Day 12

Jenkin

**Build a Docker Jenkins Pipeline to Implement CI/CD Workflow.**

Course-end Project 2

Description

Demonstrate the continuous integration and delivery by building a Docker Jenkins Pipeline.

Problem Statement Scenario:

You are a DevOps consultant in AchiStar Technologies. The company decided to implement DevOps to develop and deliver their products. Since it is an Agile organization, it follows Scrum methodology to develop the projects incrementally. You are working with multiple DevOps Engineers to build a Docker Jenkins Pipeline. During the sprint planning, you agreed to take the lead on this project and plan on the requirements, system configurations, and track the efficiency. The tasks you are responsible for:

* Availability of the application and its versions in the GitHub
  + Track their versions every time a code is committed to the repository
* Create a Docker Jenkins Pipeline that will create a Docker image from the Dockerfile and host it on Docker Hub
* It should also pull the Docker image and run it as a Docker container
* Build the Docker Jenkins Pipeline to demonstrate the continuous integration and continuous delivery workflow

Company goal is to deliver the product frequently to the production with high-end quality.

**You must use the following tools:**

* Docker: To build the application from a Dockerfile and push it to Docker Hub
* Docker Hub: To store the Docker image
* GitHub: To store the application code and track its revisions
* Git: To connect and push files from the local system to GitHub
* Linux (Ubuntu): As a base operating system to start and execute the project
* Jenkins: To automate the deployment process during continuous integration

**Following requirements should be met:**

* Document the step-by-step process from the initial installation to the final stage
* Track the versions of the code in the GitHub repository
* Availability of the application in the Docker Hub
* Track the build status of Jenkins for every increment of the project

