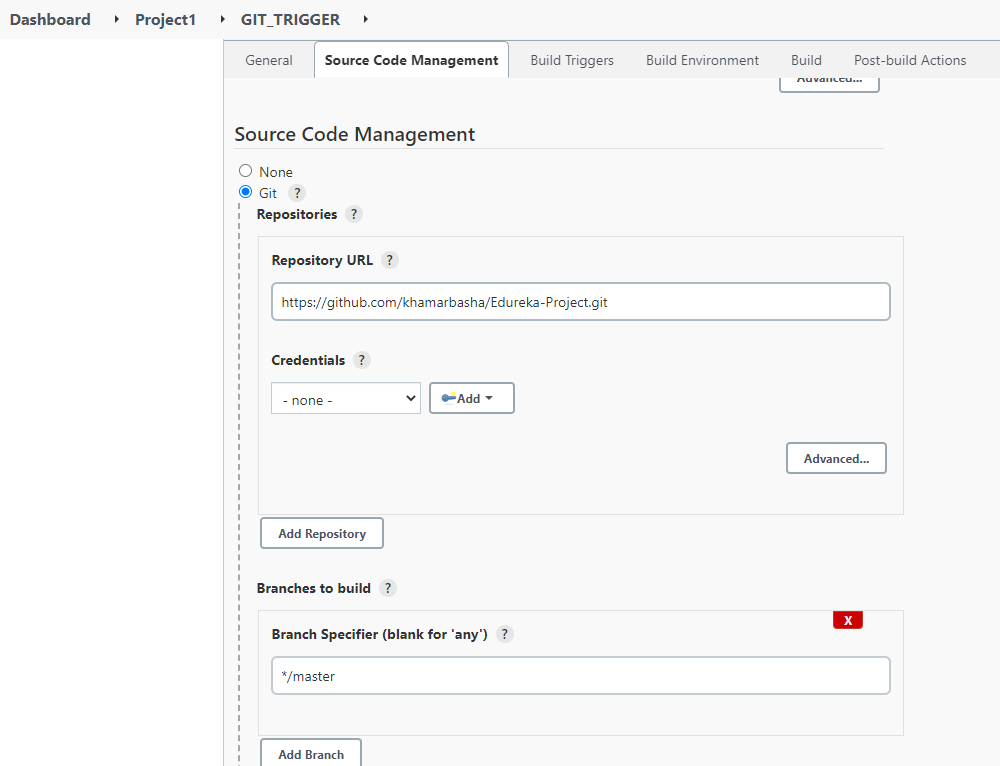
Certification Project -1

Khamar

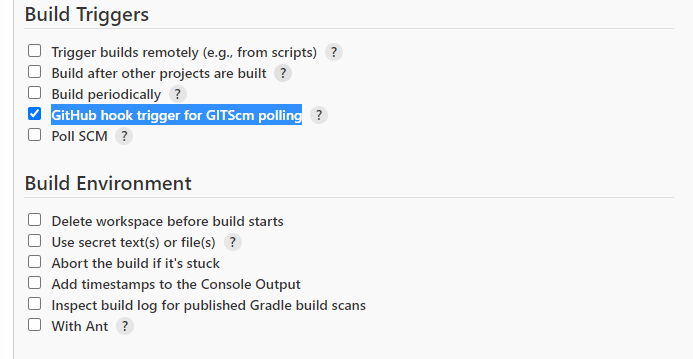
2022

Task -1 create the Jenkins job (GIT\_TRIGGER) to Auto-trigger the pipeline.

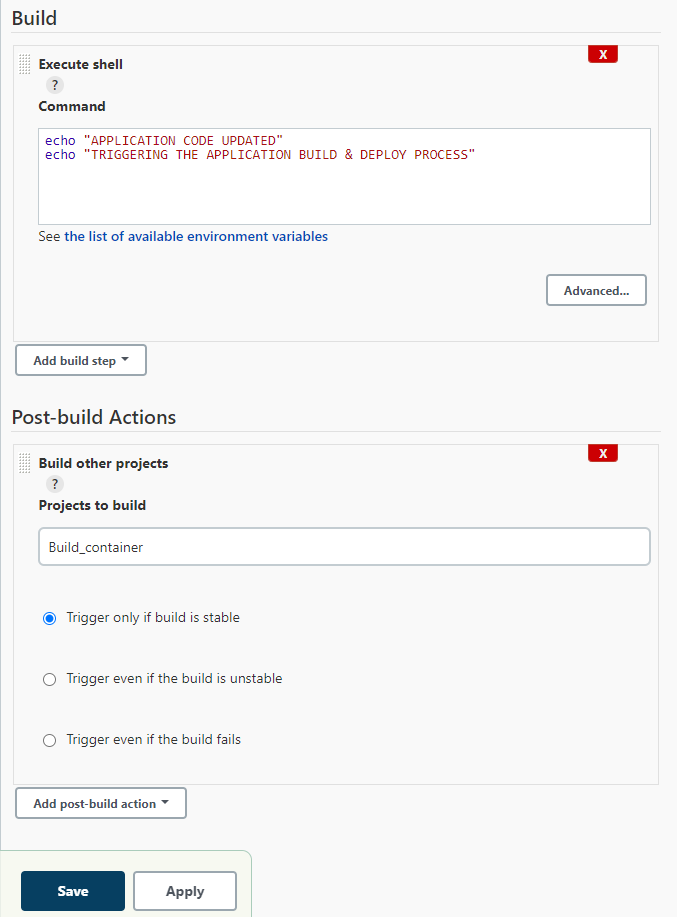
* Create the git-trigger job in the Jenkins.
* Enter the git repo url in the source code management.



