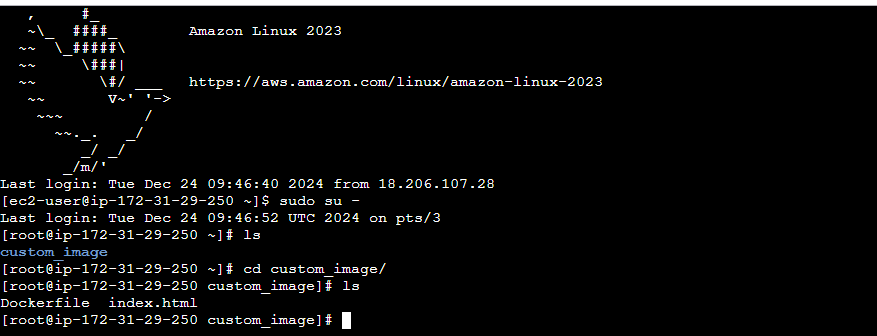
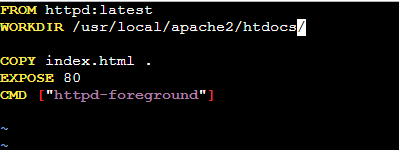
TASK

**1) Create a customized docker image by using Docker file**Created a docker file and index.html file

Also created the index.html file



Index.html

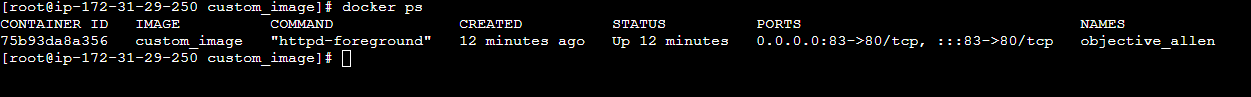


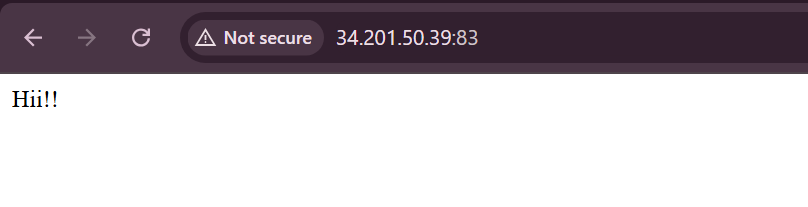
Build the created image with the command

*docker build -t custom\_image .*

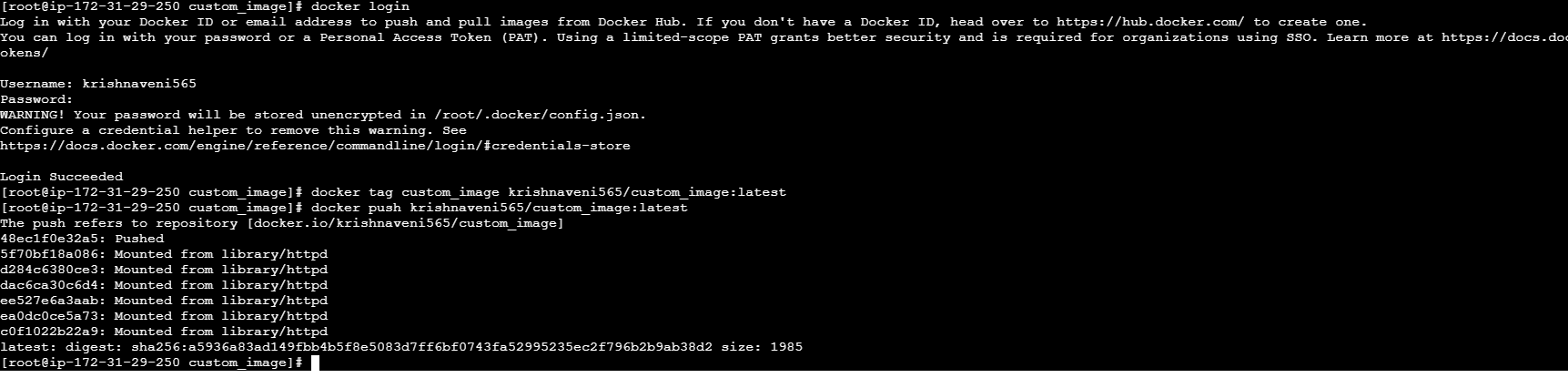
Running the container

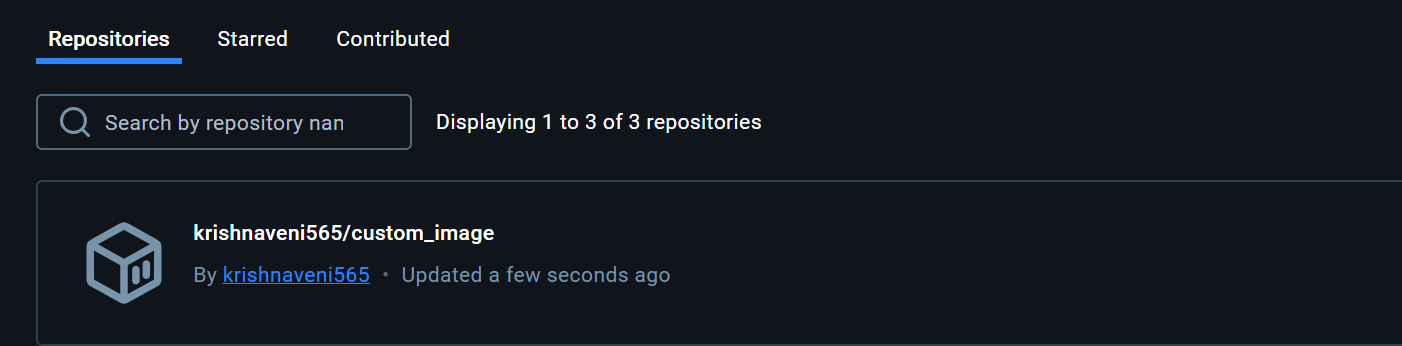
Docker container run –itd –p 83:80 <container\_id>





