TASK 1 - DevOps

Problem Statement :

JOB 1:
If Developer push to dev branch then Jenkins will fetch from dev and deploy on dev-docker environment.

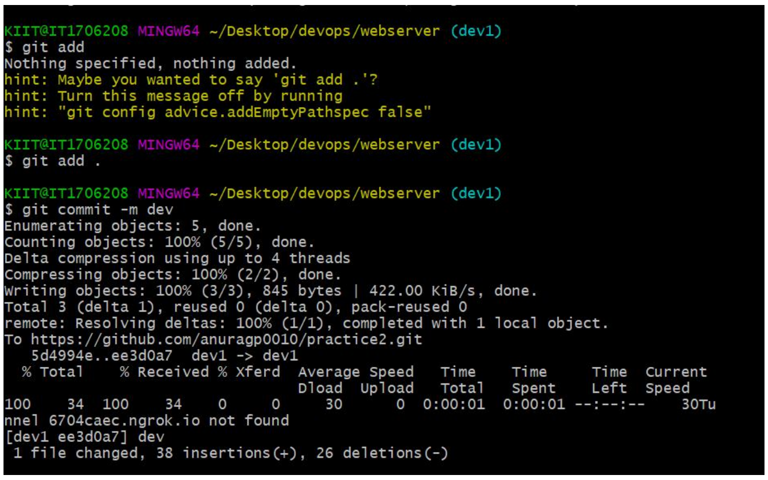
JOB 2:
If Developer push to master branch then Jenkins will fetch from master and deploy on master-docke environment.
both dev-docker and master-docker environment are on different docker containers.

JOB 3:
Manually the QA team will check (test) for the website running in dev-docker environment. If it is running fine then Jenkins will merge the dev branch to master branch and trigger #job 2

Technologies used:-

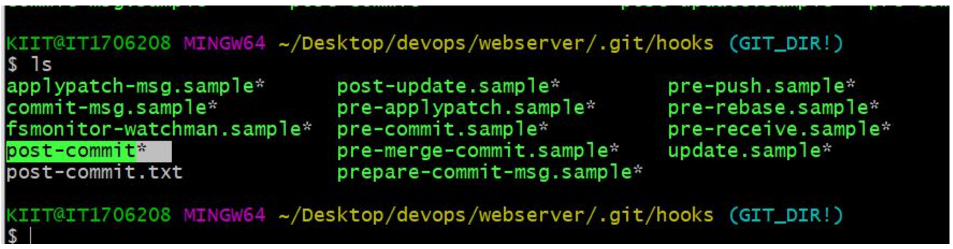
- Git and GitHub
- Docker
- Jenkins

Step 1: Developer commits the code

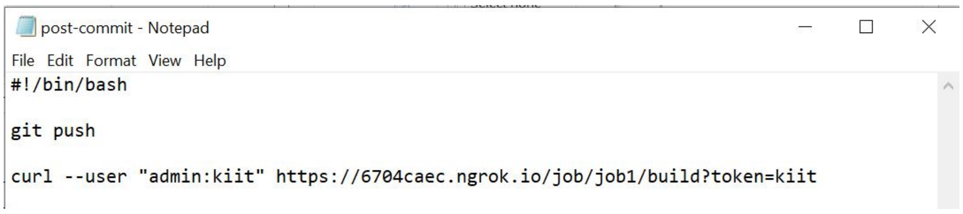


```
KIIT@IT1706208 MINGW64 ~/Desktop/devops/webserver (dev1)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'?
hint: Turn this message off by running
hint: "git config advice.addEmptyPathspec false"

KIIT@IT1706208 MINGW64 ~/Desktop/devops/webserver (dev1)
$ git add .

KIIT@IT1706208 MINGW64 ~/Desktop/devops/webserver (dev1)
$ git commit -m dev
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 845 bytes | 422.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/anuragp0010/practice2.git
   5d4994e..ee3d0a7  dev1 -> dev1
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    34  100    34    0     0     30      0  0:00:01  0:00:01 --:--:--    30Tu
nnel 6704caec.ngrok.io not found
[dev1 ee3d0a7] dev
 1 file changed, 38 insertions(+), 26 deletions(-)
```

