# Documentation of Booster_CI_CD_Project

Create CI/CD pipeline using jenkinsfile to deploy simple django web app as a microservice running on docker container locally

1) Install Jenkins, create Jenkins image and execute it

```
engysamy@Engys-MacBook-Pro Desktop % cd project
engysamy@Engys-MacBook-Pro project % docker run -p 8080:8080 -p 50000:50000 -d jenkins/jenkins:lts
74095f3e51a7933da2e3d0124b7eda21d4830139d9272e4447920e7e6d466a79
engysamy@Engys-MacBook-Pro project % docker ps
CONTAINER ID   IMAGE              COMMAND             CREATED          STATUS          PORTS                                              NAM
ES
74095f3e51a7   jenkins/jenkins:lts   "/sbin/tini -- /usr/…"   8 seconds ago    Up 8 seconds    0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp   lov
ing_snyder
engysamy@Engys-MacBook-Pro project % docker exec -it 74095f3e51a7 bash
```

2) Start to configure slave node

   A. ) First Build slave Docker file

slave_Dockerfile

```
FROM ubuntu
USER root
RUN apt-get update -qq
RUN mkdir -p jenkins_home
RUN chmod 777 jenkins_home
RUN useradd -ms /bin/bash jenkins

RUN apt-get install -qqy apt-transport-https ca-certificates curl gnupg2 software-properties-common
RUN curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -
RUN add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    focal\
    stable"
RUN apt-get update  -qq
RUN apt-get install -y docker-ce docker-ce-cli containerd.io
RUN usermod -aG docker jenkins

RUN apt-get update -qq
RUN apt-get -y install python3.6
RUN apt-get -qqy install python3-pip
RUN apt-get install openjdk-8-jdk -qq
RUN apt-get install openssh-server -qq

USER jenkins
WORKDIR jenkins_home
CMD ["/bin/bash"]
```

```
engysamy@Engys-MacBook-Pro project % docker build -f slave_Dockerfile . -t slave
Sending build context to Docker daemon  4.608kB
```

B. ) Run slave image

```
engysamy@Engys-MacBook-Pro project % docker run  -dit -v /var/run/docker.sock:/var/run/docker.sock slave
828fd6faa9abaed7066dfd66713fa9f75a93f73b9a8a961864b88495b631ada2
```

C. ) Execute image to show Remote root directory : /jenkins_home

```
engysamy@Engys-MacBook-Pro project % docker ps
CONTAINER ID       IMAGE               COMMAND                  CREATED           STATUS            PORTS
ES
828fd6faa9ab       slave               "/bin/bash"              5 seconds ago     Up 4 seconds
tifying_thompson
74095f3e51a7       jenkins/jenkins:lts   "/sbin/tini -- /usr/…"   5 minutes ago     Up 5 minutes      0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp
ing_snyder
engysamy@Engys-MacBook-Pro project % docker exec -it 828fd6faa9ab bash
jenkins@828fd6faa9ab:/jenkins_home$ cd ~
```

