INSTALL GIT AND CONFIGURE

Text

Description automatically generated

INSTALL JDK ANY VERSION jdk 16 ,64 installers

Set path in environment variables and set path with bin

A screenshot of a computer

Description automatically generated with low confidence

Install maven

[Maven – Download Apache Maven](https://maven.apache.org/download.cgi)

Set path and environment variables

Text

Description automatically generated

New item / new job

Select free style project

Build environment

Down select build- from windows bash save

Graphical user interface, text, application, email

Description automatically generated

**Create Jenkins Pipeline for automating Docker image creation and push docker image into Docker Hub**

**Step #1 - Create Credentials for Docker Hub**  
Go to your Jenkins where you have installed Docker as well. Go to credentials -->

[Graphical user interface, text, application, chat or text message

Description automatically generated](https://1.bp.blogspot.com/-XJG5VsPD8Qo/Xq-M6yxaHZI/AAAAAAAACFQ/vXDXsbe0hyM91GM0Bh5VNZXsGasOmFWeACLcBGAsYHQ/s1600/crdentials.png)

Click on Global credentials

[Graphical user interface, text, application

Description automatically generated](https://1.bp.blogspot.com/-04oZVUbk5ZY/Xq-NShoX-XI/AAAAAAAACFY/-t8z6KhrGfkaGivRrGk_IOO8-L9MHJhIwCLcBGAsYHQ/s1600/global.png)

Click on Add Credentials

[Graphical user interface, application

Description automatically generated](https://1.bp.blogspot.com/-Mvy6LFnLTsY/Xq-NSobh4VI/AAAAAAAACFc/Pkx9j31nxKobJySQegbx2uDD36JSNrHuACLcBGAsYHQ/s1600/add.png)  
  
Now Create an entry for Docker Hub credentials

[Graphical user interface, application, email

Description automatically generated](https://1.bp.blogspot.com/-aSBa_ebTebQ/Xq-N58ikwdI/AAAAAAAACFs/mbrp_V-Z_B0AYq1i9vdVeTybm_M5BZjsQCLcBGAsYHQ/s1600/user.png)

Make sure you take note of the ID as circled below:

[](https://1.bp.blogspot.com/-9ne6aQjxrNY/Xq-OeqV2rpI/AAAAAAAACF0/Lk3RdKyLiuch30Nf-ZPDJj9smeqRXzHDwCLcBGAsYHQ/s1600/id1.png)

**Step # 2 - Create a pipeline in Jenkins, name can be anything**

[Graphical user interface, text, application, email

Description automatically generated](https://1.bp.blogspot.com/-r0513ohKbYQ/Xq-UkLCFTJI/AAAAAAAACGU/A9SLAF0F5F0qJbQQtpfHEDgiAnopRrh0QCLcBGAsYHQ/s1600/pipeline.png)

**Step # 3 - Copy the pipeline code from below**  
Make sure you change red highlighted values below:  
Your docker user id should be updated.  
your registry credentials ID from Jenkins from step # 1 should be copied  
  
pipeline {  
    agent any   
    environment {  
        //once you sign up for Docker hub, use that user\_id here  
        registry = "your\_docker\_user\_id/mypythonapp"  
        //- update your credentials ID after creating credentials for connecting to Docker Hub  
        registryCredential = 'fa32f95a-2d3e-4c7b-8f34-11bcc0191d70'  
        dockerImage = ''  
    }  
      
    stages {  
        stage('Cloning Git') {  
            steps {  
                checkout([$class: 'GitSCM', branches: [[name: '\*/master']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[credentialsId: '', url: 'https://bitbucket.org/ananthkannan/mypythonrepo']]])         
            }  
        }  
      
    // Building Docker images  
    stage('Building image') {  
      steps{  
        script {  
          dockerImage = docker.build registry  
        }  
      }  
    }  
      
     // Uploading Docker images into Docker Hub  
    stage('Upload Image') {  
     steps{      
         script {  
            docker.withRegistry( '', registryCredential ) {  
            dockerImage.push()  
            }  
        }  
      }  
    }  
      
     // Stopping Docker containers for cleaner Docker run  
     stage('docker stop container') {  
         steps {  
            sh 'docker ps -f name=mypythonappContainer -q | xargs --no-run-if-empty docker container stop'  
            sh 'docker container ls -a -fname=mypythonappContainer -q | xargs -r docker container rm'  
         }  
       }  
      
      
    // Running Docker container, make sure port 8096 is opened in   
    stage('Docker Run') {  
     steps{  
         script {  
            dockerImage.run("-p 8096:5000 --rm --name mypythonappContainer")  
         }  
      }  
    }  
  }  
}

**Step # 4 - Click on Build - Build the pipeline**

Once you create the pipeline and changes values per your Docker user id and credentials ID, click on

A picture containing table

Description automatically generated