**Jenkins Lab2**

**Name: Mohamed Mostafa Mostafa Esmael Track: System Administration Branch: Aswan**

## GITHUB Repo: <https://github.com/mohamedesmael10/Jenkins_lab2/tree/terraform_ansible>

## What is Jenkins pipeline?

A **Jenkins Pipeline** is a suite of plugins that supports implementing and integrating continuous delivery pipelines into Jenkins. It allows you to define the entire build, test, and deployment process as code (**Jenkinsfile**) so that your CI/CD process becomes versioned and reproducible.

## What scripting language is Jenkins pipeline syntax based on?

The Jenkins Pipeline syntax is based on **Groovy**.

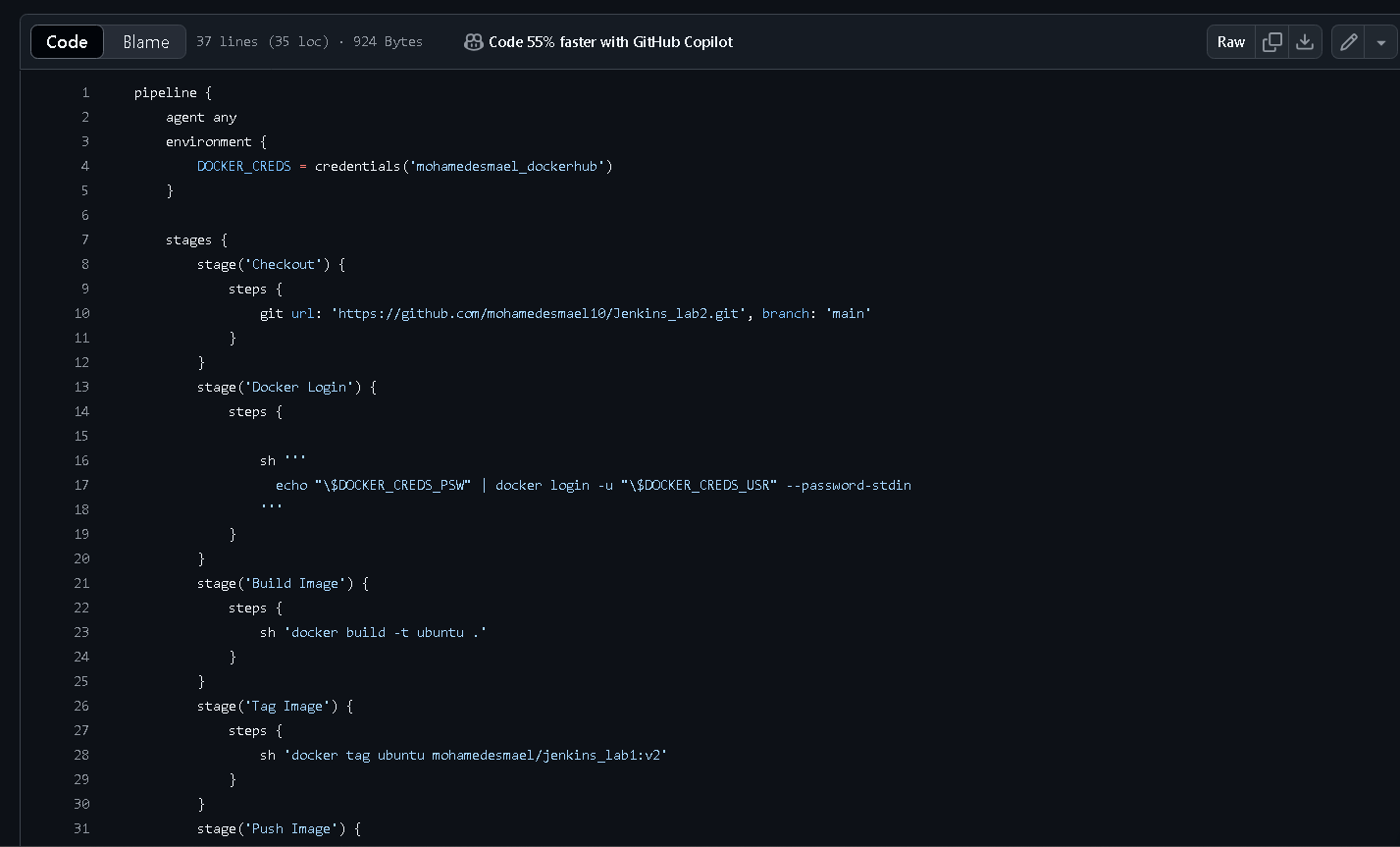
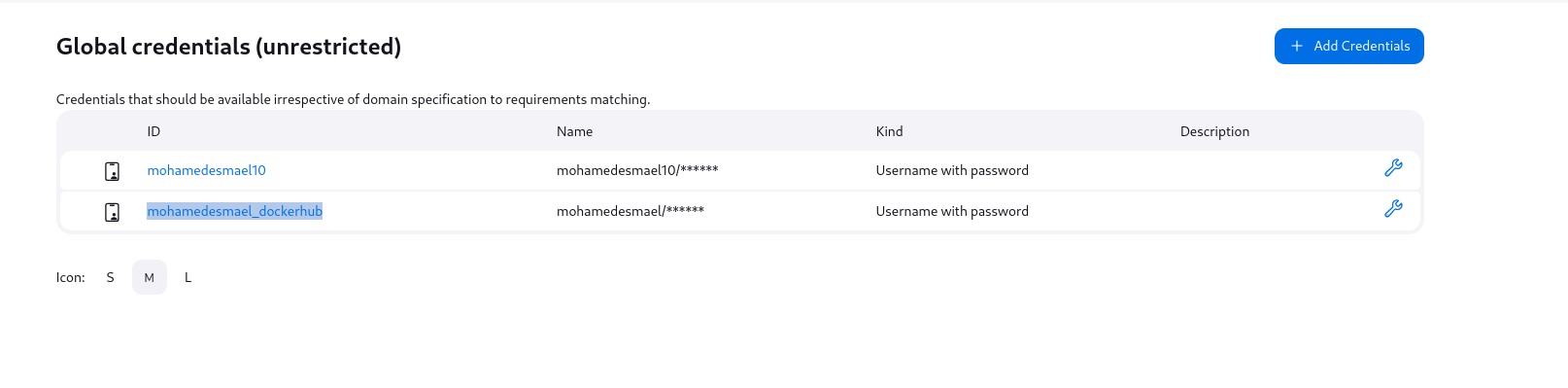
* **Declarative Pipeline** provides a simplified, opinionated syntax for creating pipelines.
* **Scripted Pipeline** uses standard Groovy syntax to define more complex pipelines with full programmatic control.

## What are the ways you can write pipeline in Jenkins?

There are primarily **two ways** to write a pipeline in Jenkins:

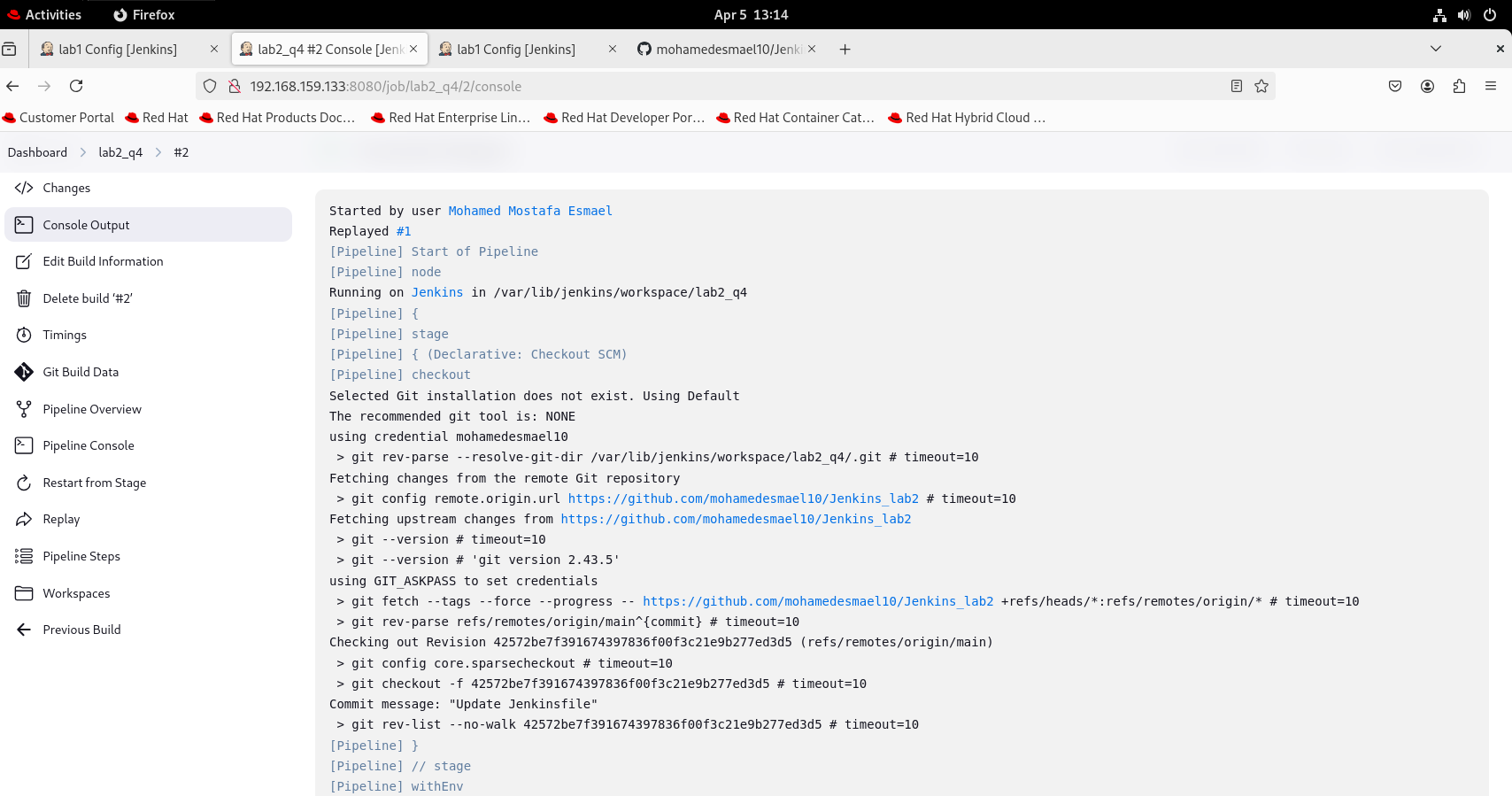
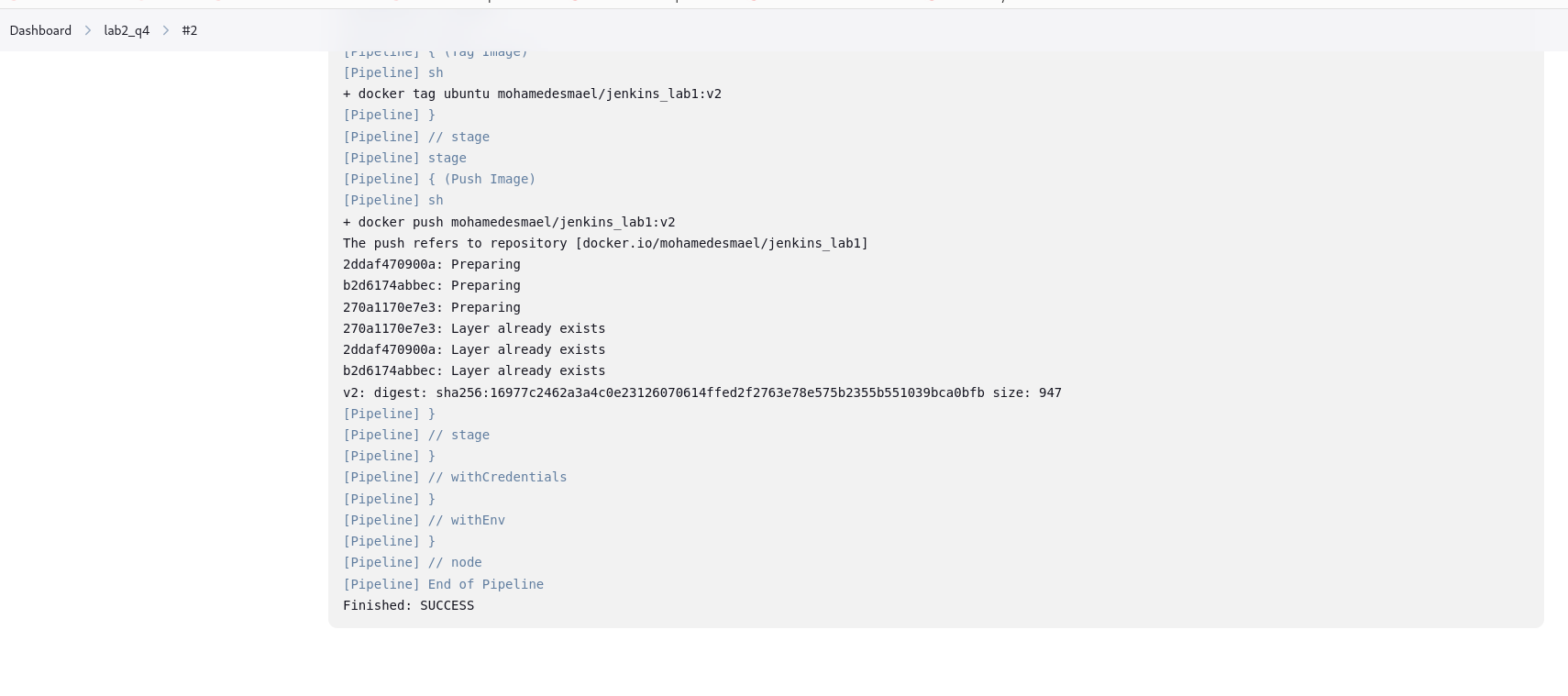
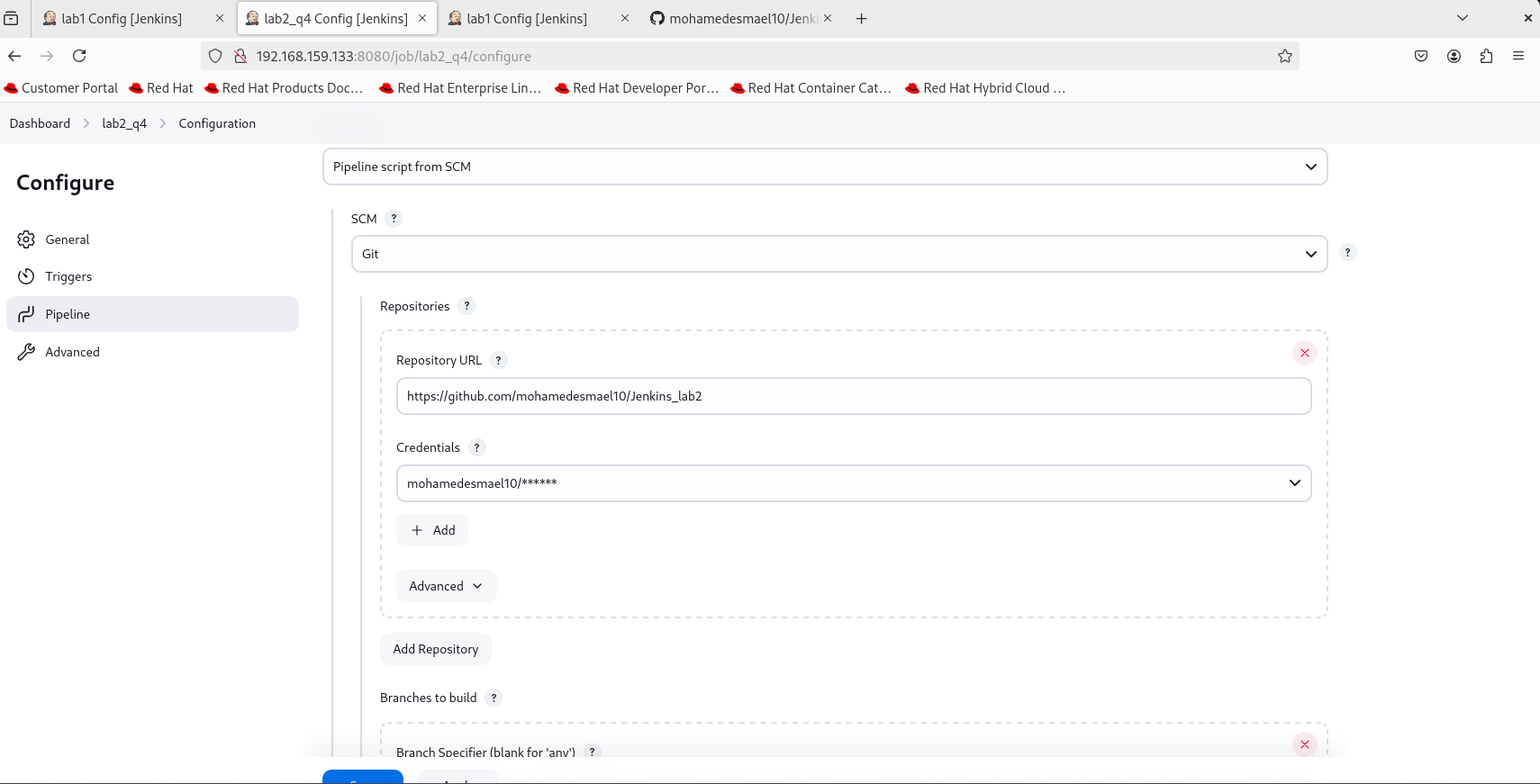
* **Declarative Pipeline:**  
  A more structured and opinionated approach that uses a predefined block syntax.
* **Scripted Pipeline:**  
  Provides more flexibility with standard Groovy code, giving you complete programmatic control.