D. ) Generate key to Create credential

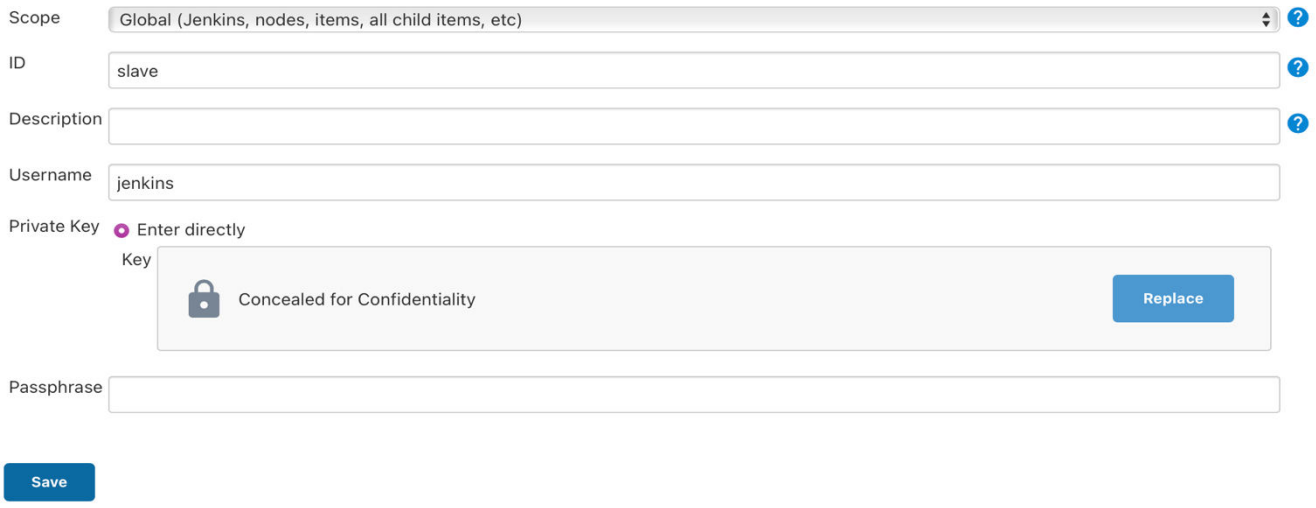
```
Last login: Sun Sep  6 12:43:16 on ttys001
engysamy@Engys-MacBook-Pro ~ % ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/engysamy/.ssh/id_rsa):
/Users/engysamy/.ssh/id_rsa already exists.
Overwrite (y/n)?
engysamy@Engys-MacBook-Pro ~ % cd ~/.ssh
engysamy@Engys-MacBook-Pro .ssh % ls
id_rsa          id_rsa.pub
engysamy@Engys-MacBook-Pro .ssh % rm *
zsh: sure you want to delete all 2 files in /Users/engysamy/.ssh [yn]? y
engysamy@Engys-MacBook-Pro .ssh % ls
engysamy@Engys-MacBook-Pro .ssh % ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/engysamy/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/engysamy/.ssh/id_rsa.
Your public key has been saved in /Users/engysamy/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:uwNGqW0pgU8TBTV3IYCqU8/ur8PUix+28PaUpYAhhaU engysamy@Engys-MacBook-Pro.local
The key's randomart image is:
+---[RSA 3072]----+
|   o+==.o o.     |
|  .oo  o o       |
|  Eo.. .         |
|  +.+oo          |
| o =.B..S .      |
|o   B B. =       |
| . +.=oo=        |
|    =+o+..       |
|   .o*=.o.       |
+----[SHA256]-----+
engysamy@Engys-MacBook-Pro .ssh % ls
id_rsa          id_rsa.pub
engysamy@Engys-MacBook-Pro .ssh % cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCtj8bLGDxWgS8oqTzUTX2wfR9LeruZWdjaUZUCwsT99EI+cdXjz9/9aZkMFmxpcaDvev3yLOhJ5R75/dGUL9GTELwVhhwS2KJJYTQV8kLDy5wP2JBbowFM9C
/qOSvfHpwPImsaemKG4i7EfPQL33uk0jQOKP/qSw8bdmcBVaXQ7ZpvbrPNNqzJvdZ/hs2QV0suVdAkpaPieXxdKUmAYdviJ23WzaPfkt+QnJXS9Jt9R4ehVxziniQmXPH0fiTSrZMpWHlUhOcnjCIU/L1hWDBt
gheiZlhXz/QLapIaYzUkLrOdROUvACT3tKEe3HnlE3AOrSfqHlbkbb+0WQ3UNNXLQMsWgLtH0QMAD315ziCF4D+mLUo0Tz9b1t5prmVEQECTkcCLPMjN9KqO7rV+OQysX98wriEgcDiRWF8Sql0= engysamy@
engysamy@Engys-MacBook-Pro .ssh % cat id_rsa
```

E. ) Echo public key in authorized_keys file in slave image

```
jenkins@828fd6faa9ab:/jenkins_home$ cd ~
jenkins@828fd6faa9ab:~$ mkdir .ssh
jenkins@828fd6faa9ab:~$ cd .ssh/
jenkins@828fd6faa9ab:~/.ssh$ ls
jenkins@828fd6faa9ab:~/.ssh$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCtj8bLGDxWgS8oqTzUTX2wfR9LeruZWdjaUZUCwsT99EI+cdXjz9/9aZkMFmxpcaDvev3yLOhJ5
hwS2KJJYTQV8kLDy5wP2JBbowFM9OCiW8mStQTG8dDzugDvKGhU70SXYsH3jREtEB1AOgYFiN/qOSvfHpwPImsaemKG4i7EfPQL33uk0jQOKP/qSw8bdmcBVaXQ7ZpvbrPNNqzJvdZ/hs2QV0suV
dviJ23WzaPfkt+QnJXS9Jt9R4ehVxziniQmXPH0fiTSrZMpWHlUhOcnjCIU/L1hWDBfUJNedW9zrL51Y/FvzRvaBsdkjfMH20qDPzkr4tc5rF7pgheiZlhXz/QLapIaYzUkLrOdROUvACT3tKEe3
b+0WQ3UNNXLQMsWgLtH0QMAD315ziCF4D+mLUo0Tz9b1t5prmVEQECTkcCLPMjN9KqO7rV+OQysX98wriEgcDiRWF8Sql0= engysamy@Engys-MacBook-Pro.local" > authorized_keys
jenkins@828fd6faa9ab:~/.ssh$ ls
authorized_keys
jenkins@828fd6faa9ab:~/.ssh$ exit
```

