INTEGRATION OF JENKINS WITH DOCKER

STEP 1: Install Docker In Our linux System using following commands

- sudo apt update
- sudo apt install docker.io
- sudo chmod 666 /var/run/docker.sock
- sudo systemctl start docker
- sudo systemctl status docker

```
DevOpsVm4@DevOpsVm4:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:5 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease
Ign:6 https://jfrog.bintray.com/artifactory-debs xenial InRelease
Get:7 https://jfrog.bintray.com/artifactory-debs xenial Release [2,665 B]
Hit:7 https://jfrog.bintray.com/artifactory-debs xenial Release
Ign:9 http://pkg.jenkins-ci.org/debian binary/ InRelease
```

```
DevOpsVm4@DevOpsVm4:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd pigz runc ubuntu-fan
Suggested packages:
  mountall aufs-tools debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker.io pigz runc ubuntu-fan
0 upgraded, 7 newly installed, 0 to remove and 3 not upgraded.
Need to get 52.2 MB of archives.
After this operation, 257 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu xenial/universe amd64 pigz amd64 2.
3.1-2 [61.1 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu xenial/main amd64 bridge-utils amd6
4 1.5-9ubuntu1 [28.6 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu xenial/universe amd64 cgroupfs-moun
t all 1.2 [4,970 B]
Get:4 http://azure.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 runc
amd64 1.0.0~rc7+git20190403.029124da-0ubuntu1~16.04.4 [1,890 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 conta
inerd amd64 1.2.6-0ubuntu1~16.04.3 [19.7 MB]
```

```
DevOpsVm4@DevOpsVm4:~$ sudo systemctl start docker
DevOpsVm4@DevOpsVm4:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Tue 2020-04-28 09:19:18 UTC; 43s ago
     Docs: https://docs.docker.com
 Main PID: 5207 (dockerd)
   CGroup: /system.slice/docker.service
           └─5207 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain

Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.070831643Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.385730701Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.477273750Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.482832365Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.634451378Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.634853979Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.635095480Z" l
Apr 28 09:19:18 DevOpsVm4 dockerd[5207]: time="2020-04-28T09:19:18.698187952Z" l
Apr 28 09:19:18 DevOpsVm4 systemd[1]: Started Docker Application Container Engin
Apr 28 09:19:50 DevOpsVm4 systemd[1]: Started Docker Application Container Engin
```

```
DevOpsVm4@DevOpsVm4:~$ sudo chmod 666 /var/run/docker.sock
DevOpsVm4@DevOpsVm4:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Tue 2020-04-28 09:19:18 UTC; 2min 7s ago
     Docs: https://docs.docker.com
 Main PID: 5207 (dockerd)
   CGroup: /system.slice/docker.service
           └─5207 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain
```

STEP 2:

