**Deleting based on range of a project**

pipeline{

agent any

stages{

stage('Delete based on build range of a given project'){

steps{

script{

def jobName = 'KarthikServerDeploy1'

def rs = Fingerprint.RangeSet.fromString("83-92", false);

def reallyDelete = false;

def job = Jenkins.instance.getItemByFullName(jobName);

println("Job: ${job.fullName}");

def builds = Jenkins.instance.getItemByFullName(jobName).getBuilds(rs);

println("Found ${builds.size()} builds");

builds.each{ b->

if (reallyDelete && b.result

== Result.SUCCESS) {

println("Deleting ${b}");

b.delete();

} else {

println("Found match ${b}");

}

}

}

script{

Jenkins.instance.getItemByFullName('KarthikServerDeploy1').builds.findAll { it.number > 83 && it.number < 92 && it.result == Result.SUCCESS}.each { it.delete() }

}

}

}

}

}

**Server Deploy and email**

pipeline{

agent any

stages{

stage('Git'){

steps{

git 'https://github.com/karthiktr147/WarDeployment.git'}

}

stage('war package'){

tools{

maven 'Maven'

}

steps{

sh 'mvn clean package'

}

}

stage('server'){

steps{

sh 'cp /var/lib/jenkins/workspace/pipeline/target/\*.war /opt/tomcat/webapps/'

}

}

}

post {

always {

echo 'build status mail from jenkins'

mail to: 'karthiktr147@gmail.com',

subject: "Result: ${currentBuild.fullDisplayName}",

body: "${env.BUILD\_URL} has result ${currentBuild.result}"

}

}

}

**Artifactory jfrog and sonarqube**

pipeline{

agent any

stages

{

stage('Source'){

steps{ git 'https://github.com/karthiktr147/SimpleProject.git'}

}

stage('Compile package'){

tools {

maven 'Maven'

}

steps{

sh 'mvn clean package'

}

}

stage('Sonar Analysis')

{

steps

{

script

{

Pipeline\_Message=env.STAGE\_NAME

echo "Sonar Analysis"

}

dir("/var/lib/jenkins/workspace/jfrogsonarkarthik")

{

withSonarQubeEnv('sonarqube')

{

sh 'mvn sonar:sonar'

}

}

}

}

stage('Artifactory')

{

steps

{

script

{

def server= Artifactory.server '1'

def buildInfo = Artifactory.newBuildInfo()

buildInfo.env.capture = true

buildInfo.env.collect()

def uploadSpec= """{

"files" :[{

"pattern": "/var/lib/jenkins/workspace/jfrogsonarkarthik/target/firstwebpipeline",

"target": "libs-snapshot-local"

}]

}"""

server.upload spec: uploadSpec, buildInfo: buildInfo

server.publishBuildInfo buildInfo

}

}

}

}

}

**Docker**

<http://52.252.102.93:8080/job/karthikdocker/configure>