentConfig: 'mong',replicaCount: '2')  } |

1. The git url in this code refers to the repository of our “mong” application. In my case I am using github as the repository.

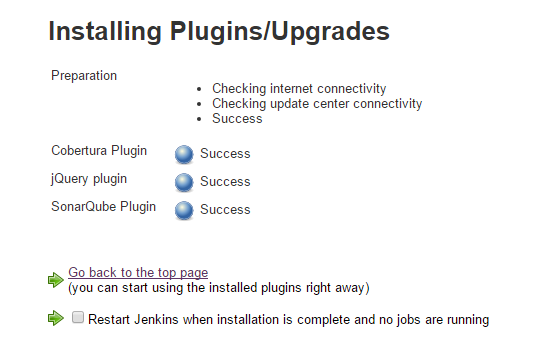


Click Save.



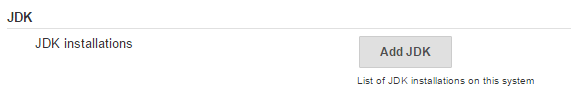
1. Now we have to add the necessary plugins in Jenkins, this includes SonarQube , Cobertura, Maven, junit etc..
2. In Jenkins Dashboard,

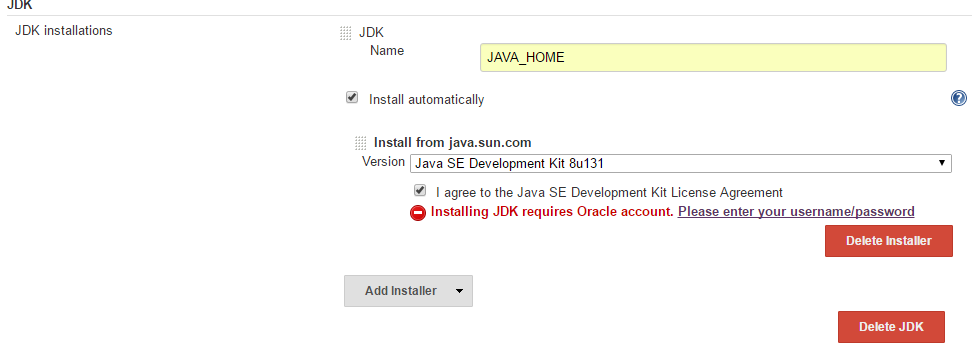
Go to Jenkins🡪manage Jenkins🡪Manage Plugins🡪Available and select all the necessary plugins mentioned above and click Install without restart.



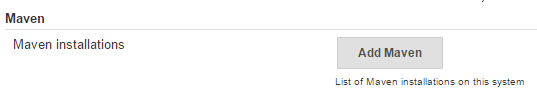
1. We now have to configure these plugins in Jenkins🡪Manage Jenkins🡪Configure System

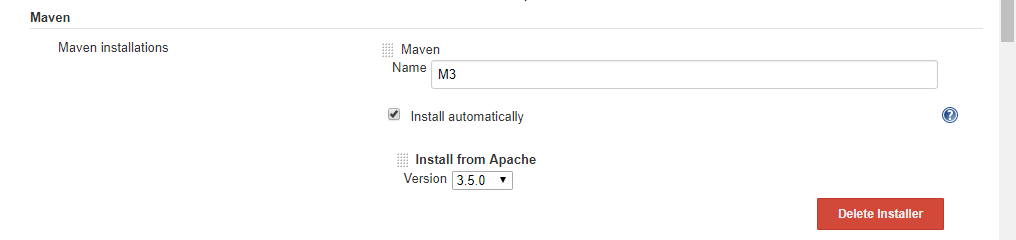
Configure jdk, use oracle account (credentials) to download Java.





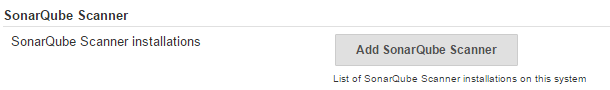
Maven section- Add Maven

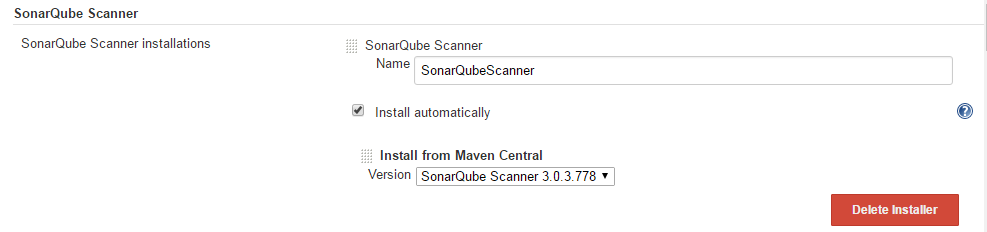




Configure Maven, “M3”is the name mentioned in the yaml section.