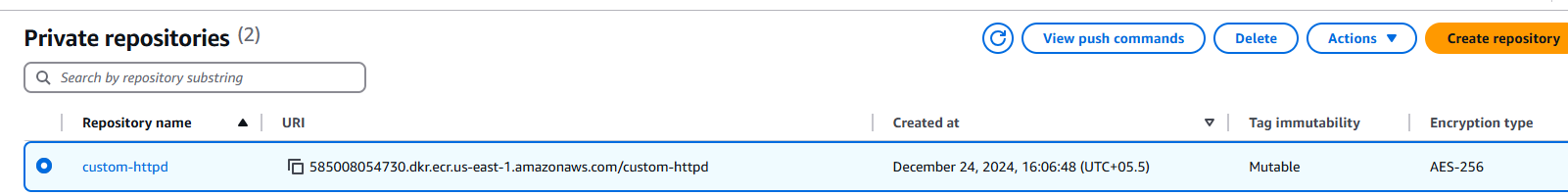
**2) Push the image to docker hub**



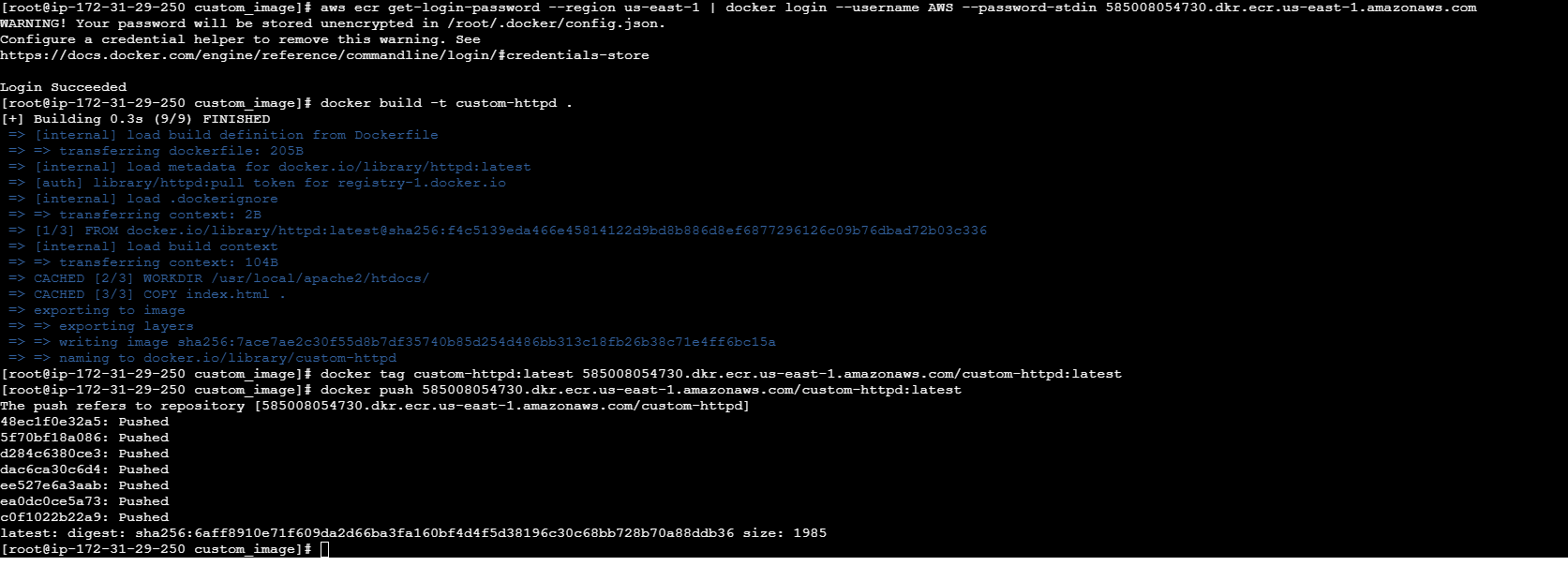


**3) Push the same Image to Amazon ECR**

Created a repo with name custom-httpd



Configured the aws cli



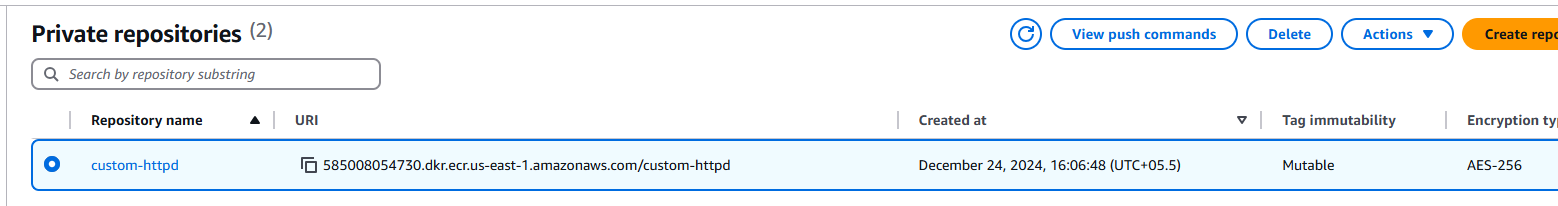
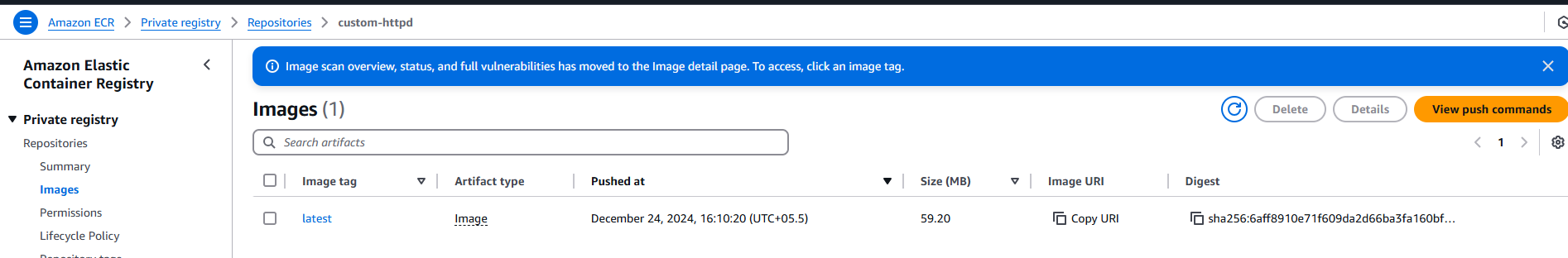


Image has pushed to the ecr



**4) Provision one ec2 using terraform and install Jenkins.**

Created main.tf to launch one ec2 and install jenkins

provider "aws" {

  region = "us-east-1"

}

resource "aws\_instance" "jenkins" {

  ami           = "ami-01816d07b1128cd2d"

  instance\_type = "t2.micro"

  key\_name      = "sample1"

  tags = {

    Name = "Jenkins-Server"

  }

  provisioner "remote-exec" {

    connection {

      type        = "ssh"

      user        = "ec2-user"

      private\_key = file("C:/Users/user/Downloads/sample1.pem")

      host        = self.public\_ip

    }

    inline = [

  "sudo yum update -y",

  "sudo yum install -y java-17-amazon-corretto-devel",

  "sudo yum install -y wget",

  "sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo",

  "sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key || echo 'Key already imported, continuing...'",

  "sudo yum install -y jenkins",

  "sudo systemctl enable jenkins ",

  "sudo systemctl start jenkins"

]