## 04) Create jenkins pipeline for your repo and use script file (jenkinsfile) to write pipeline syntax ?

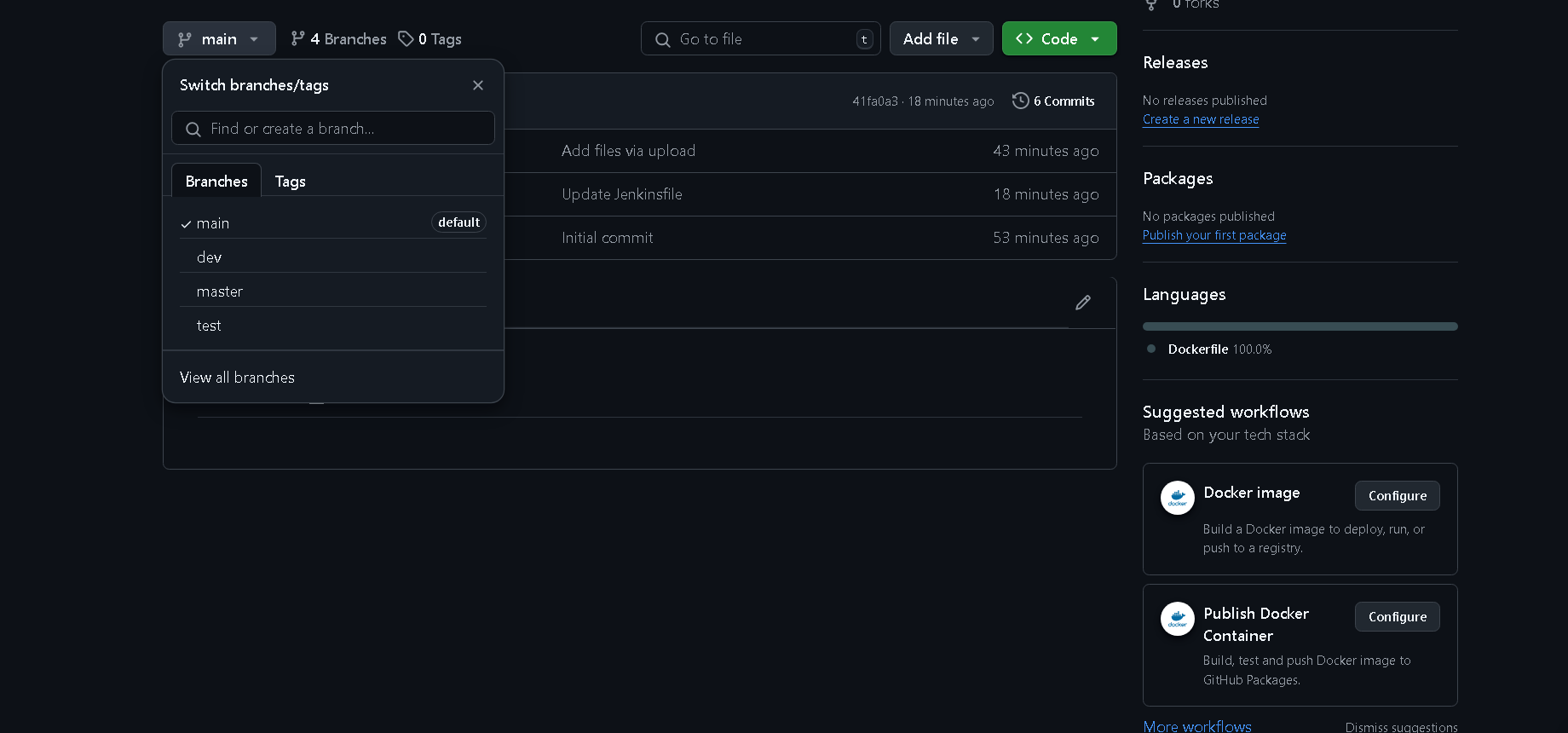


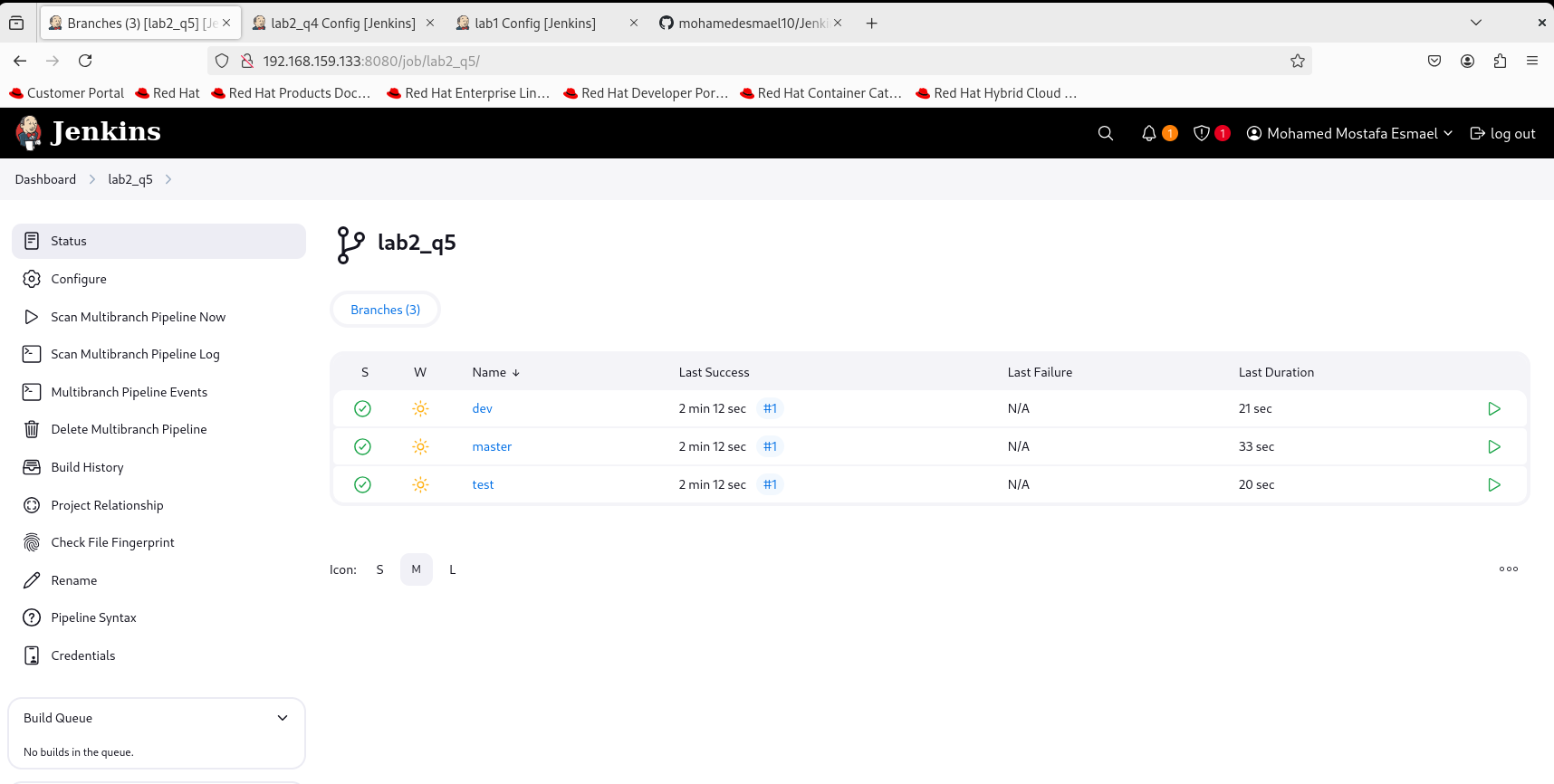
A screenshot of a computer

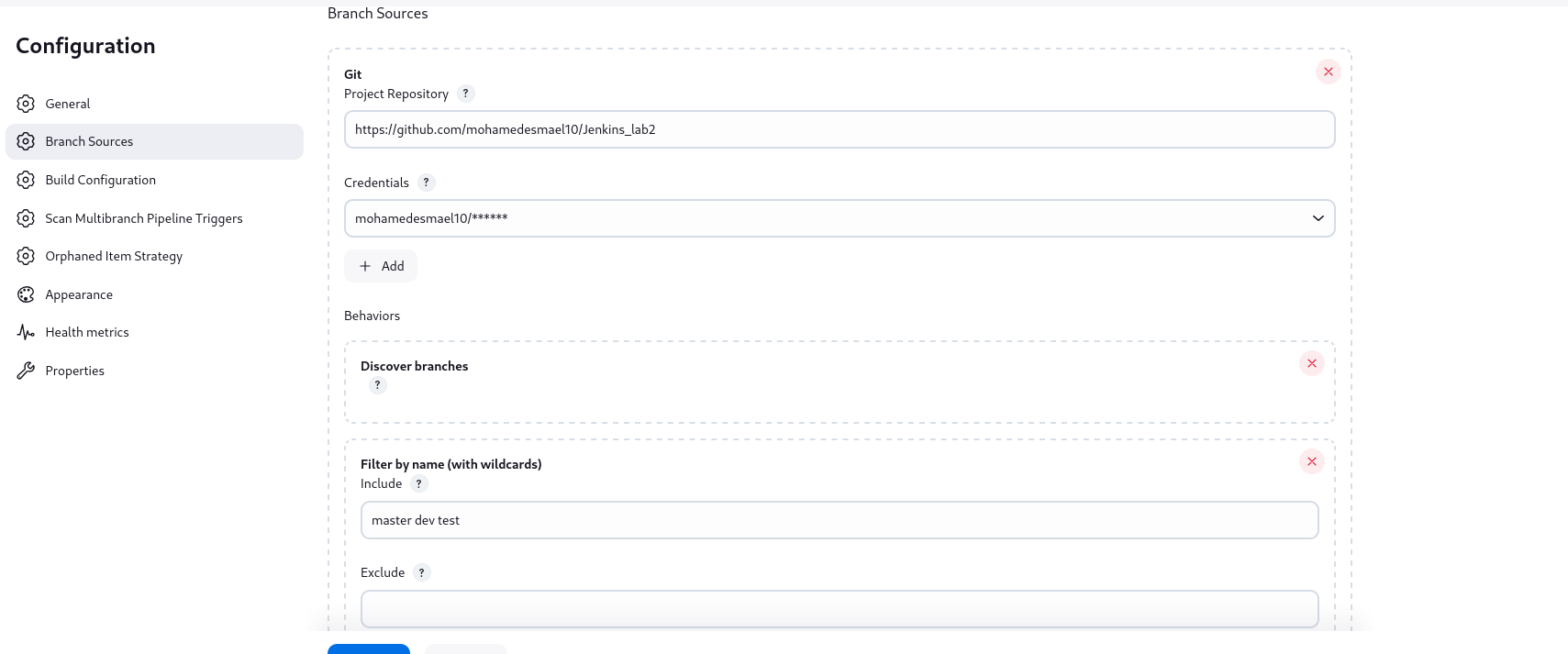
AI-generated content may be incorrect.



## Create another multibranch pipeline and filter branches to contain only (master , dev , test )?

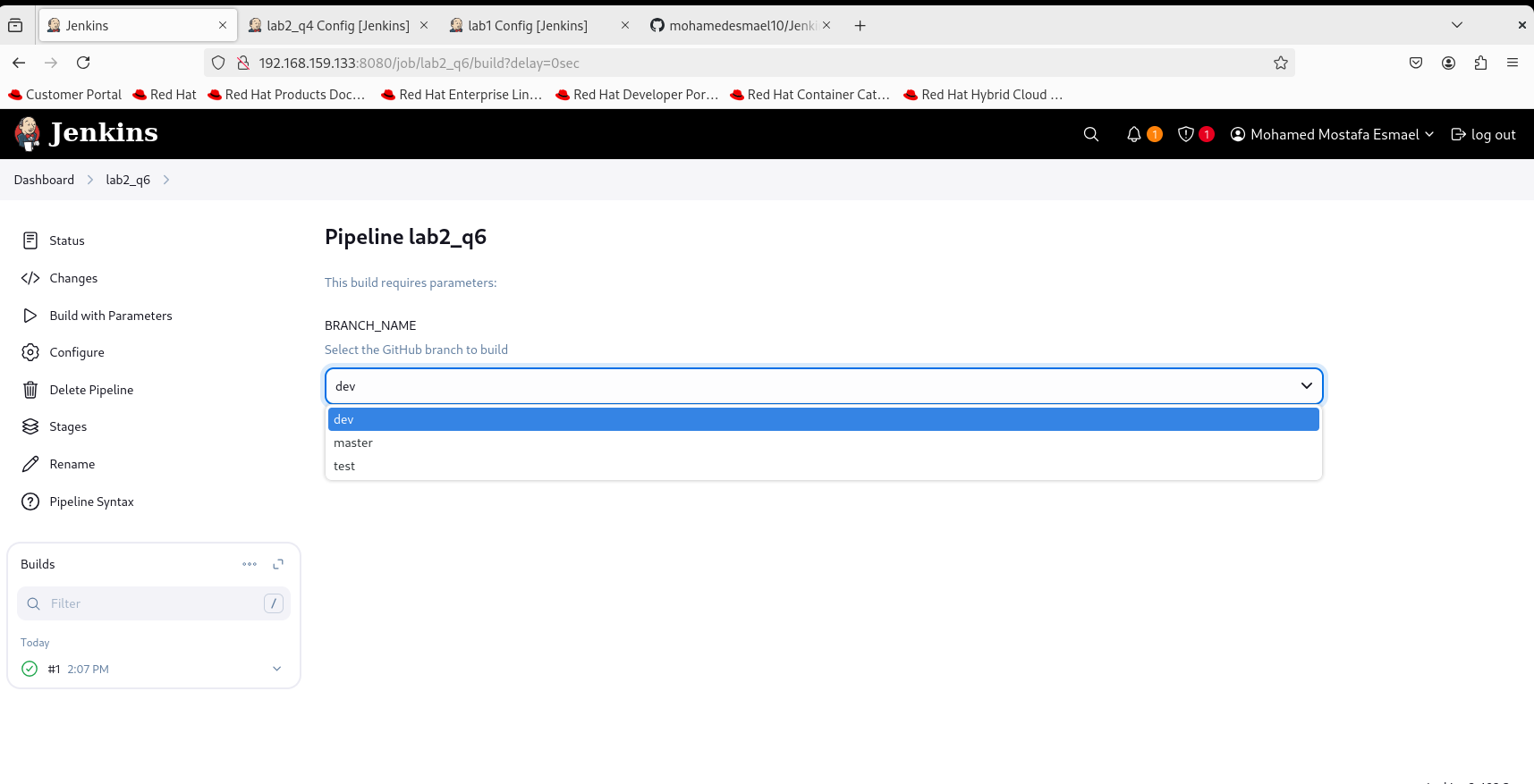






