**PROJECT 1**

**Dockerizing Jenkins Pipeline**

**STEP1: INTIALIZING AND CONFURING GIT AND GITHUB**

* LAUNCH UBUNTU INSTANCE
* OPEN COMMAND LINE TERMINAL
* CHECK FOR GIT

Git --version

* IF GIT NOT INSTALLED THEN INSTALL GIT THEN RUN FOLLOWING COMMANDS

Sudo apt-get update

Sudo apt-get install git

* After install git check version by previous command
* Create a new directory named gitdemo

mkdir gitdemo

cd git demo

* Initializing the git by

Git init

* Create a file index.html in the directory, write some code for html in it
* Create few more blank file by command

touch contact.html

* Use following commands for add and commit file to local repository

git status

git add –A (for all file adds)

git commit –m “first”

git status

Steps for github

* Open to github.com
* Create a login at github.com
* Create a repository named newdevopsproject
* Now push local repository to git repository named gitdemo

git remote add origin https://github.com/mailsanjaykumar/newdevopsproject.git

git push -u origin main

* Check the github repository
* If any change in any file in local repository then push to github repository

**STEP2 : SETUP A JENKIN SERVER AND REPOSITORY**

* Commands to install Jenkins

wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add –

sudosh -c 'echo deb http://pkg.jenkins-ci.org/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get upgrade

sudo apt-get update

sudo apt install openjdk-8-jdk –y

sudo apt-get –y install jenkins

sudo service Jenkins start

sudo service Jenkins status

ps –ef | grep Jenkins

* Jenkins can be opened at 8080 port
* To run/use Jenkins outside by azure instance address like

<https://mailsanjaykumar.centralus.cloudapp.azure.com:8080>

* Fill username and password i.e. mailsanjaykumar/goto@1234
* Click on manage Jenkins on left side panel
* In the open page click on manage plugins
* Search Docker in available sub tab and select from the list of available plugins related to docker and install
* Again to manage Jenkins ,click on configure system
* At the bottom, there is a dropdown called Add a new cloud. Select Docker from the list.
* The Docker Host URI is where Jenkins launches the agent container. In this case, we'll use the same daemon as running Jenkins, but you could split the two for scaling. Enter the URL unix: ///var/run/docker.sock
* Run a command at terminal

Sudo chmod 777 /var/run/docker.sock

* Use Test Connection to verify Jenkins can talk to the Docker Daemon. You should see the Docker version number returned.
* On the Jenkins dashboard, select Create new jobs
* Give the job a friendly name such as [newdevopsproject](http://mailsanjaykumar.centralus.cloudapp.azure.com:8080/job/newdevopsproject/), select Freestyle project then click OK.
* The build will depend on having access to Docker. Using the "Restrict where this project can be run" we can define the label we set of our configured Docker agent. The set "Label Expression" to docker-agent. You should have a configuration of "Label is serviced by no nodes and 1 cloud".
* Select the Repository type as Git and set the Repository to be https://github.com/mailsanjaykumar.
* We can now add a new build step using the Add Build Step dropdown. Select Execute Shell.
* Because the logical of how to build is specified in our Dockerfile, Jenkins only needs to call build and specify a friendly name.

docker info

docker build -t newdevopsproject:${BUILD\_NUMBER} .

docker tag newdevopsproject:${BUILD\_NUMBER} newdevopsproject:latest

docker images

* Our build is now complete. Click Save
* On the left-hand side, select Build Now
* Built is complete and image is build in docker

STEP 3: INSTALLING DOCKER AND PUSH IMAGE TO DOCKER REPOSITORY

* login to docker hub using command

Docker login –username=mailsanjaykumar

Then enter password

docker tag newdevopsproject:18 mailsanjaykumar/ newdevopsproject

docker images

docker push mailsanjaykumar/newdevopsproject

* This image is pushed to dockerhub repository
* Login to <https://hub.docker.com/>
* Click on [Repositories](https://hub.docker.com/repositories)
* We find a repository image is created with same name [mailsanjaykumar / newdevopsproject](https://hub.docker.com/repository/docker/mailsanjaykumar/newdevopsproject)