SonarQube:



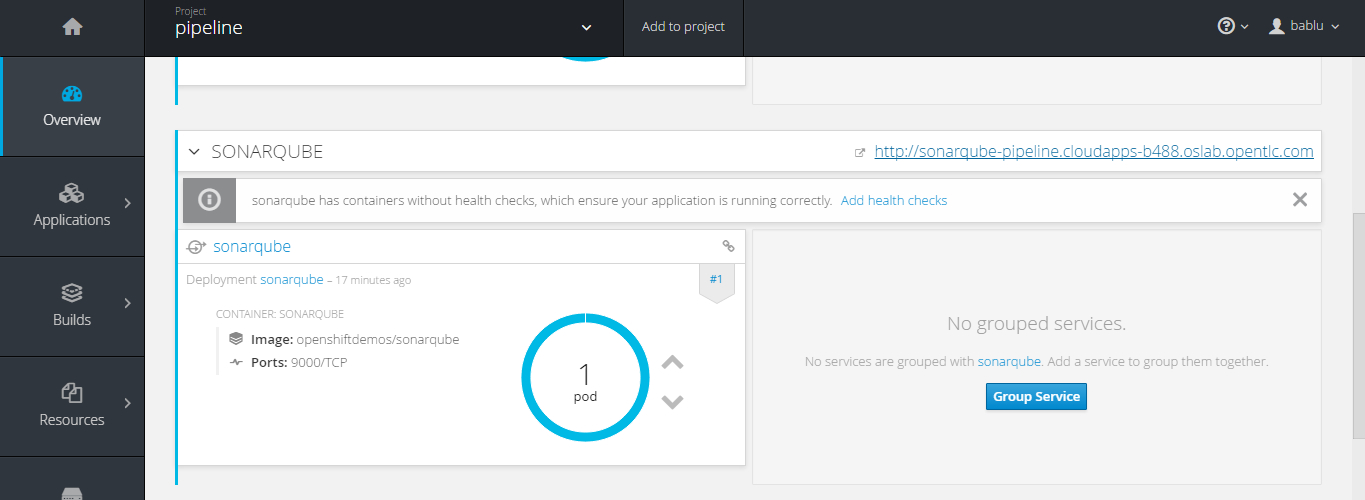


Configure SonarQube, “SonarQubeScanner” is the name mentioned in the yaml in sonar stage section and save.

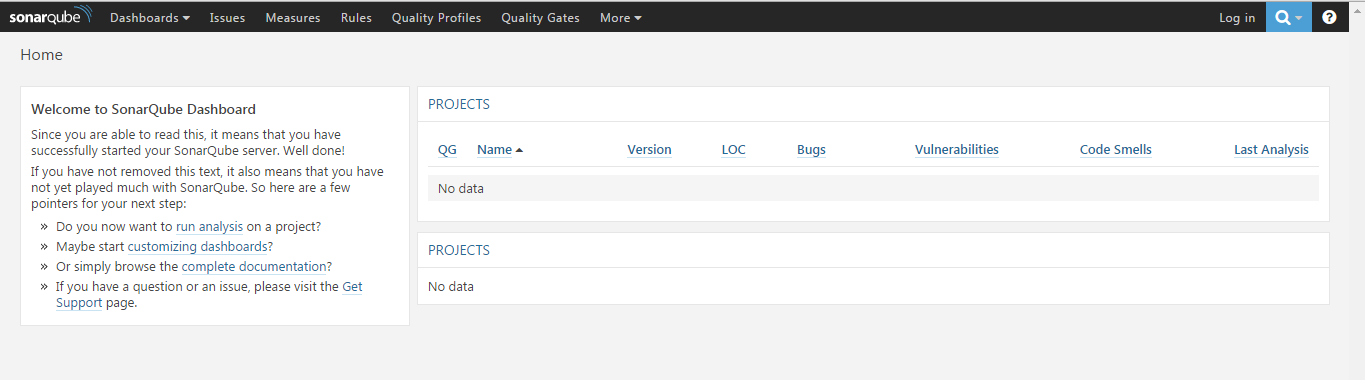
1. We now have to have the SonarQube server in Openshift. Install SonarQube using the commands in Openshift master machine.

|  |
| --- |
| oc new-app postgresql-ephemeral \  -p POSTGRESQL\_USER=sonar,POSTGRESQL\_PASSWORD=sonar,POSTGRESQL\_DATABASE=sonar |
| oc new-app docker.io/openshiftdemos/sonarqube:6.0 \  -e SONARQUBE\_JDBC\_USERNAME=sonar,SONARQUBE\_JDBC\_PASSWORD=sonar,SONARQUBE\_JDBC\_URL=jdbc:postgresql://postgresql/sonar |

This will run a new pod in openshift for SonarQube.