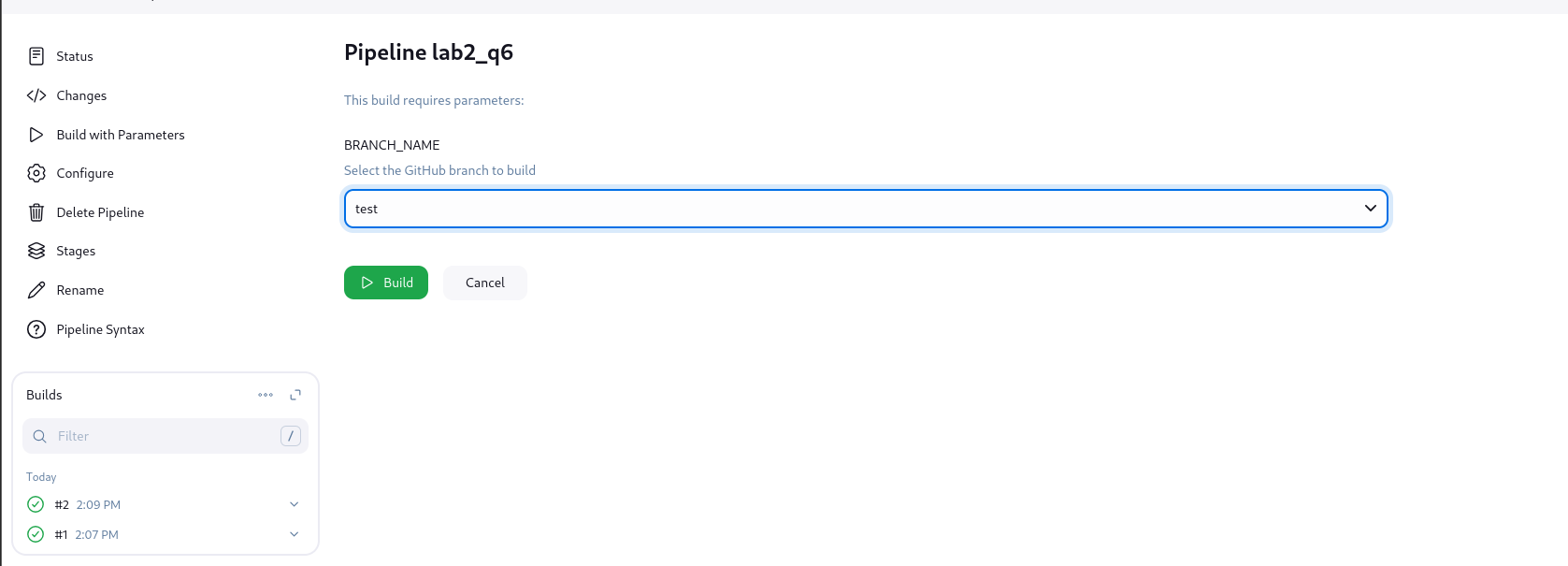
## Create another multibranch pipeline and filter branches to contain only (master , dev , test ) ?

A screenshot of a computer

AI-generated content may be incorrect.

The image is tagged differently depending on the branch:

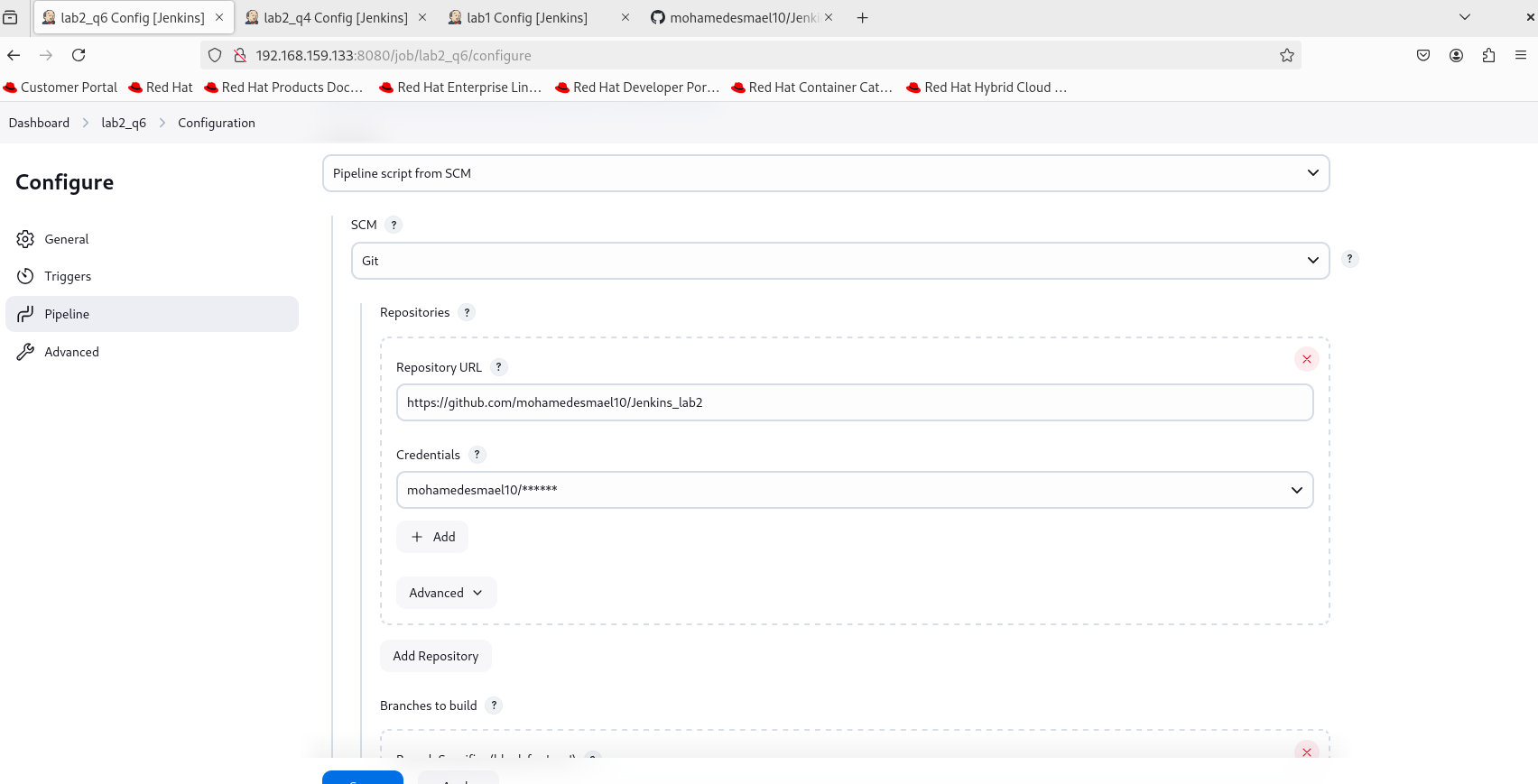
* **dev** → tag v4
* **master** → tag v5
* **test** → tag v3



A screenshot of a computer

AI-generated content may be incorrect.