F. ) Cat the private key to make credential. In Jenkins, make type of credential ssh with Private key

```
engysamy@Engys-MacBook-Pro .ssh % cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEArY/Gyxg8VoEvKKk81E19sH0fS3q7mVnY2lGVAsLE/fRCPnHV48/f
/WmZDBZsaXGq73r88izoSeUe+f3RlC/RkxC8FYYcFtjiSWF0Ef1Cw8ucD9iOW6MBIRIqol
```



G. ) Bring host which is the ipv4 of mystifying_thamson (172.17.0.3)

```
engysamy@Engys-MacBook-Pro .ssh % docker inspect bridge
[
    {
        "Name": "bridge",
        "Id": "2b12264972c927496ea62cba4d2cae58252e123a31d35af76becefb20adff7f0",
        "Created": "2020-09-06T10:43:21.952531714Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "74095f3e51a7933da2e3d0124b7eda21d4830139d9272e4447920e7e6d466a79": {
                "Name": "loving_snyder",
                "EndpointID": "c4280b9343b14e07045d43df2774365c9b73a0526ea9941852c4aca2de7ed9d2",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            },
            "828fd6faa9abaed7066dfd66713fa9f75a93f73b9a8a961864b88495b631ada2": {
                "Name": "mystifying_thompson",
                "EndpointID": "1bdbbdfd5c806d2cbaa59ba6f9216489b8a560e609b819f27641dd5450e06711",
                "MacAddress": "02:42:ac:11:00:03",
                "IPv4Address": "172.17.0.3/16",
                "IPv6Address": ""
```