Index.html or java or python

<html>

<head>

</head>

<body>

<p>Welcome to HTML Web Page</p>

</body>

</html>

With simple message

Dockerfile for index.html app with help of nginx server

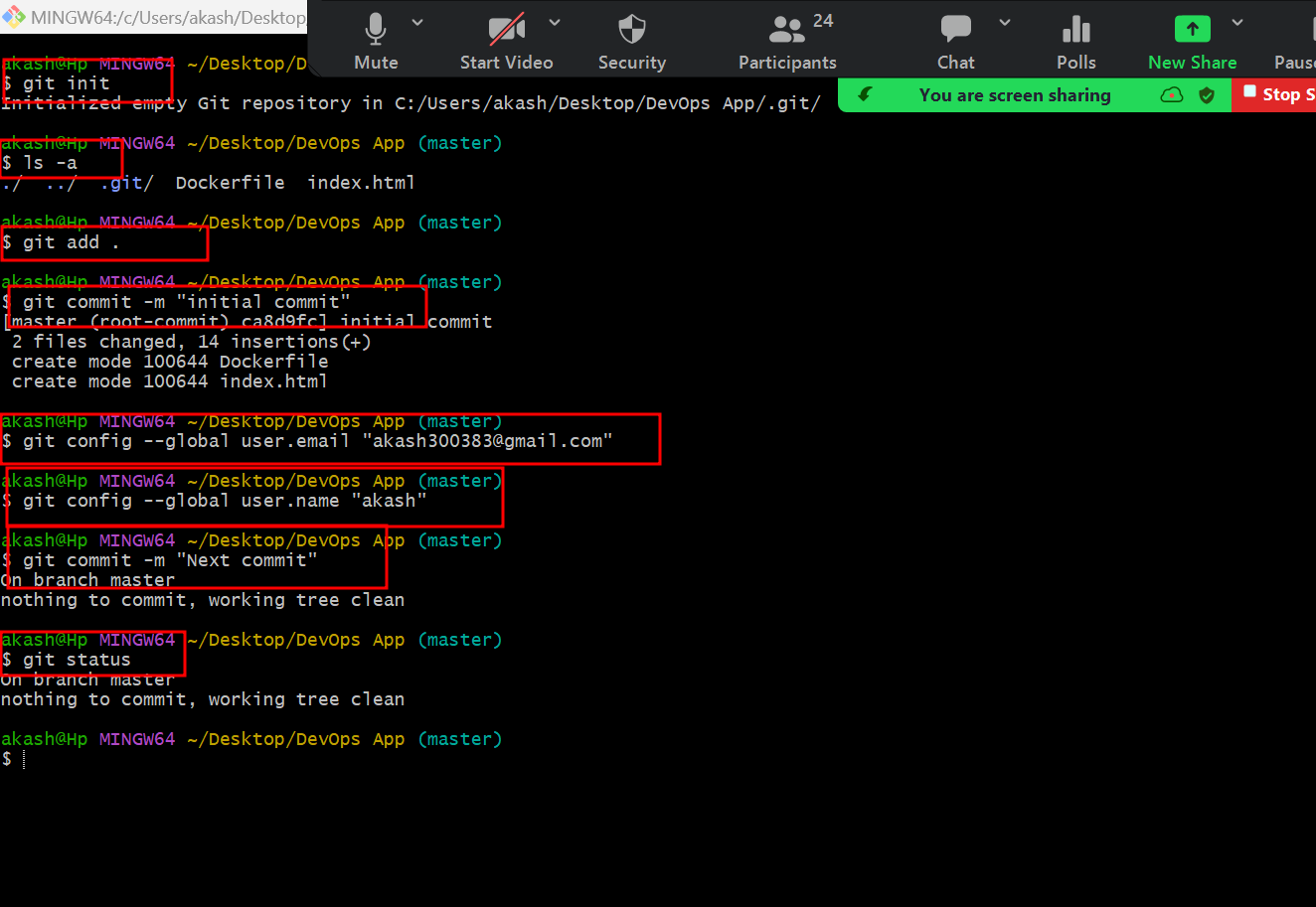
Push this code to git hub

Git can be in your local machine or VM machine etc.

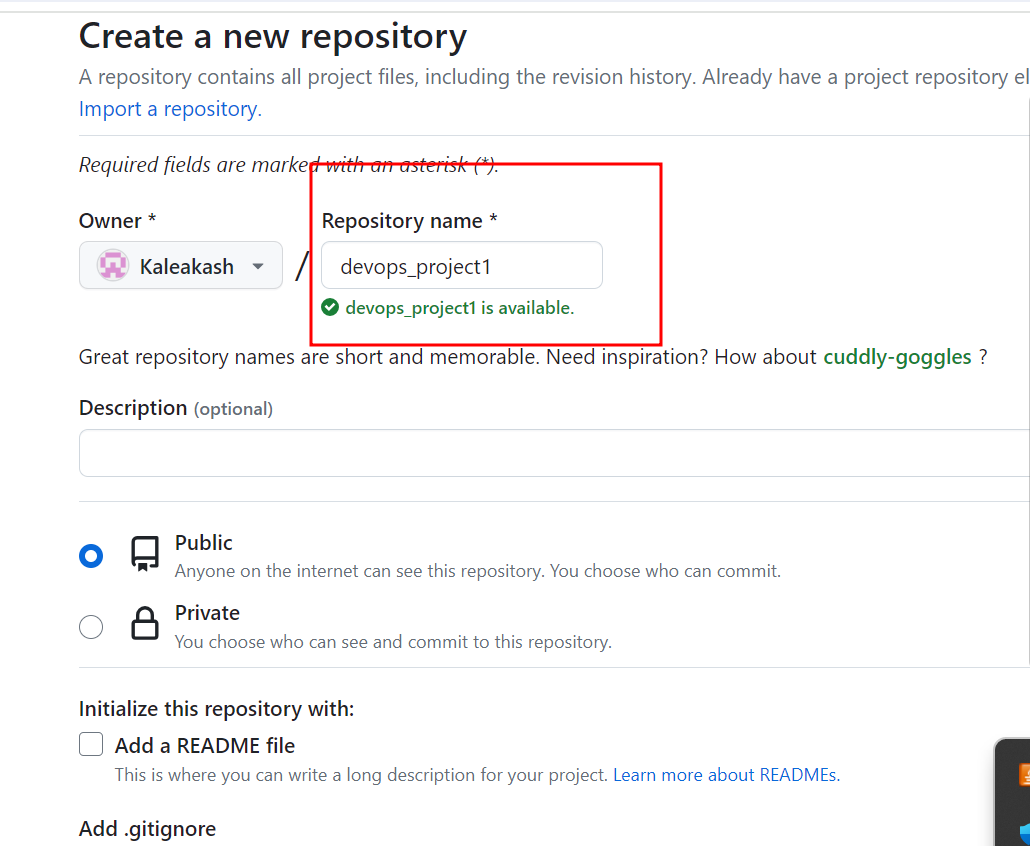
Dockerfile

FROM nginx

COPY index.html /usr/share/nginx/html



Now we need to create remote repository in github account.