  }

}

output "instance\_public\_ip" {

  value = aws\_instance.jenkins.public\_ip

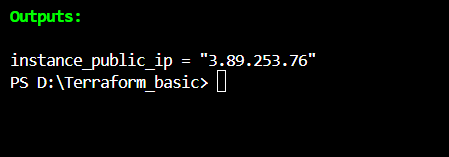
}

Commands

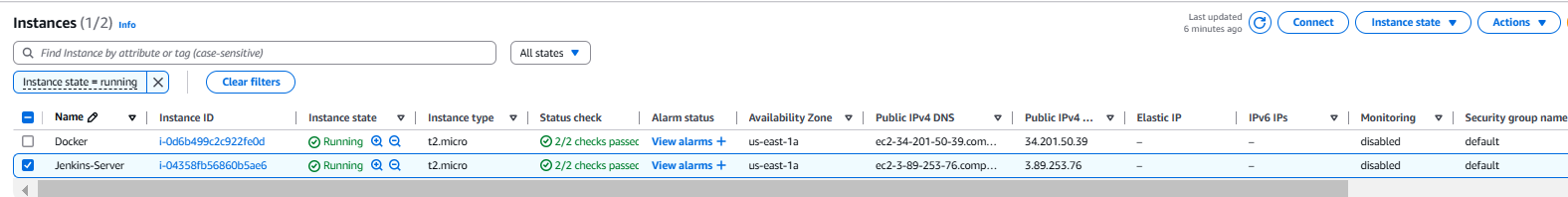
terraform init

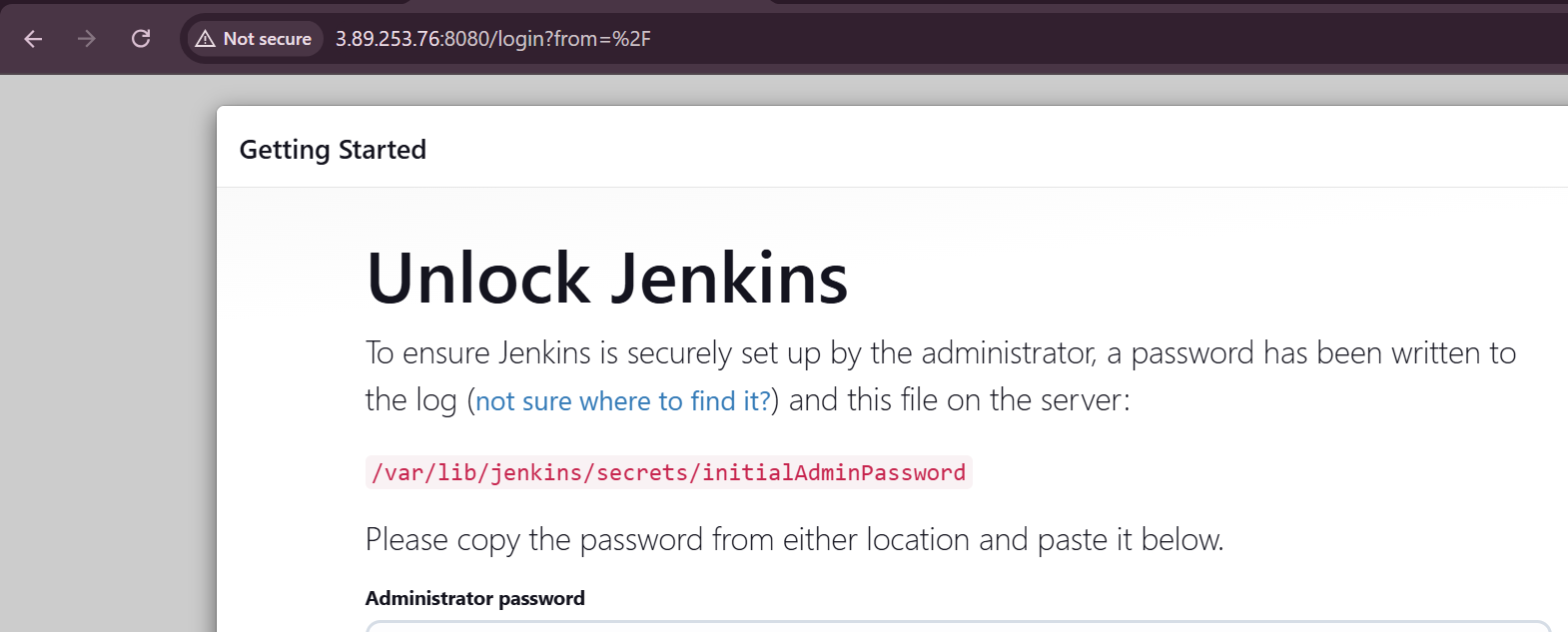
terraform plan

terraform apply



**Outputs in aws console**

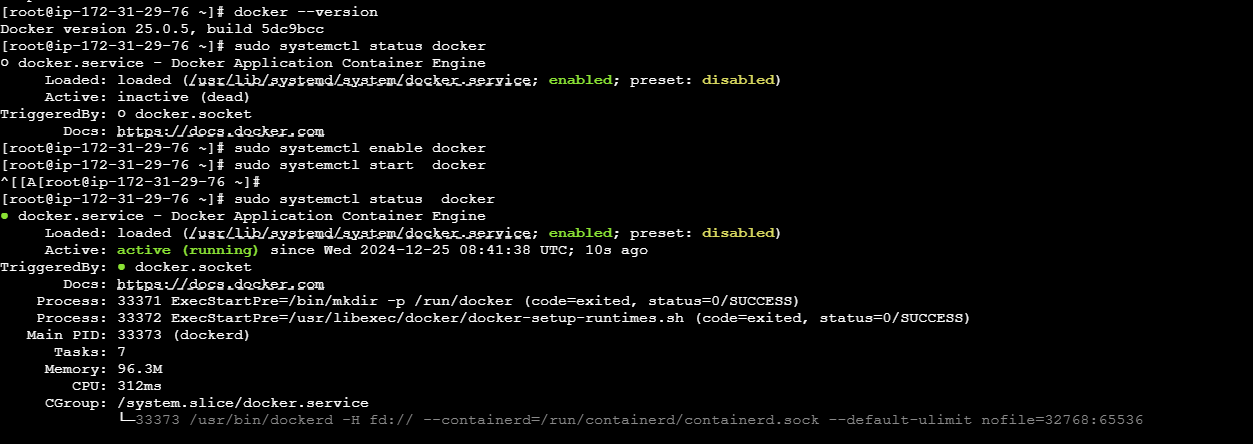




**5) Create One Jenkins job to Build and push the Docker image to Docker Hub. (**[**https://github.com/betawins/Python-app.git**](https://github.com/betawins/Python-app.git)**)**