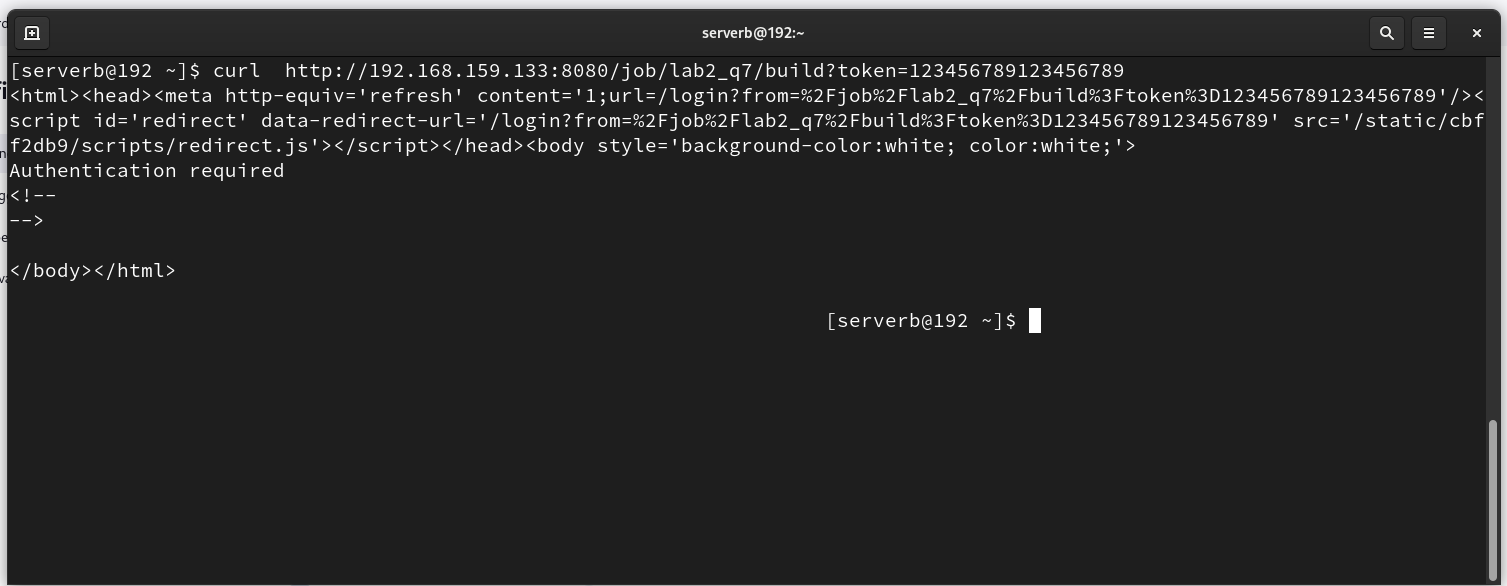


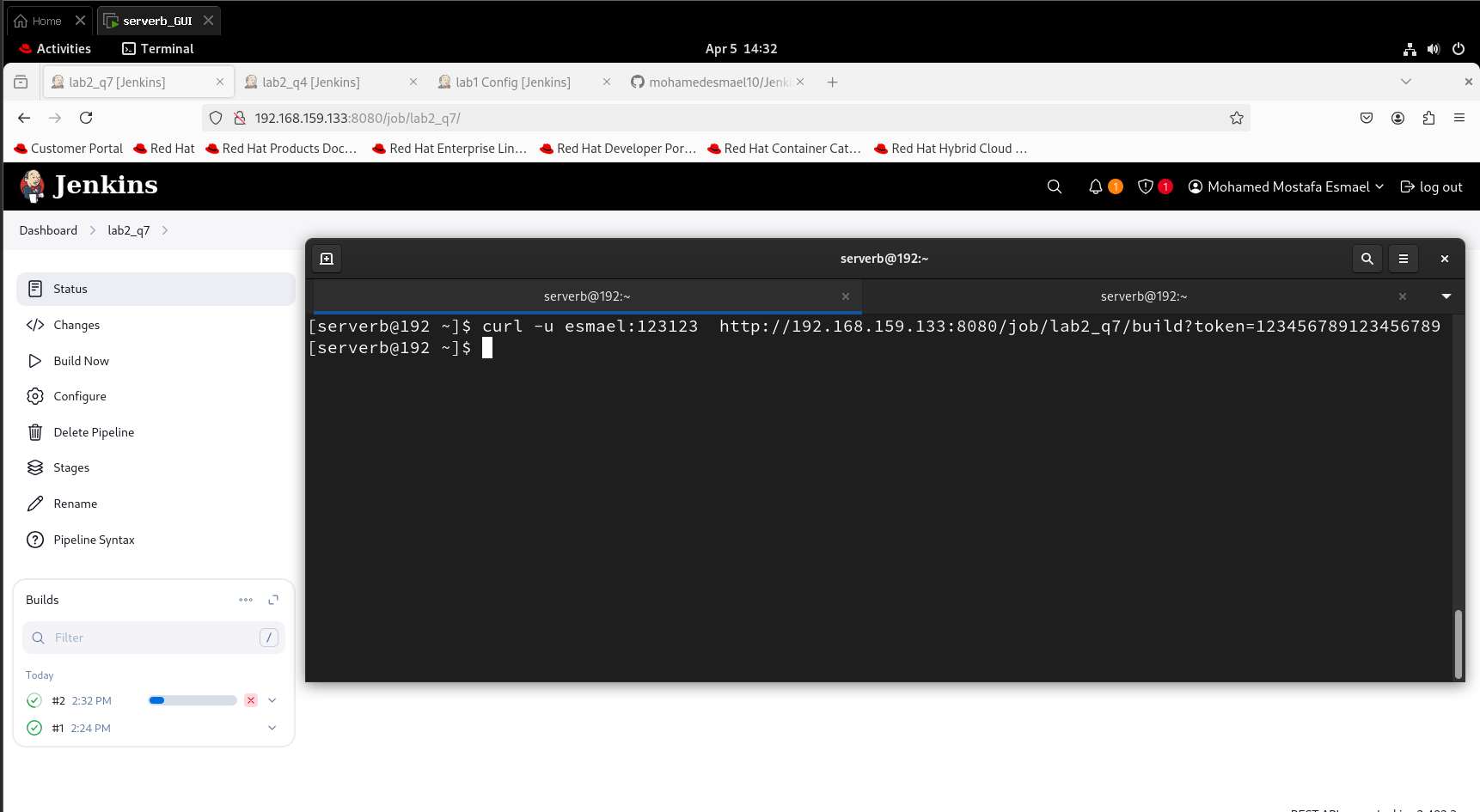


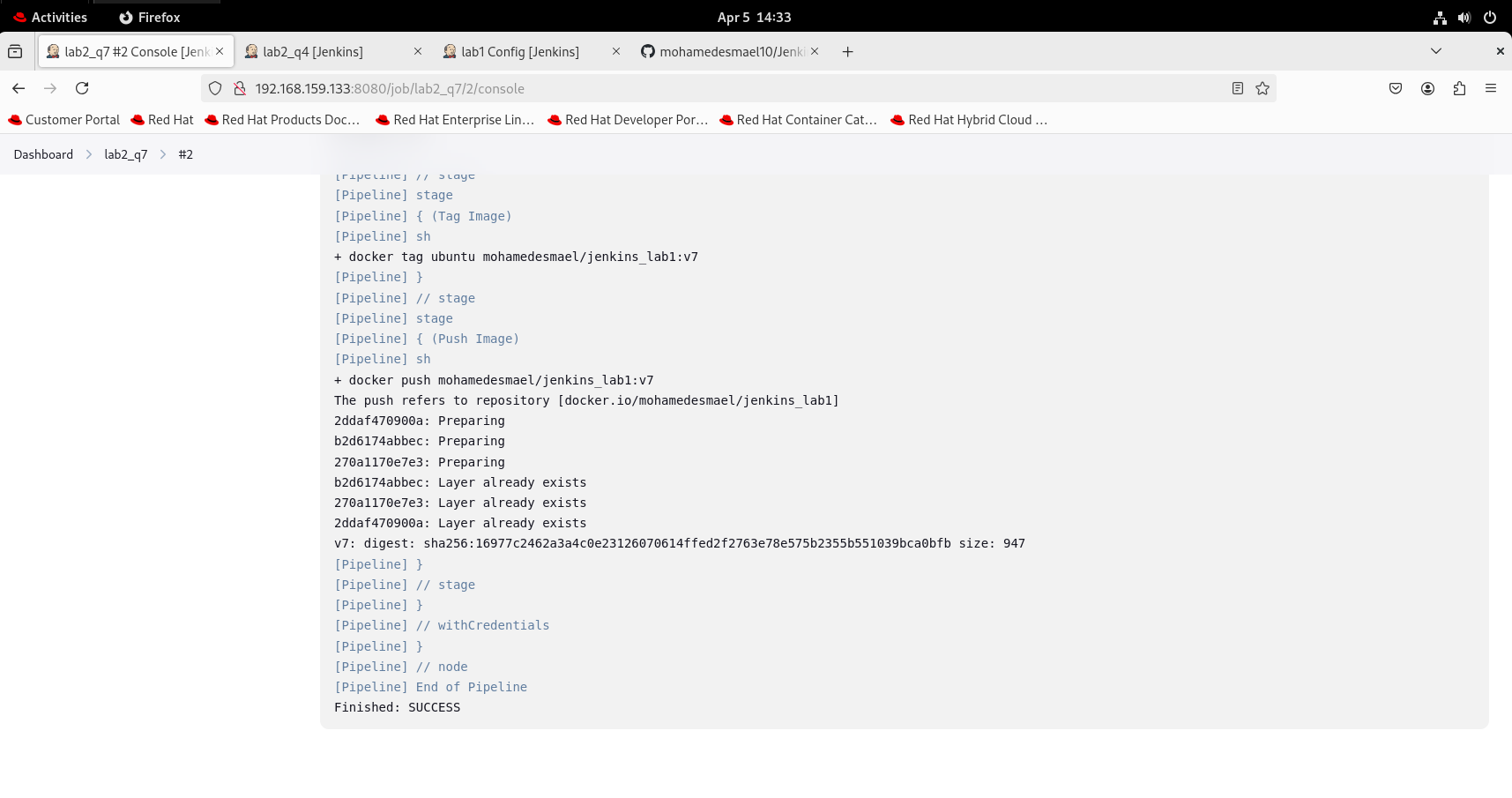
A white page with blue lines

AI-generated content may be incorrect.

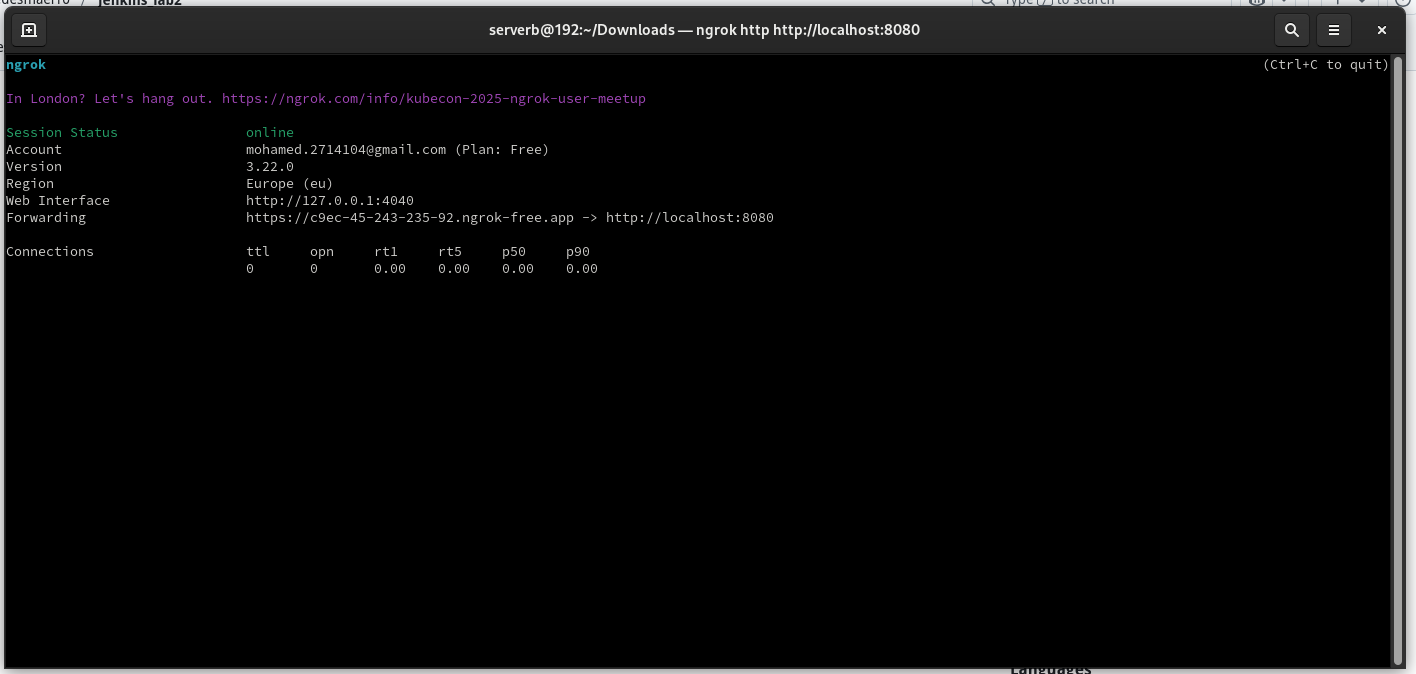
## trigger jenkins pipeline from terminal