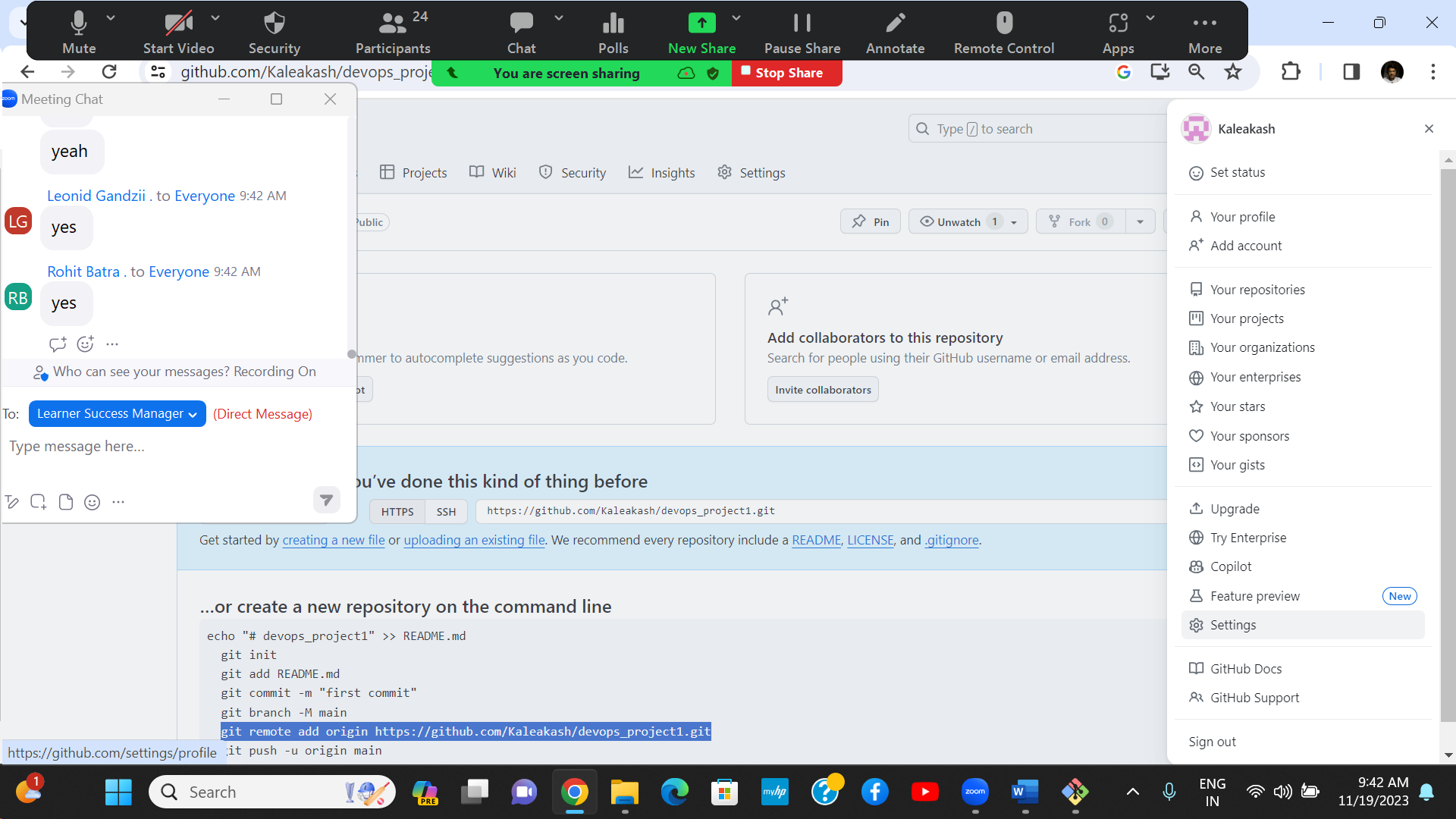
git remote add origin https://github.com/Kaleakash/devops\_project1.git