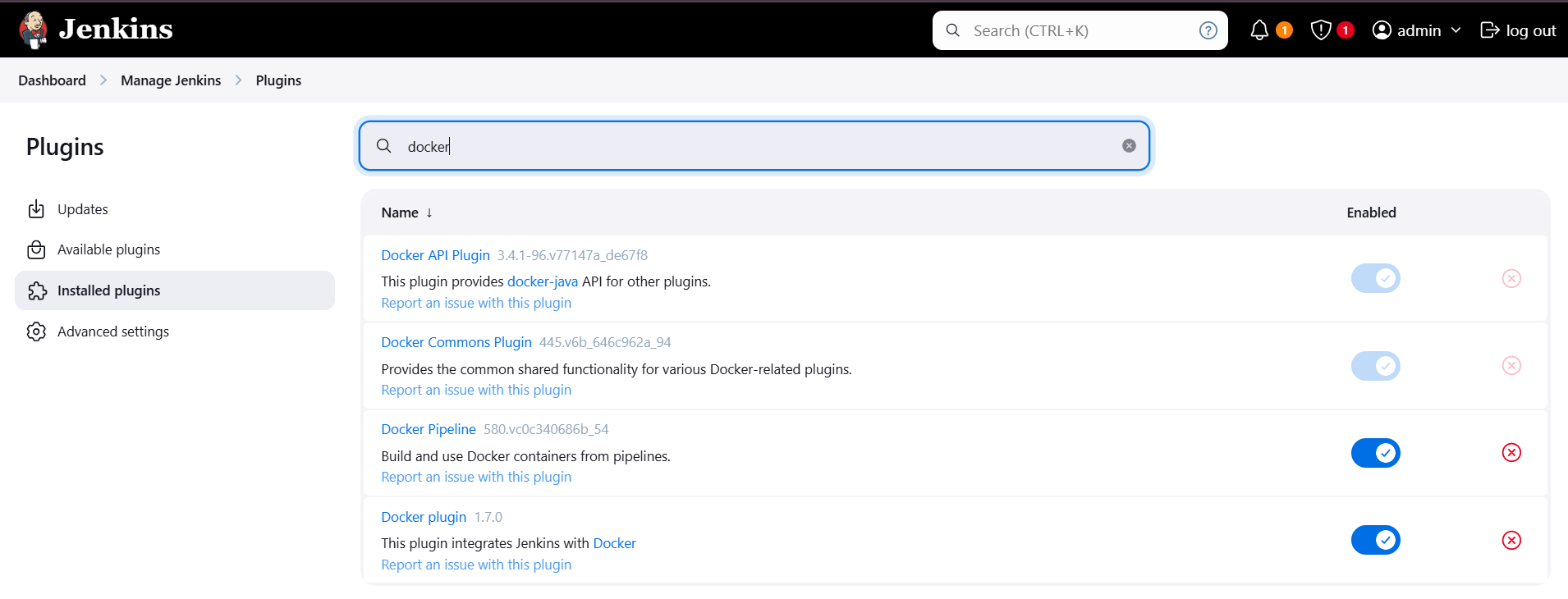
In Jenkins server,

Install docker and start the docker service

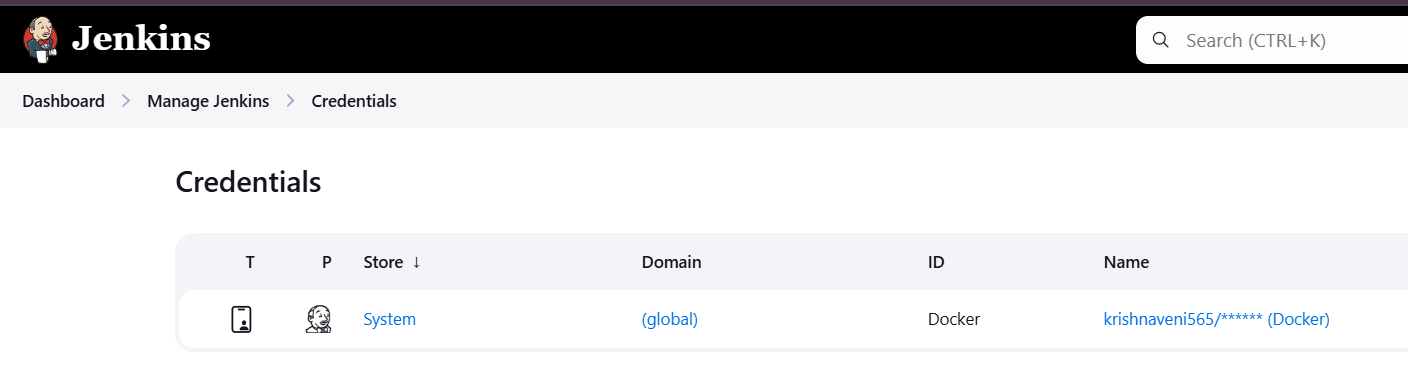


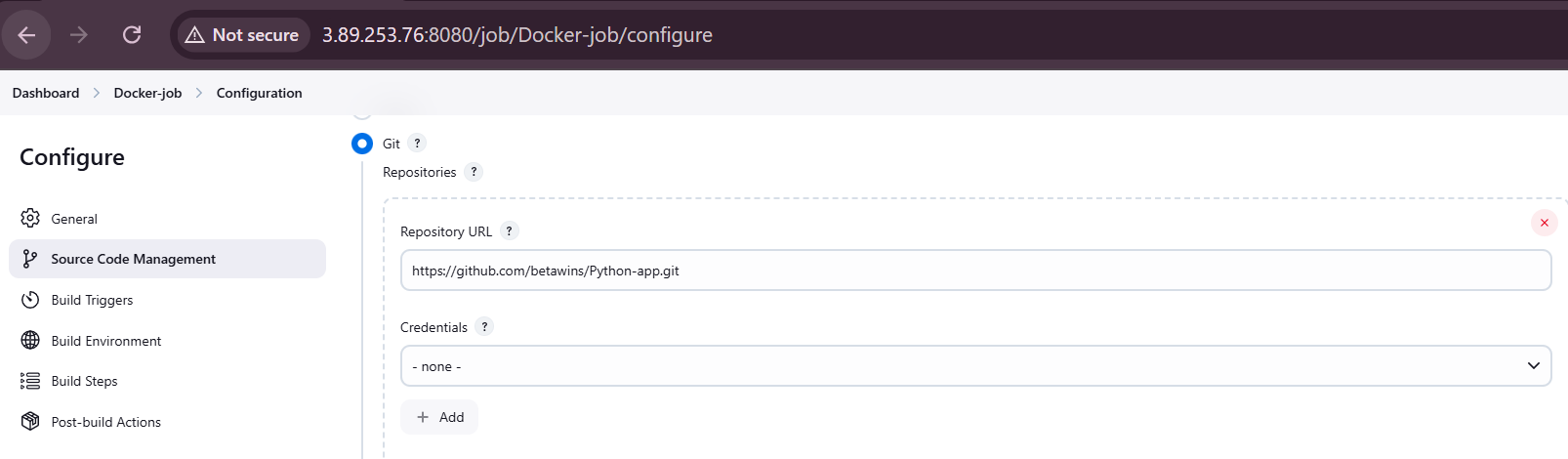


Download the plugin in Jenkins “Docker”

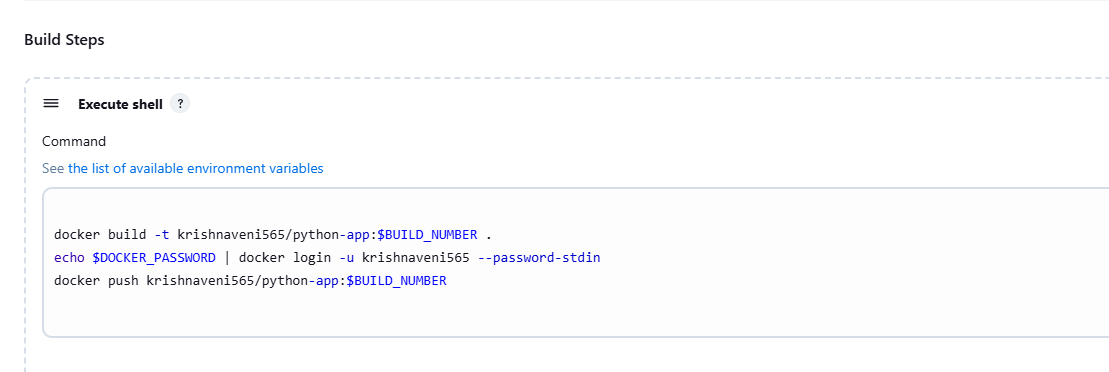


Also add the credentials of Docker “username and password”



