Website Deployment Automation using Jenkins and in Docker containers

Overview: The following article will provide you a clear idea of the process of launching a web app can using Jenkins. It involves the automation of the process from the Developer to the end user of the website. Here as soon as the developer makes a commit in the github repository the push is made automatically. Web-hooks linking github to Jenkins will automatically start a job in Jenkins which will the automatically pull the latest changes and copy them in the local repository in the server. This job will then automatically trigger another job which will start a new docker container running the httpd image. The website is hosted in this docker container and the container will be exposed to be accessible by the end users. The detailed steps to do the same are provided below along with some of the necessary screenshots attached.

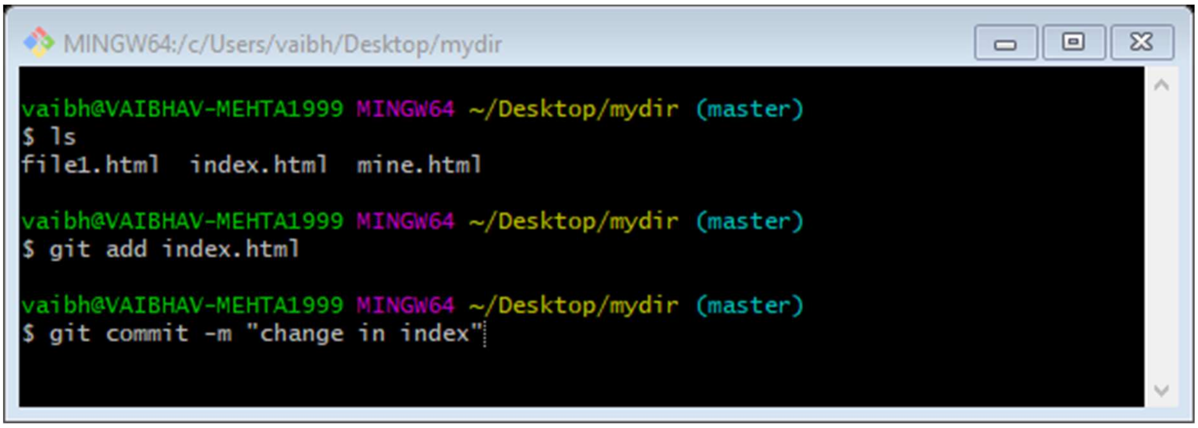
Software required:

- Git, Jenkins and Docker in RHEL or any other OS.
- Github account is necessary
- Docker httpd image available in dockerhub.
- ngrok software for tunnelling(a way for free web hosting)

Procedure:

1. The first step is to setup Github account, Jenkins and Docker environment in RedHat or any other OS.

2. Initialize a local git repository using “`git init`” command; create some files in the repo. Add the files to the staging area using “`git add file_name`” as:

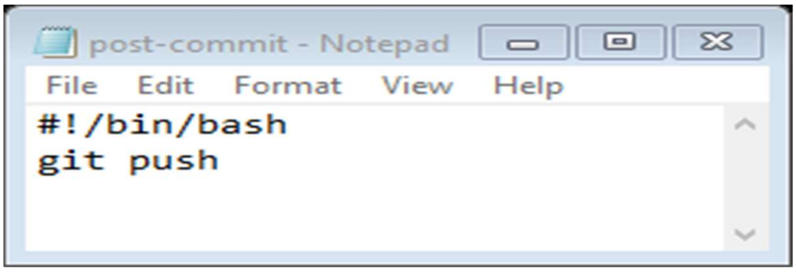


```
vaibh@VAIBHAV-MEHTA1999 MINGW64 ~/Desktop/mydir (master)
$ ls
file1.html  index.html  mine.html

vaibh@VAIBHAV-MEHTA1999 MINGW64 ~/Desktop/mydir (master)
$ git add index.html

vaibh@VAIBHAV-MEHTA1999 MINGW64 ~/Desktop/mydir (master)
$ git commit -m "change in index"
```

3. In the repo use "`cd .git/hooks`". Create a new file there named "`post-commit`" and the following lines in the file. These will create a script to automatically push as soon as a commit is made.