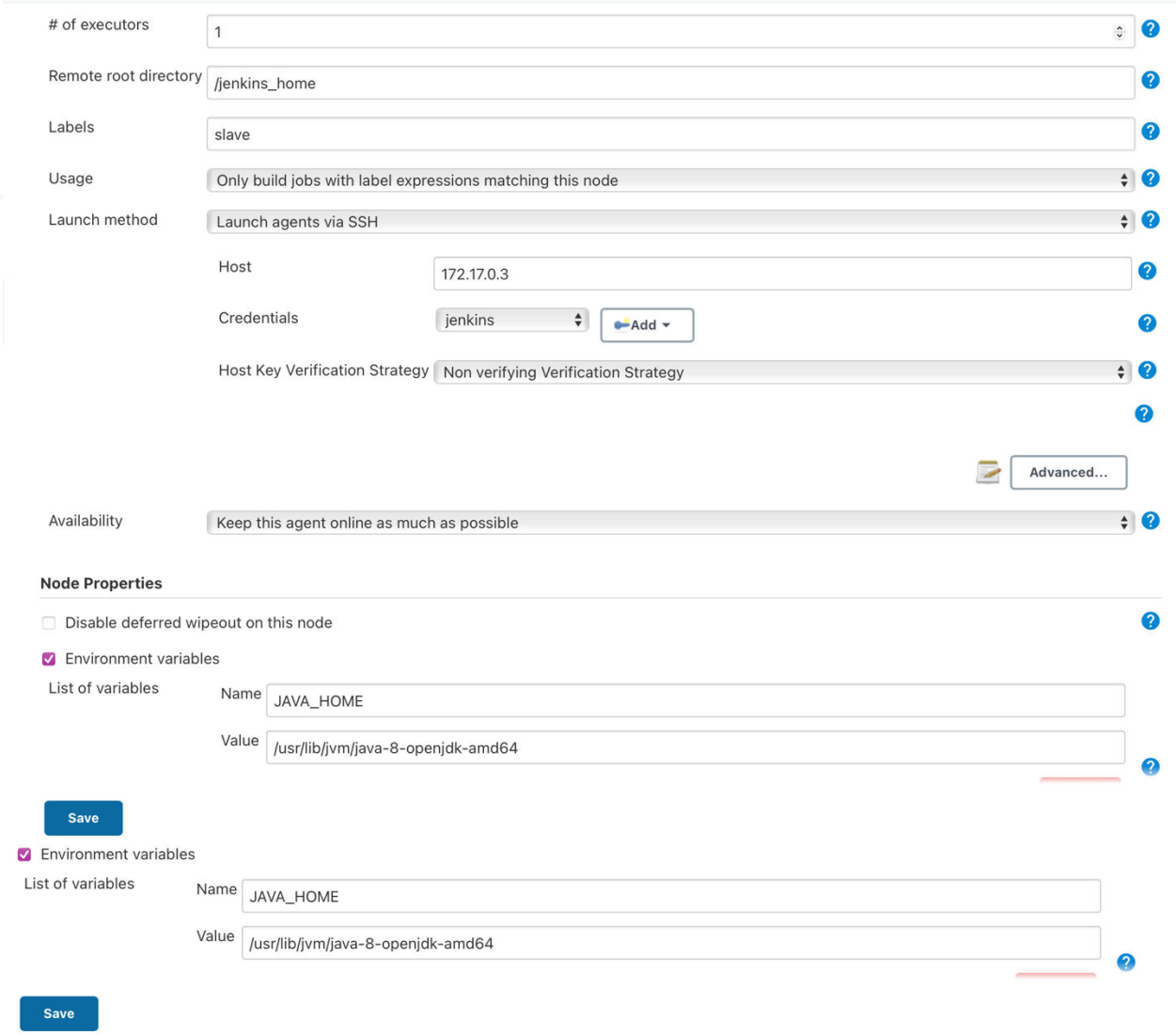
H. ) Bring path of the environment variable java home and start the service ssh

```
root@828fd6faa9ab:/jenkins_home# cd /usr/lib
root@828fd6faa9ab:/usr/lib# cd jvm
root@828fd6faa9ab:/usr/lib/jvm# ls
java-1.8.0-openjdk-amd64  java-8-openjdk-amd64
root@828fd6faa9ab:/usr/lib/jvm# exit
exit
engysamy@Engys-MacBook-Pro ~ % docker exec -it -u root 828fd6faa9ab bash
root@828fd6faa9ab:/jenkins_home# service ssh start
 * Starting OpenBSD Secure Shell server sshd
root@828fd6faa9ab:/jenkins_home# exit
exit
```

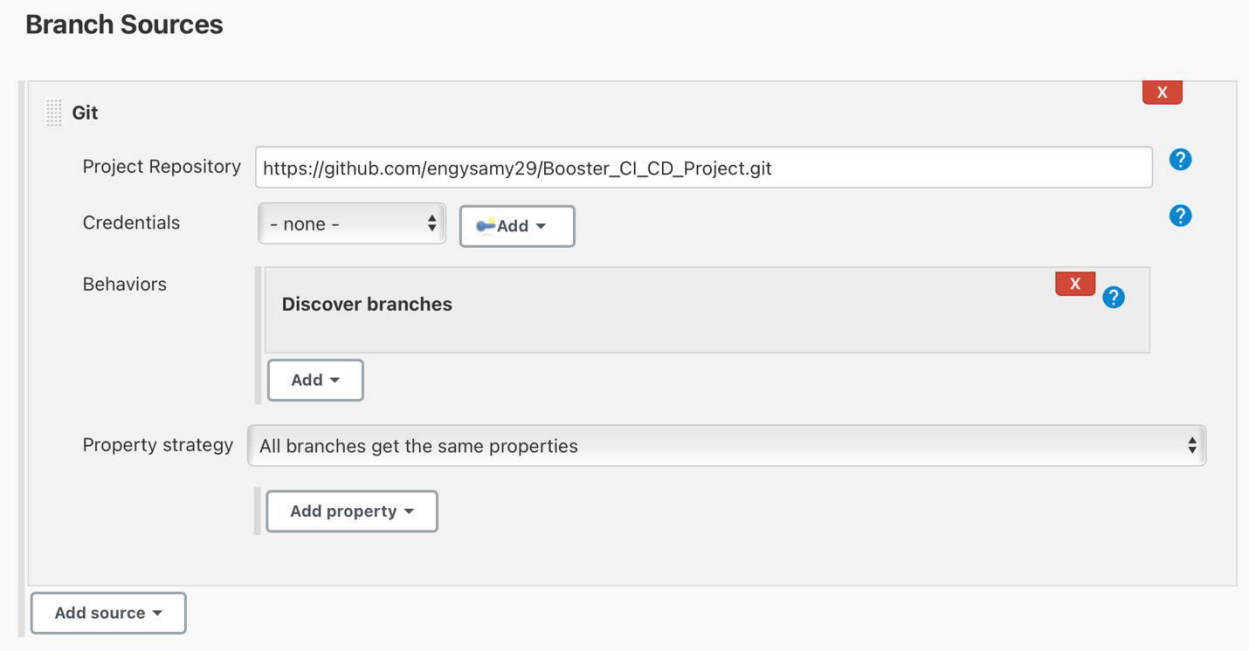
I. ) Change mod of the image to have permission

```
engysamy@Engys-MacBook-Pro project % docker exec -it -u root 828fd6faa9ab bash
root@828fd6faa9ab:/jenkins_home# chmod 777 /var/run/docker.sock
```