Now Create a account in DockerHub using below links

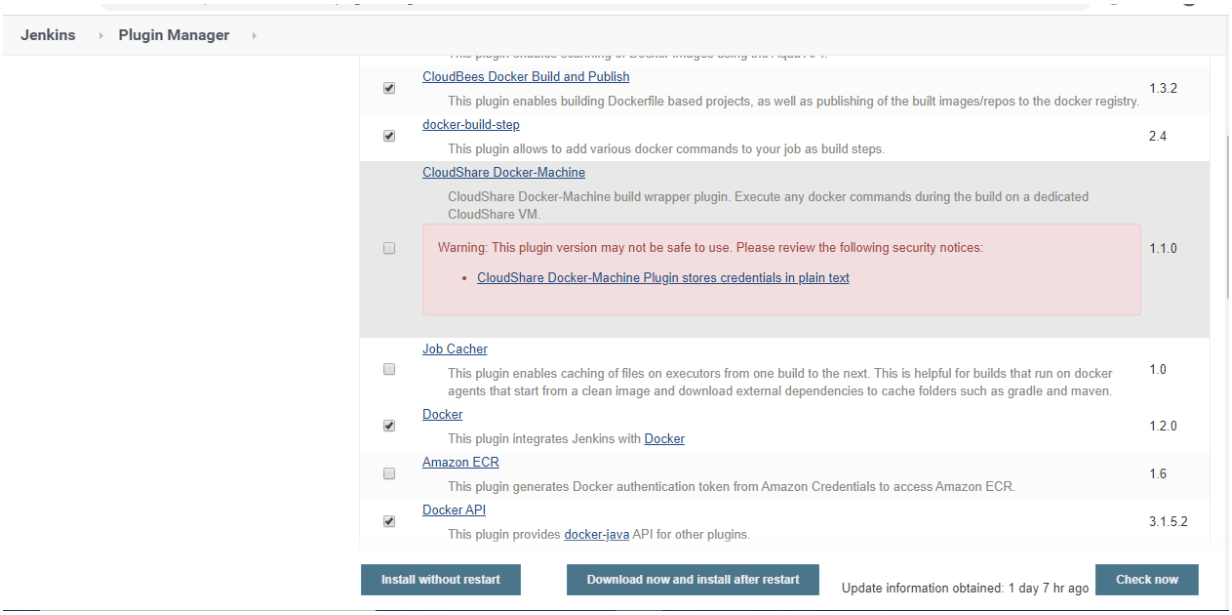
- https://labs.play-with-docker.com/

STEP 3: Now In your Github, create one repository add push maven project make sure that project contain target folder in that .jar file must be present. After that add one docker file in the same repository.
Add this in the docker file.

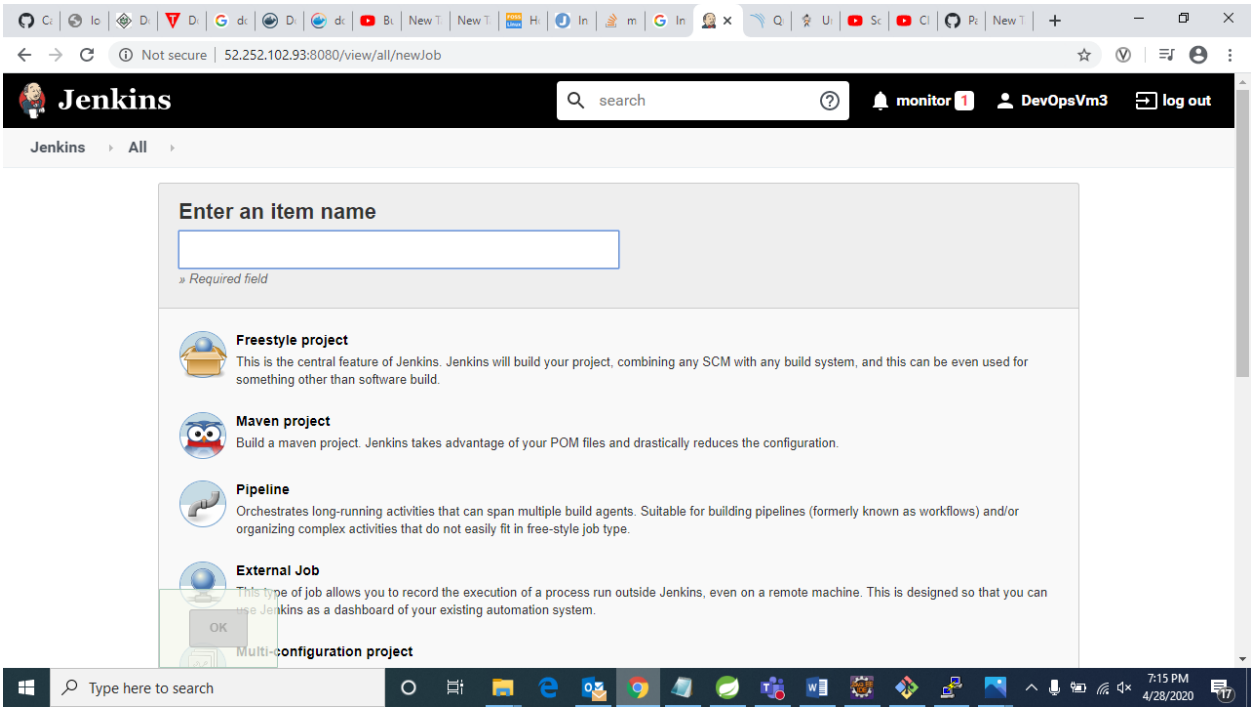
```
FROM
openjdk:8
        EXPOSE 8080
        ADD target/employeemanagement-0.0.1-SNAPSHOT-jar-with-
        dependencies.jar employeemanagement-0.0.1-SNAPSHOT-jar-
        with-dependencies.jar
        ENTRYPOINT ["java","-jar","/employeemanagement-0.0.1-
        SNAPSHOT-jar-with-dependencies.jar"]
```