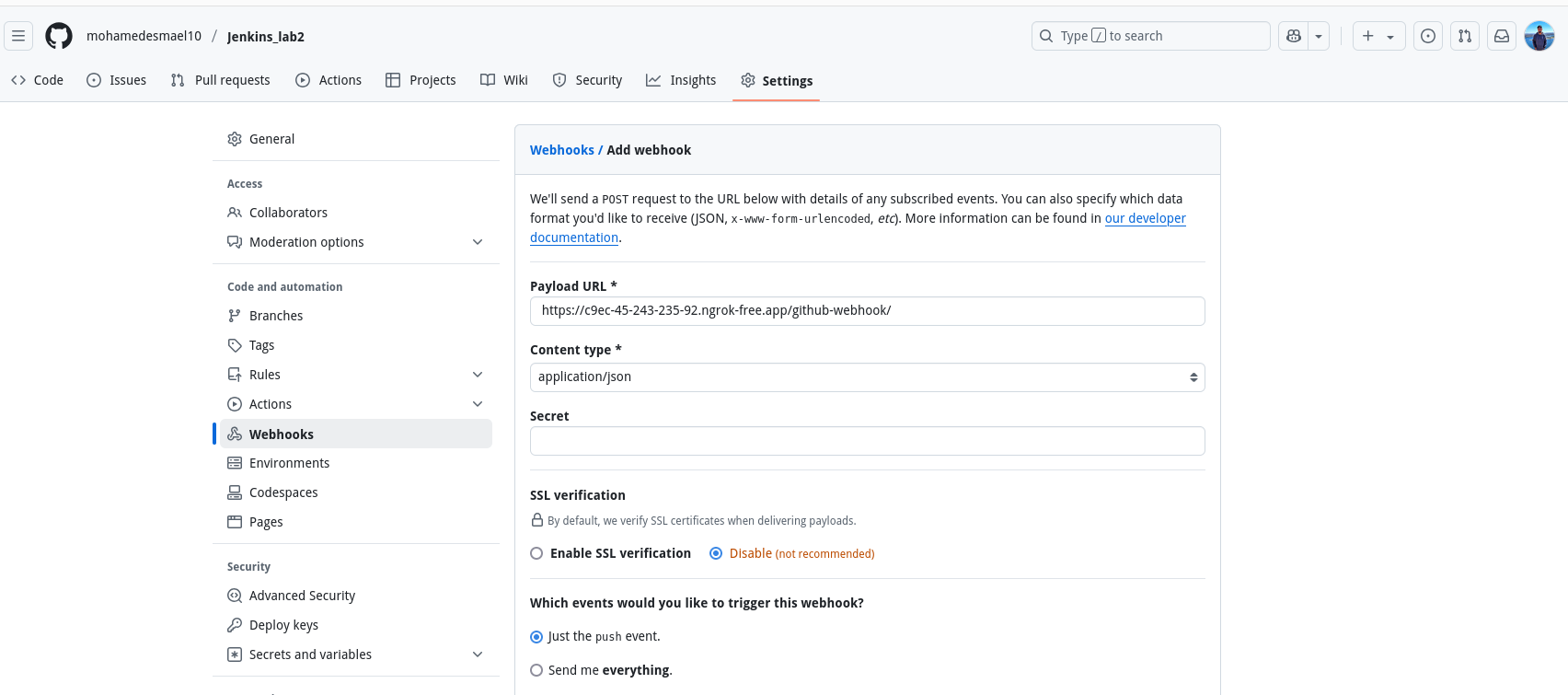
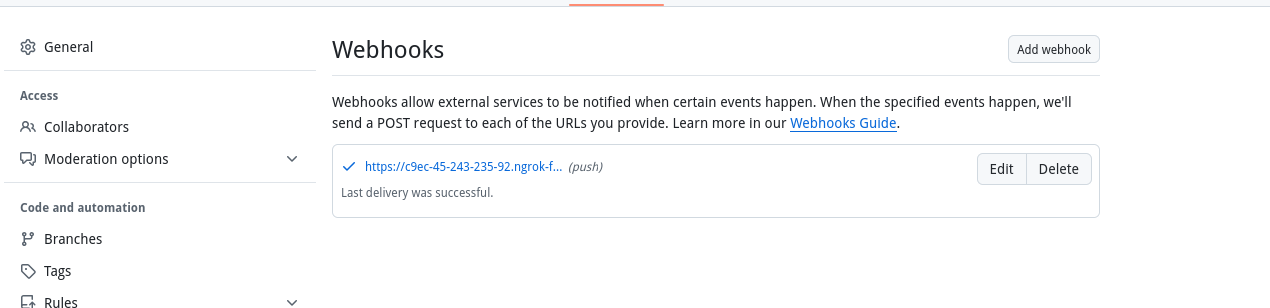


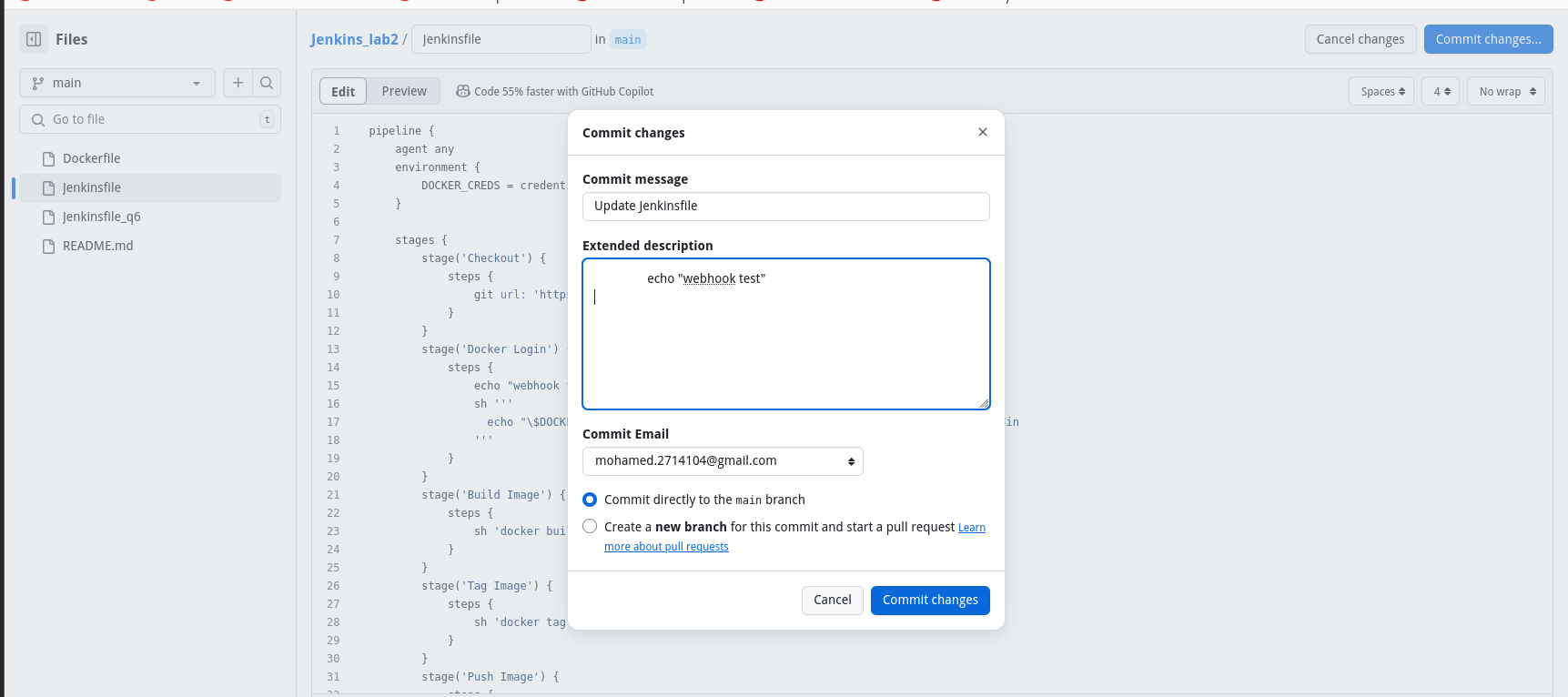




## 08) activate webhook in github to send commits to jenkins

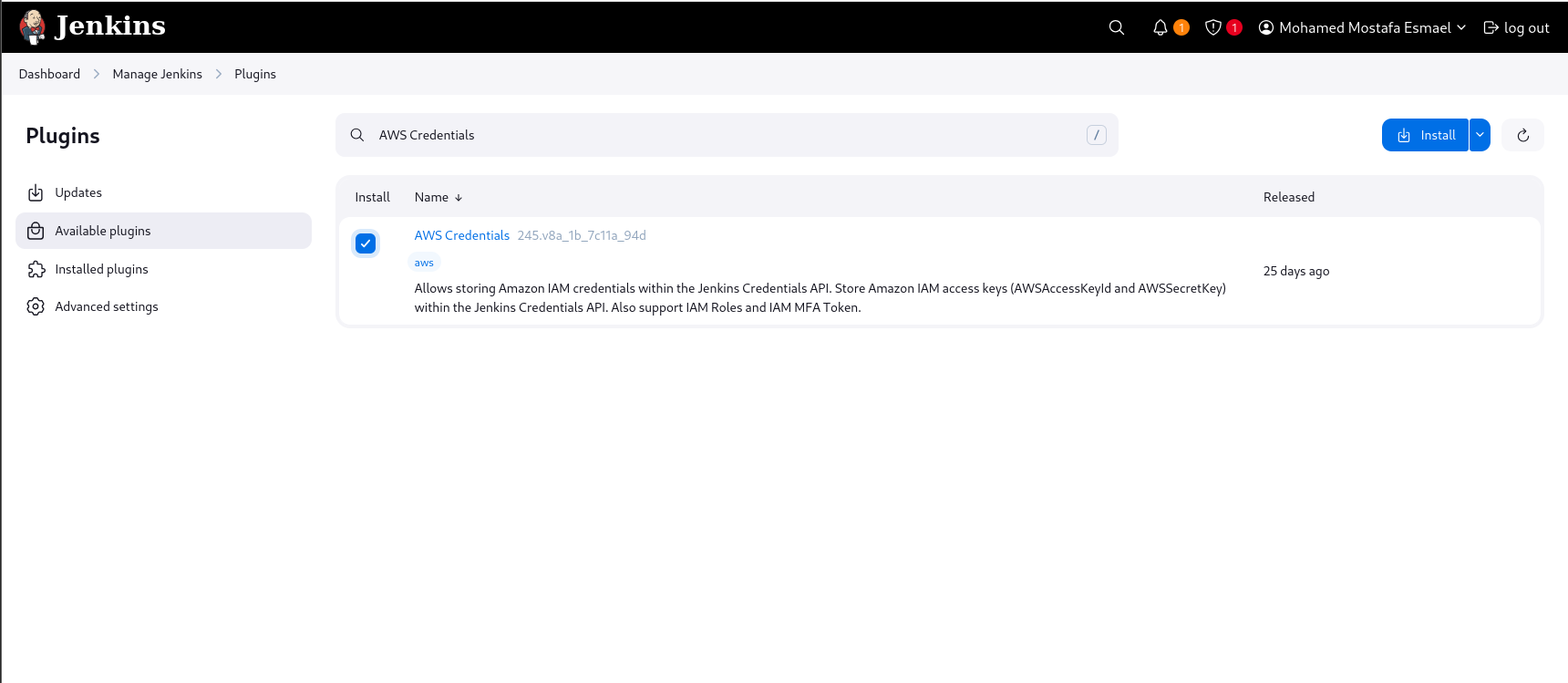
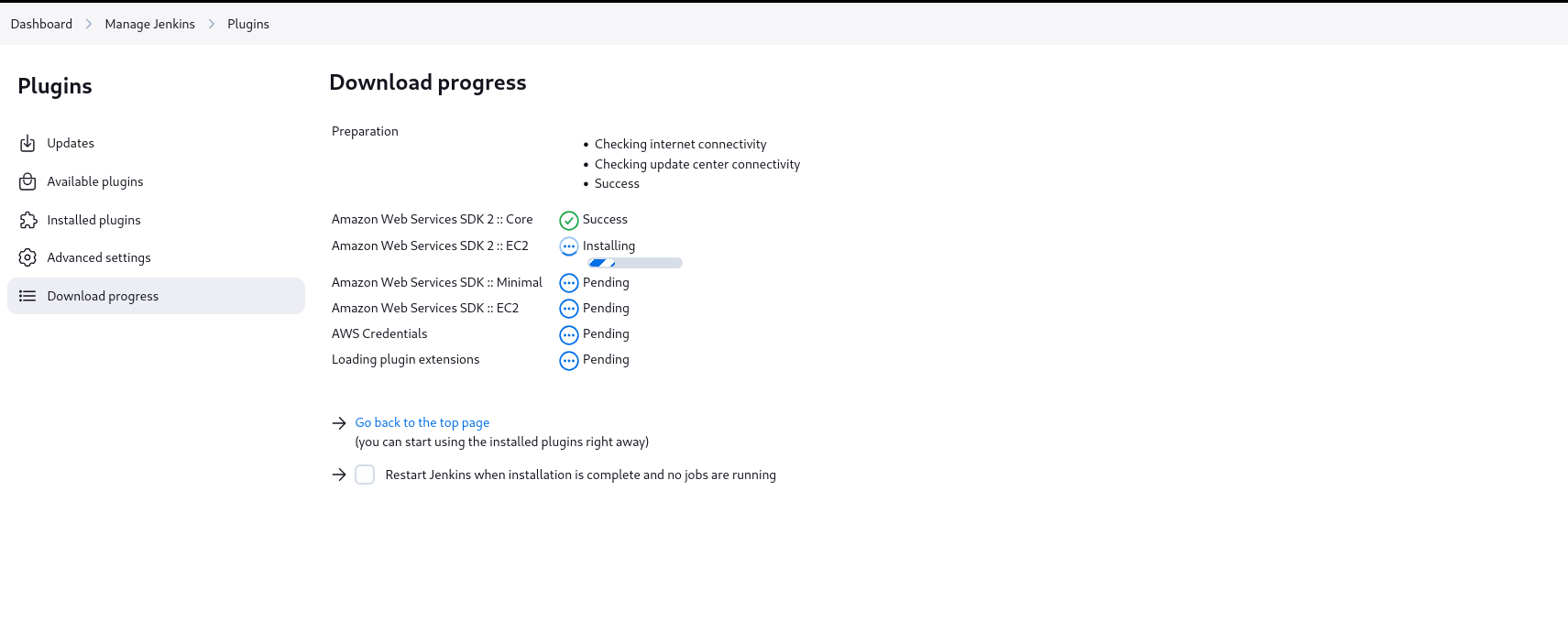


