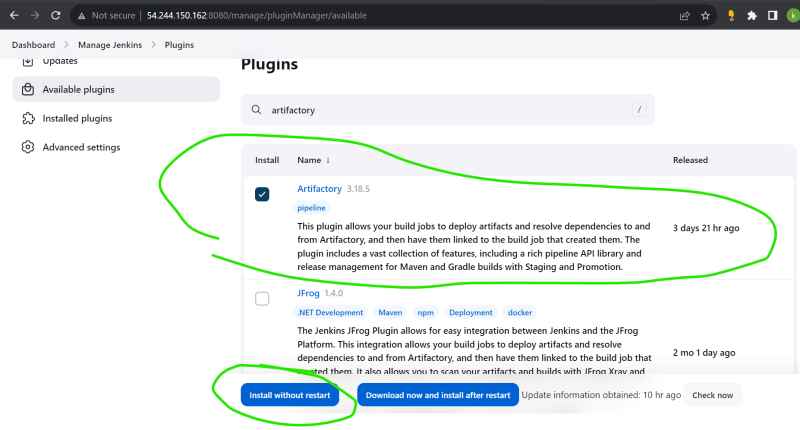
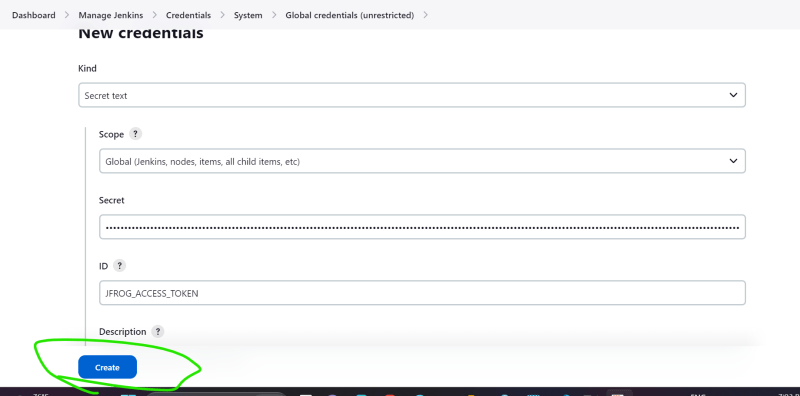
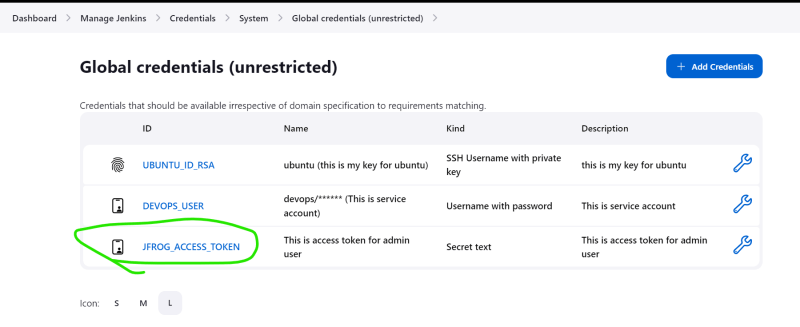
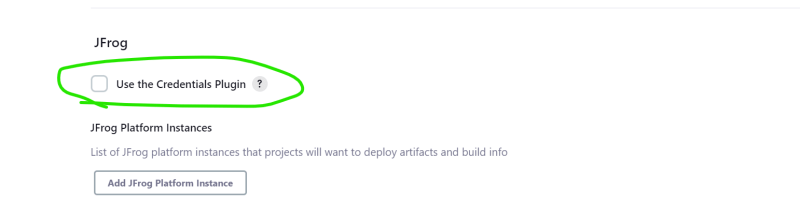
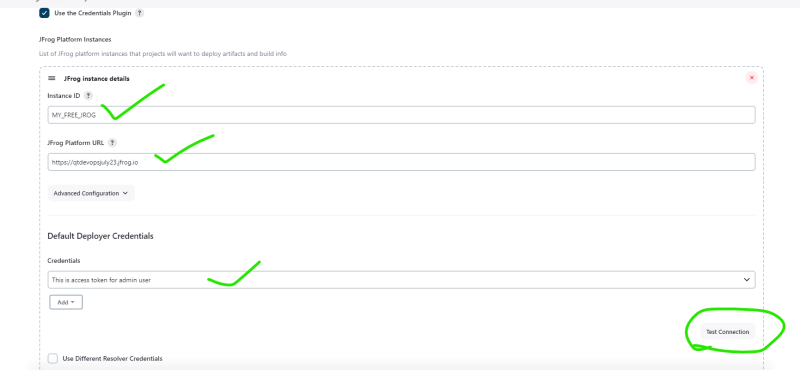
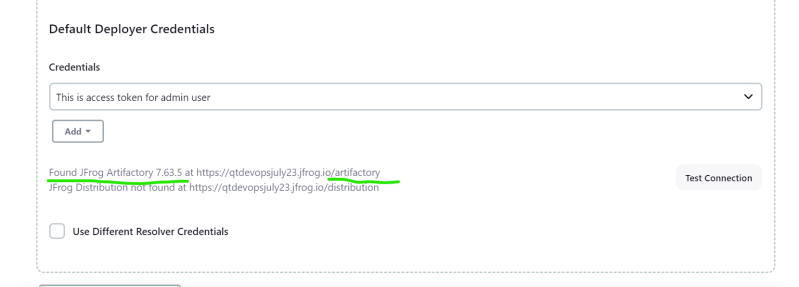
**Artifactory Jenkins integration**

* Create an access token after frog account creation
* Install artifactory plugin in jenkins  
  
* Now Navigate to Manage Jenkins -> Credentials and create a credential with secret text  
    