STEP 4:

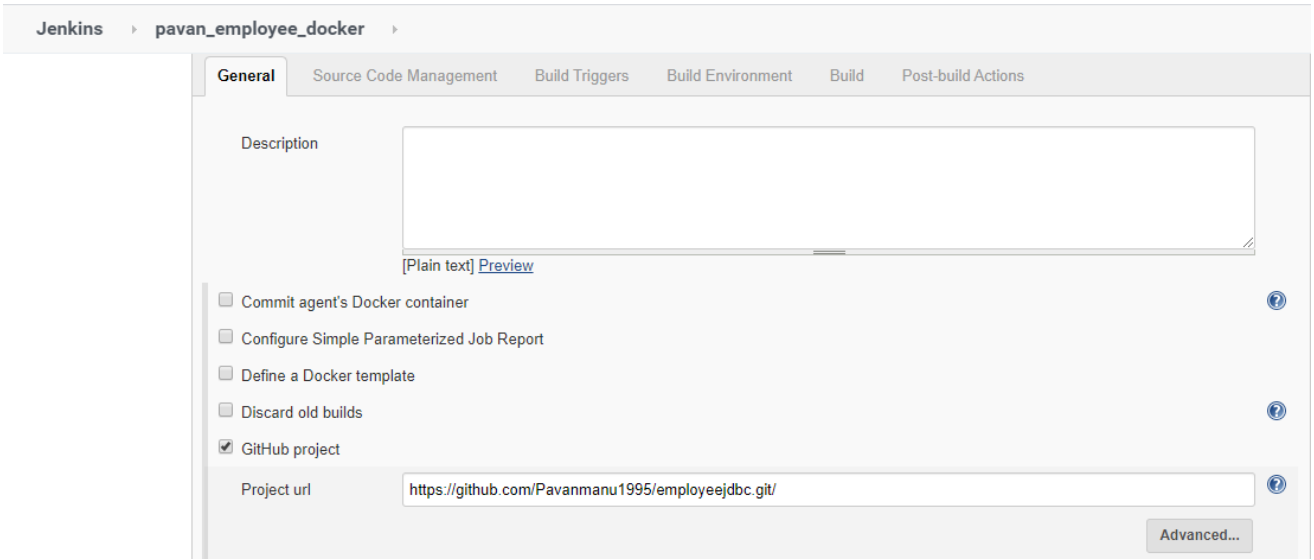
Now Open Jenkins → manage plugins → available → download required plugins.

