**Working with Declarative Pipelines**

**1.Install Docker and Start the Service**

yum -y install docker

service docker start

**2. Update the Group properties of Jenkins user id as below (Permissons to access docker)**

[root@ip-172-31-32-46 CICD]# id jenkins

uid=498(jenkins) gid=497(jenkins) groups=497(jenkins)

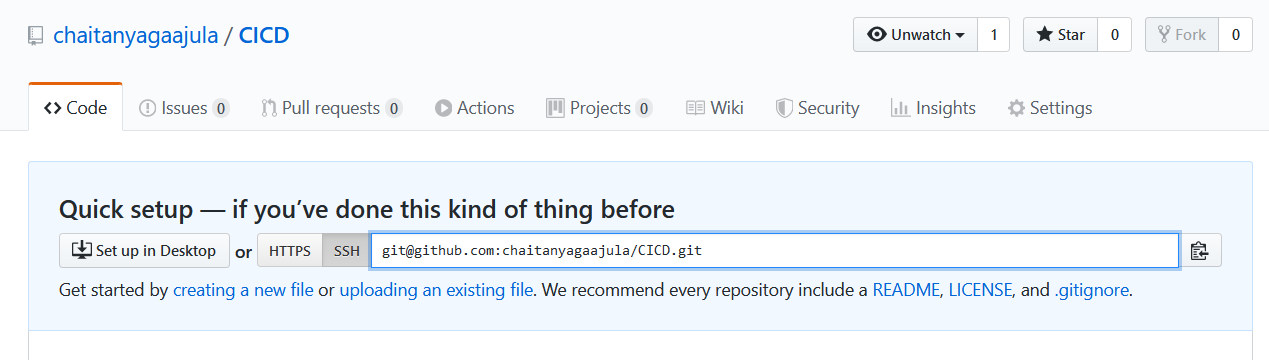
[root@ip-172-31-32-46 CICD]# usermod -G docker jenkins

[root@ip-172-31-32-46 CICD]# id jenkins

uid=498(jenkins) gid=497(jenkins) groups=497(jenkins),496(docker)

[root@ip-172-31-32-46 CICD]#

**3.Fork the repo -**<https://github.com/chaitanyagaajula/CICD>



4. **Check if the below files are there in /root/.ssh**

[root@ip-172-31-32-46 .ssh]# pwd

/root/.ssh

[root@ip-172-31-32-46 .ssh]# ls -lrt

total 8

-rw------- 1 root root 547 Apr 14 13:50 authorized\_keys

-rw-r--r-- 1 root root 884 Apr 15 10:56 known\_hosts

5. **Generating public/private rsa key pair**

[root@ip-172-31-32-46 .ssh]# ssh-keygen

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:a2dzgN0XBkePTQsshiU/rKqR+rk11QumsBJotqKmf/I root@ip-172-31-32-46

The key's randomart image is:

+---[RSA 2048]----+

| .o.ooo .|

| .+o +.=.|

| .+. +.o|

| . oo... . |

| + . . S=o.. . |

|o . . + =..... |

|.. . + =o +.. |

|o.. + =..o o |

|=..=E=. |

+----[SHA256]-----+

**6. List the files to check details of Public and Private Keys**

[root@ip-172-31-32-46 .ssh]# ls -lrt

total 16

-rw------- 1 root root 547 Apr 14 13:50 authorized\_keys

-rw-r--r-- 1 root root 884 Apr 15 10:56 known\_hosts

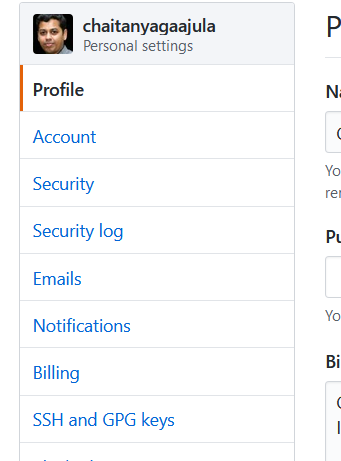
-rw-r--r-- 1 root root 402 Apr 15 10:57 id\_rsa.pub