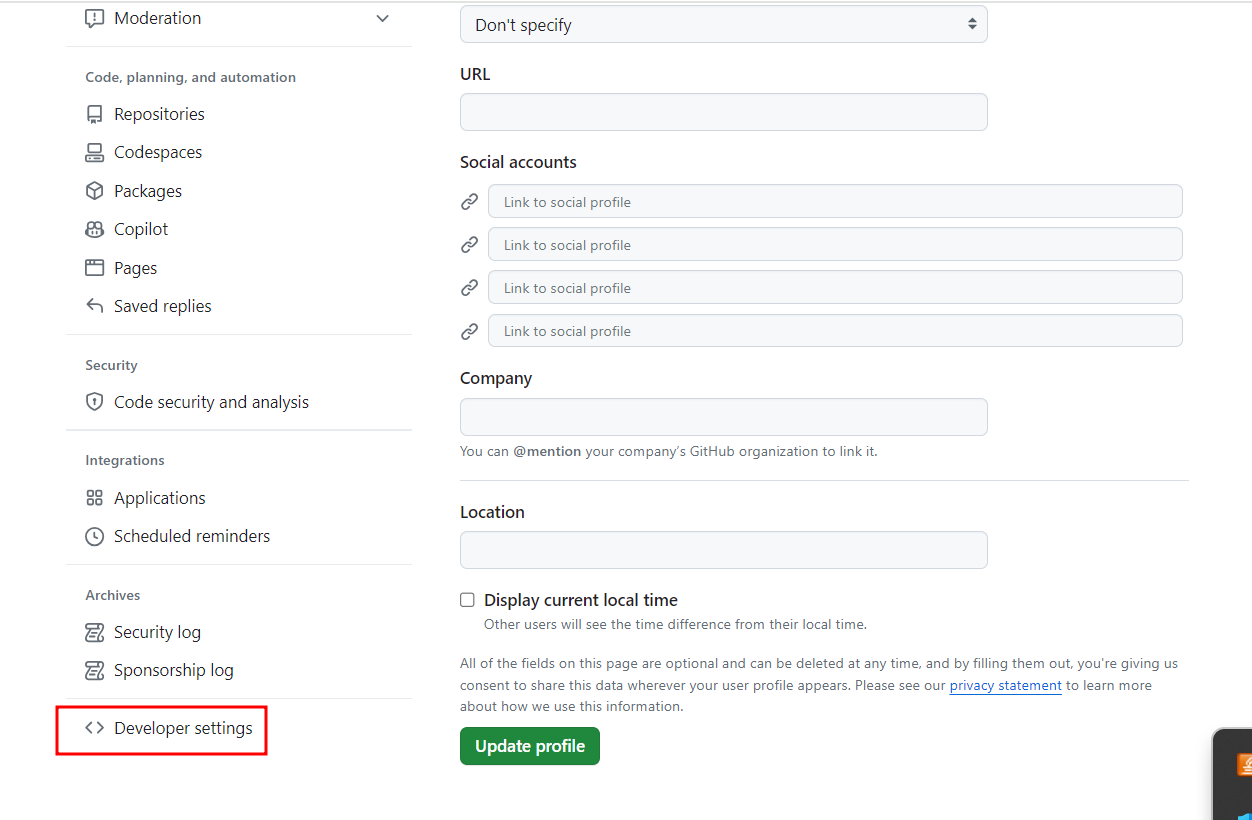
git remote add origin https://token@github.com/Kaleakash/devops\_project1.git

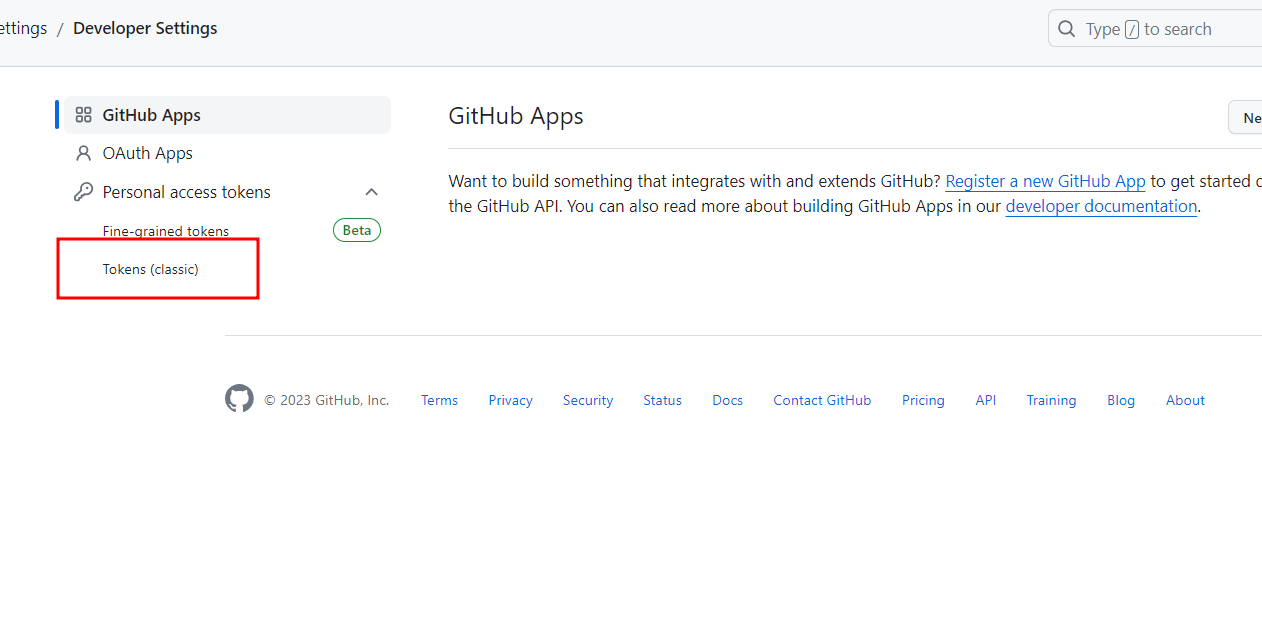
git remote add origin https://ghp\_sdUvfK6x4nDQVpH6DTR823aiv35XwA24nGZK@github.com/Kaleakash/devops\_project1.git