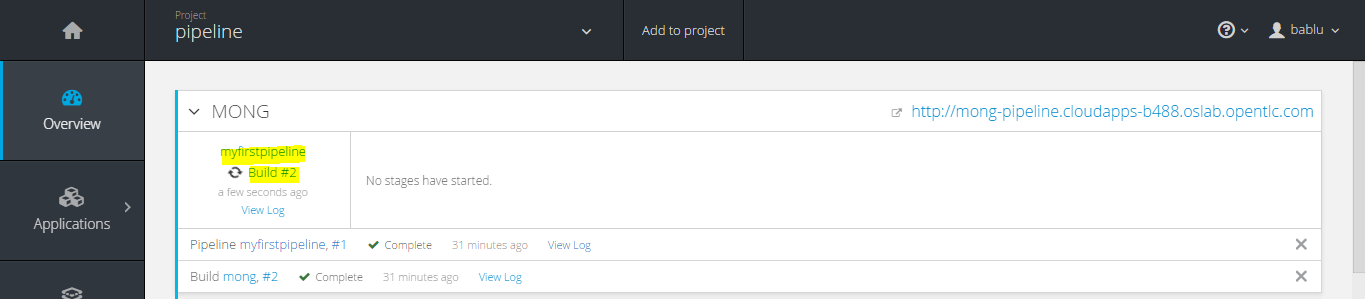


Use the http route to access the sonar dashboard.

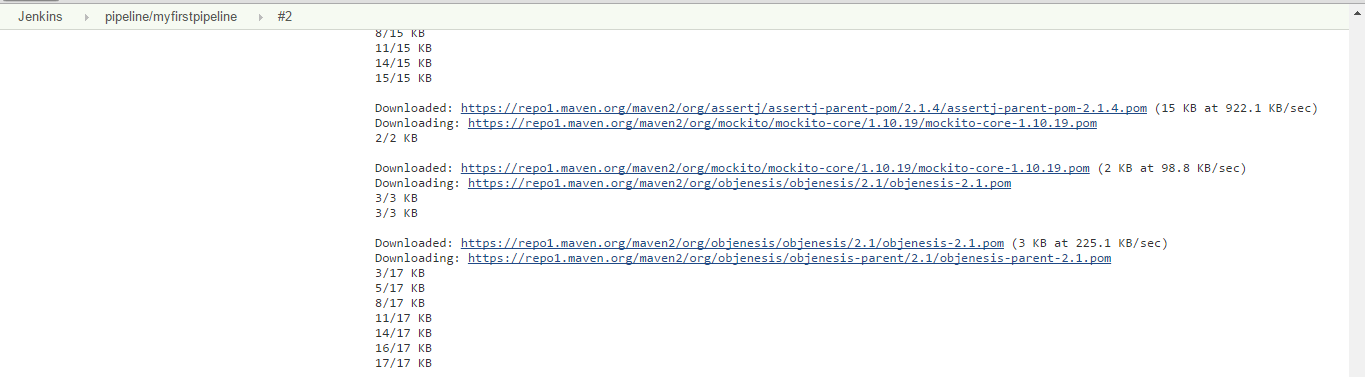


1. In Openshift console, Choose your pipeline and edit the –Dsonar.host.url with the sonar dashboard http link. Pipeline🡪actions🡪edit section. Click Save.
2. Now start the pipeline in Openshift. Pipelines 🡪Start Pipeline

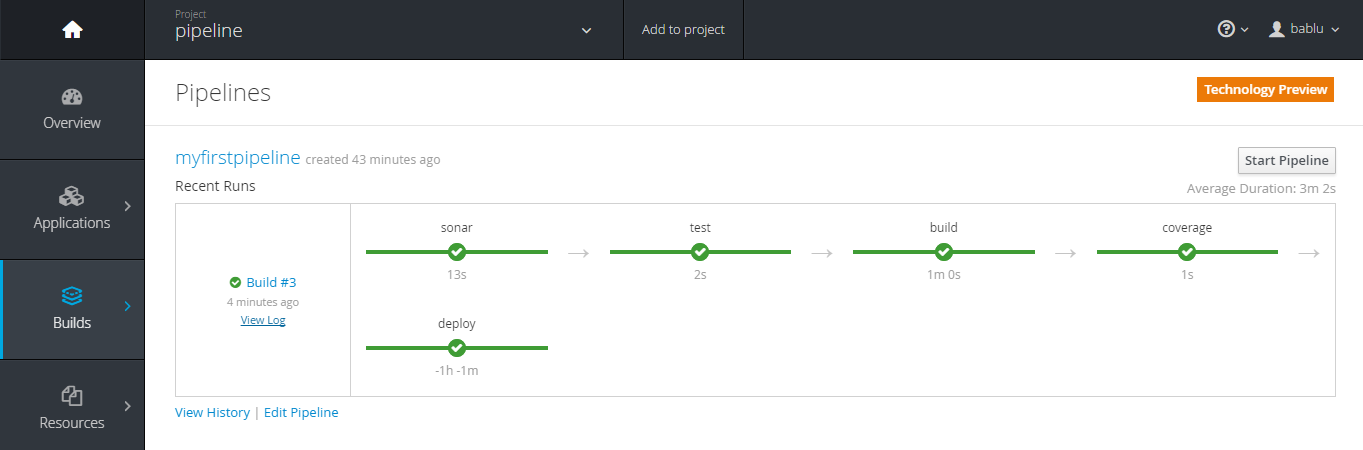




This will start a new build in openshift. View log will redirect us to the Jenkins log dashboard.



View the result in openshift console.



Also in Jenkins dashboard.