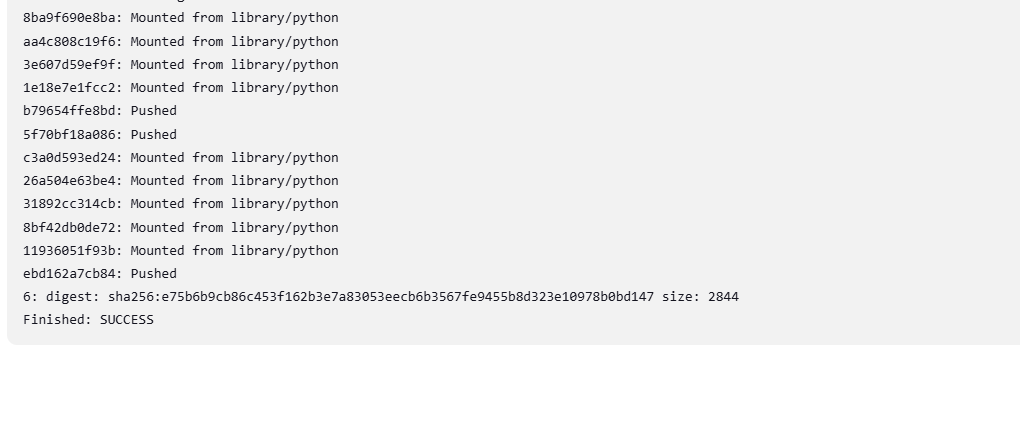
Now create the “Job”[pipeline] 



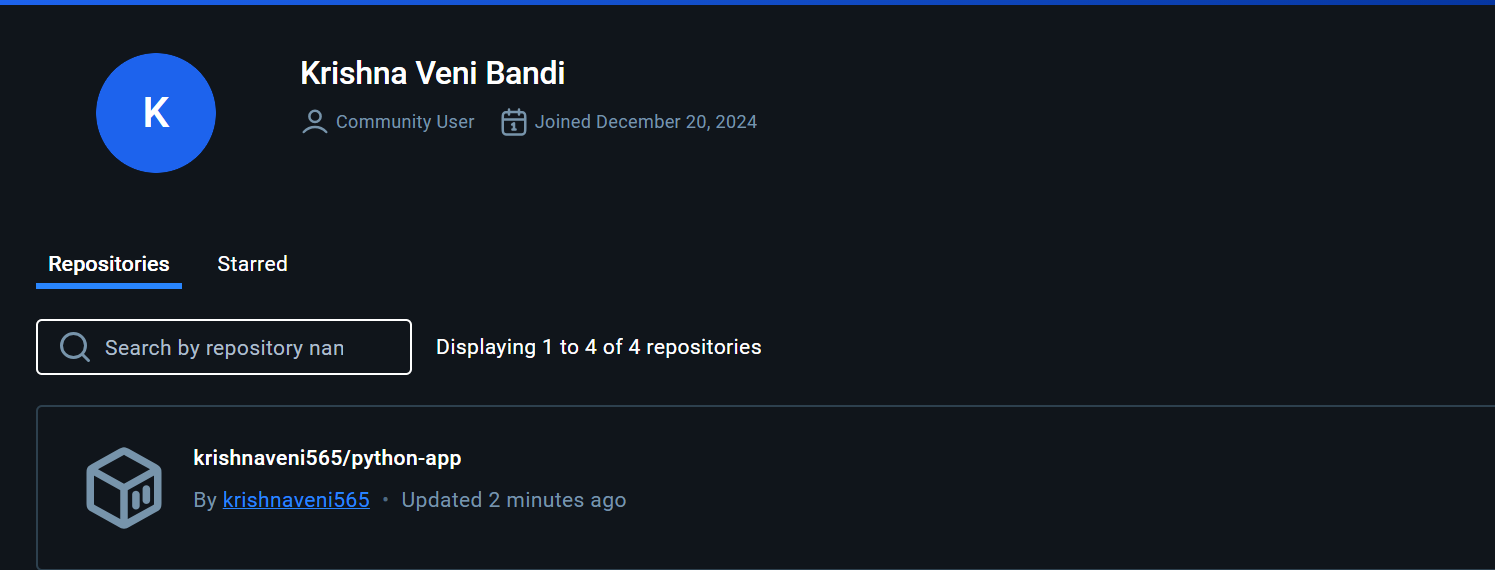


Save

And build the job

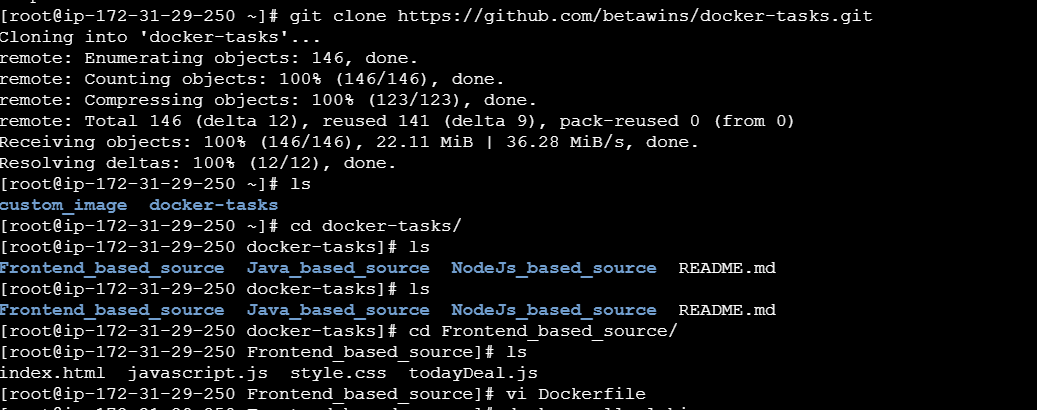


We can also check that image is pushed to dockerhub.com

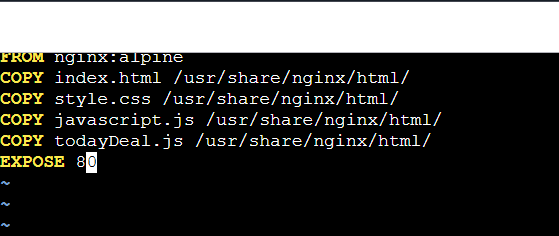


**Source Code:**[**https://github.com/betawins/docker-tasks.git**](https://github.com/betawins/docker-tasks.git)

**1. From the source code of the frontend, Using that write a Docker file & build a docker image, run & push that image to your docker registry**