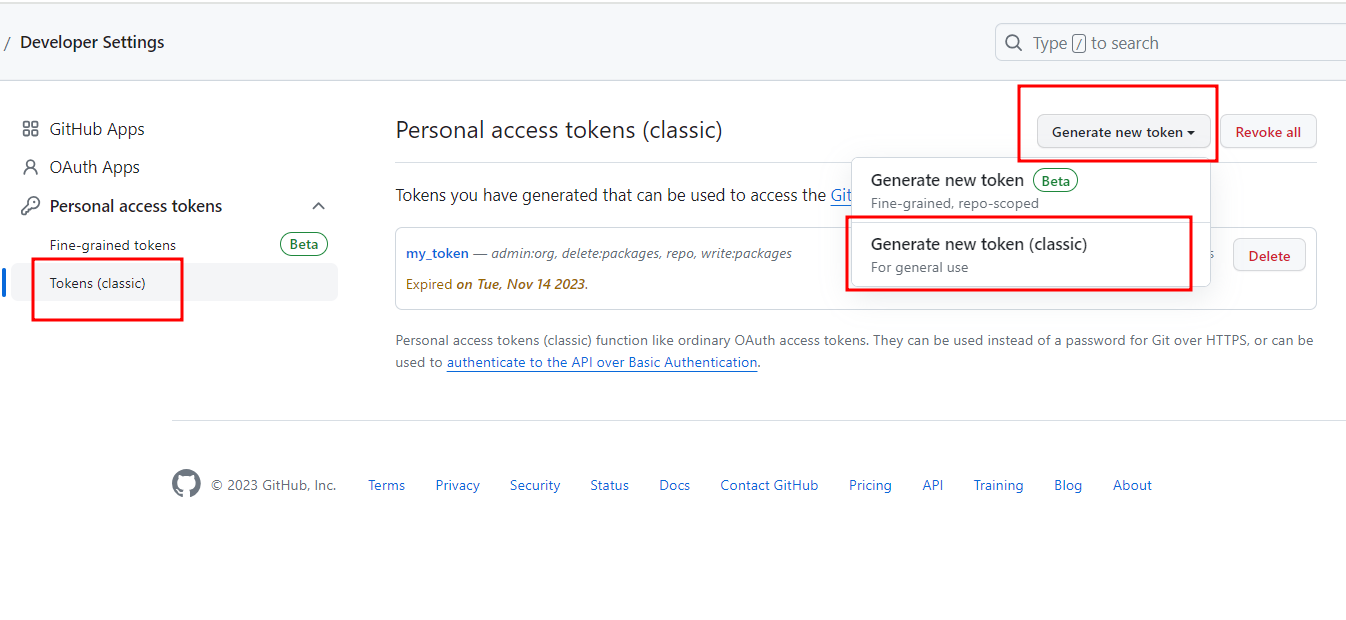
ghp\_sdUvfK6x4nDQVpH6DTR823aiv35XwA24nGZK