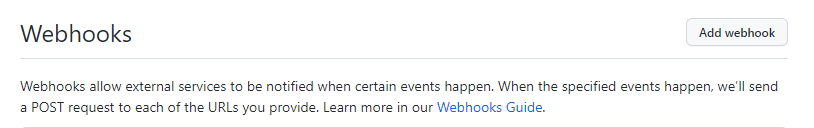
* Go to build trigger section and enable “GitHub hook trigger for GITScm polling” to auto trigger the pipeline when new changes will push to git repo using git webhook feature.



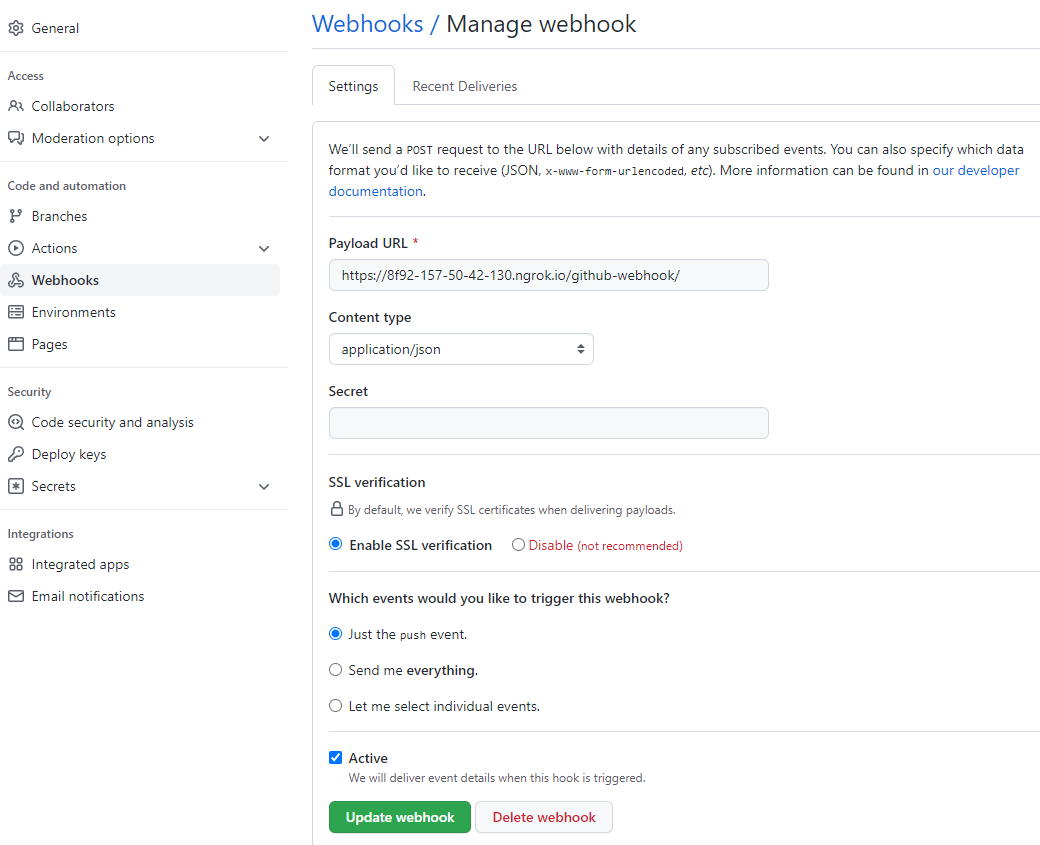
* Add shell script as shown below and save the job.



* Go to git repository of the application to be deployed, then go to setting and click on add webhook.

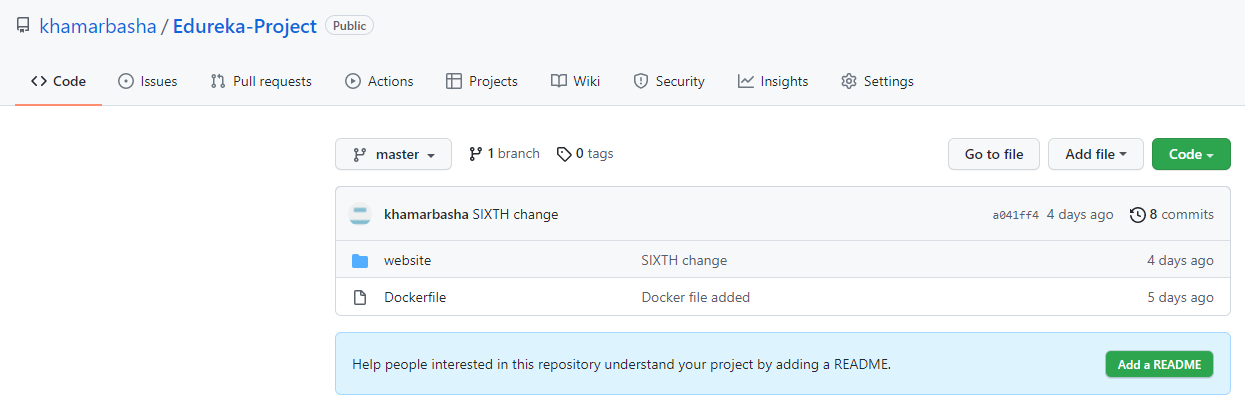


* Enter the Jenkins payload URL and select the content type to application/json, enable ssl verification and select the event which trigger the webhook “just the push event” as shown below.



Task -2 create the Jenkins job (BUILD\_CONTAINER) to build the container for that application form git repo.

* Create docker file and push it to git repro as shown below.