Step 2: Create a hook to push



```
KIIT@IT1706208 MINGW64 ~/Desktop/devops/webserver/.git/hooks (GIT_DIR!)
$ ls
applypatch-msg.sample*       post-update.sample*          pre-push.sample*
commit-msg.sample*           pre-applypatch.sample*       pre-rebase.sample*
fsmonitor-watchman.sample*   pre-commit.sample*           pre-receive.sample*
post-commit*                 pre-merge-commit.sample*     update.sample*
post-commit.txt              prepare-commit-msg.sample*

KIIT@IT1706208 MINGW64 ~/Desktop/devops/webserver/.git/hooks (GIT_DIR!)
$ |
```

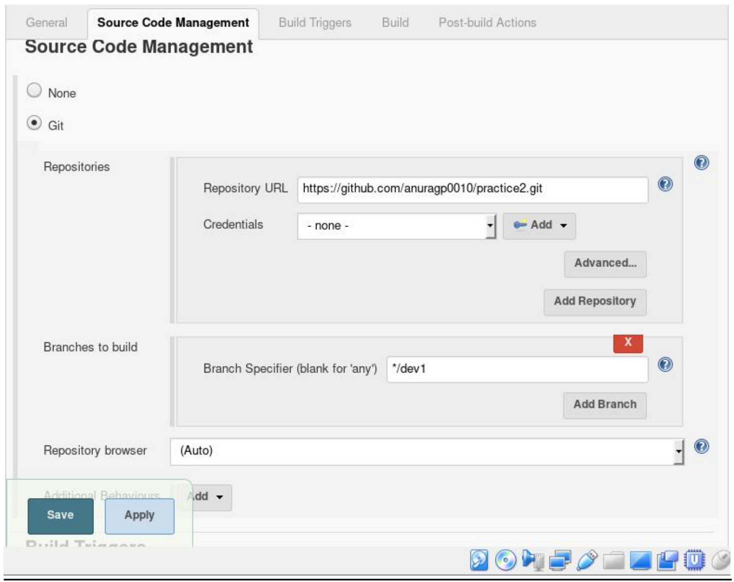


```
#!/bin/bash

git push

curl --user "admin:kiit" https://6704caec.ngrok.io/job/job1/build?token=kiit
```

Fig: Inside post-commit file (used ngrok for tunneling the network)

So, after this when ever the Developer will commit the code then "post-commit" will directly push the code to GitHub and after that the Jenkins will start their Job or their Build Pipeline.

Job 1:



Source Code Management

None [✓ Git]

Repositories

Repository URL: https://github.com/anuragp0010/practice2.git

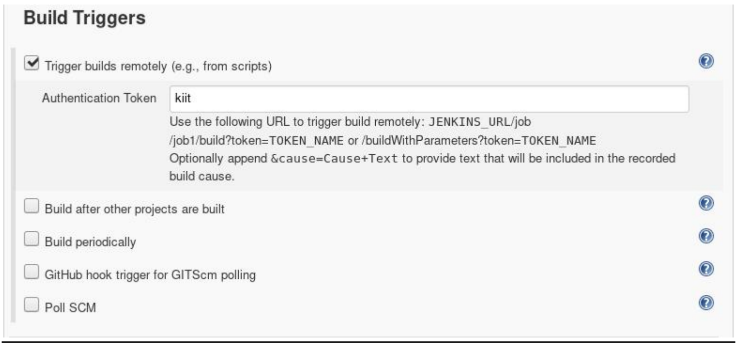
Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */dev1

Repository browser: (Auto)

Fig: Pulling from Dev1 node



Build Triggers

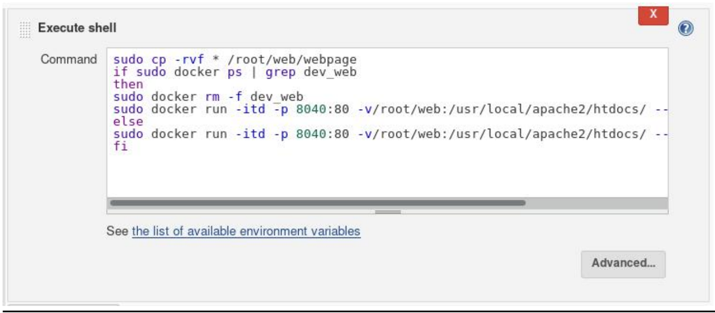
[✓] Trigger builds remotely (e.g., from scripts)

Authentication Token: kiit

Use the following URL to trigger build remotely: `JENKINS_URL`/job/job1/build?token=`TOKEN_NAME` or /buildWithParameters?token=`TOKEN_NAME`
Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Fig: Used Authentication Token for trigger