-rw------- 1 root root 1675 Apr 15 10:57 id\_rsa

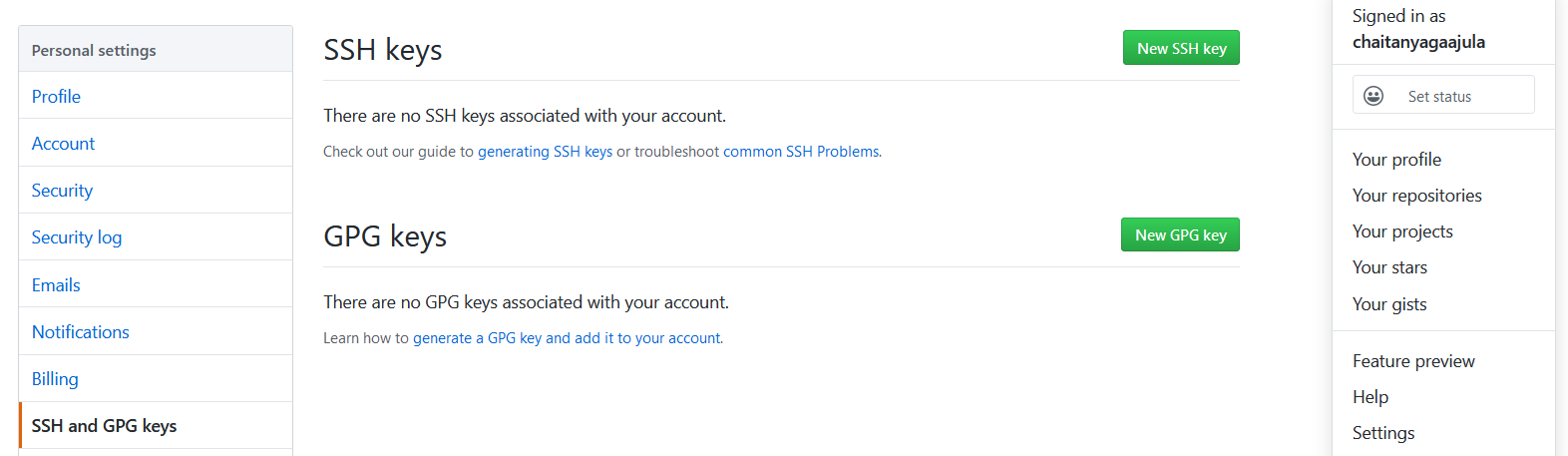
[root@ip-172-31-32-46 .ssh]#

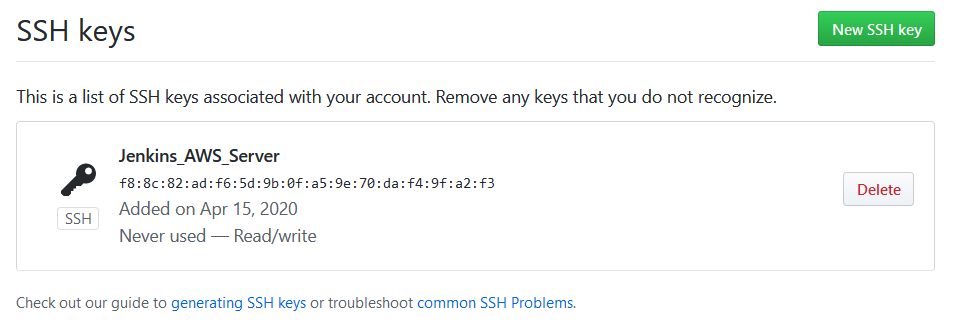
6) **Update Public Key contents in GitHub.**

User Profile -> Settings -> SSH and GPG keys , Create new SSH Key and copy contents of Public key (id\_rsa.pub)



Add New ssh key





**4. Installing GIT client in EC2 server as below**

**The 4th step is for Trainer machine only.**

[root@ip-172-31-32-46 CICD]# yum update

[root@ip-172-31-32-46 CICD]# git --version

git version 2.14.5

[root@ip-172-31-32-46 CICD]# git config --global user.name "chaitanyagaajula"

[root@ip-172-31-32-46 CICD]# git config --global user.email "chaitanyagaajula@gmail.com"

[root@ip-172-31-32-46 CICD]# git config --global --list

user.name=chaitanyagaajula

user.email=chaitanyagaajula@gmail.com

[root@ip-172-31-32-46 CICD]#

[root@ip-172-31-32-46 CICD]# git init

Initialized empty Git repository in /root/CICD/.git/

[root@ip-172-31-32-46 CICD]# ls -lrt

total 0

[root@ip-172-31-32-46 CICD]# ls -lrta

total 12

dr-xr-x--- 6 root root 4096 Apr 15 10:54 ..

drwxr-xr-x 3 root root 4096 Apr 15 10:54 .

drwxr-xr-x 7 root root 4096 Apr 15 10:54 .git

[root@ip-172-31-32-46 CICD]#

[root@ip-172-31-32-46 CICD]# echo "# CICD" >> README.md

[root@ip-172-31-32-46 CICD]# ls

README.md

[root@ip-172-31-32-46 CICD]# git add README.md

[root@ip-172-31-32-46 CICD]# git commit -m "first commit"

[master (root-commit) 31db9d5] first commit

1 file changed, 1 insertion(+)

create mode 100644 README.md

[root@ip-172-31-32-46 CICD]# git remote add origin git@github.com:chaitanyagaajula/CICD.git

[root@ip-172-31-32-46 CICD]# git push -u origin master

Counting objects: 3, done.

Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.

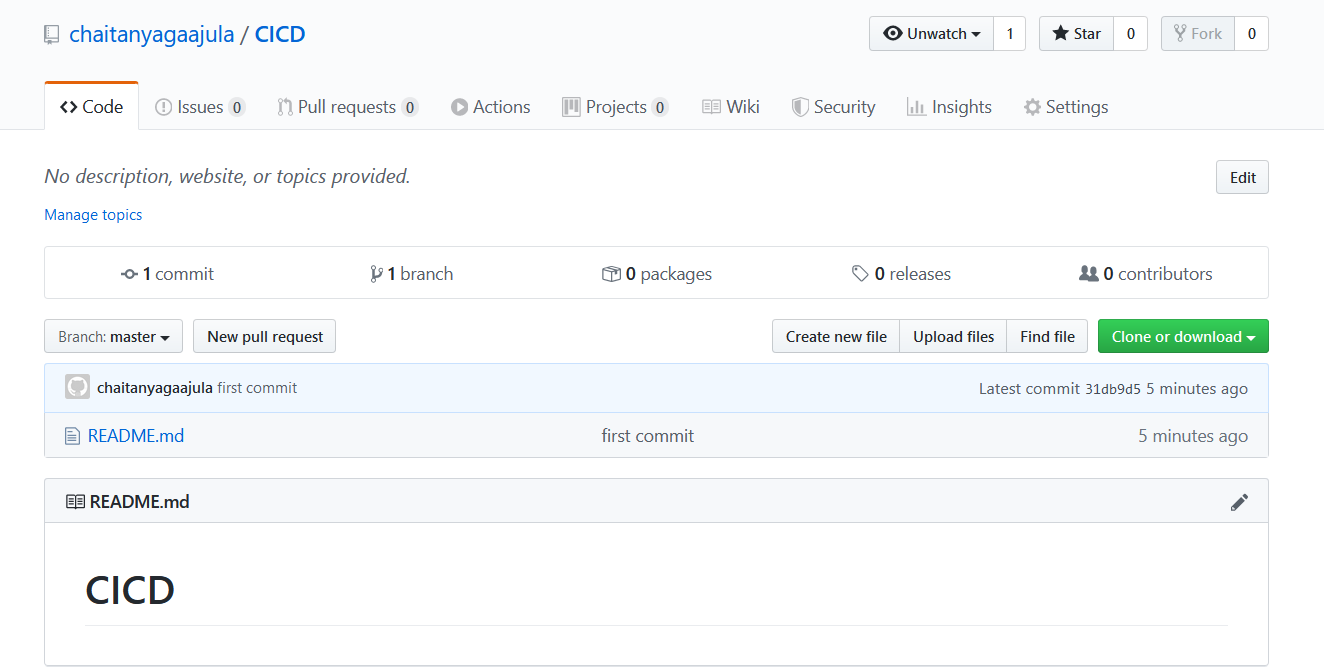
Total 3 (delta 0), reused 0 (delta 0)

To github.com:chaitanyagaajula/CICD.git

\* [new branch] master -> master

Branch master set up to track remote branch master from origin.