Docker file

FROM devopsedu/webapp

ADD website /var/www/html

RUN rm /var/www/html/index.html

CMD ["apache2ctl","-D","FOREGROUND"]

EXPOSE 80

* Create Jenkins job to build, save the docker image and create docker container by using git docker file of test server by using following shell script.

pwd

ls -ltr

docker build -t khamarbasha/build:1.0 .

docker images

docker rm app --force

docker run -d -p 5200:80 --name app khamarbasha/build:1.0

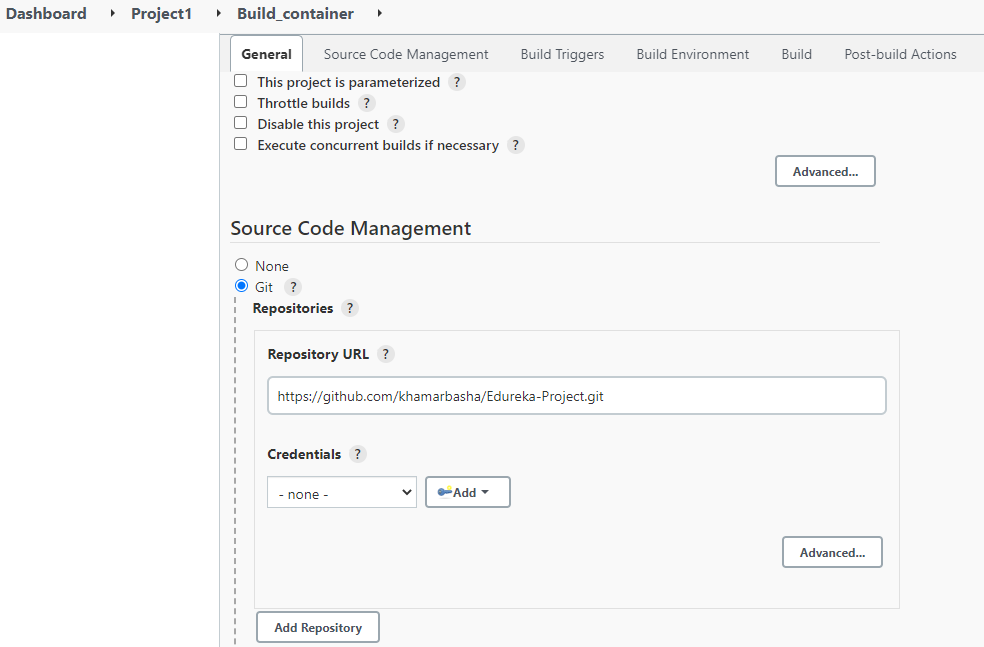
docker ps

mkdir -p /root/Desktop/Docker

rm -f /root/Desktop/Docker/\*

docker save -o /root/Desktop/Docker/app.tar khamarbasha/build:1.0

* Enter the git repo url in the source code management.



* Enter the shell command to build the docker image and container as shown below and save the job.



Task -3 create the Jenkins job (DEPLOY\_CONTAINER) to deploy the container to production server.

* Create Jenkins job setup docker infrastructure and delete existing container and deploy new container to the production server using ansible playbook as shown below.

Ansible Playbook for infrastructure setup and container deployment

---

- name: install docker

hosts: edureka

become: yes

become\_user: root

gather\_facts: yes

tasks:

- name: copy docker repositary

copy:

src: /etc/yum.repos.d/docker-ce.repo

dest: /etc/yum.repos.d/docker-ce.repo

- name: install docker

command: yum install docker-ce docker-ce-cli containerd.io -- allowerasing -y

- name: start the docker service

service:

name: docker

state: started

enabled: yes

- name: remove existing docker image file and containers

shell: |

mkdir -p /root/Desktop/Docker

cd /root/Desktop/Docker

rm -f \*

- name: copy docker image

copy:

src: /root/Desktop/Docker/app.tar

dest: /root/Desktop/Docker/app.tar

- name: container deployment

shell: |

cd /root/Desktop/Docker

docker load -i app.tar

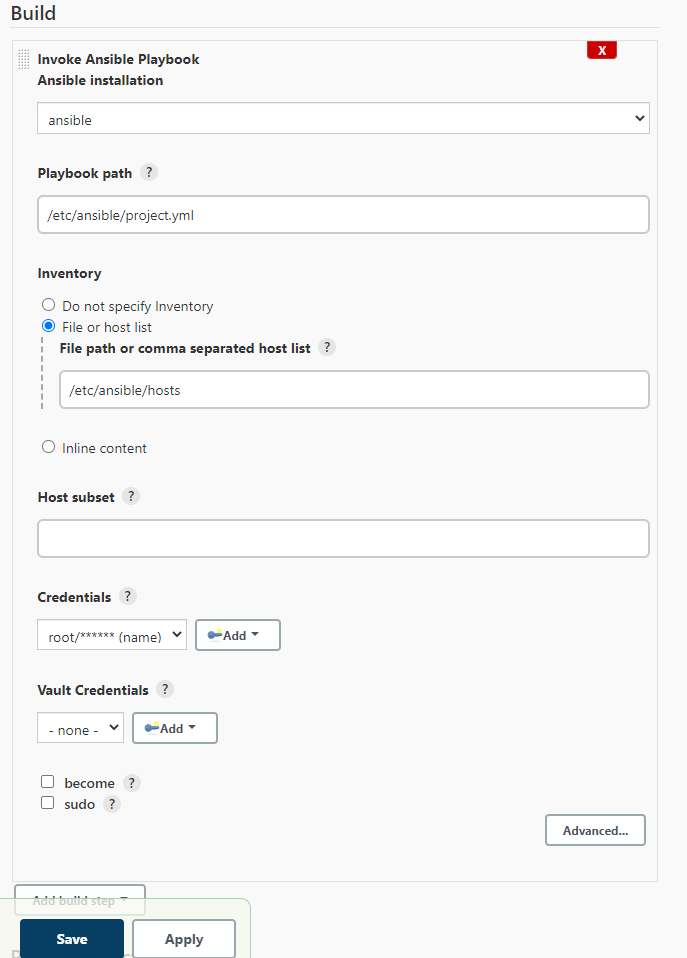
docker images

docker rm app --force

docker run -d -p 5200:80 --name app khamarbasha/build:1.0

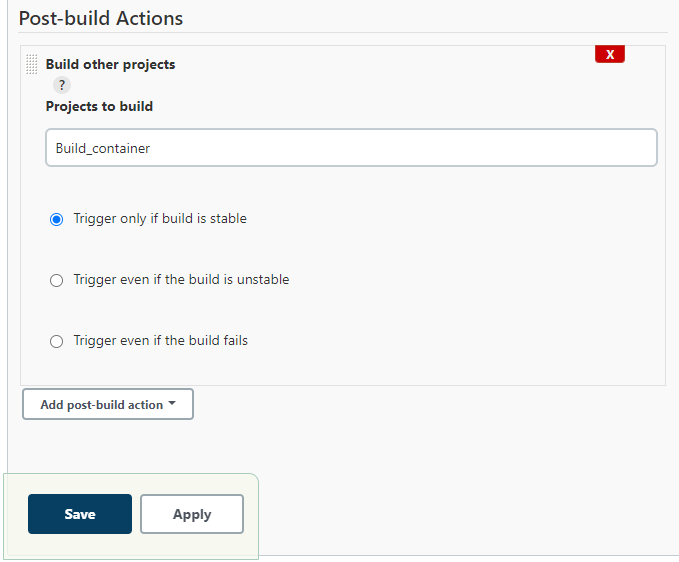
docker ps

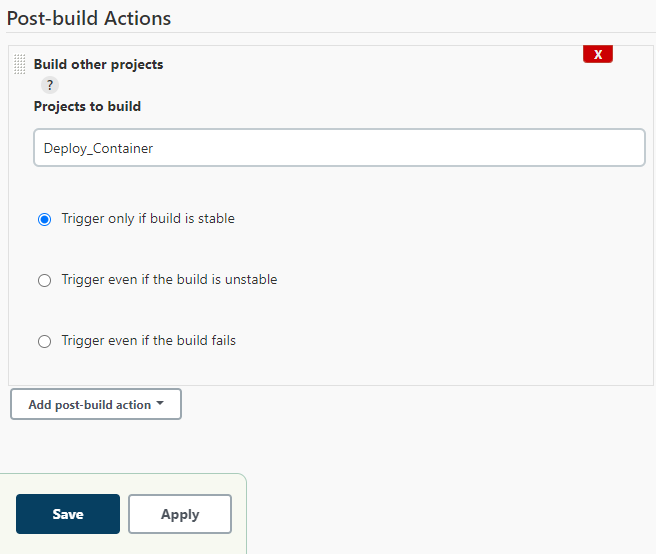
* Create the Deploy\_Container jobs and create a build to invoke ansible playbook as shown below.
* Select the path for playbook and ansible inventory file as shown below.



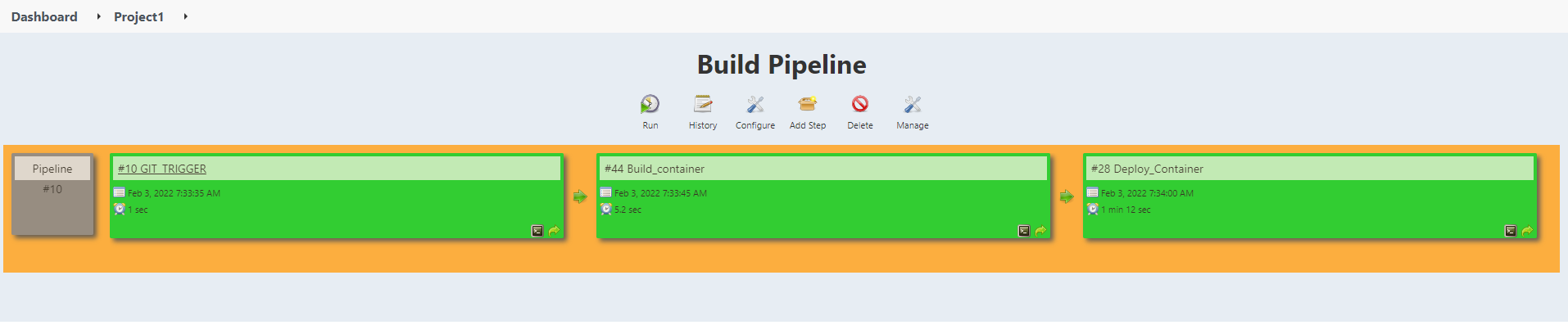
Task -4 create the pipeline.

* Create the pipeline by selecting the post build actions for the jobs “GIT\_TRIGGER” & “BUILD\_CONTAINER” as shown below.



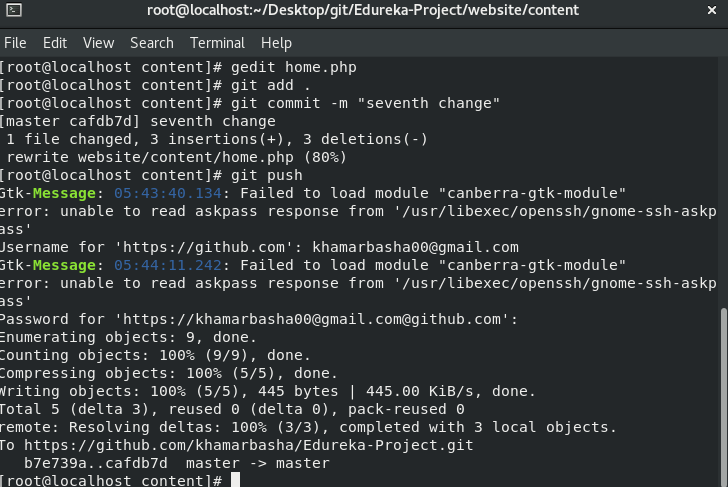


* Create the Jenkins pipeline view “Project1” as shown below.