J. ) final look of slave node



3) Make multi-branch pipeline called project




4) Run Master Branch

## Console Output

```
Started by user engy
 > git rev-parse --is-inside-work-tree # timeout=10
Setting origin to https://github.com/engysamy29/Booster_CI_CD_Project.git
 > git config remote.origin.url https://github.com/engysamy29/Booster_CI_CD_Project.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git --version # 'git version 2.11.0'
 > git config --get remote.origin.url # timeout=10
 > git fetch --tags --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/dev
Seen branch in repository origin/master
Seen 2 remote branches
Obtained Jenkinsfile from acfb2077a7bc913660cd369bcb61fece75d6a0d1
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on slave in /jenkins_home/workspace/project_master
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
Fetching changes from the remote Git repository
Fetching without tags
 > git rev-parse --is-inside-work-tree # timeout=10
 > git config remote.origin.url https://github.com/engysamy29/Booster_CI_CD_Project.git # timeout=10
```

```
24a7c2c9bc62: Waiting
4b562a21ac3e: Pushed
f77249ed9761: Pushed
b58a9f7974a2: Pushed
24a7c2c9bc62: Layer already exists
e126d63f0afe: Pushed
46a6fbcfe6ac: Layer already exists
001e4a80973b: Layer already exists
2ba5b91ca2b0: Layer already exists
2f37d1102187: Layer already exists
79bde4d54386: Layer already exists
9dce532948fc: Pushed
2ebc4148c469: Pushed
v1.0: digest: sha256:f5ee5430ea299a7af83961db075d3382fc95473c383c951fbbb73f4cb807495d size: 2833
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] sh
+ docker run -d -p 5000:5000 engysamy/dajngoo:v1.0
2ea364c5f93e2fc2c3f96f76c439062f6d237fe142ffa5a8d31ef2ac367d424a
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

5) Run Dev branch

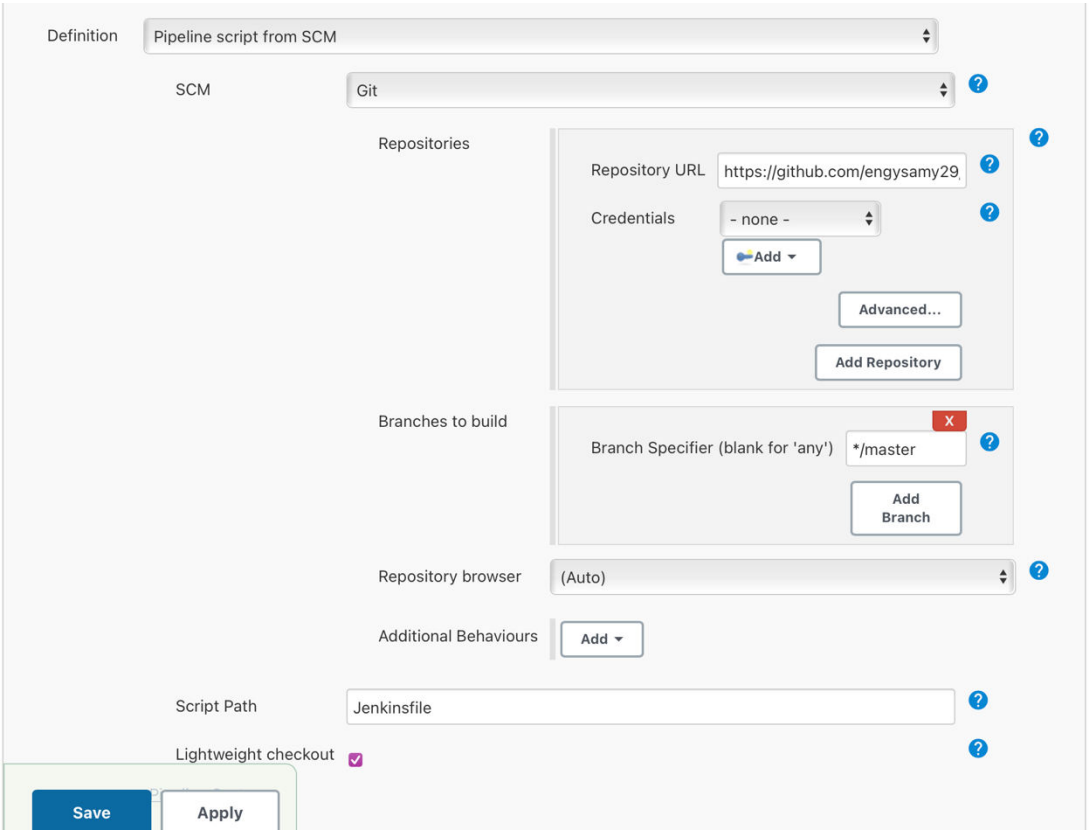
## Console Output