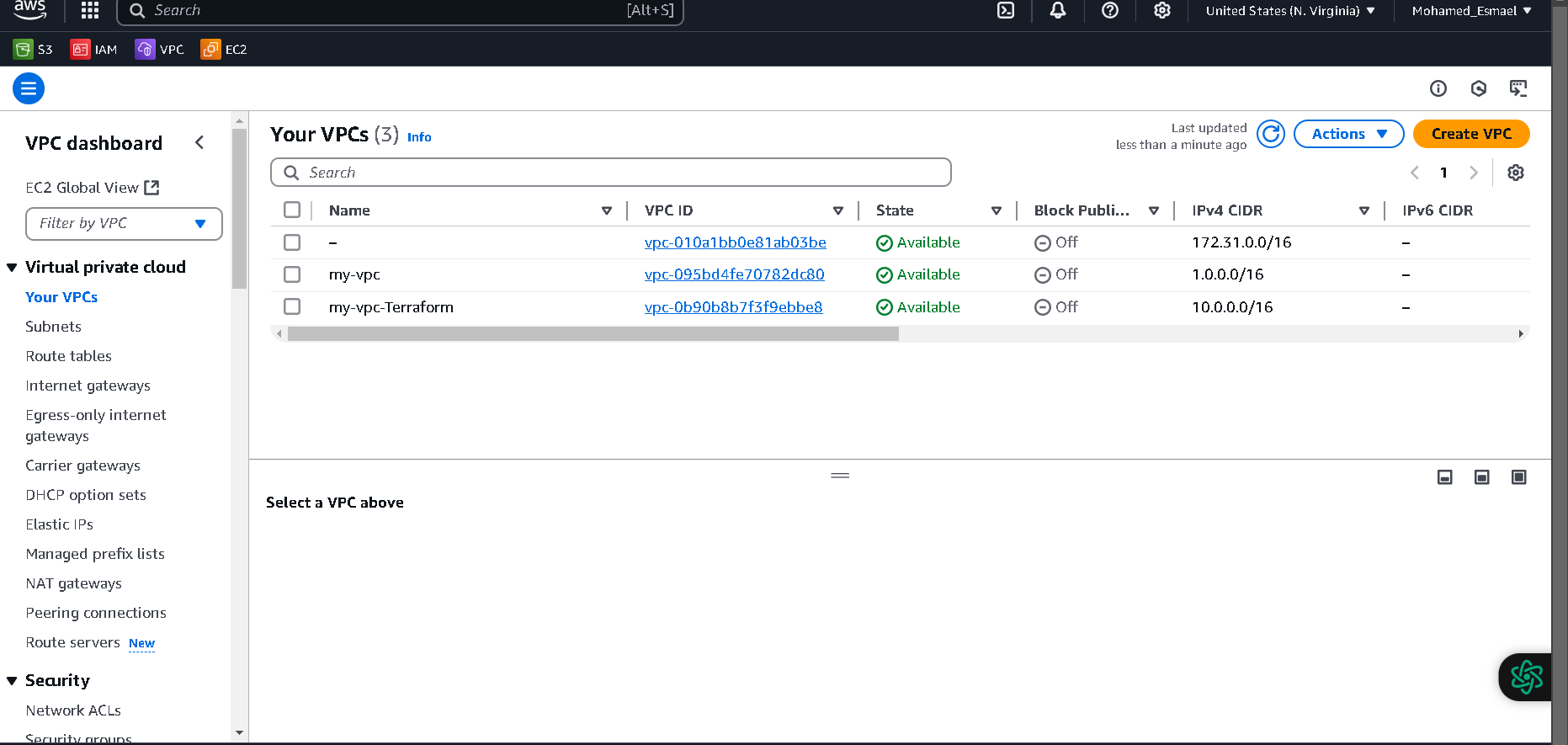


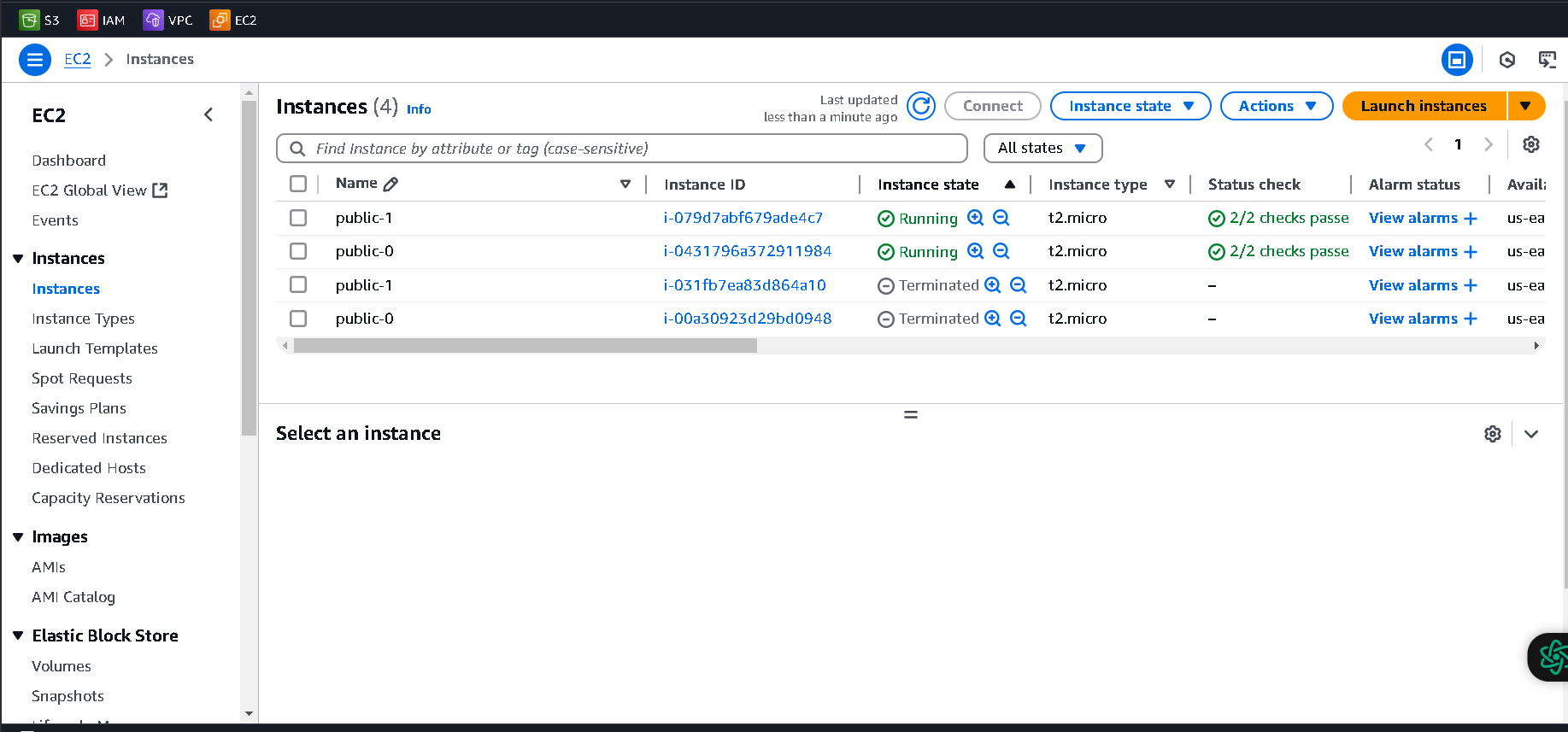


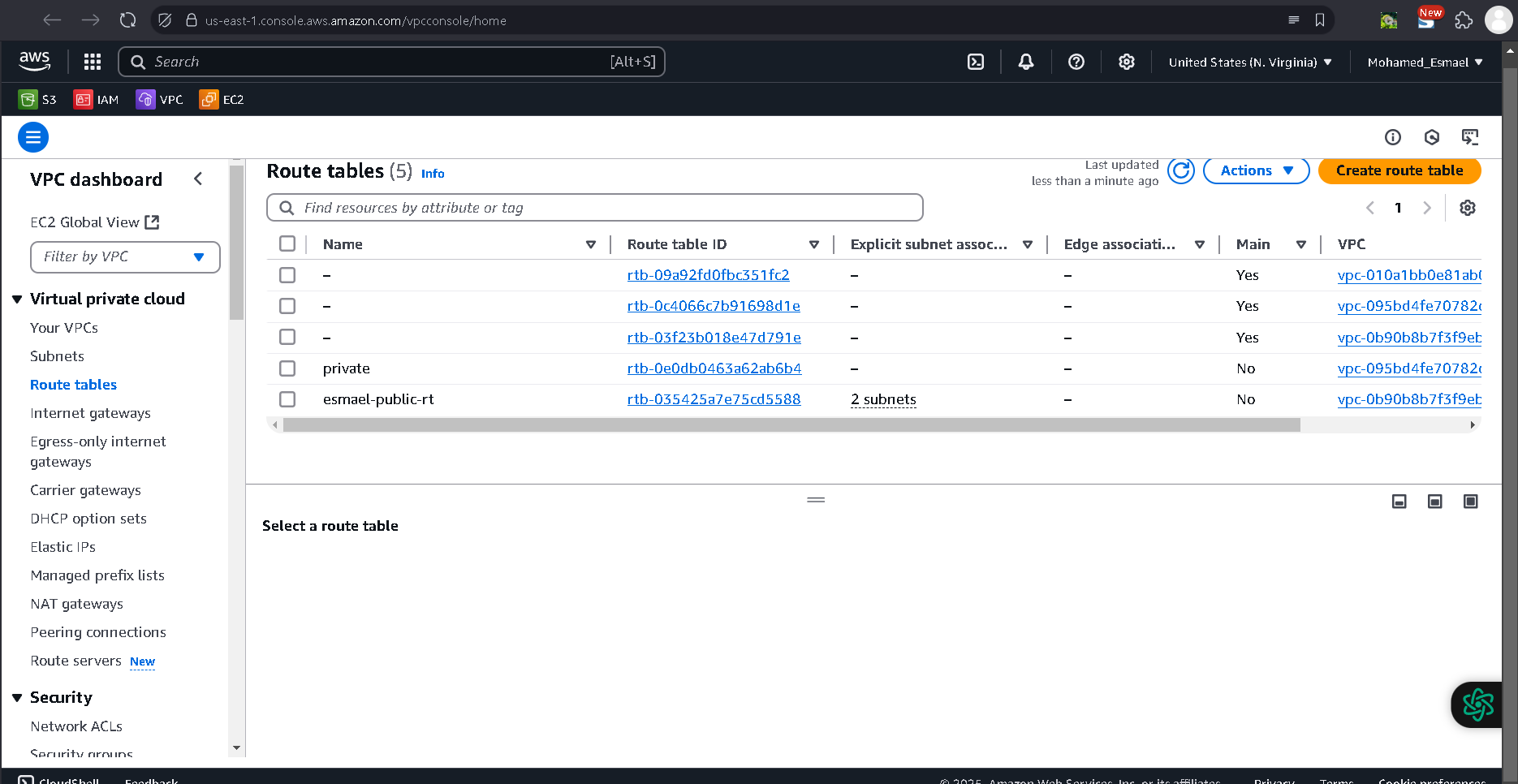


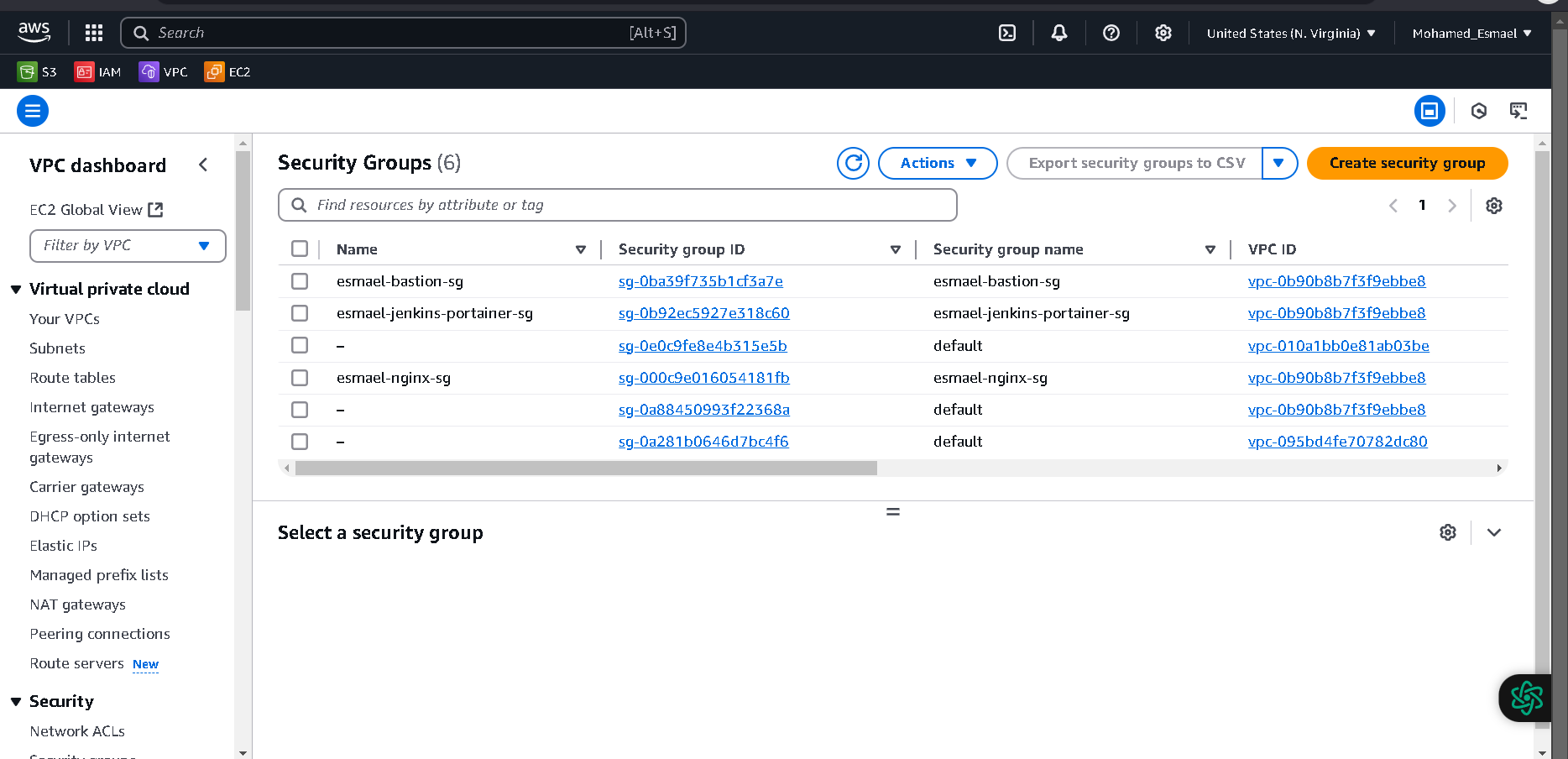