[root@ip-172-31-32-46 CICD]#



7) **Clone the repo by following the below steps**,

Student machine ## cd /root/

mkdir JenkinsProject

Now cd to /root/JenkinsProject/

And do the git clone as below

git clone <https://github.com/chaitanyagaajula/CICD.git> .

It will create directory named ‘CICD’ with all necessary project files,

**8) List the files , you can see the contents of the repo**

[root@ip-172-31-85-119 CICD]# pwd

/root/JenkinsProject/CICD

[root@ip-172-31-85-119 CICD]# ls -lrt

total 28

drwxr-xr-x 3 root root 4096 May 4 08:38 tests

-rw-r--r-- 1 root root 26 May 4 08:38 requirements.txt

-rw-r--r-- 1 root root 7 May 4 08:38 README.md

-rw-r--r-- 1 root root 196 May 4 08:38 Dockerfile

-rw-r--r-- 1 root root 331 May 4 08:38 app.py

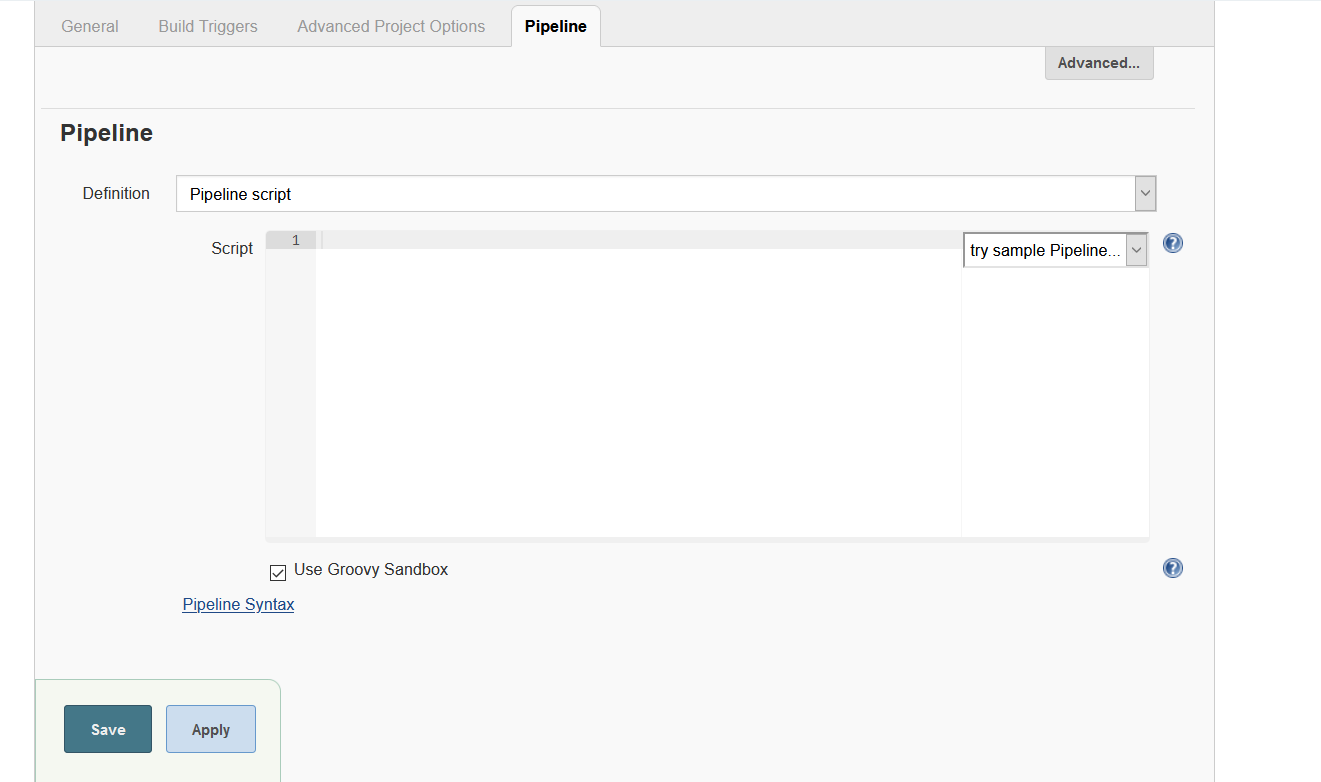
drwxr-xr-x 3 root root 4096 May 4 11:17 buzz

-rw-r--r-- 1 root root 1226 May 4 11:25 Jenkinsfile

9) **Now login to Jenkins**

Jenkins -> New item -> type Project name as ‘CICID\_Pipeline1’