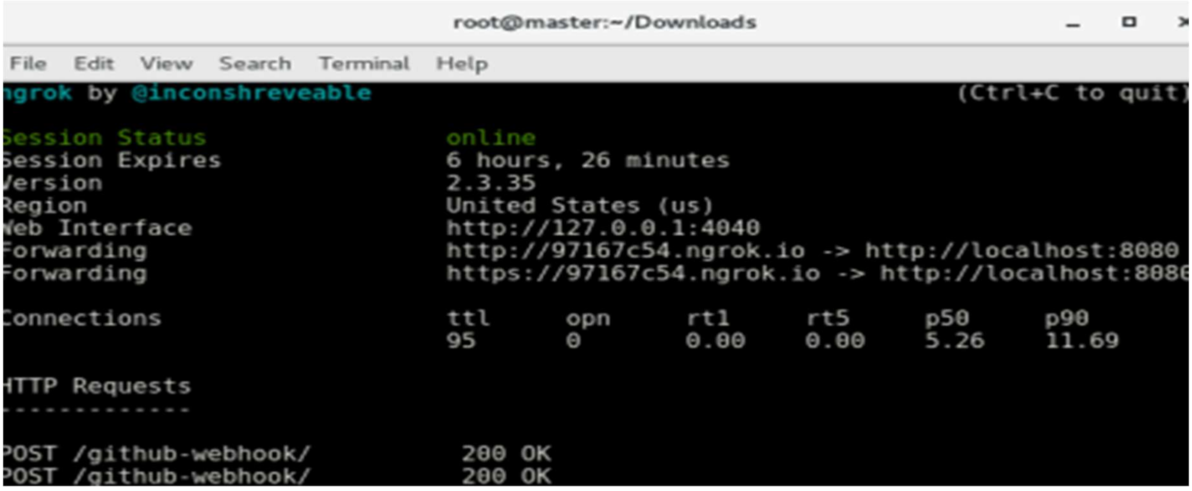


```
#!/bin/bash
git push
```

4. In the RHEL OS install ngrok software and in the command give the following command:

```
>./ngrok http 8080
```

It will provide you this kind of screen along with an ip on which your Jenkins is hosted on the web.



```
ngrok by @inconshreveable                                  (Ctrl+C to quit)

Session Status                online
Session Expires               6 hours, 26 minutes
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://97167c54.ngrok.io -> http://localhost:8080
Forwarding                    https://97167c54.ngrok.io -> http://localhost:808

Connections                   ttl     opn     rt1     rt5     p50     p90
                              95      0       0.00    0.00    5.26    11.69

HTTP Requests
-------------

POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
```

5. On your project in git go to Settings> Webhooks. Give the URL as the one obtained above followed by "/guthub-webhook/". Select the content type as json, active and add webhook.

6. In your server(RHEL OS here) system give unlimited power to the jenkins user.

7. JOB1: Job for auto pulling changes in the github to local repo in the server

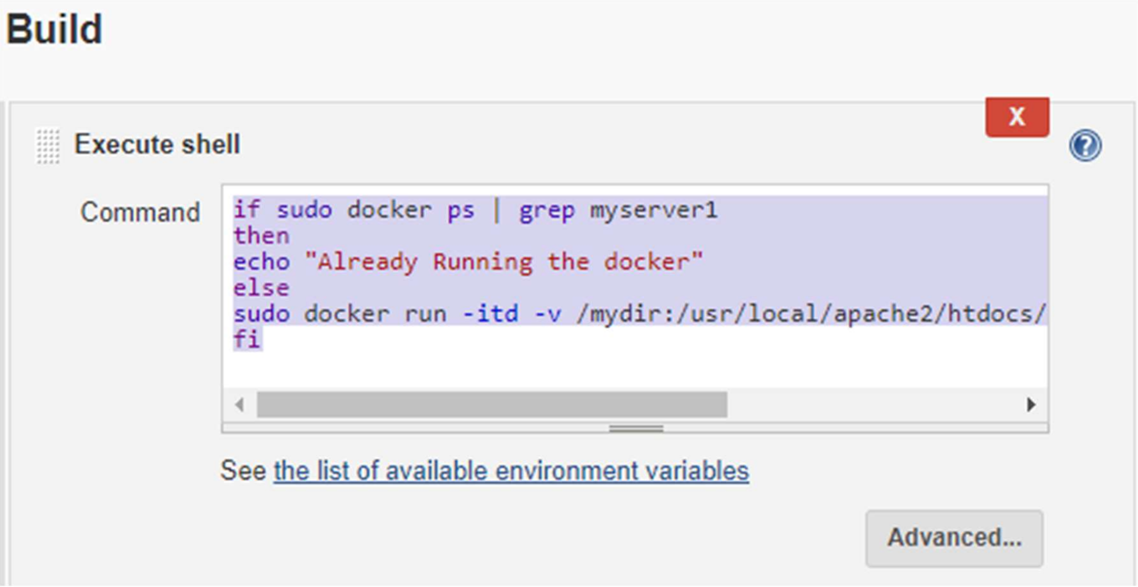
- Open Jenkins and create a new freestyle project. In 'Source Control Management' add the github repo url.
- In '"Build Triggers" select "GitHub hook trigger for GITScm polling".
- In "Build" select "Execute Shell" and give "sudo cp * /mydir", where mydir is the name of the folder where your repo is built in the server system.
- save and apply