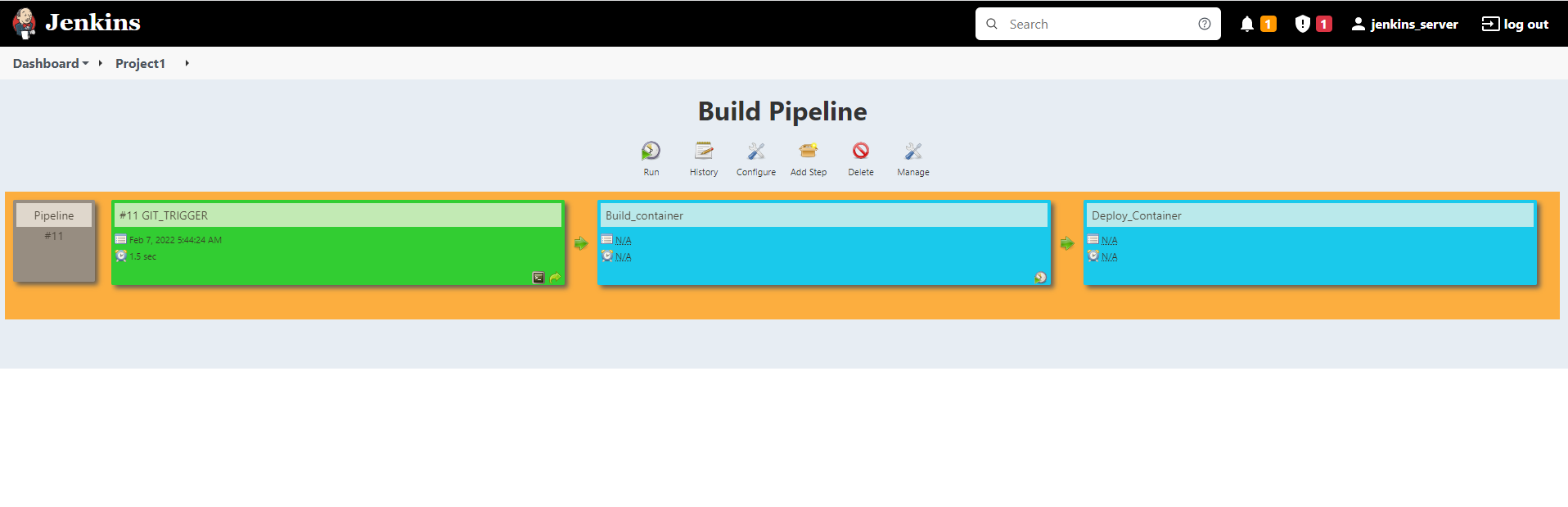


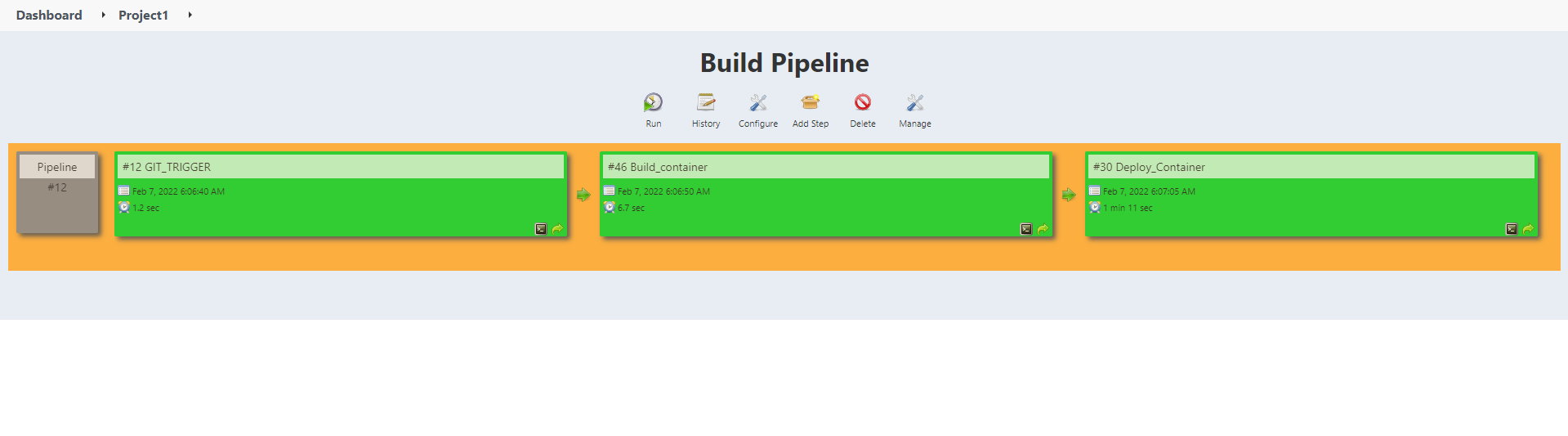
Task -5 check the CI/CD pipeline.

* Change the content of the application form the local git repo and push the change to central git repo as shown below.