Execute shell

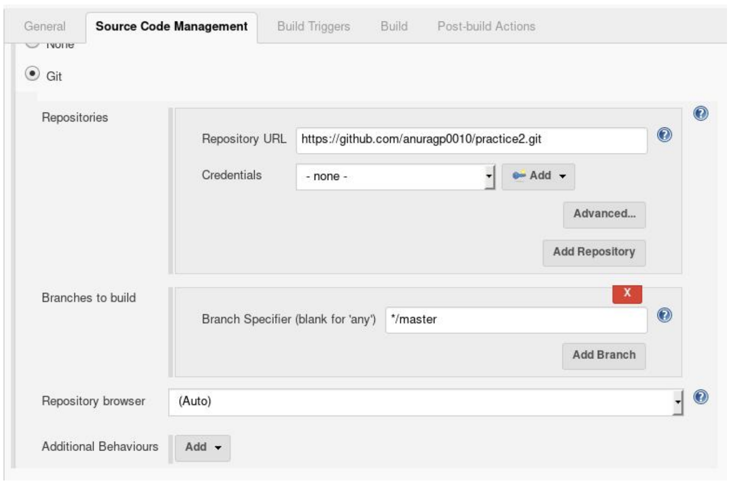
Command

```
sudo cp -rvf * /root/web/webpage
if sudo docker ps | grep dev_web
then
sudo docker rm -f dev_web
sudo docker run -itd -p 8040:80 -v/root/web:/usr/local/apache2/htdocs/ --
else
sudo docker run -itd -p 8040:80 -v/root/web:/usr/local/apache2/htdocs/ --
fi
```

See the list of available environment variables

Fig: Copying the Dev1 content in web folder

Job 2:



General | **Source Code Management** | Build Triggers | Build | Post-build Actions

( ) None

(●) Git

Repositories

Repository URL: https://github.com/anuragp0010/practice2.git

Credentials: - none -

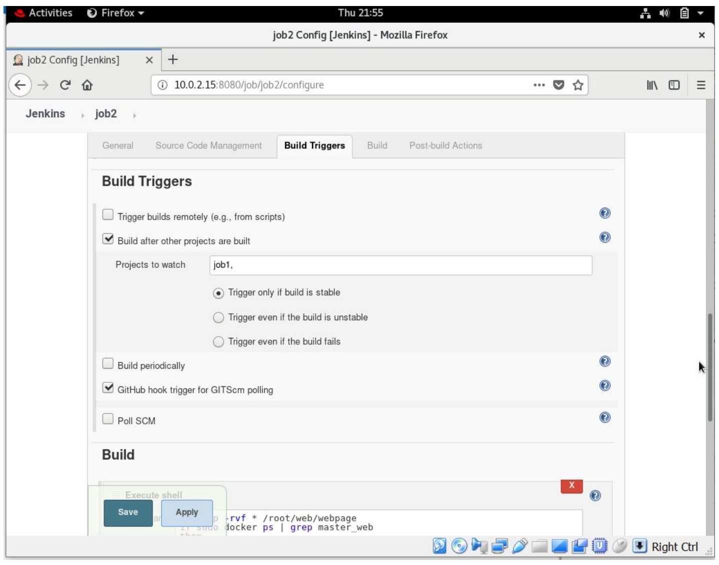
Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

Fig: Pulling from master node



job2 Config [Jenkins] - Mozilla Firefox

10.0.2.15:8080/job/job2/configure

Jenkins › job2 ›

General | Source Code Management | **Build Triggers** | Build | Post-build Actions

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☑ Build after other projects are built

Projects to watch: job1,

(●) Trigger only if build is stable

( ) Trigger even if the build is unstable

( ) Trigger even if the build fails

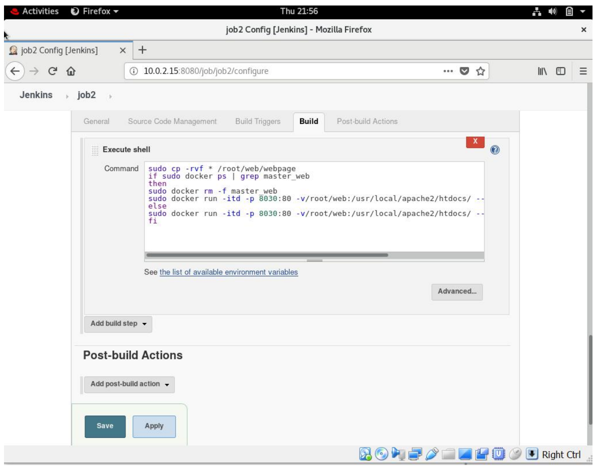
☐ Build periodically

☑ GitHub hook trigger for GITScm polling

☐ Poll SCM

Build

Fig: Job2 will trigger only after Job1 runs fine



job2 Config [Jenkins] - Mozilla Firefox

10.0.2.15:8080/job/job2/configure

Jenkins › job2 ›

General | Source Code Management | Build Triggers | **Build** | Post-build Actions