8. JOB2: Job to automatically launch a new docker if one does not exist for hosting the website.

- Open Jenkins and create a new freestyle project.
- In "Biuld Triggers" select "Build after other projects are built" and give the name of the previous job. Also select "Trigger only if build is stable".
- In the "Build" select "Execute Shell" and give the following command:

➢

```
if sudo docker ps | grep myserver1
then
   echo "Already Running the docker"
   else
   sudo docker run -itd -v /mydir:/usr/local/apache2/htdocs/
-p 8000:80 --name myserver1 httpd
   fi
```



Build

Execute shell

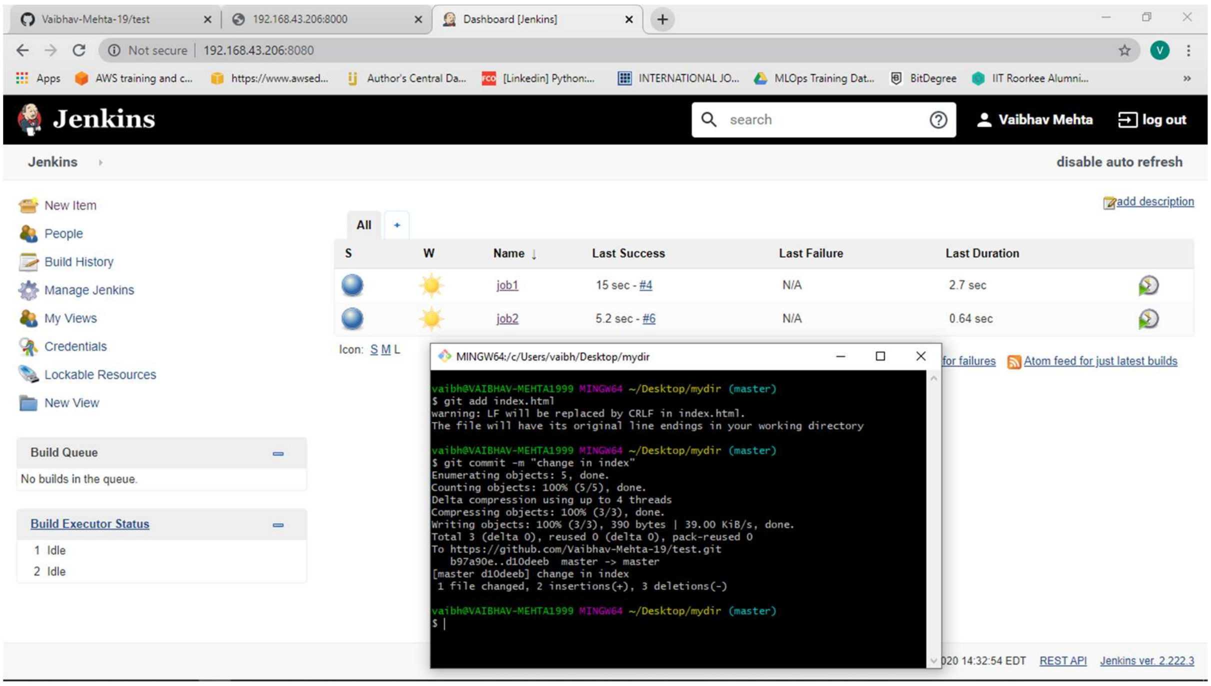
Command

```
if sudo docker ps | grep myserver1
then
echo "Already Running the docker"
else
sudo docker run -itd -v /mydir:/usr/local/apache2/htdocs/
fi
```

See the list of available environment variables

save and apply.