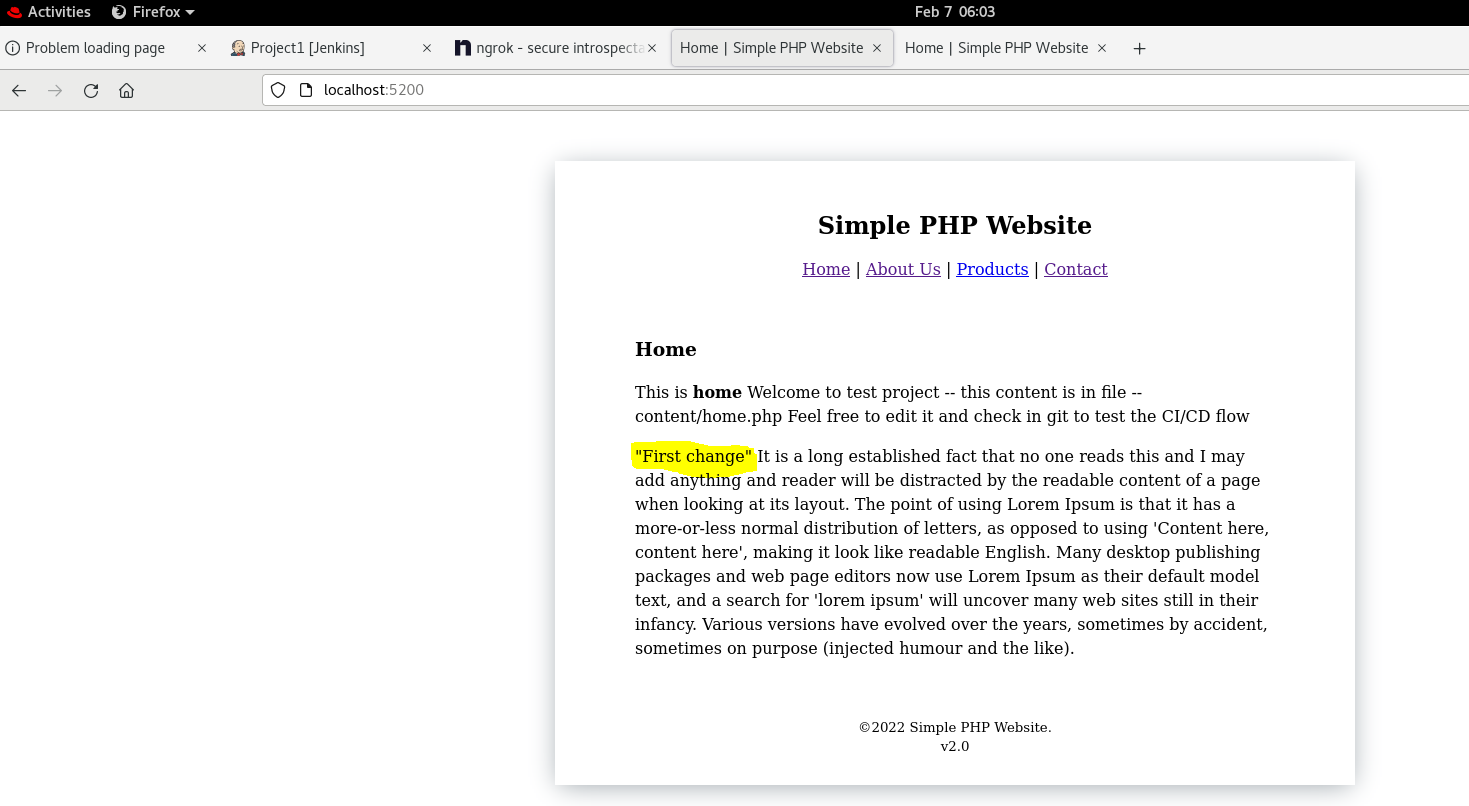
* The git push will trigger the Jenkins pipeline by triggering the GIT\_TRIGGER job as shown below.