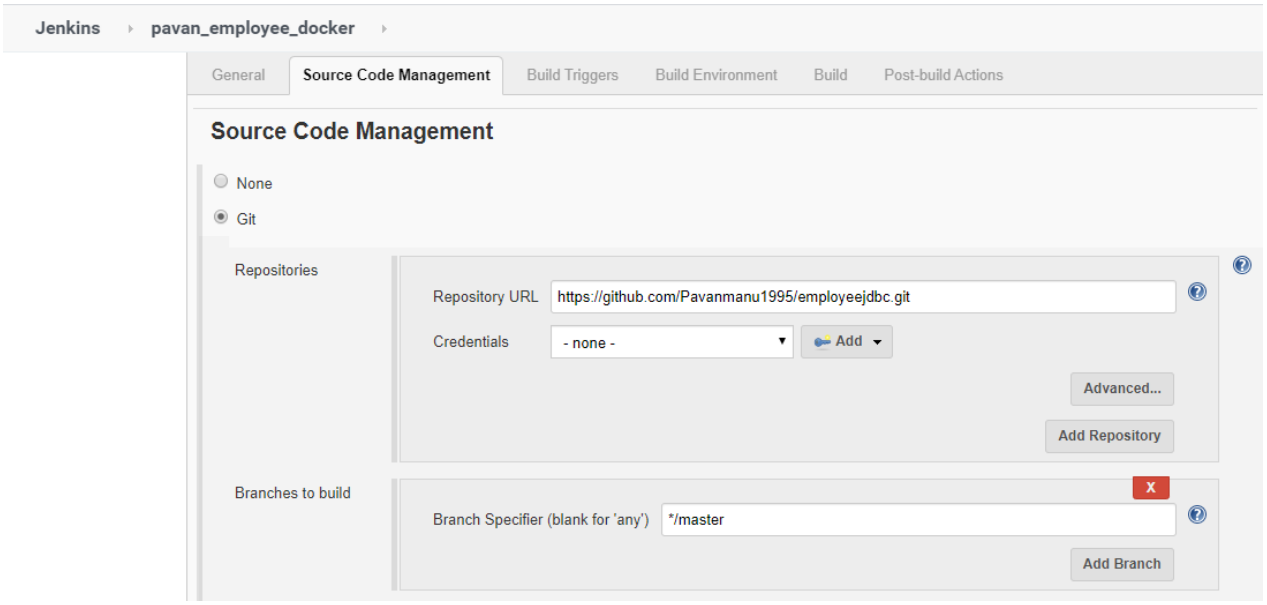


STEP 5: Now create new item using freestyle project and configure project.

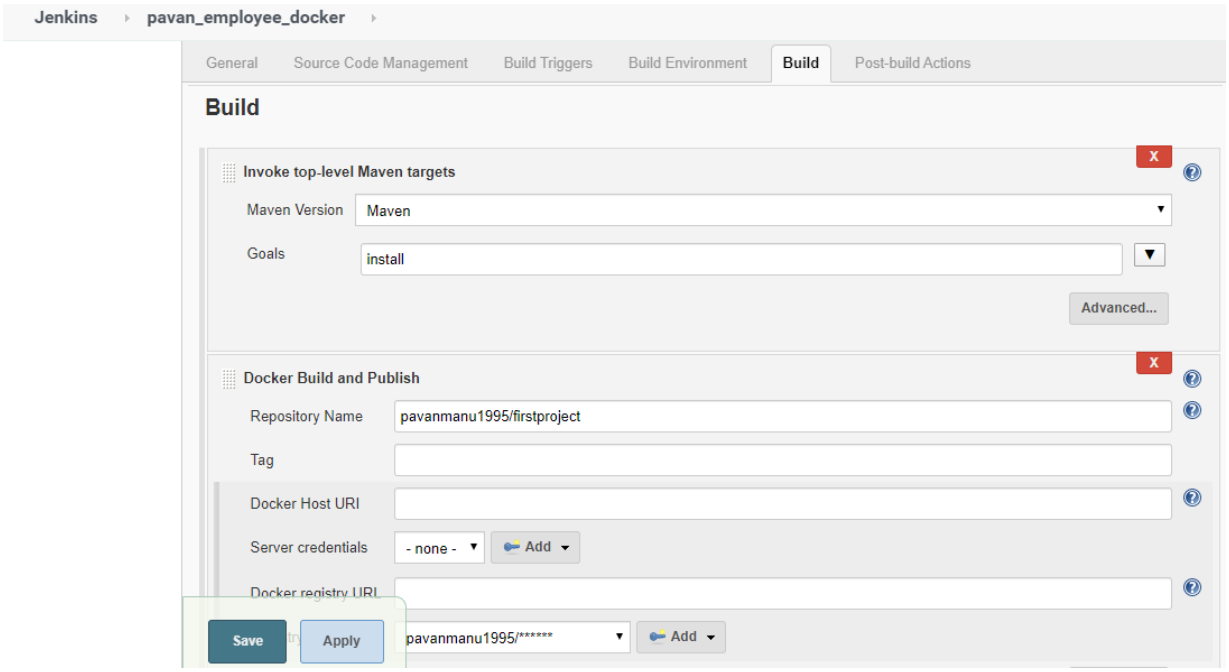


General → github project → put the github url → source code management → git → put the github url