## 09) run pipeline to make ec2 using terraform and by ansible run any script on it all steps done through pipeline

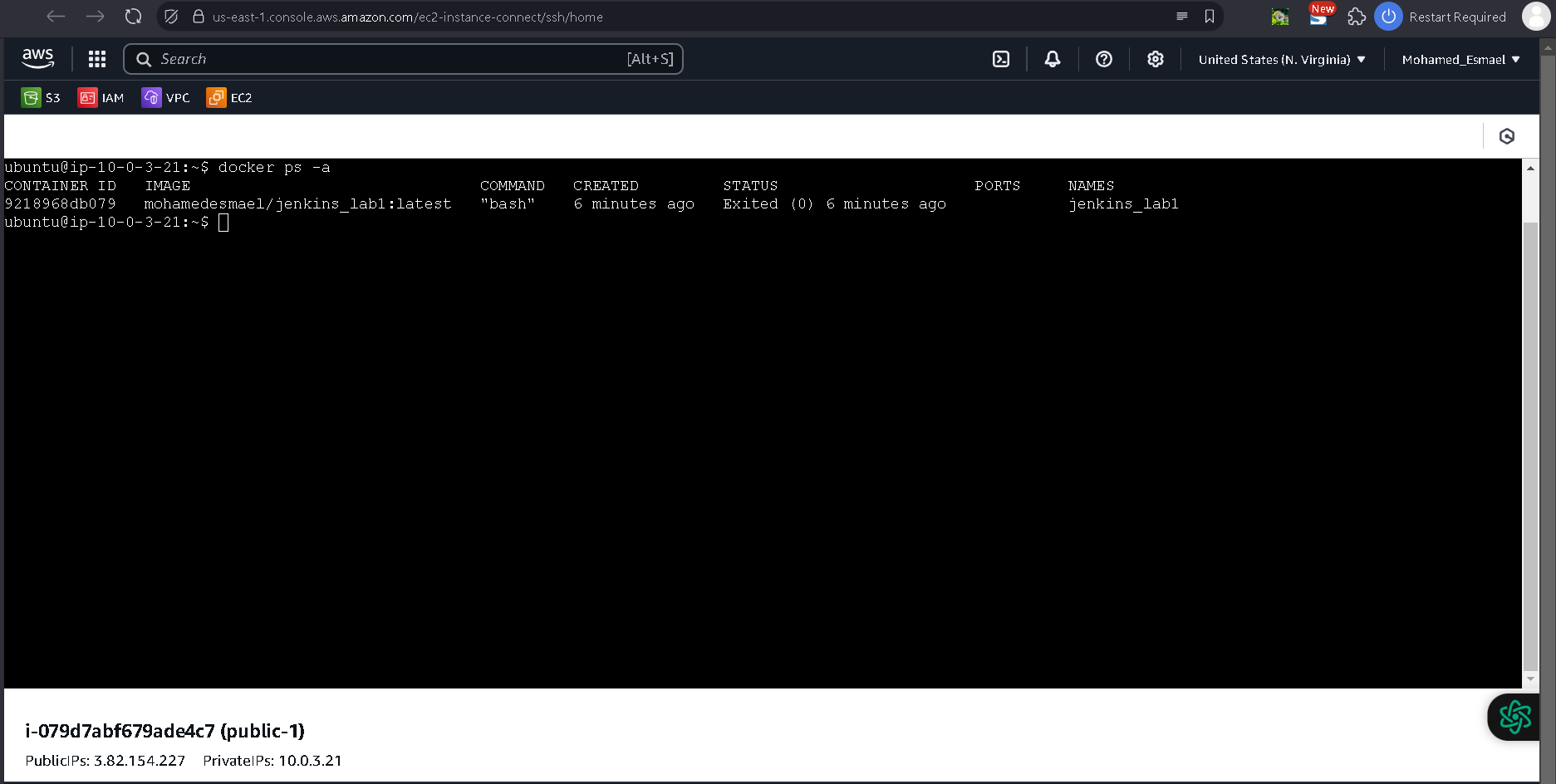


A screenshot of a computer

AI-generated content may be incorrect.