**10). Under Pipeline Tab, Click on Pipeline Syntax link**

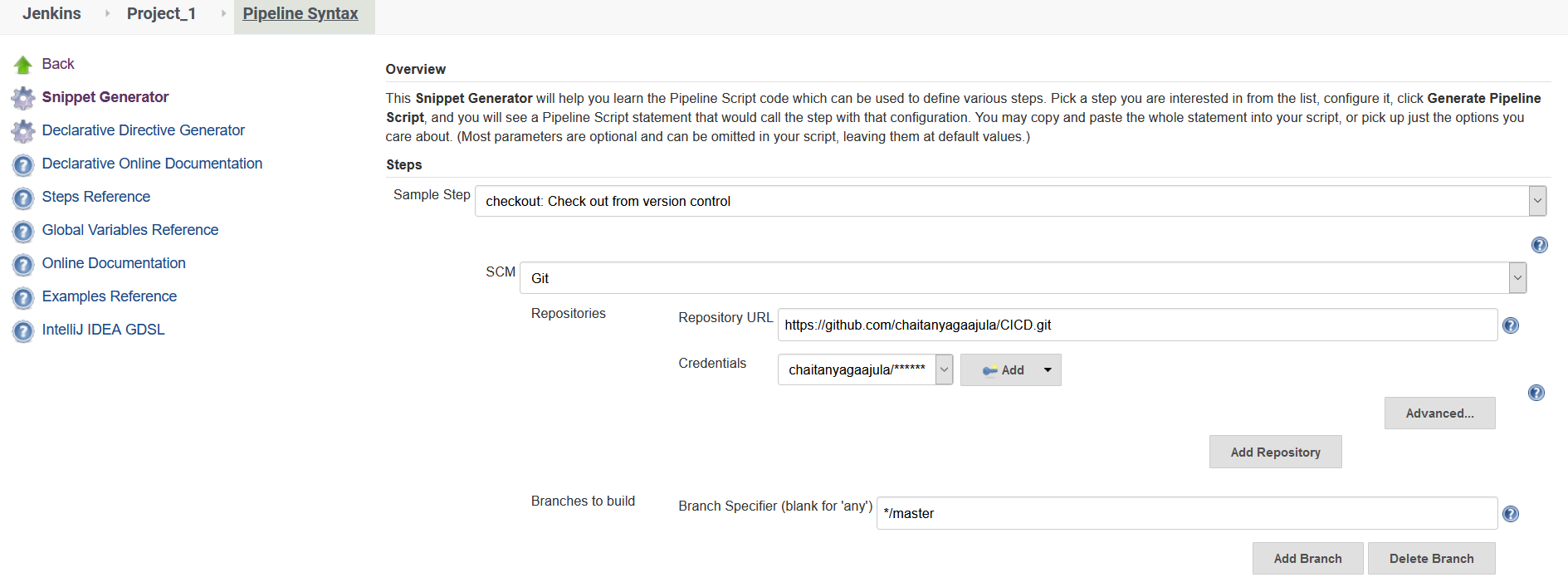


11. We are creating Declarative Pipeline script using Groovy script as below,

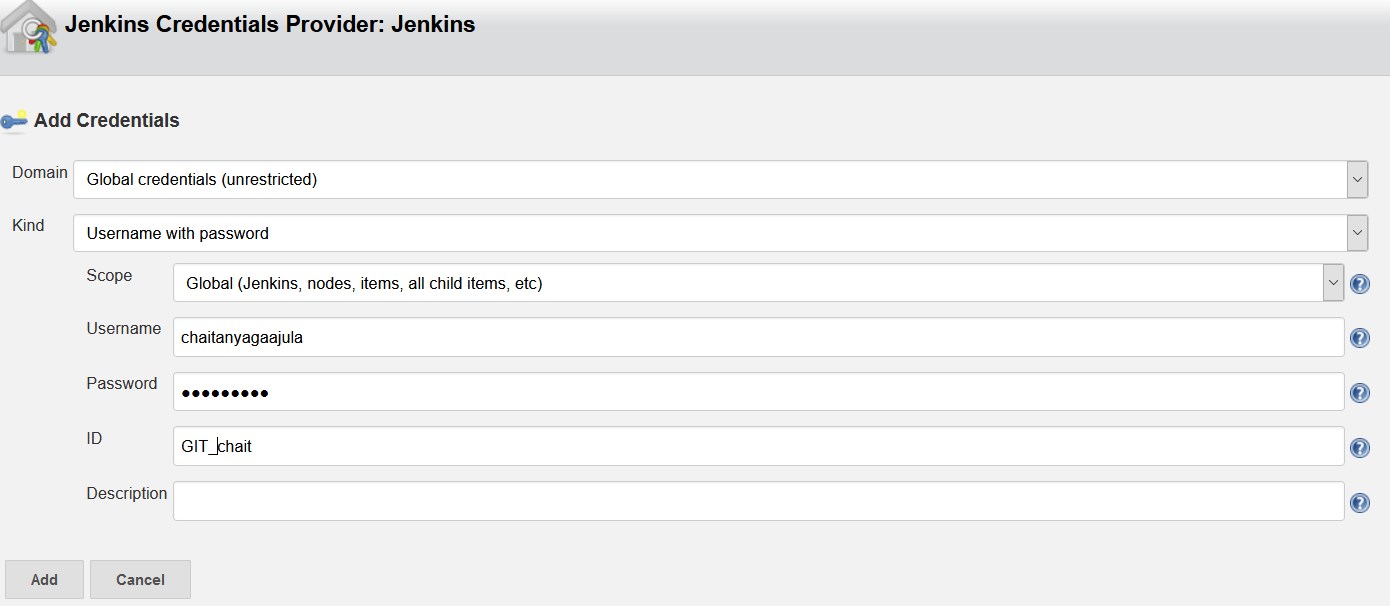
Under Pipeline Syntax (Groovy) do the below changes

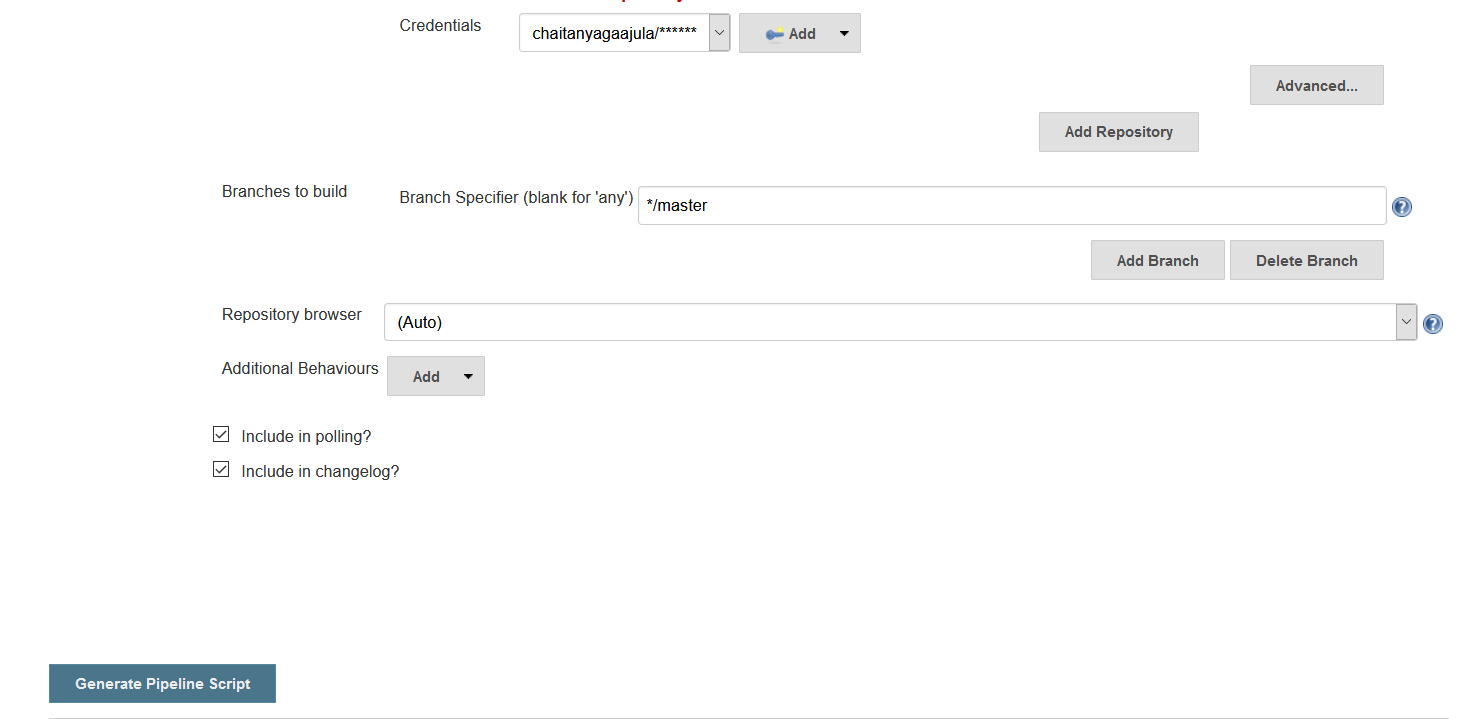
checkout: Check out from version control