Execute shell

Command

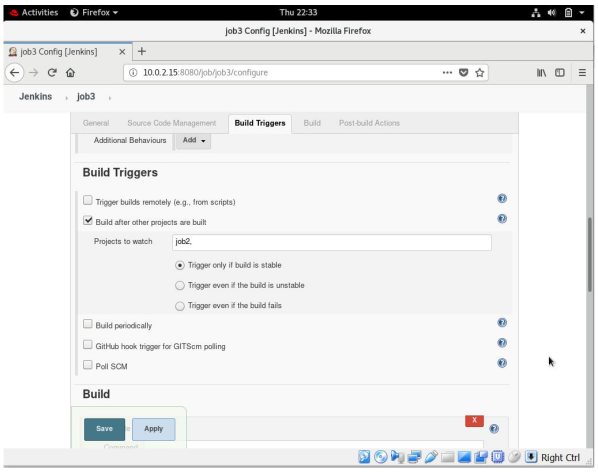
```
sudo cp -rvf * /root/web/webpage
if sudo docker ps | grep master_web
then
sudo docker rm -f master_web
sudo docker run -itd -p 8030:80 -v/root/web:/usr/local/apache2/htdocs/ --
else
sudo docker run -itd -p 8030:80 -v/root/web:/usr/local/apache2/htdocs/ --
fi
```

See the list of available environment variables

Post-build Actions

Fig: Copying the Master content in web folder

Job 3:

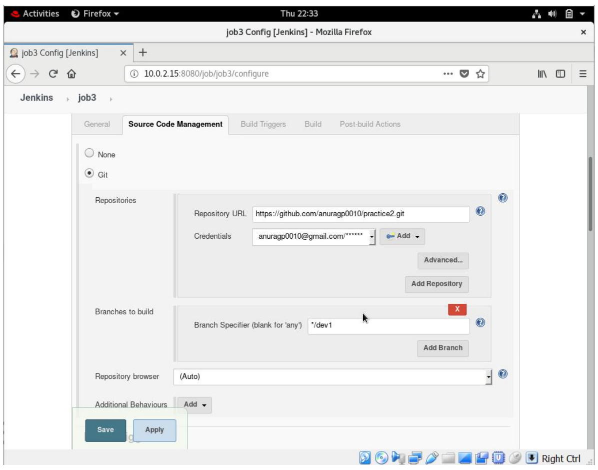


Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☑ Build after other projects are built
  - Projects to watch: job2,
  - ● Trigger only if build is stable
  - ○ Trigger even if the build is unstable
  - ○ Trigger even if the build fails
- ☐ Build periodically
- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

Build

Fig: Job3 will trigger only after Job2 runs fine



- ○ None
- ● Git

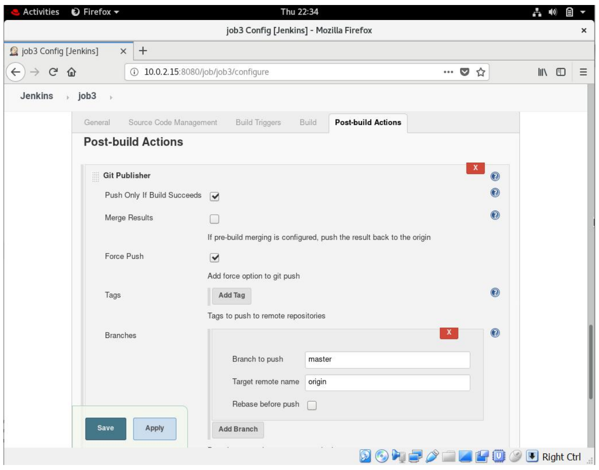
Repositories
- Repository URL: https://github.com/anuragp0010/practice2.git
- Credentials: anuragp0010@gmail.com/******

Branches to build
- Branch Specifier (blank for 'any'): */dev1

Repository browser: (Auto)

Additional Behaviours

Fig: Giving credentials to login into GitHub



Post-build Actions

Git Publisher
- Push Only If Build Succeeds ☑
- Merge Results ☐
  If pre-build merging is configured, push the result back to the origin
- Force Push ☑
  Add force option to git push
- Tags
  Tags to push to remote repositories
- Branches
  - Branch to push: master
  - Target remote name: origin
  - Rebase before push ☐