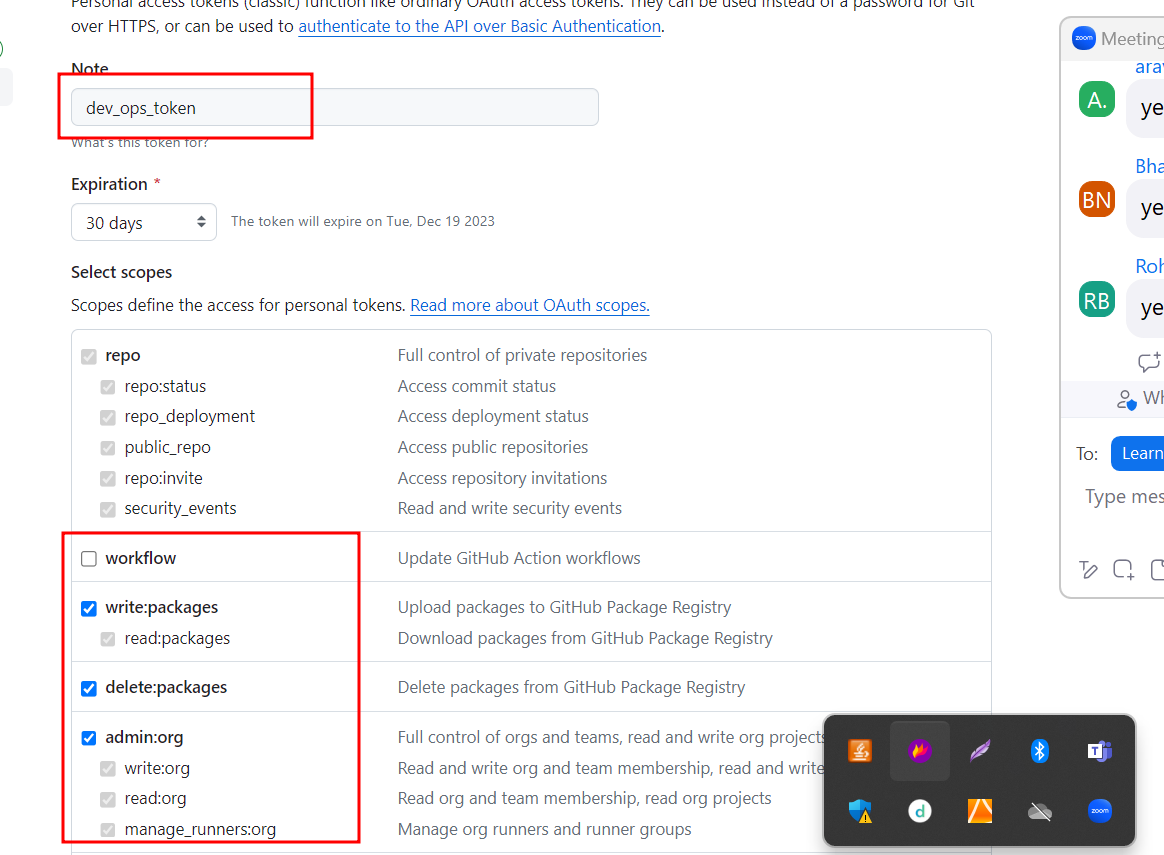
we will generate token in git hub account.