  
* Now Navigate to Manage Jenkins -> System -> Jfrog  
    
    
  
* <https://jfrog.com/help/r/jfrog-integrations-documentation/jenkins-artifactory-plug-in> for official docs of jfrog artifactory pipeline
* <https://github.com/jfrog/project-examples/tree/master/jenkins-examples/pipeline-examples/declarative-examples> for samples of jfrog jenkins pipelines
* <https://github.com/jfrog/project-examples/blob/master/jenkins-examples/pipeline-examples/declarative-examples/maven-example/Jenkinsfile> for specific jenkinsfile
* The pipeline example

pipeline {

agent { label 'JDK-17' }

options {

timeout(time: 30, unit: 'MINUTES')

}

triggers {

pollSCM('\* \* \* \* \*')

}

tools {

jdk 'JDK\_17'

}

stages {

stage('vcs') {

steps {

git url: 'https://github.com/dummyrepos/spring-petclinic-1.git',

branch: 'develop'

}

}

stage('build and package') {

steps {

rtMavenDeployer (

id: "SPC\_DEPLOYER",

serverId: "JFROG\_CLOUD",

releaseRepo: 'qt-app-libs-snapshot-local',

snapshotRepo: 'qt-app-libs-snapshot-local'

)

rtMavenRun (

tool: 'MAVEN\_3.9', // Tool name from Jenkins configuration

pom: 'pom.xml',

goals: 'clean install',

deployerId: "SPC\_DEPLOYER"

//,

//buildName: "${JOB\_NAME}",

//buildNumber: "${BUILD\_ID}"

)

rtPublishBuildInfo (

serverId: "JFROG\_CLOUD"

)

}

}

stage('reporting') {

steps {

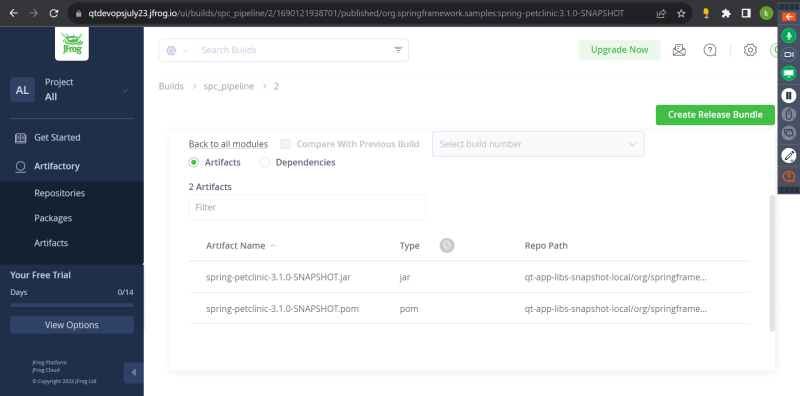
junit testResults: '\*\*/target/surefire-reports/TEST-\*.xml'

}

}

}

}



**Static code Analysis**

* For static code analysis lets use sonar cloud
* <https://docs.sonarcloud.io/advanced-setup/ci-based-analysis/jenkins-extension-for-sonarcloud/#:~:text=Configure%20SonarCloud%3A,created%20as%20a%20> for configuration docs
* <https://www.sonarsource.com/products/sonarcloud/signup/> to create sonarqube cloud account
* Now lets create a SonarQube static code analysis
* <https://directdevops.blog/2019/01/05/sonarqube/> for configuring and installing sonar qube
* The pipeline

pipeline {

agent any

options {

timeout(time: 30, unit: 'MINUTES')

}

triggers {

pollSCM('\* \* \* \* \*')

}

tools {

jdk 'JDK\_17\_UBUNTU'

maven 'MAVEN\_3.9'

}

stages {

stage('vcs') {

steps {

git url: 'https://github.com/dummyrepos/spring-petclinic-1.git',

branch: 'develop'

}

}

stage('SonarQube analysis') {

steps {

// performing sonarqube analysis with "withSonarQubeENV(<Name of Server configured in Jenkins>)"

withSonarQubeEnv('SONAR\_CLOUD') {

// requires SonarQube Scanner for Maven 3.2+

sh 'mvn clean package sonar:sonar -Dsonar.organization=khajaprojectsjuly23 -Dsonar.token=67d5cbb26a76f3a1c2c669a0d7be62e66722c488 -Dsonar.projectKey=springpetclinic'

}

}

}

stage('reporting') {

steps {

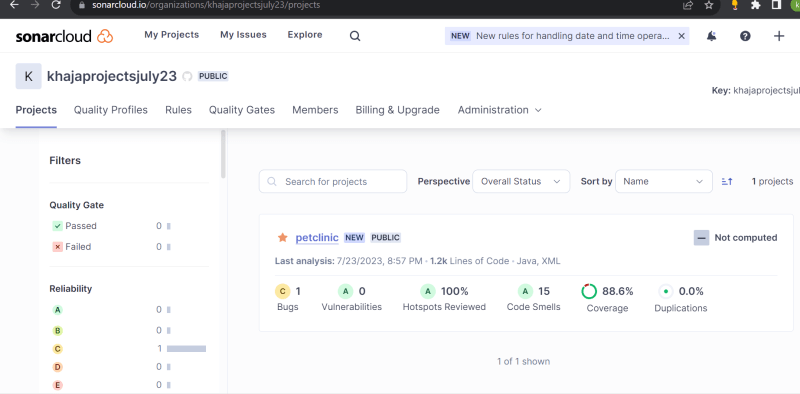
junit testResults: '\*\*/target/surefire-reports/TEST-\*.xml'

}

}

}

}



Note: <https://learn.microsoft.com/en-us/azure/devops/pipelines/ecosystems/ecosystems?view=azure-devops> for different ecosystem