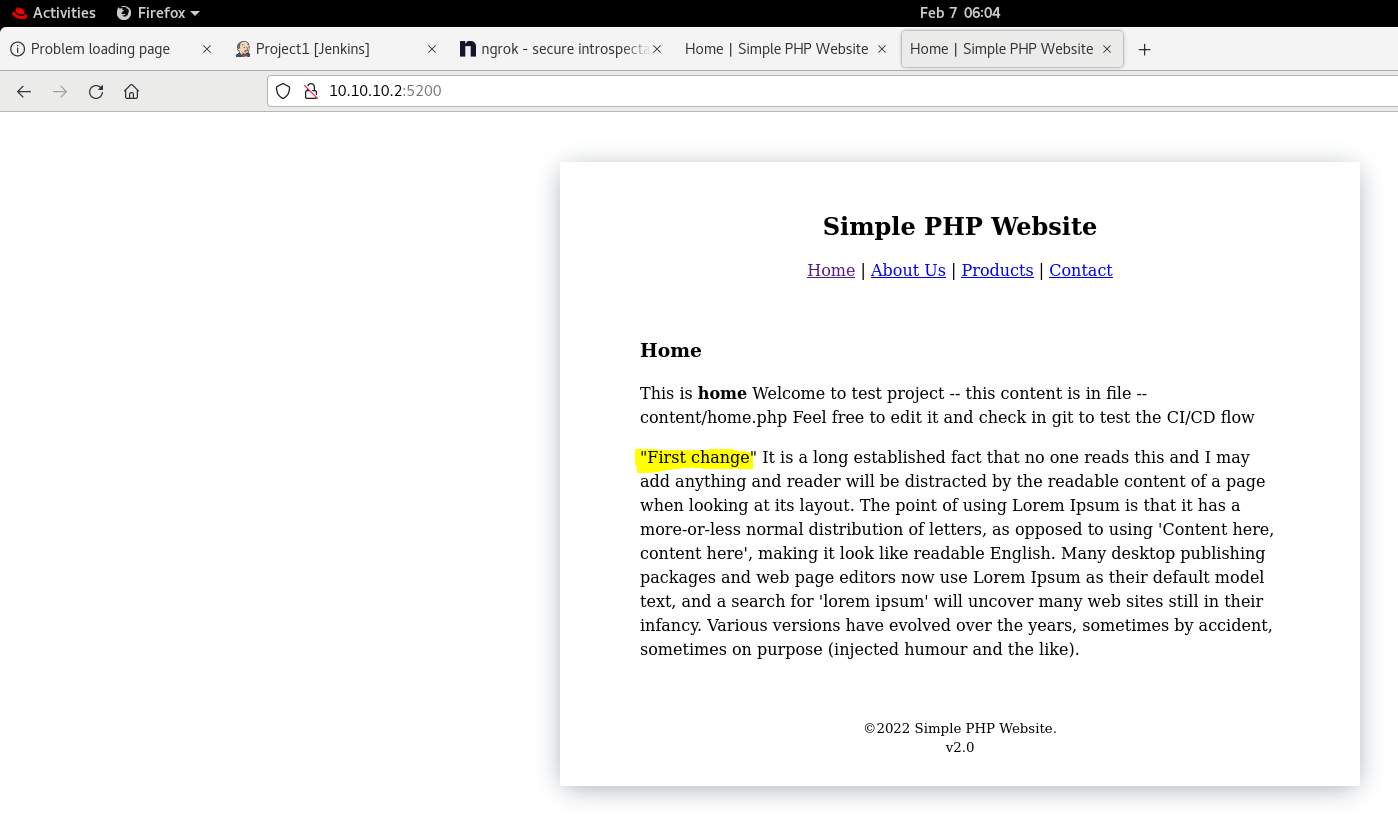
`

* After the successful exicution of pipeline check the application deployed in test server and production server using the web url as shown below

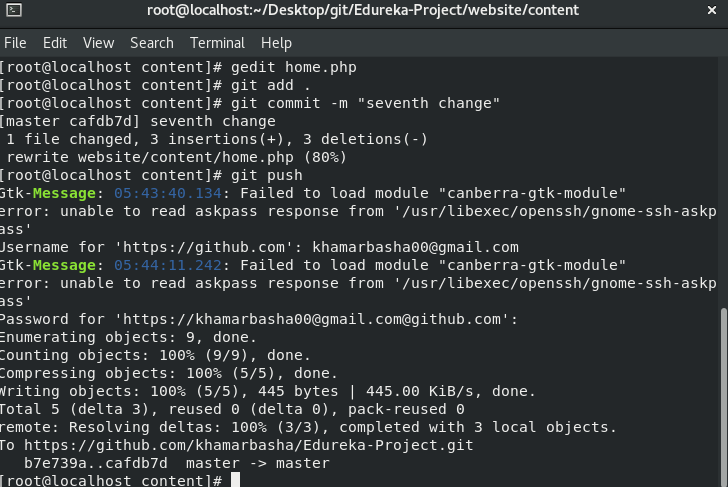
Test server url : <https://localhost:5200/>



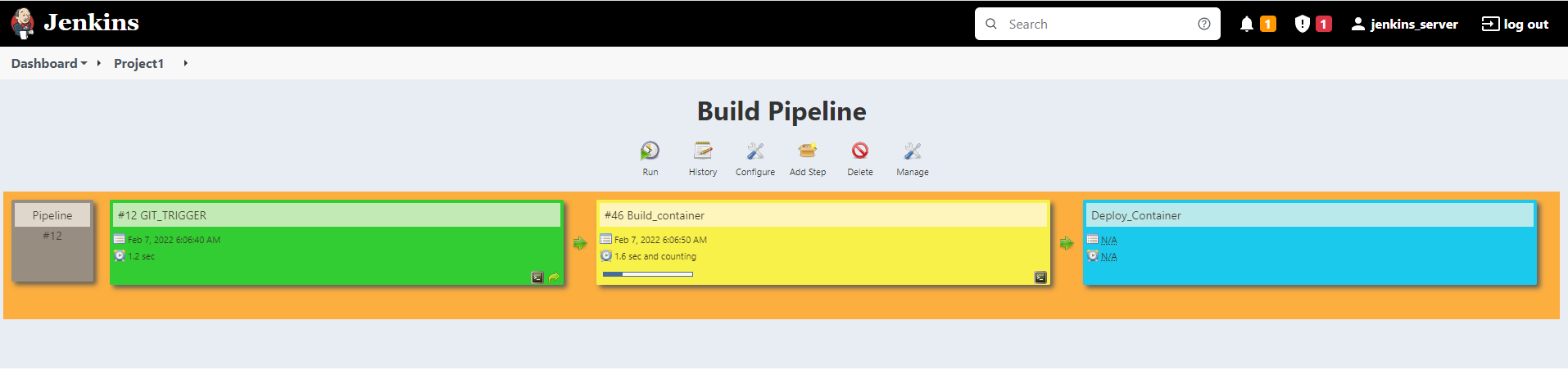
Prod server url : <https://10.10.10.2:5200/>