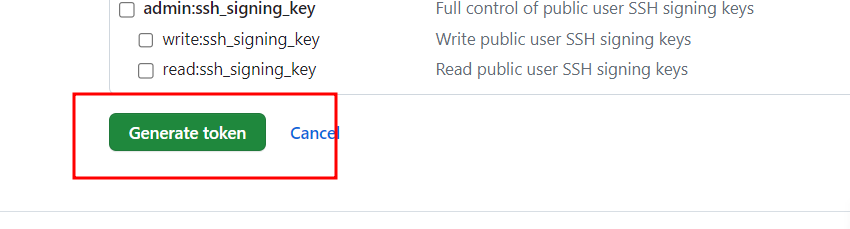










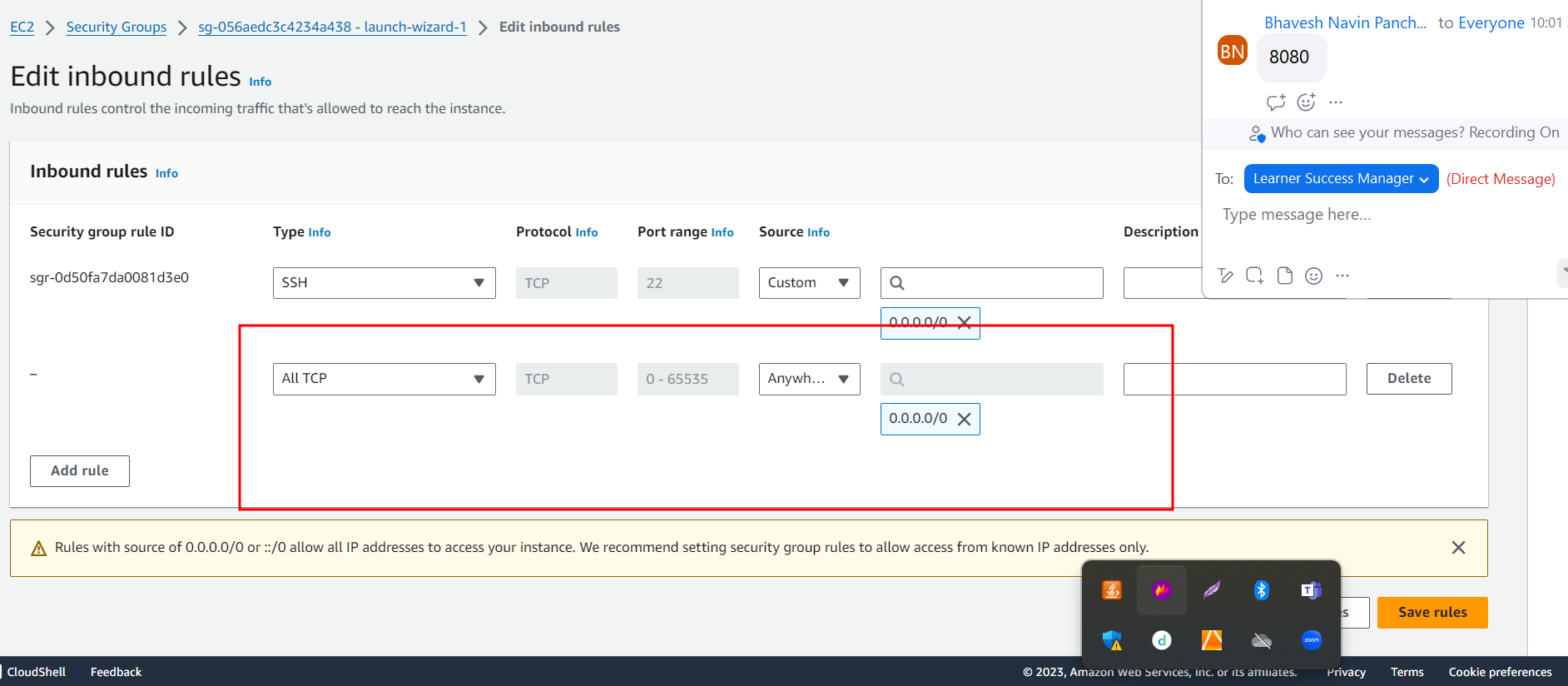


git remote add origin https://token@github.com/Kaleakash/devops\_project1.git

git push -u origin HEAD

now you need to create EC2 instance

After created EC2 instance now you need to open port number



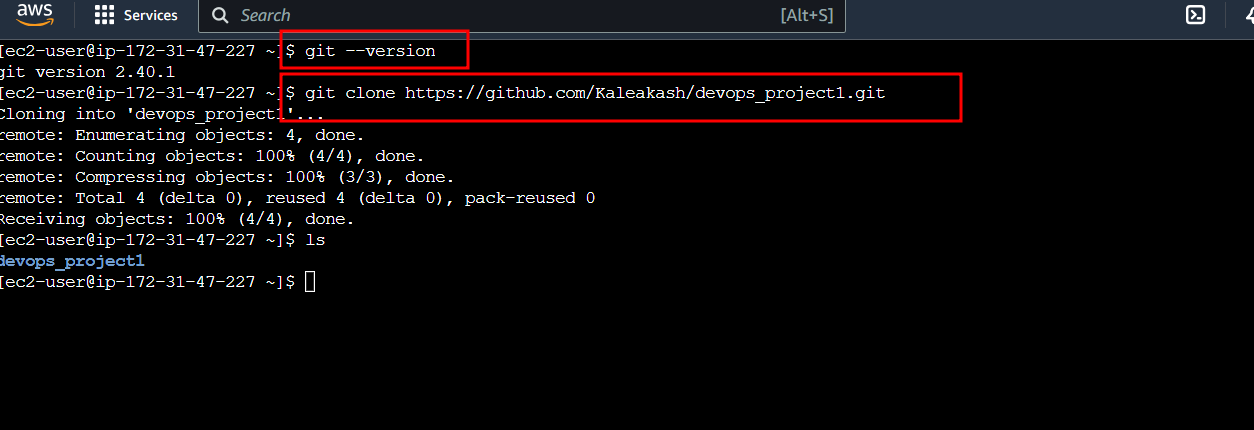
Connect EC2 instance

Then install required software.