Dockerfile



Save the file

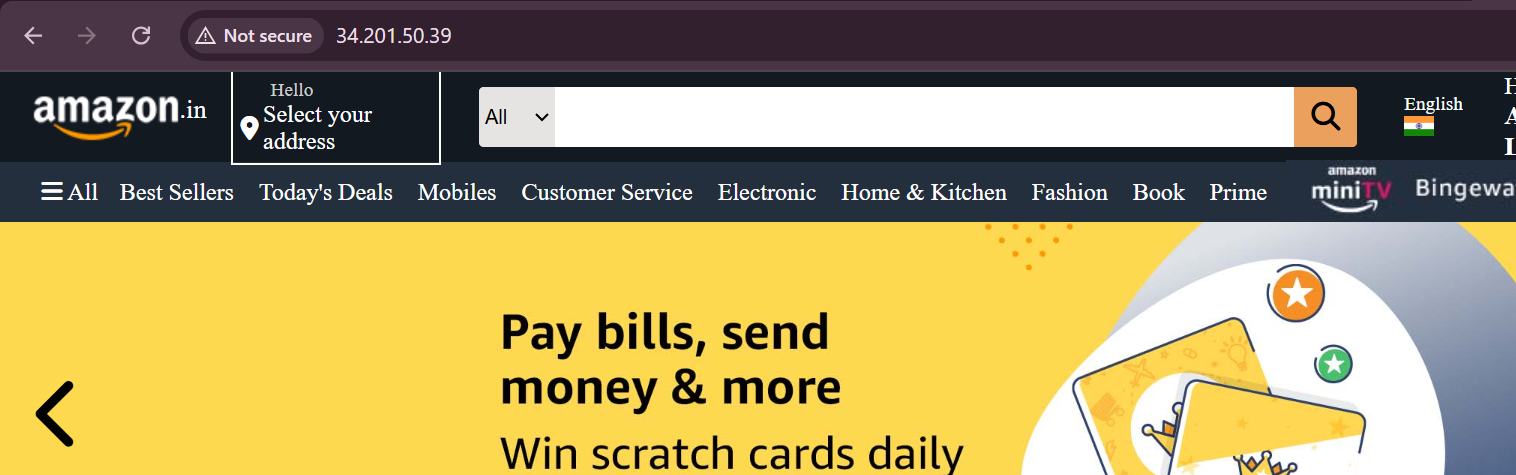
Next build the image

docker build -t frontend:v1 .