```
Started by user engy
 > git rev-parse --is-inside-work-tree # timeout=10
Setting origin to https://github.com/engysamy29/Booster_CI_CD_Project.git
 > git config remote.origin.url https://github.com/engysamy29/Booster_CI_CD_Project.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git --version # 'git version 2.11.0'
 > git config --get remote.origin.url # timeout=10
 > git fetch --tags --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/dev
Seen branch in repository origin/master
Seen 2 remote branches
Obtained Jenkinsfile from cd21f89170c39dcf14a9661ef96bbde05cf396d5
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on slave in /jenkins_home/workspace/project_dev
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
Fetching changes from the remote Git repository
Fetching without tags
 > git rev-parse --is-inside-work-tree # timeout=10
 > git config remote.origin.url https://github.com/engysamy29/Booster_CI_CD_Project.git # timeout=10
Fetching upstream changes from https://github.com/engysamy29/Booster_CI_CD_Project.git
```

```
79bde4d54386: Waiting
80ae5e1eccd8: Waiting
6d32133e8ce1: Pushed
24d470c6cc7c: Pushed
cdec99512f73: Pushed
5f1ac4f2cede: Pushed
24a7c2c9bc62: Layer already exists
46a6fbcfe6ac: Layer already exists
001e4a80973b: Layer already exists
2ba5b91ca2b0: Layer already exists
2f37d1102187: Layer already exists
79bde4d54386: Layer already exists
80ae5e1eccd8: Pushed
940ee442a0c1: Pushed
v1.0: digest: sha256:4eacd95d1fadc55b96dbfde5f011671384f899fac5043fa4279a9263ec4a7ea7 size: 2835
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] sh
+ docker run -d -p 6000:3000 engysamy/dajngoo_dev:v1.0
7b48c364a1a62372f352943f87e1e2f023e8af6ff89b0b9a2d080a2223674fa4
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

6) Message sent to slack when build start

- I have done a single pipeline called pip1 to run master branch in git hub




Output:

```
79bde4d54386: Preparing
93e76ce3645b: Waiting
24a7c2c9bc62: Waiting
46a6fbcfe6ac: Waiting
001e4a80973b: Waiting
2ba5b91ca2b0: Waiting
2f37d1102187: Waiting
79bde4d54386: Waiting
2da53b84b4c4: Pushed
478b51a7f7e9: Pushed
a7563be517f2: Pushed
a1268997915d: Pushed
24a7c2c9bc62: Layer already exists
46a6fbcfe6ac: Layer already exists
001e4a80973b: Layer already exists
2ba5b91ca2b0: Layer already exists
2f37d1102187: Layer already exists
79bde4d54386: Layer already exists
93e76ce3645b: Pushed
3709d90e6f68: Pushed
v1.0: digest: sha256:56626cf684a1298dd2ea02f7dc0a48dada66db2d8f7eaa9b72da878d3fa21ebd size: 2833
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] sh
+ docker run -d -p 7000:3000 engysamy/dajngoo:v1.0
12d0000536dc70317a56fd68c14faed2103549b652fe1fd0fec9d40c3d30e7e7
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```