**Dockerizing Jenkins Pipeline**

**STEP 1** Setup a VS code workspace and the Github repository.

1. Open VS code.
2. Create a directory “simplilearn-devops-certification” in the terminal and change direcory.
3. Run “git init” to initilize repository.
4. Create a reposiroty “simplilearn-devops-certification “ in the github.com
5. Create a file name “README.md” add details of the project in the file.
6. Execute the steps to do Initial commit, This will add the project in the github master branch.
   1. git remote add origin https://github.com/aktechthoughts/simplilearn-devops-certification.git
   2. git add .
   3. git commit –m “ Initial Commit”
   4. git push --set-upstream origin master

**STEP 2** Setup docker and Jenkins running on docker container I the same machine.

1. Install a docker on the machine with docker destop.
2. Install Jenkins on the same machine using docker container by running the following command in powershell

docker run -u root --rm -d -p 8080:8080 -p 50000:50000 -v C:\Users\.docker\Jenkins:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock --name jenkins jenkinsci/blueocean

1. The previous command will run the Jenkin server at localhost:8080 port and it can be accessed in the browser using <http://localhost:8080/>
2. Select “Install Selected packages” and wait for installation to finish.
3. Create a new user after installation is finished.
4. same machine where Jenkin is installed.

**STEP 3**  Setup Jenkinsfile in the repository

1. Create Jenkinsfile in the project root directory.
2. Add following content in the file.

pipeline {

  environment {

    registry = "sathishsubramanian/dockerising\_jenkins\_pipeline"

    registryCredential = 'dockerhub'

  }

  agent any

  stages {

    stage('Building image') {

      steps{

        script {

          dockerImage = docker.build registry + ":$BUILD\_NUMBER"

        }

      }

    }

    stage('Deploy Image') {

      steps{

        script {

          docker.withRegistry( '', 'dockerhub' ) {

            dockerImage.push()

          }

        }

      }

    }

    stage('Remove Image') {

      steps{

        sh "docker rmi $registry:$BUILD\_NUMBER"

      }

    }

  }

}

node {

stage('Execute Image'){

def customImage = docker.build("sathishsubramanian/dockerising\_jenkins\_pipeline:${env.BUILD\_NUMBER}")

customImage.inside {

sh 'echo Hello'

}

}

}

1. There are four stages in the Jenkins.
   1. Building Image
   2. Deploying the image in the dockerhub repository
   3. Removing the Image from Jenkin node.
   4. Executing Image from dockerhub.

**STEP 4** Register and open <https://hub.docker.com/> with your own login.

1. Create a new docker reposiory named ‘dockerising\_jenkins\_pipeline’.
2. Create a file ‘Dockerfile’ in the project created in STEP1
3. Add following conent in the file.

# Dockerfile

FROM busybox

CMD echo "Hello world! This is my first Docker image."

1. Run this git hub project as pipeline script in Jenkins by running the Jenkins script file.