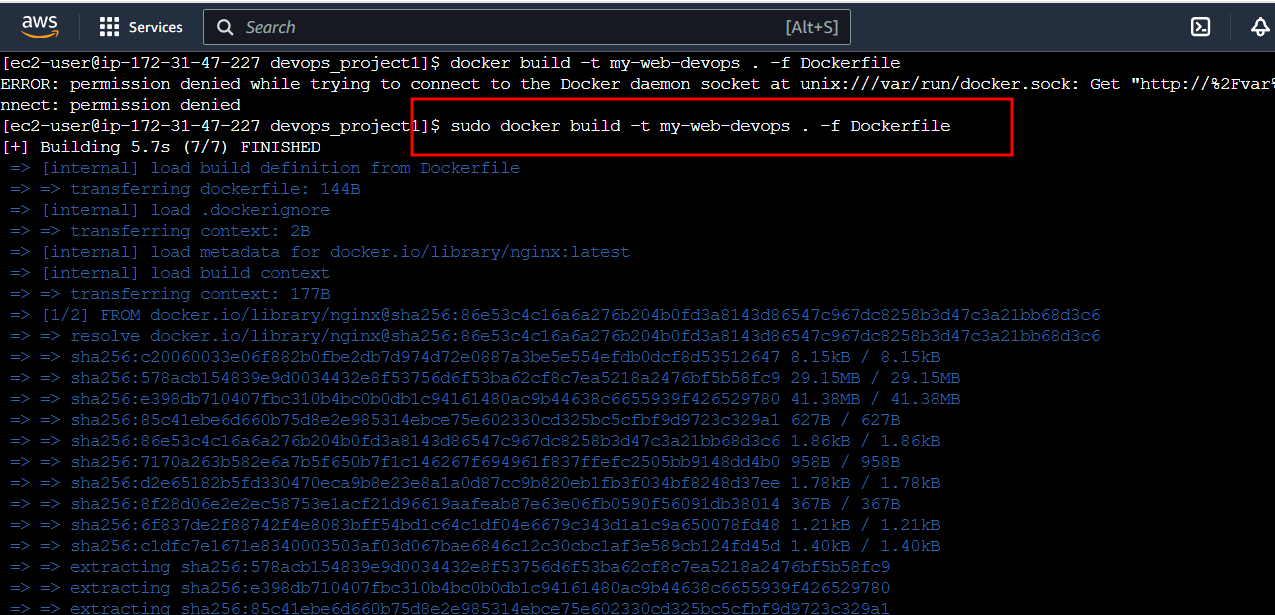
Install the git

sudo yum install git -y

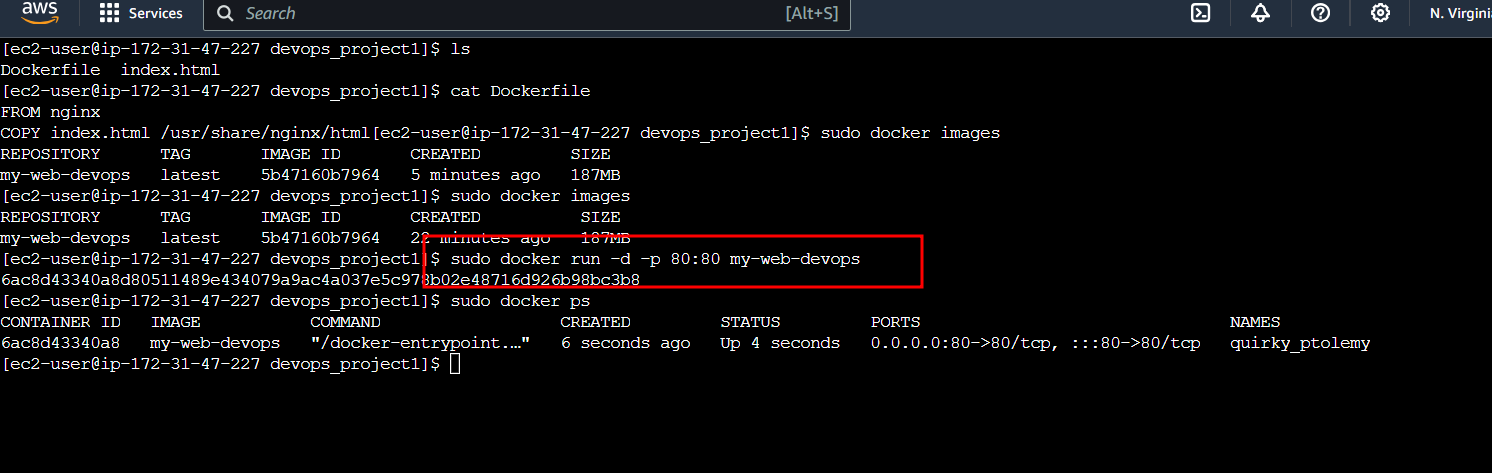
Now you need to clone this project in EC2 instance.



sudo docker build -t my-web-devops . -f Dockerfile



After image created successfully please run the image



Please check your application running or not

Using EC2 instance public ip address

<http://publicIpAddress:80>

Now we run docker container using Jenkin

First install java software.

EC2 instance running two service

* 1. Docker
  2. Jenkin

Inside Jenkin Pipeline or jobs we want to run Docker.

We do the changes in local machine in the application ie

Index.html

Then we check status, add, commit and push

In Jenkin you need to create job or pipeline job which pull the project

Whenever we push new changes in remote repository

Jenkin job pull the project from git hub account

And build the docker and run docker container.