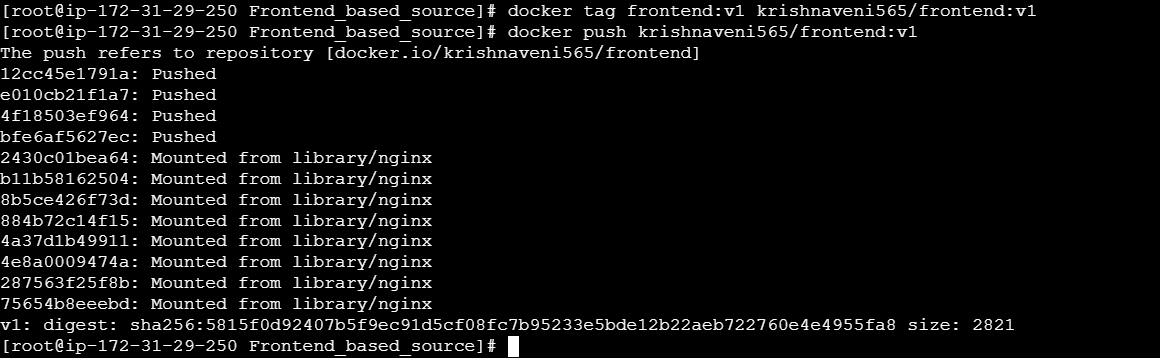
After building the image

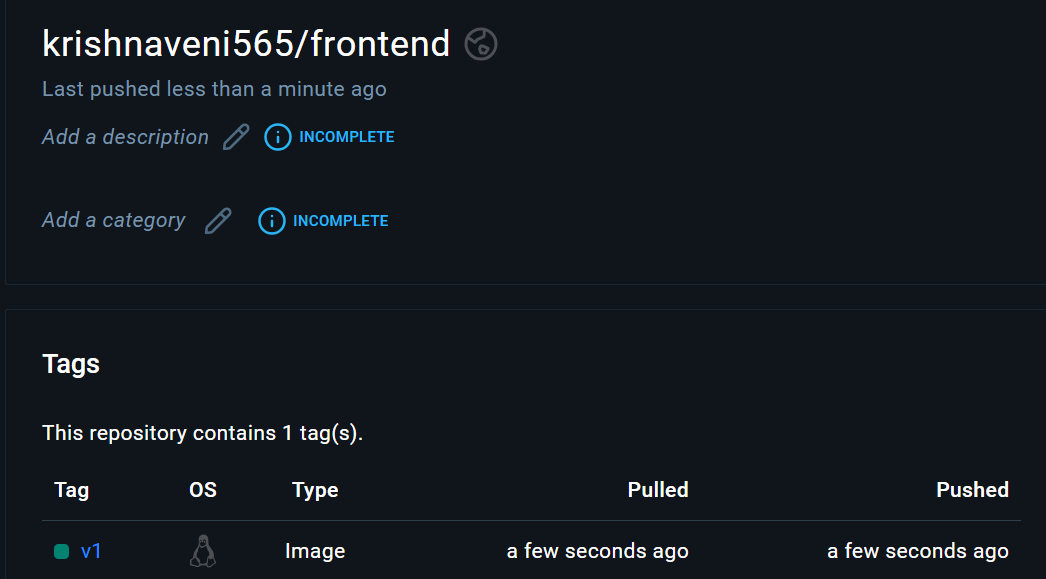
Run it

With cmd: docker run -itd -p 80:80 frontend:v1

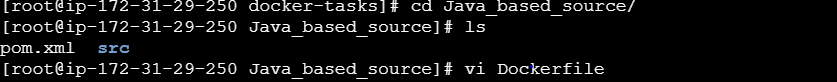


Pushing the created image with the docker.hub

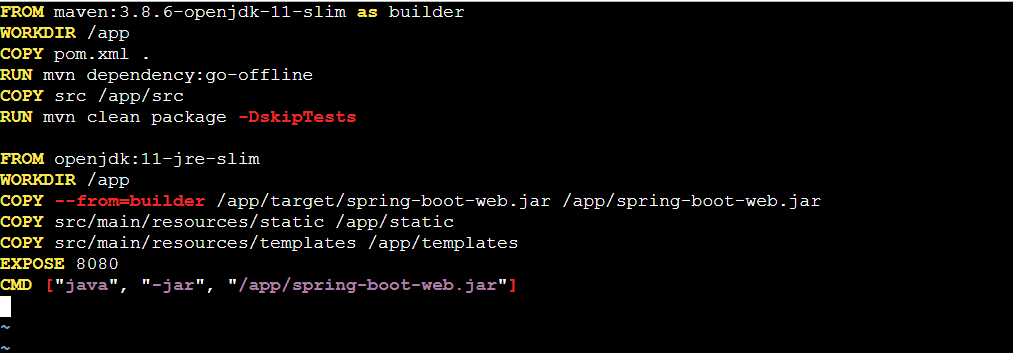




**2. From the Java Based Source Code, Write a Docker file, build, run & push to docker registry.**



Dockerfile



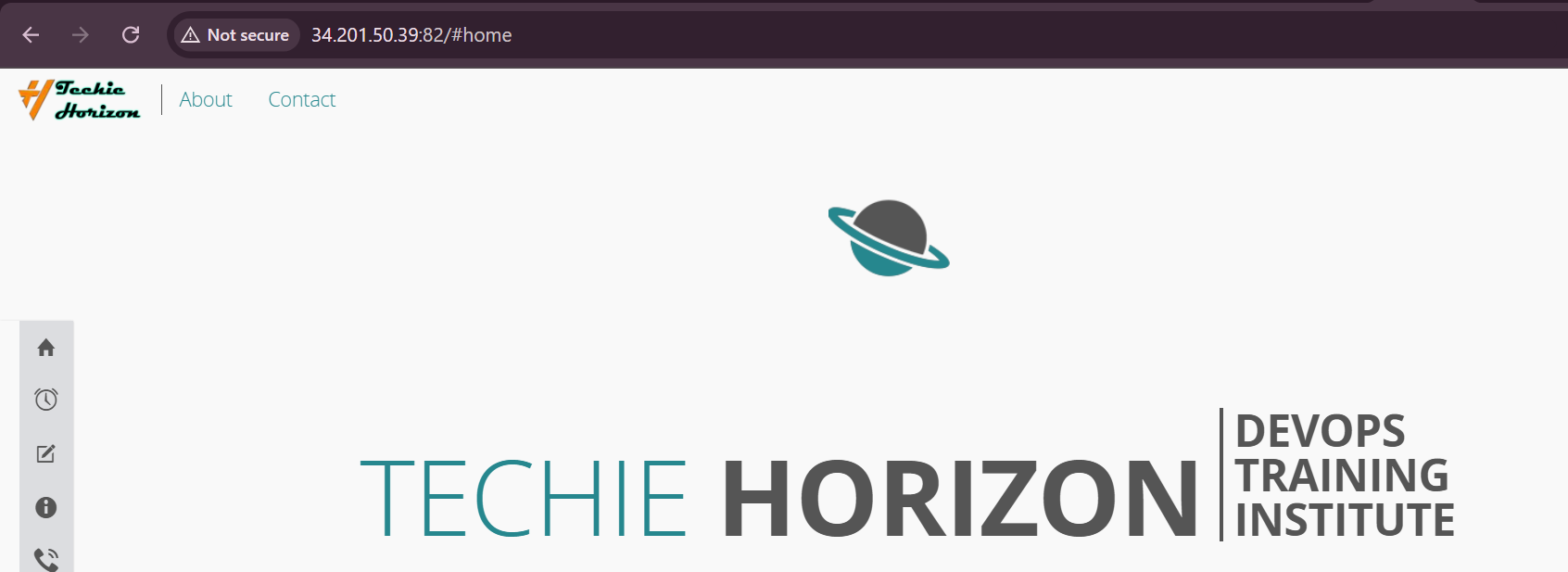
Build the image

docker build -t javabased:v1 .