**Once you enter all the details -Click on “Generate Pipeline Script’**





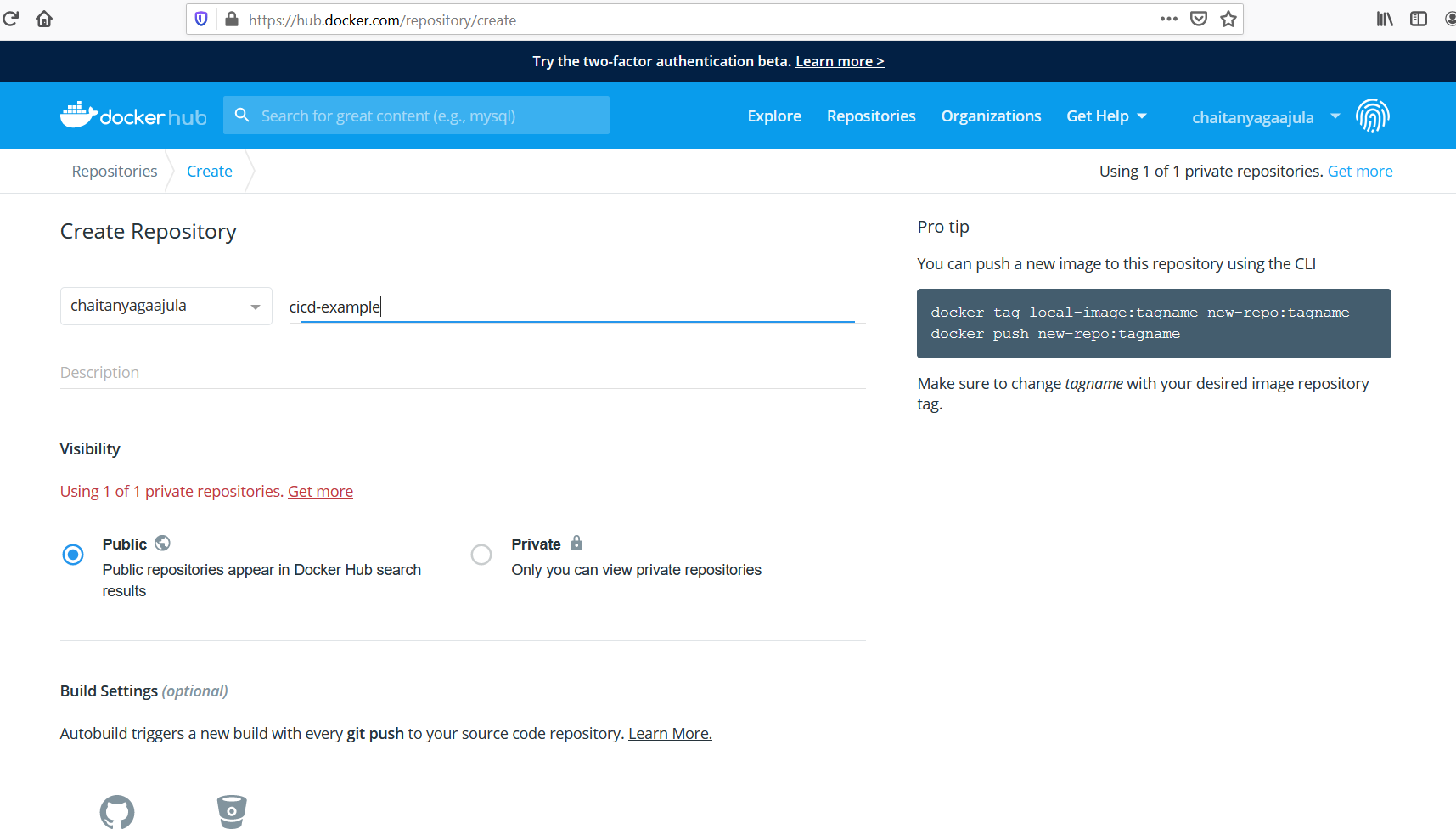


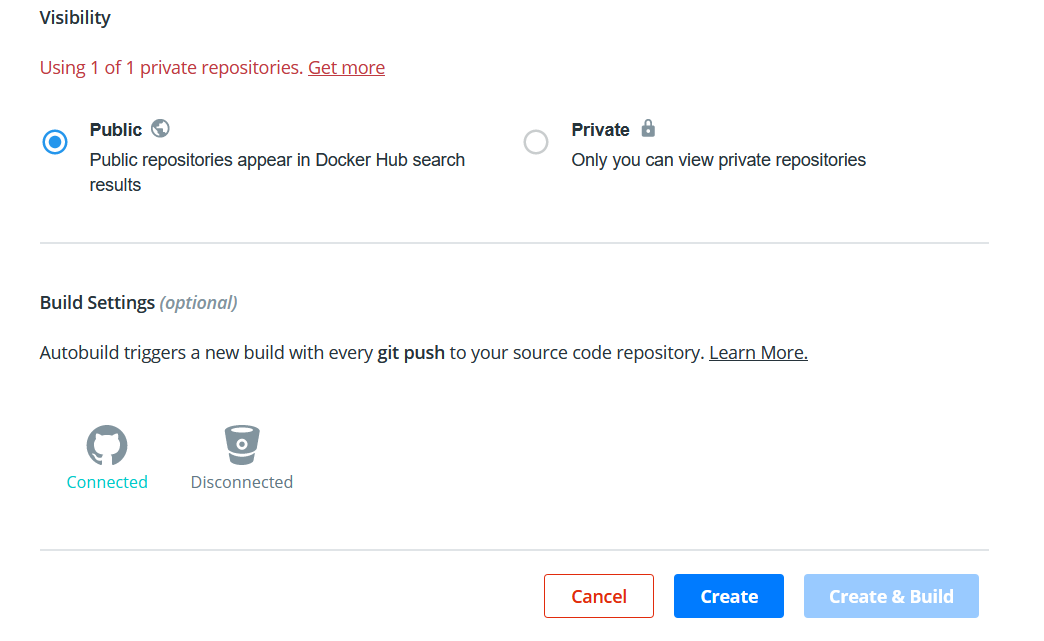


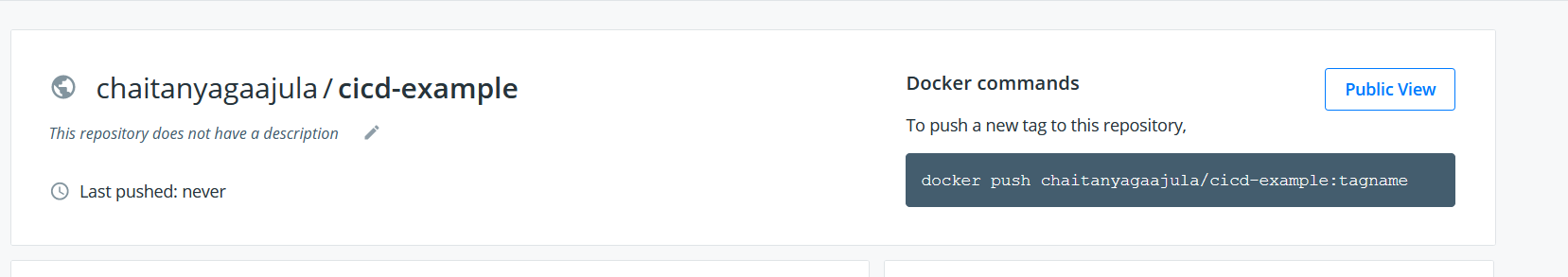
12. We get output - checkout([$class: 'GitSCM', branches: [[name: '\*/master']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[credentialsId: 'GIT\_chait', url: 'https://github.com/chaitanyagaajula/CICD.git']]])

Copy the contents to a notepad

13.Create a Public Dockerhub repository





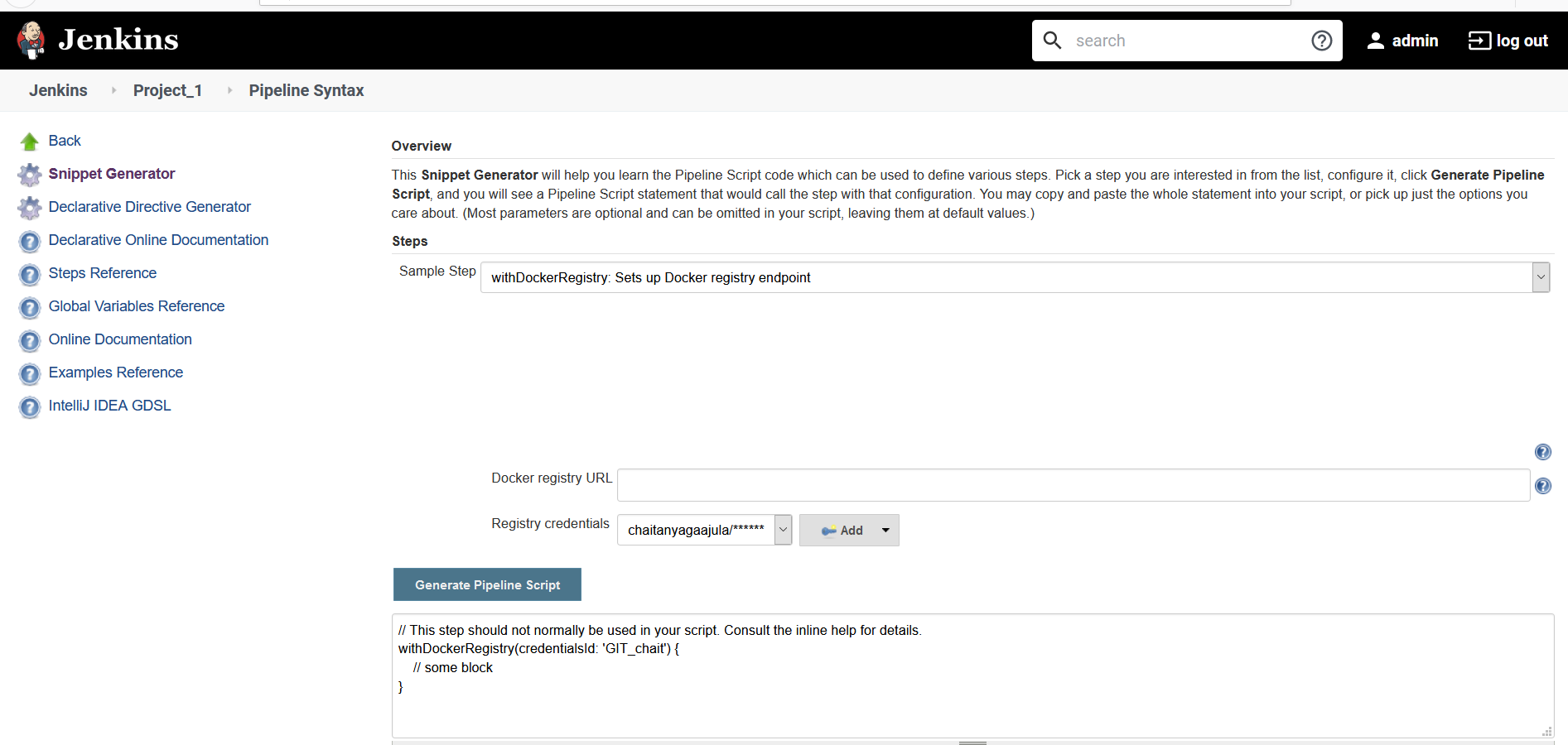


**14. To integrate Docker with Jenkins**

Under Pipeline Syntax (Groovy) do the below changes

**withDockerRegistry: Sets up Docker registry endpoint**

**Click on the button which says “Generate Pipeline script”**



**10. Add the o/p of both scm script o /p and docker script /opt in the below script**

Final Groovy script is as follows,

**pipeline {**

**agent any**

**stages {**

**stage('checkout\_Code\_integration') {**

**steps {**

**checkout([$class: 'GitSCM', branches: [[name: '\*/master']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRemoteConfigs: [[credentialsId: '65adad77-7981-400f-922b-74cc6ba0c88e', url: 'https://github.com/chaitanyagaajula/CICD.git']]])**