9. Now as soon as the developer makes any new commit all these process will run automatically in the background and the end user will be able to see the webpage as seen in the images available below.



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	job1	15 sec - #4	N/A	2.7 sec
●	☀	job2	5.2 sec - #6	N/A	0.64 sec

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

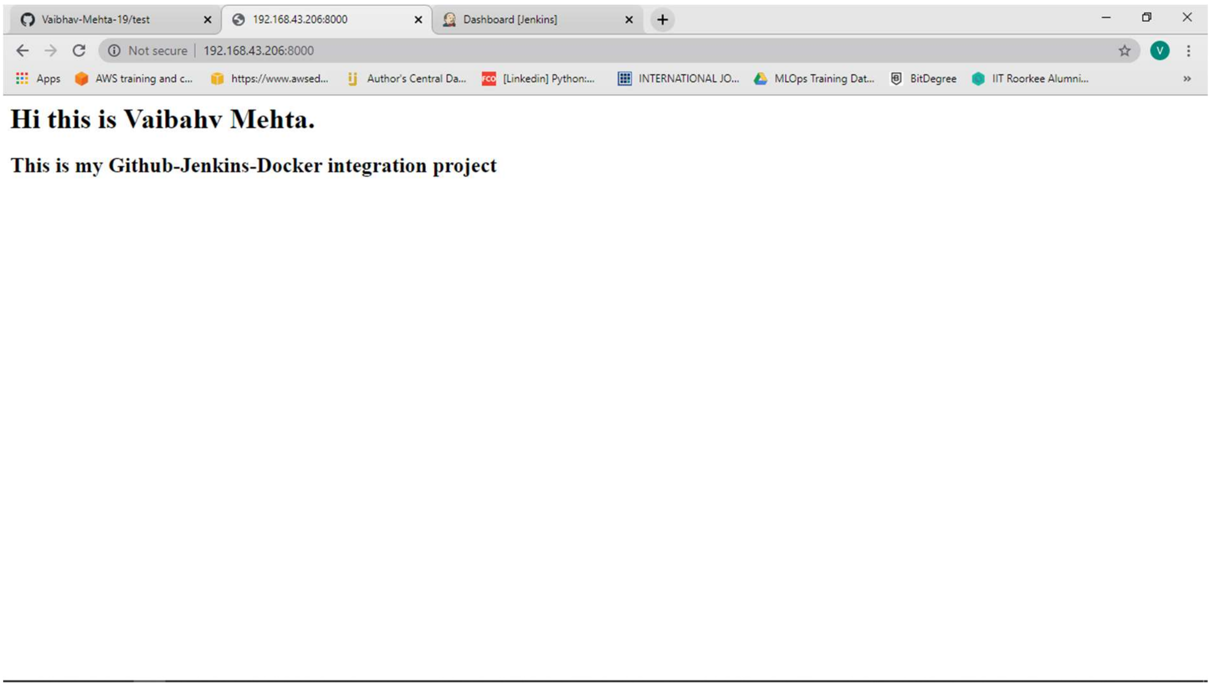
2 Idle

MINGW64:/c/Users/vaibh/Desktop/mydir

```
vaibh@VAIBHAV-MEHTA1999 MINGW64 ~/Desktop/mydir (master)
$ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

vaibh@VAIBHAV-MEHTA1999 MINGW64 ~/Desktop/mydir (master)
$ git commit -m "change in index"
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 390 bytes | 39.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Vaibhav-Mehta-19/test.git
   b97a90e..d10deeb  master -> master
[master d10deeb] change in index
 1 file changed, 2 insertions(+), 3 deletions(-)

vaibh@VAIBHAV-MEHTA1999 MINGW64 ~/Desktop/mydir (master)
$ |
```



Hi this is Vaibahv Mehta.

This is my Github-Jenkins-Docker integration project

<u>Future Work:</u> I am working on the project for integrating more procedures and tools for further automation of the process of hosing a webpage and to completely eliminate the need for any manual steps.

For further references, suggestions or doubts:

Github: https://github.com/Vaibhav-Mehta-19/test.git

LinkedIn: https://www.linkedin.com/in/vaibhavmehta1999