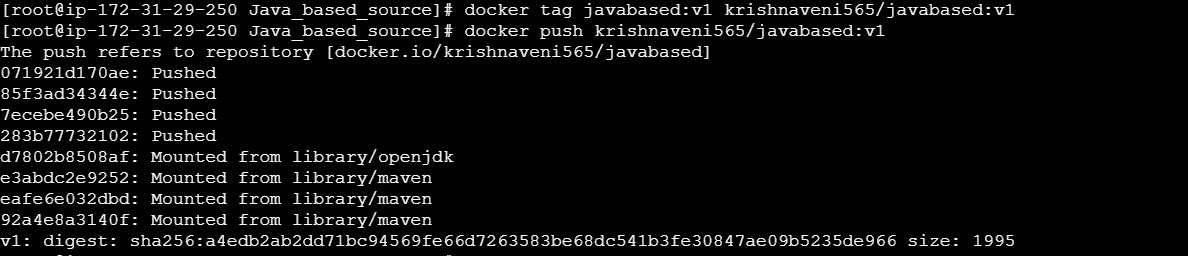
run the container

docker run -itd -p 82:8080 javabased:v1

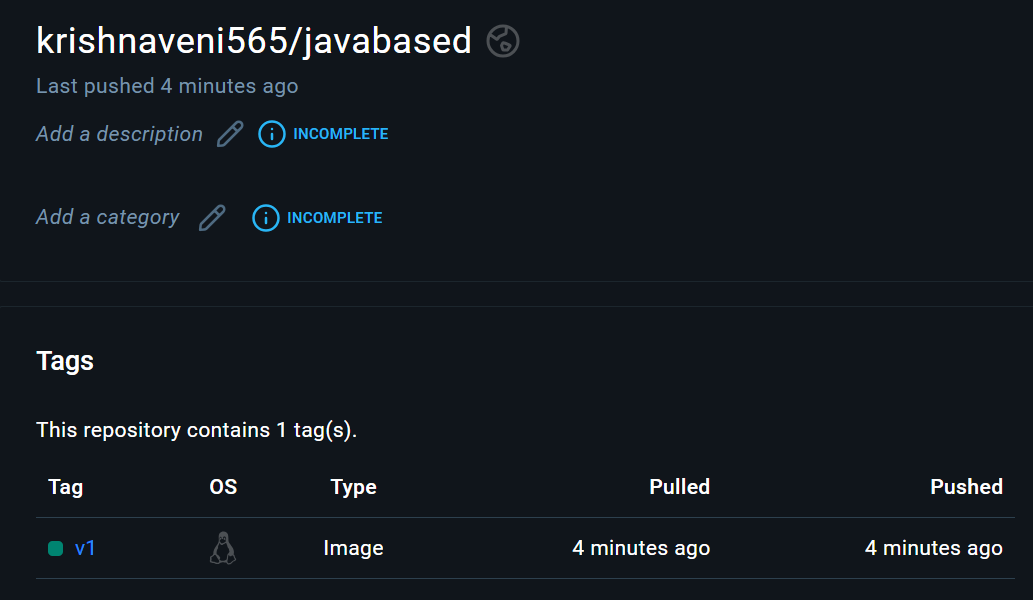
we can check in the browser:publicip:82



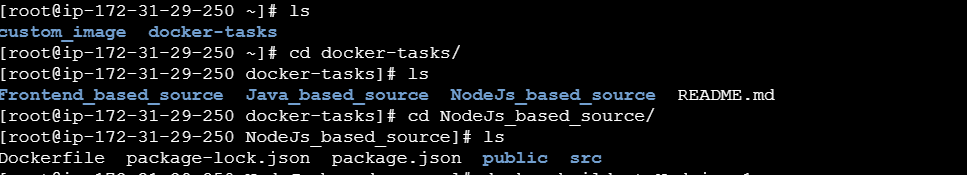
Pushing the image to docker registry



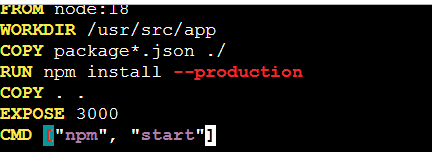
We can check in the dockerhub registry



**3. From the NodeJS Based Source Code, Write a docker file, build with tag v1, run & push to docker registry.**

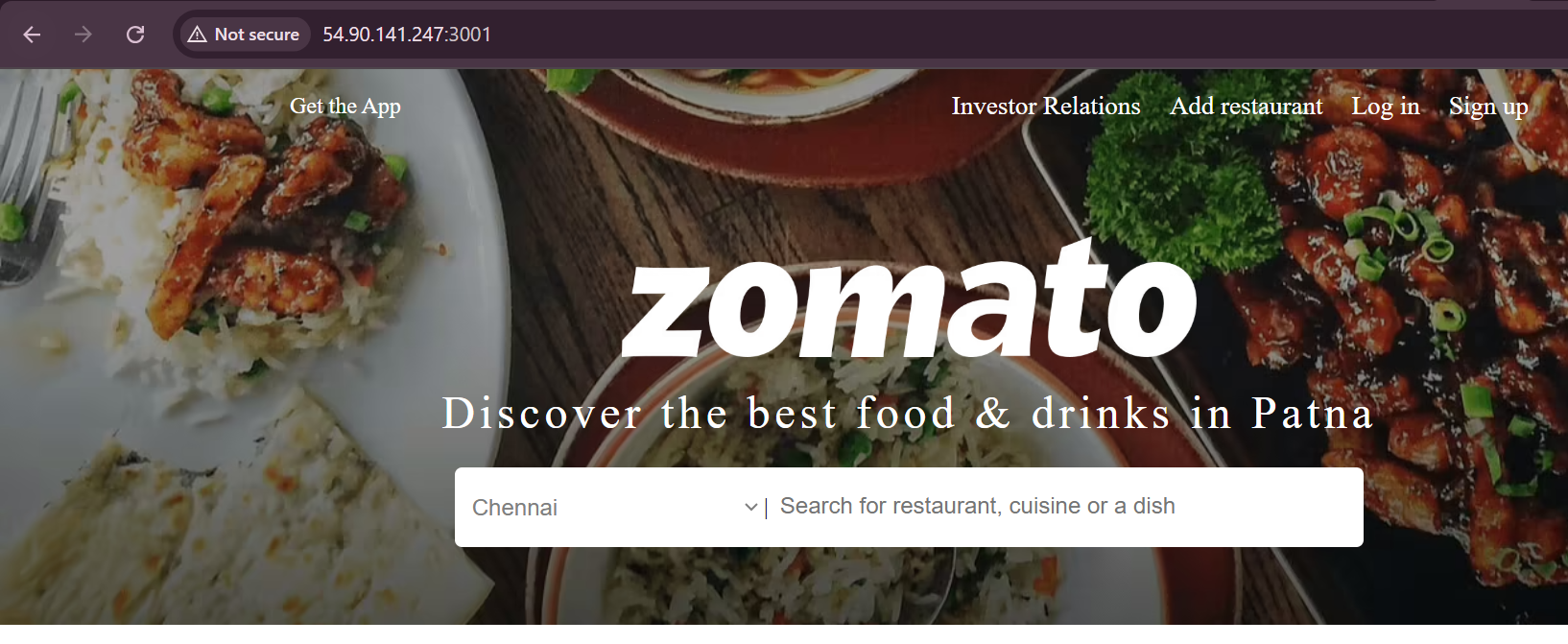


Dockerfile:

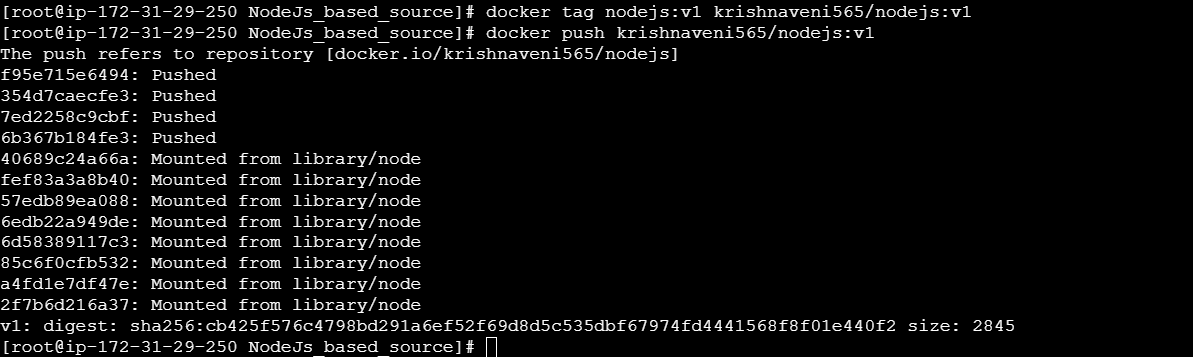


Save the dockerfile

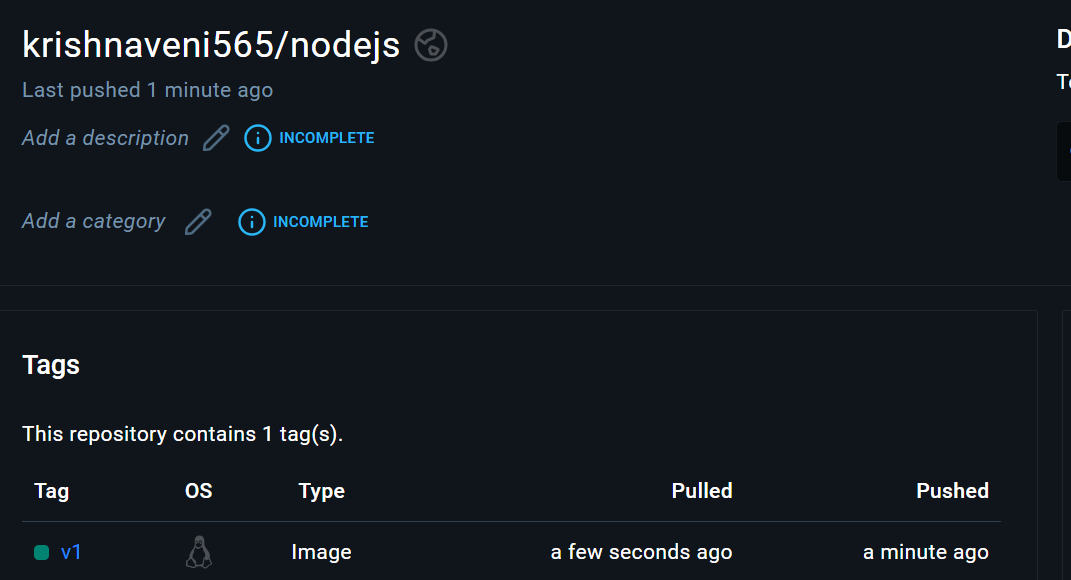
Build and run the image



Pushing the image to the dockerhub

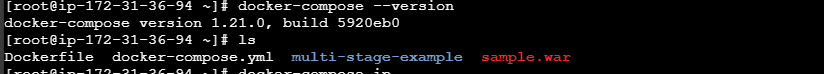


In the dockerhub registry



**4. Write a docker-compose docker file to setup WordPress with MySQL database**

**Installing the docker compose**



Creating a script docker-compose.yml file and write a script

version: '3'

services:

db:

image: mysql:5.7

volumes:

- db\_data:/var/lib/mysql

restart: always

environment:

- MYSQL\_ROOT\_PASSWORD=somewordpress

- MYSQL\_DATABASE=wordpress

- MYSQL\_USER=wordpress

- MYSQL\_PASSWORD=wordpress

wordpress:

depends\_on:

- db

image: wordpress:latest

ports:

- "8000:80"

restart: always

environment:

- WORDPRESS\_DB\_HOST=db:3306

- WORDPRESS\_DB\_USER=wordpress

- WORDPRESS\_DB\_PASSWORD=wordpress

- WORDPRESS\_DB\_NAME=wordpress

volumes:

db\_data: { }