**}**

**}**

**stage('Unit\_Testing') {**

**steps {**

**sh "/usr/local/bin/pip install -r requirements.txt"**

**sh "/usr/bin/python -m pytest -v tests/test\_generator.py"**

**}**

**}**

**stage('Docker\_image\_Build') {**

**steps {**

**sh "/usr/bin/docker build -t chaitanyagaajula/cicd-example:latest ."**

**}**

**}**

**stage('publish') {**

**steps {**

**withDockerRegistry(credentialsId: '65adad77-7981-400f-922b-74cc6ba0c88e', url: 'https://index.docker.io/v1/') {**

**sh "/usr/bin/docker push chaitanyagaajula/cicd-example:latest"**

**}**

**}**

**}**

**stage('Running\_image\_from\_DockerHub') {**

**steps {**

**sh "/usr/bin/docker run -p 5000:5000 --rm chaitanyagaajula/cicd-example:latest"**

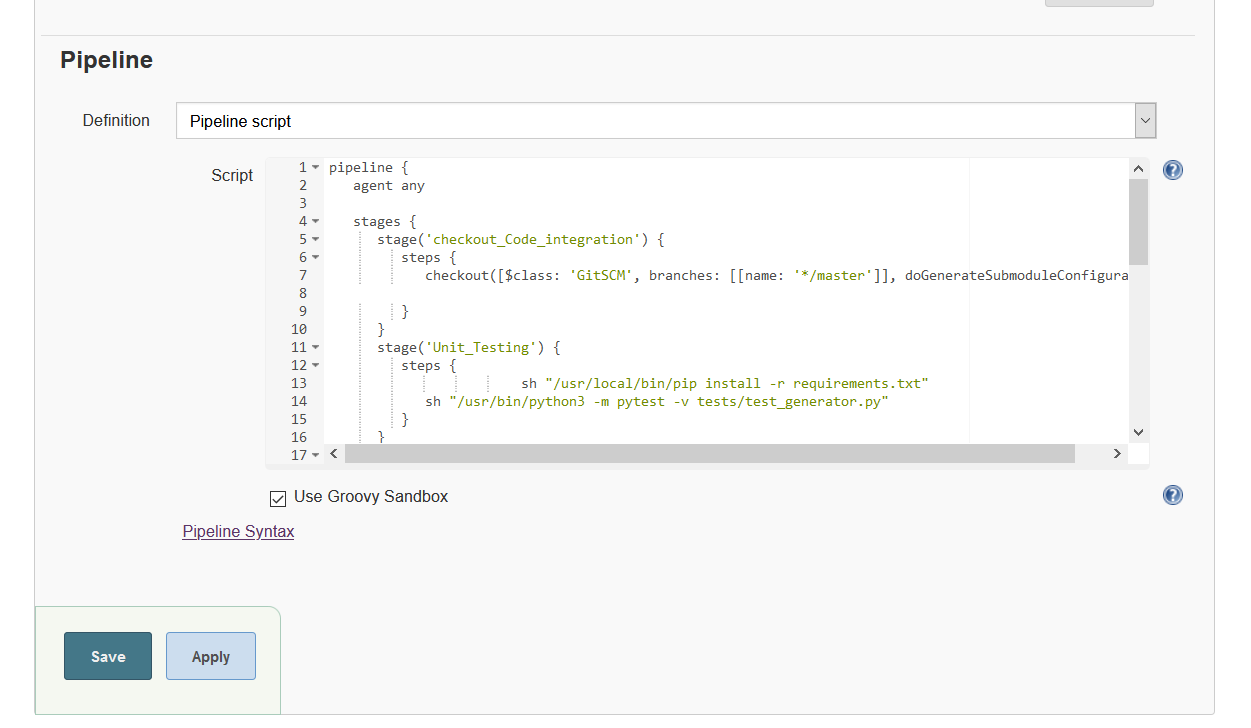
**}**

**}**

**}**

**}**

11) Copy the script and update the script under Pipeline Script , Apply and Save



12. On the Jenkins Console under ‘CICID\_Pipeline1’ , Click on Build now

####################################################

**Console output:**

Started by user unknown or anonymous

Running in Durability level: MAX\_SURVIVABILITY

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](http://3.84.145.85:8080/computer/(master)/) in /var/lib/jenkins/workspace/CICD\_Pipeline1

[Pipeline] {

[Pipeline] stage

[Pipeline] { (checkout\_Code\_integration)

[Pipeline] checkout

using credential 65adad77-7981-400f-922b-74cc6ba0c88e

> git rev-parse --is-inside-work-tree # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/chaitanyagaajula/CICD.git> # timeout=10

Fetching upstream changes from <https://github.com/chaitanyagaajula/CICD.git>

> git --version # timeout=10

using GIT\_ASKPASS to set credentials

> git fetch --tags --progress -- <https://github.com/chaitanyagaajula/CICD.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/master^{commit} # timeout=10

> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10

Checking out Revision d6762b7f5f695c9ed4d5faba6126e65c4763b520 (refs/remotes/origin/master)

> git config core.sparsecheckout # timeout=10

> git checkout -f d6762b7f5f695c9ed4d5faba6126e65c4763b520 # timeout=10

Commit message: " commiting jenkisn file"

> git rev-list --no-walk d6762b7f5f695c9ed4d5faba6126e65c4763b520 # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Unit\_Testing)

[Pipeline] sh

+ /usr/local/bin/pip install -r requirements.txt

DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip, can be found at <https://pip.pypa.io/en/latest/development/release-process/#python-2-support>

Defaulting to user installation because normal site-packages is not writeable

Requirement already satisfied: pytest==3.0.6 in /var/lib/jenkins/.local/lib/python2.7/site-packages (from -r requirements.txt (line 1)) (3.0.6)