Fig: Merging the Dev1 branch to Master branch

Build Pipeline of the Task

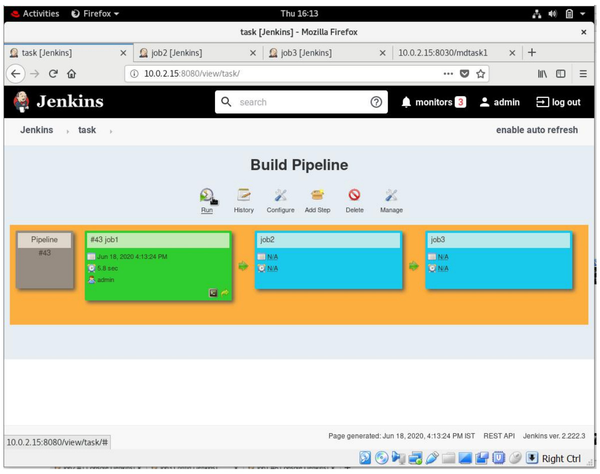


Fig: Job1 runs successfully

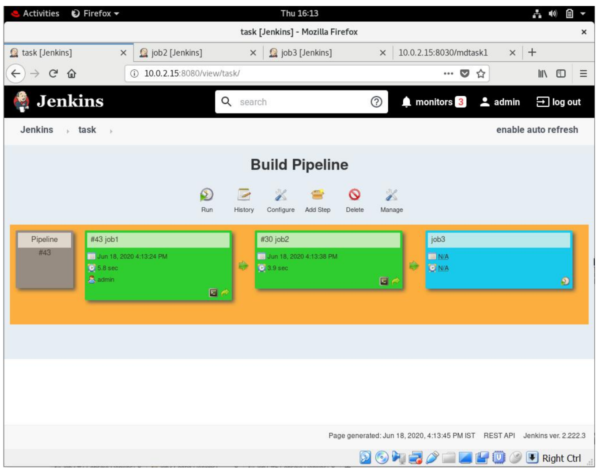


Fig: Job2 runs successfully

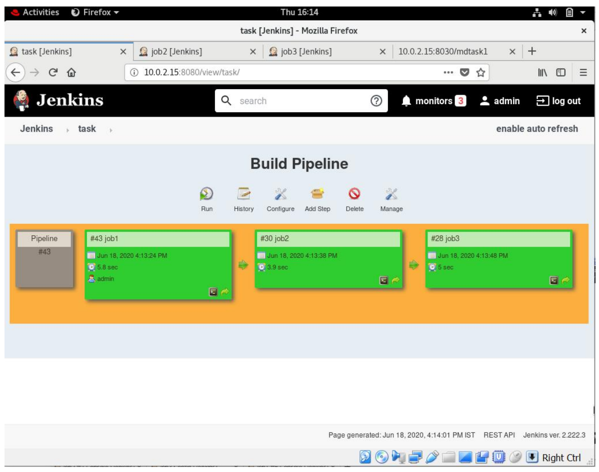


Fig: Job3 runs successfully

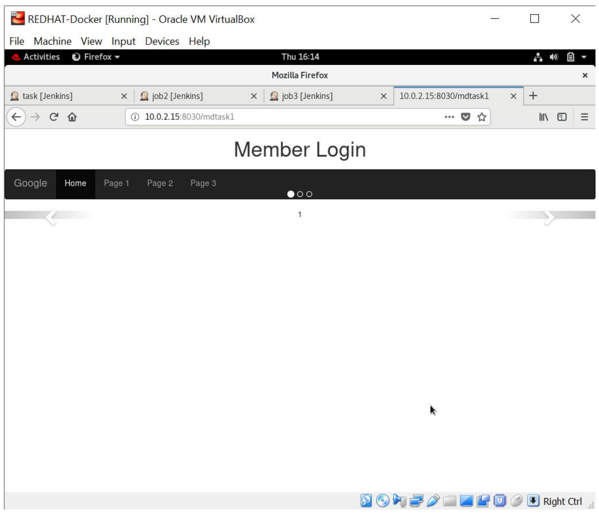


Fig: Final Webpage

In the end I would like to thank Vimal Daga Sir for this task and all the knowledge he shared with us. He is the reason I am capable of create such a great project by our own .

Thank you Everyone For Reading !