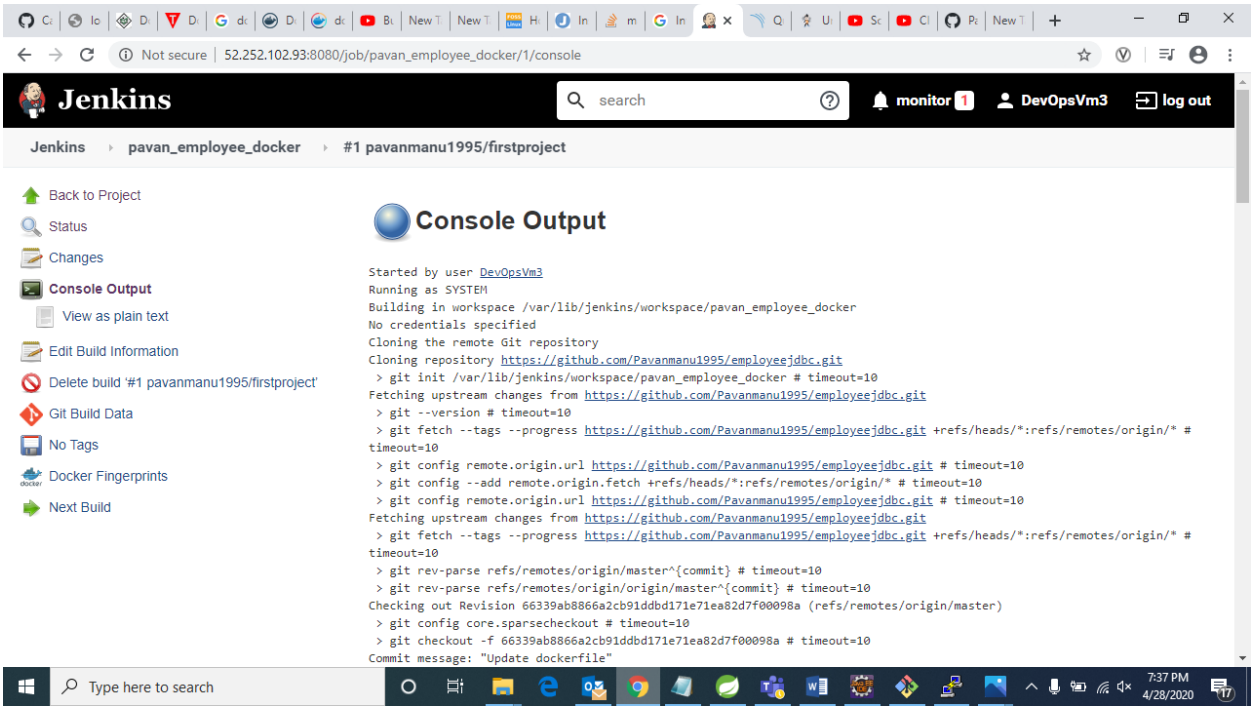




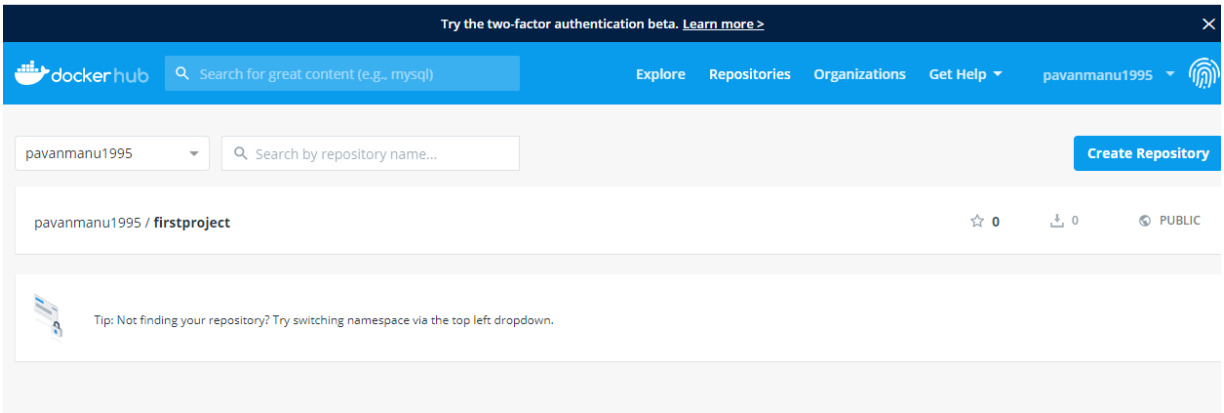
Build → add build steps → invoke top-level maven targets → maven version →select from drop down → goals :install → add build steps → Docker build and publish → repository name: give the dockerhub username/repository name → apply → save.



After saving click on build now, after successful build the project will be pushed to dockerhub



Check the dockerhub the pushed project will be seen.



To check the project is pushed successfully, in linux execute docker images u can see your project.