Requirement already satisfied: Flask==0.12 in /var/lib/jenkins/.local/lib/python2.7/site-packages (from -r requirements.txt (line 2)) (0.12)

Requirement already satisfied: py>=1.4.29 in /var/lib/jenkins/.local/lib/python2.7/site-packages (from pytest==3.0.6->-r requirements.txt (line 1)) (1.8.1)

Requirement already satisfied: setuptools in /usr/lib/python2.7/dist-packages (from pytest==3.0.6->-r requirements.txt (line 1)) (36.2.7)

Requirement already satisfied: Jinja2>=2.4 in /usr/lib/python2.7/dist-packages (from Flask==0.12->-r requirements.txt (line 2)) (2.7.2)

Requirement already satisfied: click>=2.0 in /var/lib/jenkins/.local/lib/python2.7/site-packages (from Flask==0.12->-r requirements.txt (line 2)) (7.1.2)

Requirement already satisfied: itsdangerous>=0.21 in /var/lib/jenkins/.local/lib/python2.7/site-packages (from Flask==0.12->-r requirements.txt (line 2)) (1.1.0)

Requirement already satisfied: Werkzeug>=0.7 in /var/lib/jenkins/.local/lib/python2.7/site-packages (from Flask==0.12->-r requirements.txt (line 2)) (1.0.1)

Requirement already satisfied: markupsafe in /usr/lib64/python2.7/dist-packages (from Jinja2>=2.4->Flask==0.12->-r requirements.txt (line 2)) (0.11)

[Pipeline] sh

+ /usr/bin/python -m pytest -v tests/test\_generator.py

============================= test session starts ==============================

platform linux2 -- Python 2.7.16, pytest-3.0.6, py-1.8.1, pluggy-0.4.0 -- /usr/bin/python

cachedir: .cache

rootdir: /var/lib/jenkins/workspace/CICD\_Pipeline1, inifile:

collecting ... collected 3 items

tests/test\_generator.py::test\_sample\_single\_word PASSED

tests/test\_generator.py::test\_sample\_multiple\_words PASSED

tests/test\_generator.py::test\_generate\_buzz\_of\_at\_least\_five\_words PASSED

=========================== 3 passed in 0.03 seconds ===========================

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Docker\_image\_Build)

[Pipeline] sh

+ /usr/bin/docker build -t chaitanyagaajula/cicd-example:latest .

Sending build context to Docker daemon 131.1kB

Step 1/7 : FROM alpine:3.5

---> f80194ae2e0c

Step 2/7 : RUN apk add --update python py-pip

---> Using cache

---> 7eed49c67413

Step 3/7 : COPY requirements.txt /src/requirements.txt

---> Using cache

---> ef60a458c1c6

Step 4/7 : RUN pip install -r /src/requirements.txt

---> Using cache

---> 4abc0fb62417

Step 5/7 : COPY app.py /src

---> Using cache

---> 372034b4377c

Step 6/7 : COPY buzz /src/buzz

---> Using cache

---> 0106906d73ba

Step 7/7 : CMD python /src/app.py

---> Using cache

---> 754bf7b3b823

Successfully built 754bf7b3b823

Successfully tagged chaitanyagaajula/cicd-example:latest

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (publish)

[Pipeline] withDockerRegistry

$ docker login -u chaitanyagaajula -p \*\*\*\*\*\*\*\* <https://index.docker.io/v1/>

WARNING! Using --password via the CLI is insecure. Use --password-stdin.

WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/CICD\_Pipeline1@tmp/a298b061-0c63-4edd-a8bd-f2b1f18d1c6e/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

[Pipeline] {

[Pipeline] sh

+ /usr/bin/docker push chaitanyagaajula/cicd-example:latest

The push refers to repository [docker.io/chaitanyagaajula/cicd-example]

fe03a7d948c5: Preparing

6897df7e7fa4: Preparing

94b66e97aa78: Preparing

913bbf975077: Preparing

4d2356b10af6: Preparing

f566c57e6f2d: Preparing

f566c57e6f2d: Waiting

4d2356b10af6: Layer already exists

6897df7e7fa4: Layer already exists

fe03a7d948c5: Layer already exists

913bbf975077: Layer already exists

94b66e97aa78: Layer already exists

f566c57e6f2d: Layer already exists

latest: digest: sha256:2a1a8365cd9d02c442261a07d20738c3b30b6ec38a714b68117241f010fe837d size: 1573

[Pipeline] }

[Pipeline] // withDockerRegistry

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Running\_image\_from\_DockerHub)

[Pipeline] sh

+ /usr/bin/docker run -p 5000:5000 --rm chaitanyagaajula/cicd-example:latest

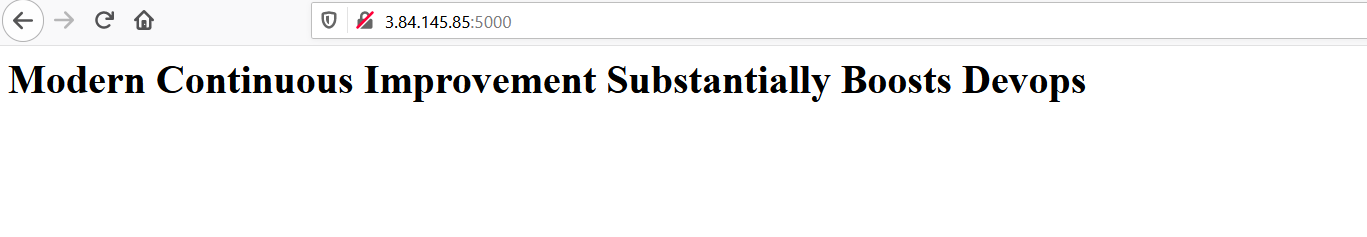
\* Running on <http://0.0.0.0:5000/> (Press CTRL+C to quit)



**Output:**

Now we can see the output using the URL:

http://3.84.145.85:5000/



And refresh the page,



**Stopping the application:**

[root@ip-172-31-85-119 CICD]# docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

4db6f2bce9c2 chaitanyagaajula/cicd-example:latest "/bin/sh -c 'python …" 2 minutes ago Up 2 minutes 0.0.0.0:5000->5000/tcp vibrant\_dewdney

[root@ip-172-31-85-119 CICD]# docker stop vibrant\_dewdney

vibrant\_dewdney

[root@ip-172-31-85-119 CICD]#