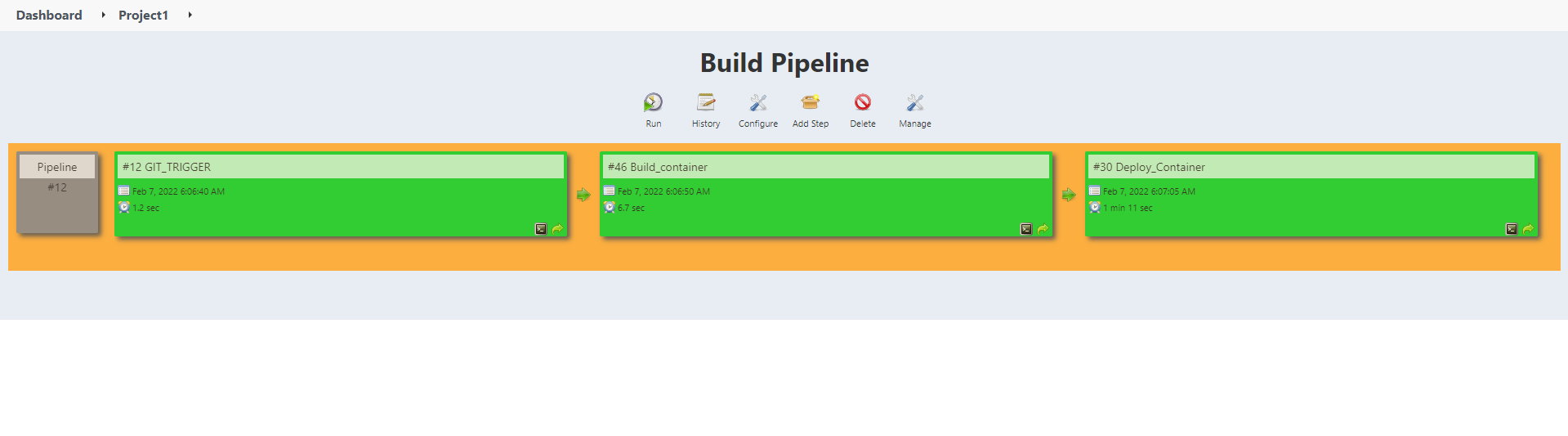


* Again change the content of the application form the local git repo and push the change to central git repo as shown below.



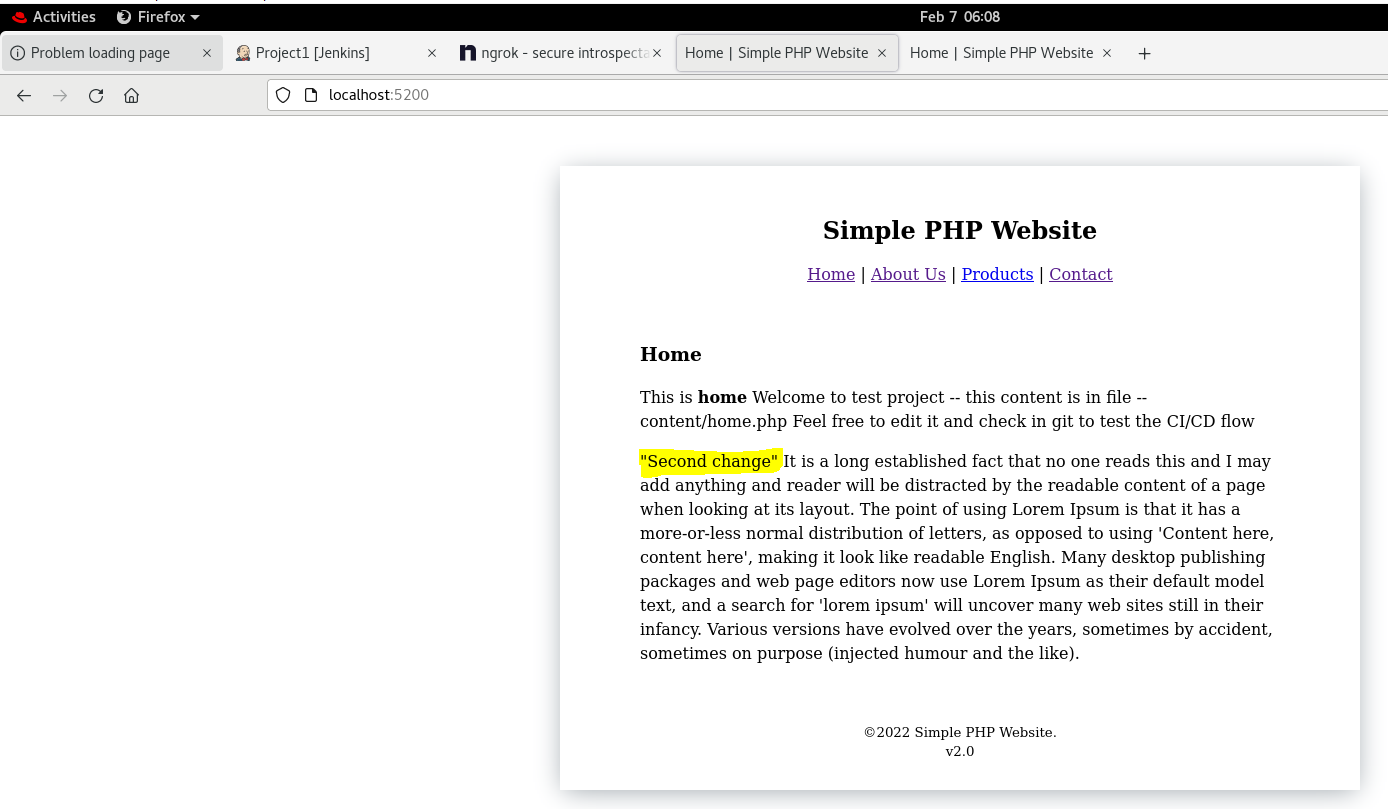
* The git push will trigger the Jenkins pipeline by triggering the GIT\_TRIGGER job as shown below.



`

* After the successful exicution of pipeline check the application deployed in test server and production server using the web url as shown below

Test server url : <https://localhost:5200/>



Prod server url : <https://10.10.10.2:5200/>