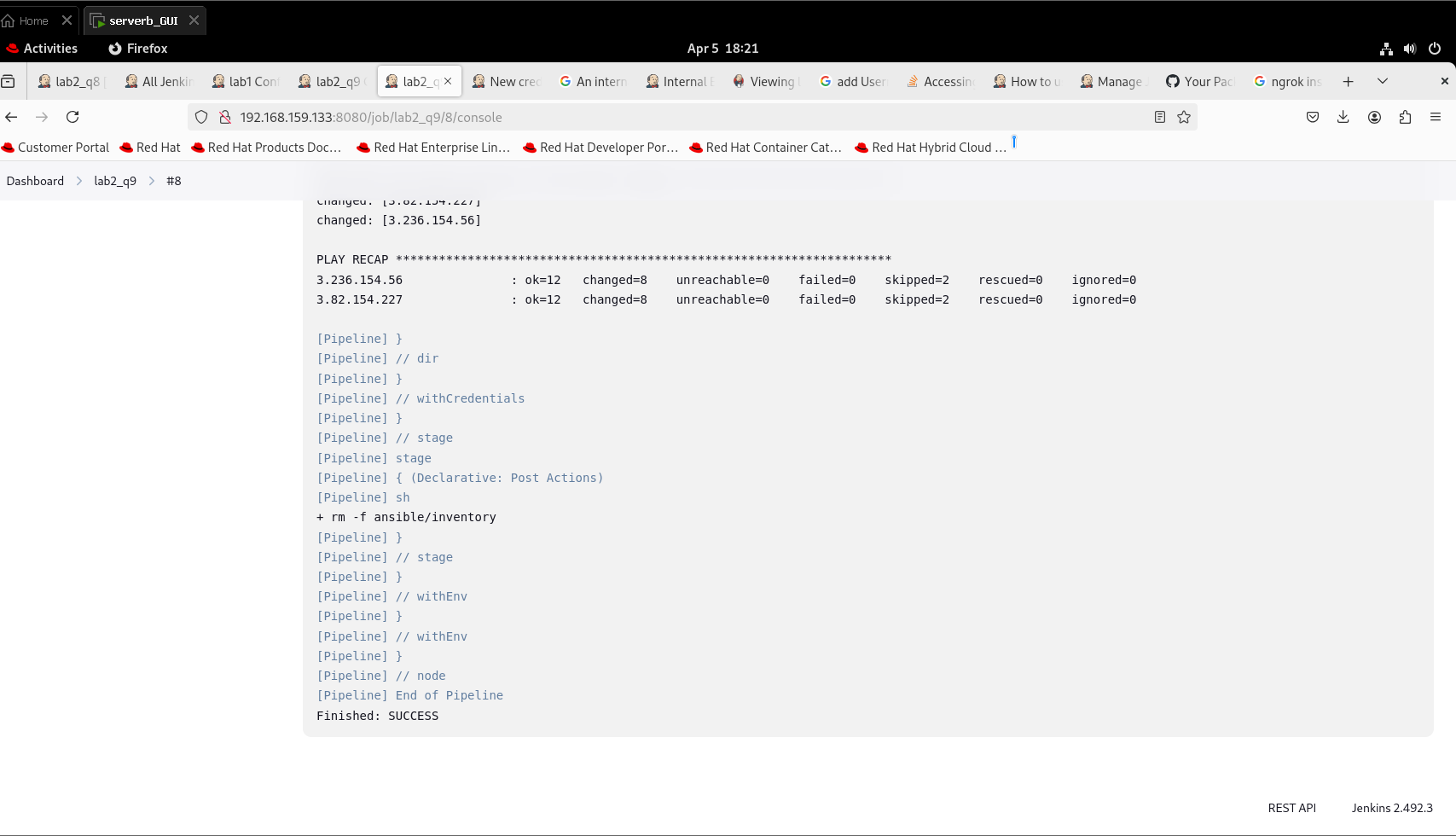


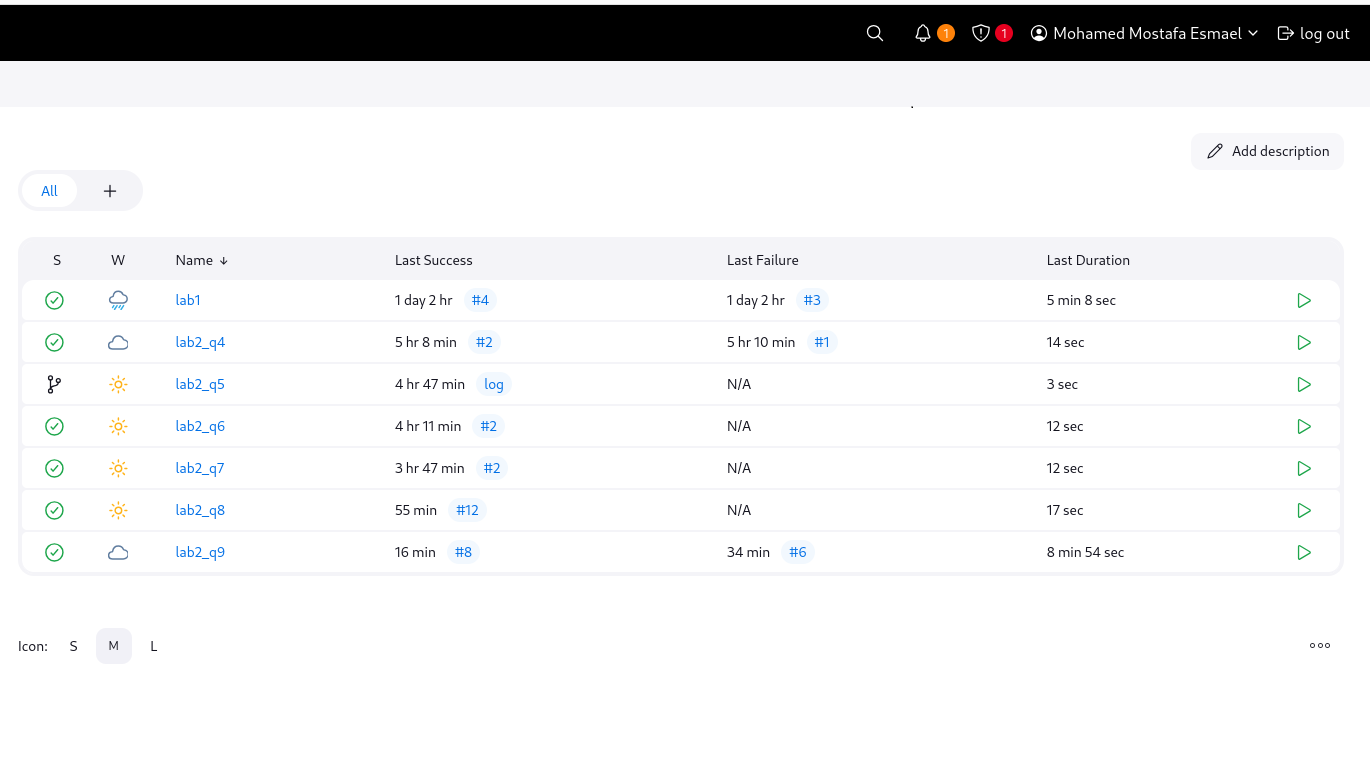




A screenshot of a computer

AI-generated content may be incorrect.





#### Console Output:

Started by user Mohamed Mostafa Esmael

Obtained Jenkinsfile from git https://github.com/mohamedesmael10/Jenkins\_lab2

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins in /var/lib/jenkins/workspace/lab2\_q9

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential mohamedesmael\_dockerhub

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/lab2\_q9/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url https://github.com/mohamedesmael10/Jenkins\_lab2 # timeout=10

Fetching upstream changes from https://github.com/mohamedesmael10/Jenkins\_lab2

> git --version # timeout=10

> git --version # 'git version 2.43.5'

using GIT\_ASKPASS to set credentials

> git fetch --tags --force --progress -- https://github.com/mohamedesmael10/Jenkins\_lab2 +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/terraform\_ansible^{commit} # timeout=10

Checking out Revision 568db1d32f6510d3c238926f94b21a44aba88d4e (refs/remotes/origin/terraform\_ansible)

> git config core.sparsecheckout # timeout=10

> git checkout -f 568db1d32f6510d3c238926f94b21a44aba88d4e # timeout=10

Commit message: "Update react.yaml"

> git rev-list --no-walk 8170cf580d17c2689be3a74c93c3788dfe6d0cff # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Docker Build)

[Pipeline] sh

+ docker build Docker/. -t mohamedesmael/jenkins\_lab1:7

#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile

#1 transferring dockerfile: 300B 0.0s done

#1 DONE 0.1s

#2 [internal] load metadata for docker.io/library/ubuntu:22.04

#2 ...

#3 [auth] library/ubuntu:pull token for registry-1.docker.io

#3 DONE 0.0s

#2 [internal] load metadata for docker.io/library/ubuntu:22.04

#2 DONE 1.8s

#4 [internal] load .dockerignore

#4 transferring context: 2B done

#4 DONE 0.0s

#5 [1/3] FROM docker.io/library/ubuntu:22.04@sha256:ed1544e454989078f5dec1bfdabd8c5cc9c48e0705d07b678ab6ae3fb61952d2

#5 DONE 0.0s

#6 [2/3] RUN apt-get update && apt-get install -y --no-install-recommends curl vim && apt-get clean && rm -rf /var/lib/apt/lists/\*

#6 CACHED

#7 [3/3] WORKDIR /app

#7 CACHED

#8 exporting to image

#8 exporting layers done

#8 writing image sha256:e53e4c1d788b14080bac22aa68d53f8109c52f4e56515097f117a8e2a7fd7b81 done

#8 naming to docker.io/mohamedesmael/jenkins\_lab1:7 done

#8 DONE 0.0s

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Docker Tag as Latest)

[Pipeline] sh

+ docker tag mohamedesmael/jenkins\_lab1:7 mohamedesmael/jenkins\_lab1:latest

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Docker Push)

[Pipeline] withCredentials

Masking supported pattern matches of $DOCKER\_PASSWORD

[Pipeline] {

[Pipeline] sh

+ echo \*\*\*\*

+ docker login -u mohamedesmael --password-stdin

Login Succeeded

+ docker push mohamedesmael/jenkins\_lab1:7

The push refers to repository [docker.io/mohamedesmael/jenkins\_lab1]

2ddaf470900a: Preparing

b2d6174abbec: Preparing

270a1170e7e3: Preparing

270a1170e7e3: Layer already exists

2ddaf470900a: Layer already exists

b2d6174abbec: Layer already exists

7: digest: sha256:16977c2462a3a4c0e23126070614ffed2f2763e78e575b2355b551039bca0bfb size: 947

+ docker push mohamedesmael/jenkins\_lab1:latest

The push refers to repository [docker.io/mohamedesmael/jenkins\_lab1]

2ddaf470900a: Preparing

b2d6174abbec: Preparing

270a1170e7e3: Preparing

2ddaf470900a: Layer already exists

270a1170e7e3: Layer already exists

b2d6174abbec: Layer already exists

latest: digest: sha256:16977c2462a3a4c0e23126070614ffed2f2763e78e575b2355b551039bca0bfb size: 947

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Run Terraform)

[Pipeline] withCredentials

Masking supported pattern matches of $AWS\_ACCESS\_KEY\_ID or $AWS\_SECRET\_ACCESS\_KEY

[Pipeline] {

[Pipeline] dir

Running in /var/lib/jenkins/workspace/lab2\_q9/terraform

[Pipeline] {

[Pipeline] sh

+ export AWS\_ACCESS\_KEY\_ID=\*\*\*\*

+ AWS\_ACCESS\_KEY\_ID=\*\*\*\*

+ export AWS\_SECRET\_ACCESS\_KEY=\*\*\*\*

+ AWS\_SECRET\_ACCESS\_KEY=\*\*\*\*

+ terraform init

[0m[1mInitializing the backend...[0m

[0m[1mInitializing modules...[0m

[0m[1mInitializing provider plugins...[0m

- Reusing previous version of hashicorp/aws from the dependency lock file

- Reusing previous version of hashicorp/tls from the dependency lock file

- Reusing previous version of hashicorp/local from the dependency lock file

- Using previously-installed hashicorp/aws v5.94.1

- Using previously-installed hashicorp/tls v4.0.6

- Using previously-installed hashicorp/local v2.5.2

[0m[1m[32mTerraform has been successfully initialized![0m[32m[0m

[0m[32m

You may now begin working with Terraform. Try running "terraform plan" to see

any changes that are required for your infrastructure. All Terraform commands

should now work.

If you ever set or change modules or backend configuration for Terraform,

rerun this command to reinitialize your working directory. If you forget, other

commands will detect it and remind you to do so if necessary.[0m

+ terraform apply -auto-approve

[0m[1mmodule.vpc.aws\_vpc.main: Refreshing state... [id=vpc-02397187b2e5abc0a][0m

[0m[1mmodule.internet\_gateway.aws\_internet\_gateway.gw: Refreshing state... [id=igw-05d355c59eb19dbf6][0m

[0m[1mmodule.bastion\_security\_group.aws\_security\_group.sg: Refreshing state... [id=sg-0d25d2e89240a81dd][0m

[0m[1mmodule.public\_subnet\_1.aws\_subnet.subnets: Refreshing state... [id=subnet-0a8d4f80929cb96f8][0m

[0m[1mmodule.jenkins\_portainer\_sg.aws\_security\_group.sg: Refreshing state... [id=sg-03709420fdeaa5f82][0m

[0m[1mmodule.public\_subnet\_2.aws\_subnet.subnets: Refreshing state... [id=subnet-01d07981ce20a6381][0m

[0m[1mmodule.key.tls\_private\_key.rsa: Refreshing state... [id=5981330f47e7f2d4119081807692ab306c171911][0m

[0m[1mmodule.nginx\_security\_group.aws\_security\_group.sg: Refreshing state... [id=sg-0edb945be1ff5a935][0m

[0m[1mmodule.key.aws\_key\_pair.TF\_key: Refreshing state... [id=key-pair][0m

[0m[1mmodule.key.local\_file.TF-key: Refreshing state... [id=0ea4cea05ad57a9a0a2373ffff89da1e927c316b][0m

[0m[1mmodule.load\_balancer.aws\_lb\_target\_group.lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:025066251600:targetgroup/my-target-group/10c2f50cf97d4cb6][0m

[0m[1mmodule.public\_route\_table.aws\_route\_table.public: Refreshing state... [id=rtb-0d03d9932f76f14e8][0m

[0m[1mmodule.public\_instances.aws\_instance.instance[0]: Refreshing state... [id=i-00a30923d29bd0948][0m

[0m[1mmodule.public\_instances.aws\_instance.instance[1]: Refreshing state... [id=i-031fb7ea83d864a10][0m

[0m[1mmodule.load\_balancer.aws\_lb.lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:025066251600:loadbalancer/app/my-load-balancer/d490abb3b715419a][0m

[0m[1mmodule.public\_route\_table.aws\_route\_table\_association.public\_association[1]: Refreshing state... [id=rtbassoc-06d27a5e3cadf1122][0m

[0m[1mmodule.public\_route\_table.aws\_route\_table\_association.public\_association[0]: Refreshing state... [id=rtbassoc-0183daf0fe520e38f][0m

[0m[1mmodule.load\_balancer.aws\_lb\_listener.lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:025066251600:listener/app/my-load-balancer/d490abb3b715419a/1ecf9e75211640cf][0m

[0m[1mmodule.load\_balancer.aws\_lb\_target\_group\_attachment.lb[1]: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:025066251600:targetgroup/my-target-group/10c2f50cf97d4cb6-20250405141824283900000004][0m

[0m[1mmodule.load\_balancer.aws\_lb\_target\_group\_attachment.lb[0]: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:025066251600:targetgroup/my-target-group/10c2f50cf97d4cb6-20250405141824319300000005][0m

[0m[1m[32mNo changes.[0m[1m Your infrastructure matches the configuration.[0m

[0mTerraform has compared your real infrastructure against your configuration

and found no differences, so no changes are needed.

[0m[1m[32m

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

[0m[0m[1m[32m

Outputs:

[0mbastion\_public\_ip\_1 = "3.239.222.193"

bastion\_public\_ip\_2 = "3.84.120.51"

public\_lb\_dns = "my-load-balancer-1067543872.us-east-1.elb.amazonaws.com"

+ chmod +x inventory.sh

+ ./inventory.sh

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Run Ansible)

[Pipeline] withCredentials

Masking supported pattern matches of $AWS\_ACCESS\_KEY\_ID or $AWS\_SECRET\_ACCESS\_KEY

[Pipeline] {

[Pipeline] dir

Running in /var/lib/jenkins/workspace/lab2\_q9/ansible

[Pipeline] {

[Pipeline] sh

+ cat inventory

[web]

3.239.222.193

3.84.120.51

+ ansible-playbook -i inventory react.yaml

PLAY [Setup Docker and run container] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.239.222.193]

ok: [3.84.120.51]

TASK [Update apt cache] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [3.239.222.193]

changed: [3.84.120.51]

TASK [Install prerequisites] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Add Docker GPG key] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [3.84.120.51]

changed: [3.239.222.193]

TASK [Add Docker repository] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Install Docker] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Ensure Docker service is running] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Add user to Docker group] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Restart Docker service] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [3.84.120.51]

changed: [3.239.222.193]

TASK [Pull the latest version of the Docker image] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Check if the container is already running] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [3.84.120.51]

ok: [3.239.222.193]

TASK [Stop the existing Docker container if it is running] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [3.239.222.193]

skipping: [3.84.120.51]

TASK [Remove the existing Docker container if it exists] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [3.239.222.193]

skipping: [3.84.120.51]

TASK [Run the latest version of the Docker image] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [3.84.120.51]

changed: [3.239.222.193]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3.239.222.193 : ok=12 changed=4 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0

3.84.120.51 : ok=12 changed=4 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Declarative: Post Actions)

[Pipeline] sh

+ rm -f ansible/inventory

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS