Additional Task

**Name:Shaik Khaja Basha**

**Batch : Batch 11**

**Date : 26.07.2025**

**Task : terraform with jenkins**

**Task:**

**Create two files and two directors using terraform with help of jenkins pipeline**

**Create a Jenkins pipeline that provisions infrastructure using Terraform. The pipeline should:**

1. **Automatically perform terraform init, validate, plan, and apply stages.**
2. **All Terraform files (main.tf, variables.tf, terraform.tfvars, etc.) should be stored in a GitHub repository (e.g., terraform-jenkins-pipeline).**
3. **Clone this repository in your Jenkins pipeline.**
4. **Design the Jenkinsfile to execute the Terraform steps sequentially.**
5. **The files should be stored in the Jenkins workspace (agent) during execution.**
6. **After running the pipeline, verify that the Terraform resources are created successfully two files and two directories in your local or target environment.**
7. **Take screenshots of the output/results as proof**

Jenkins Pipeline for Provisioning Files and Directories with Terraform

**Below is a step-by-step guide to create two files and two directories using Terraform, orchestrated via a Jenkins Pipeline. This approach integrates version control (GitHub), automated infrastructure provisioning (Terraform), and CI/CD (Jenkins).**

1. Prepare GitHub Repository

Store all Terraform configuration files (main.tf, variables.tf, terraform.tfvars, etc.) in a GitHub repository, such as Jenkins-terraform-pipeline/.

Example file structure:

Jenkins-terraform-pipeline/

terraform-jenkins-pipeline/

|--main.tf

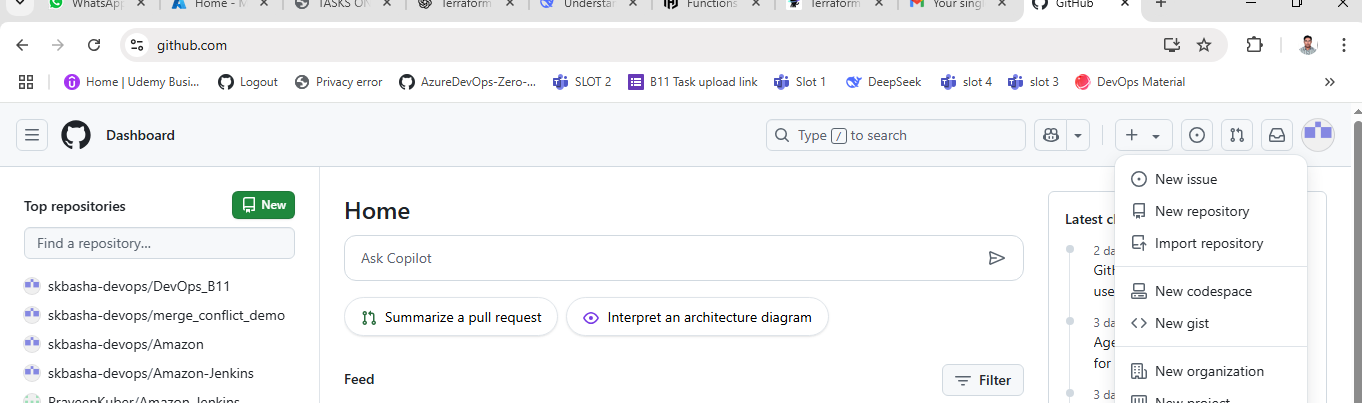
|--Jenkinsfile

|-- outputs.tf (optional)

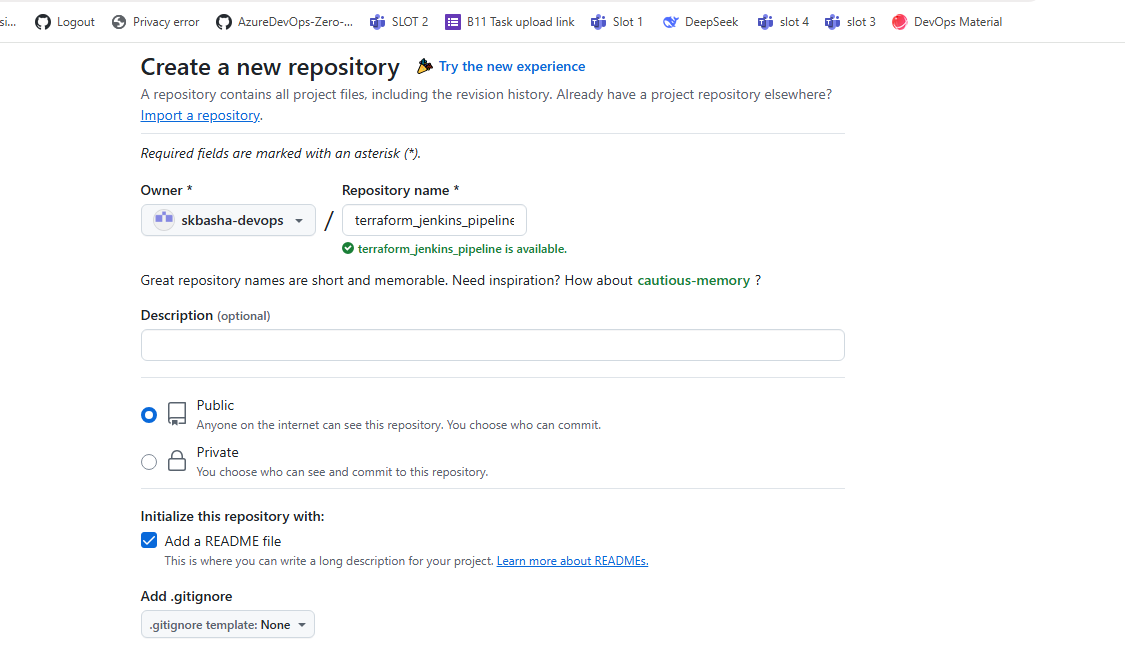
|-- .gitignore (optional)

└── README.md (optional)

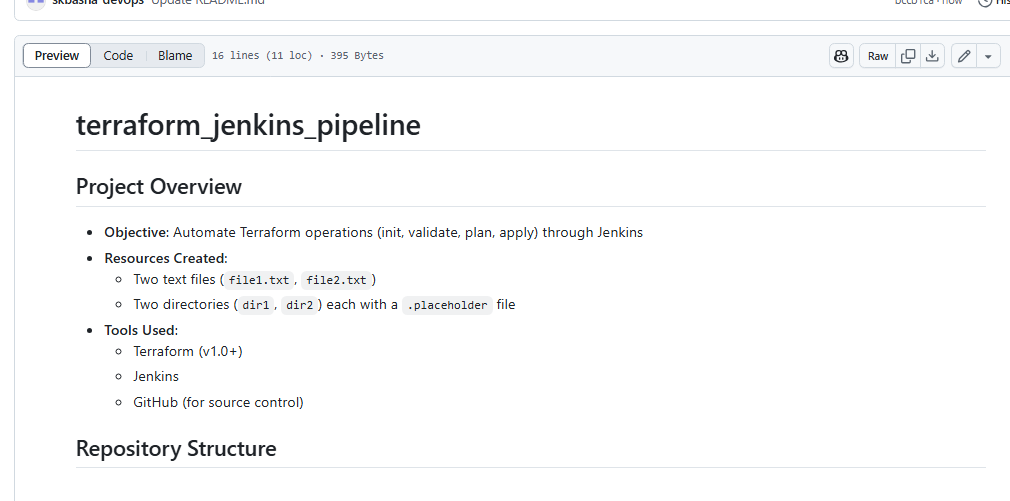
Create a new repository by clicking on New repository



Enter the details as follows

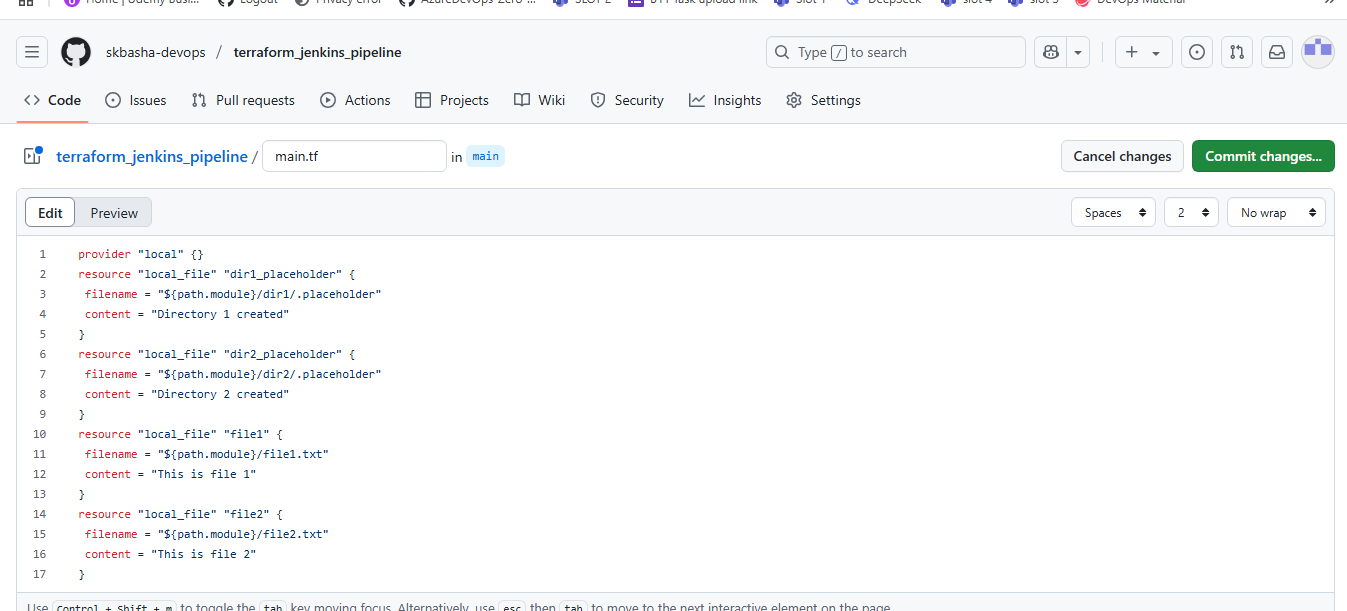


Enter the details for README.md file

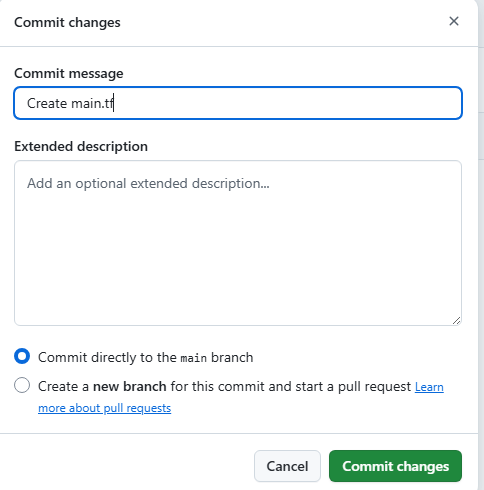


Create a new file and enter the name main.tf

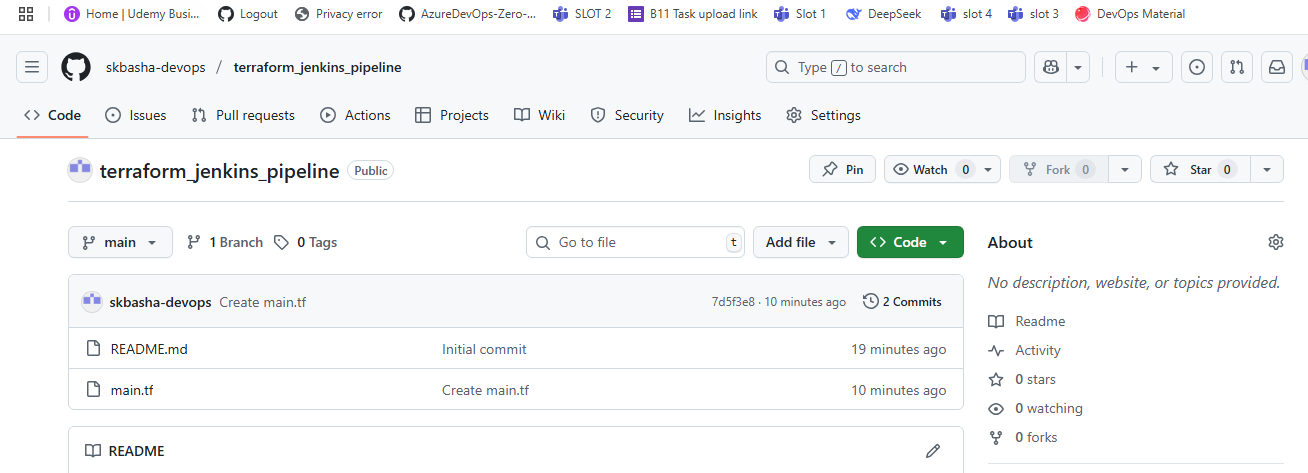
Add the contents to file and click on Commit changes



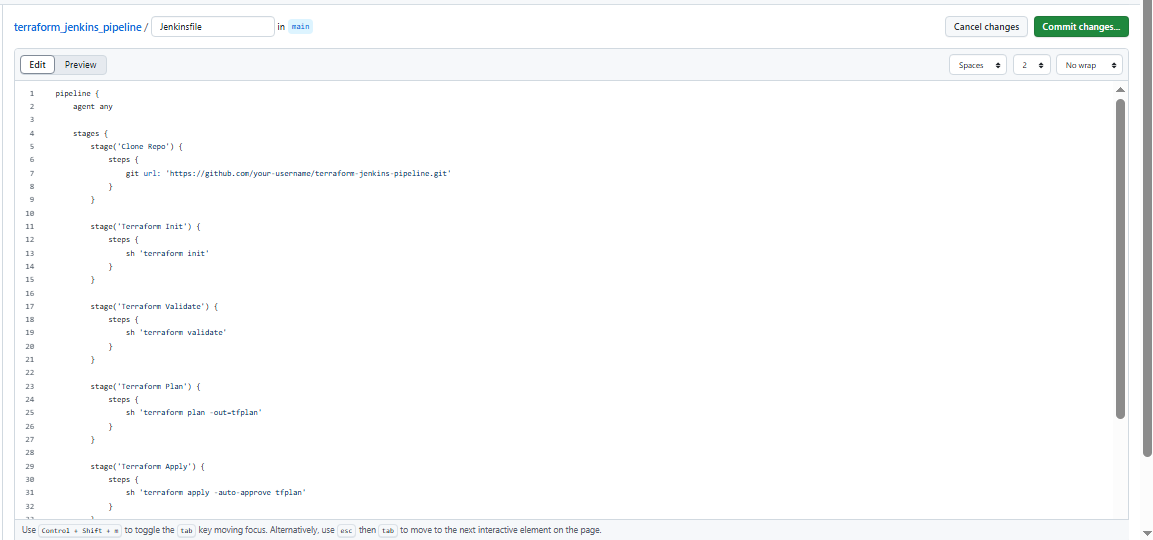
If we use ${path.module} in the root module, it resolves to . (the current directory).

Enter the Commit message and click on Commit changes 

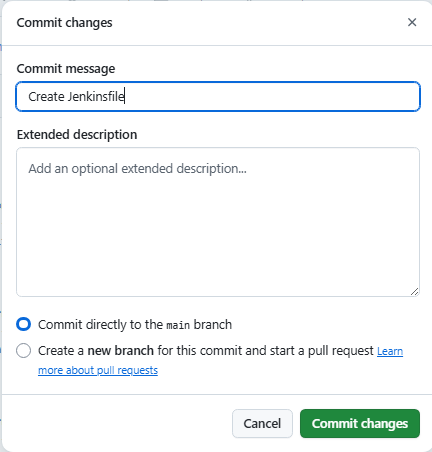
main.tf will be added in main branch



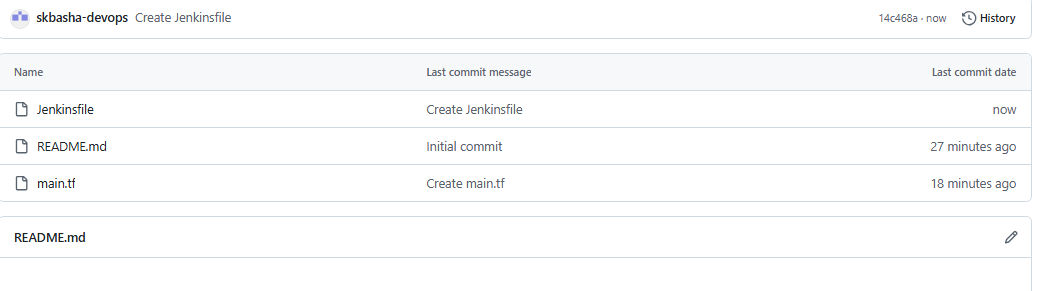
Now we need to add the Jenkinsfile, enter the details as follows and click on Commit changes



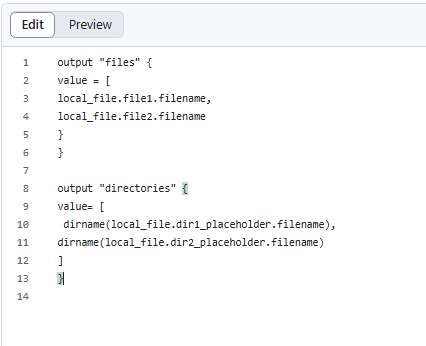
Enter the Commit message and click on Commit changes



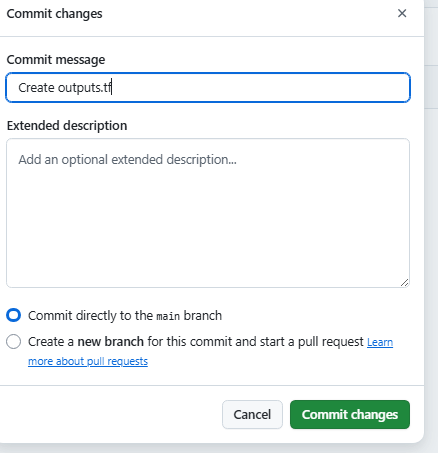
Jenkinsfile is added



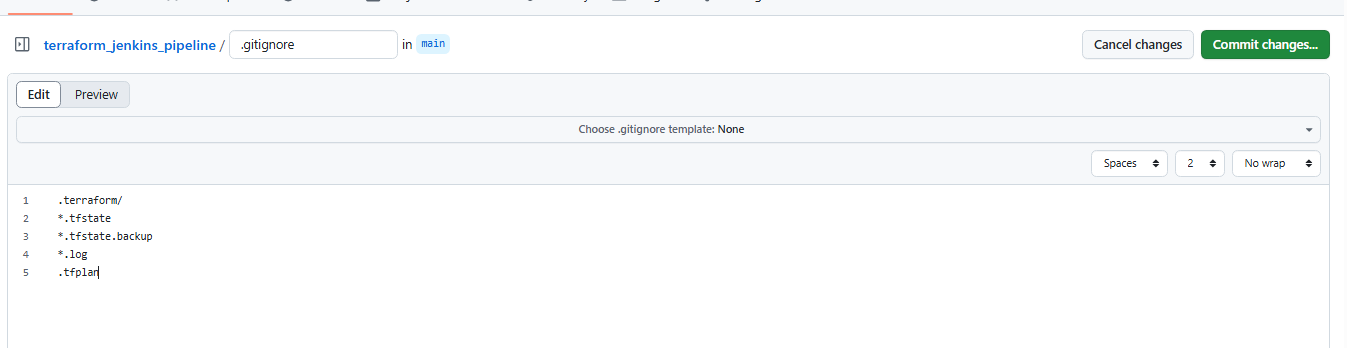
Add the outputs.tf file enter the details as follows and click on Commit changes



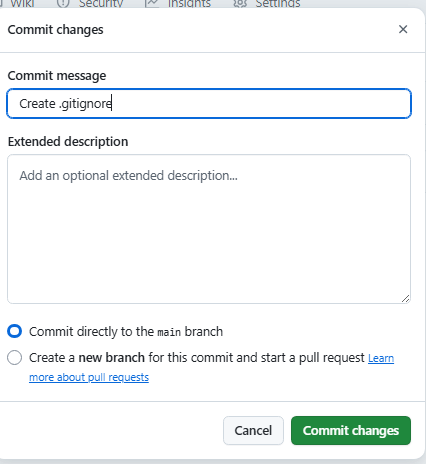
Enter the commit message and click on Commit changes



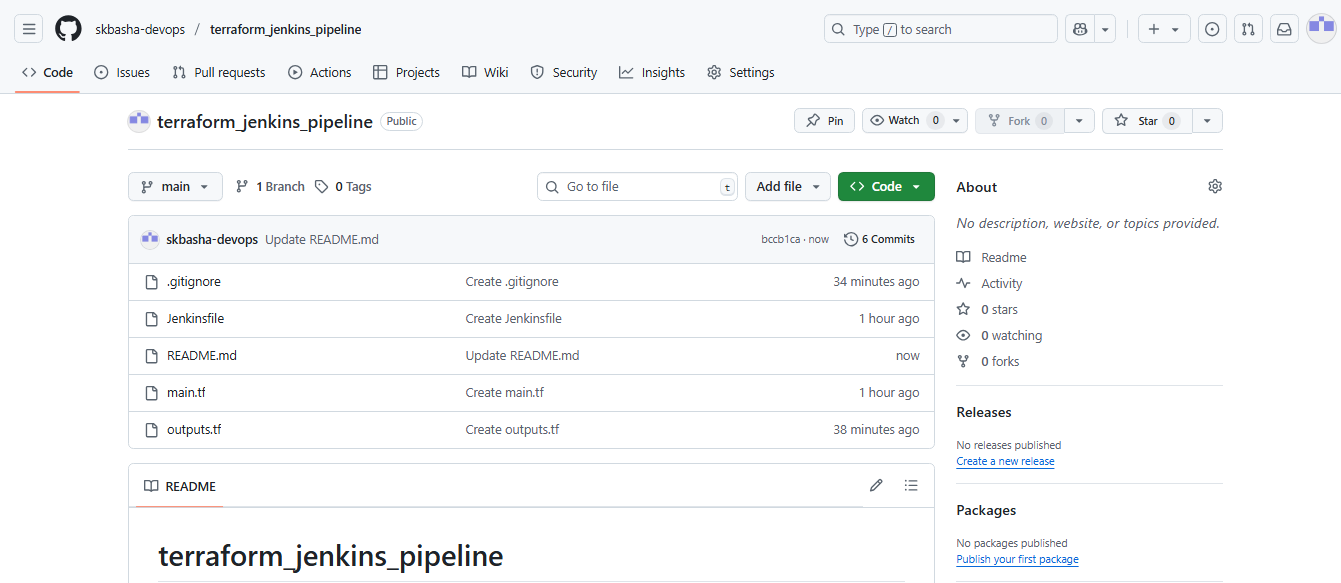
Add the .gitignore file and enter the details as follows and click on Commit changes



Enter the Commit message and click on Commit changes



All the files required are added



Final Repo Structure:

terraform-jenkins-pipeline/

├── .gitignore (optional)

├── Jenkinsfile

├── README.md (optional)

├── main.tf

└── outputs.tf (optional)

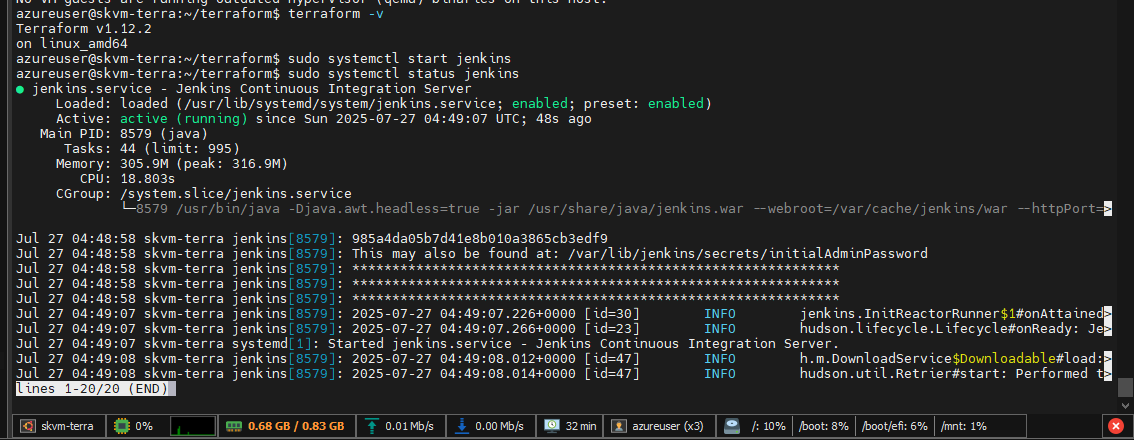
Connect Jenkins to GitHub Repo

In the linux CLI check if terraform is installed and start Jenkins

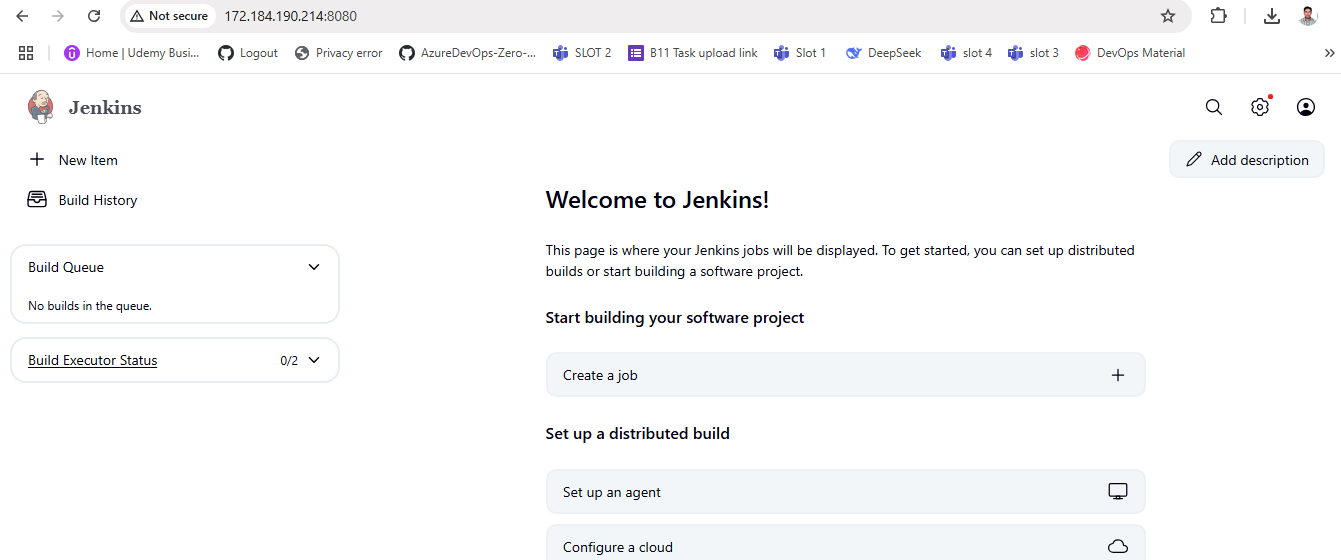
Run commands terraform --version

sudo systemctl start jenkins

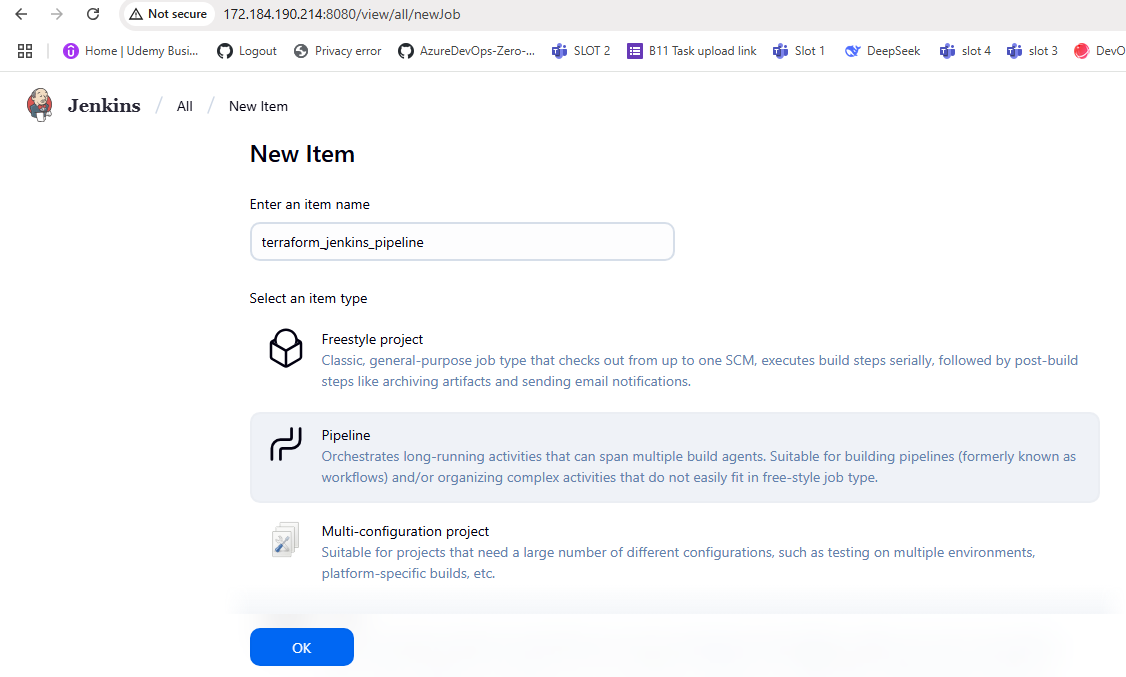
sudo systemctl status jenkins



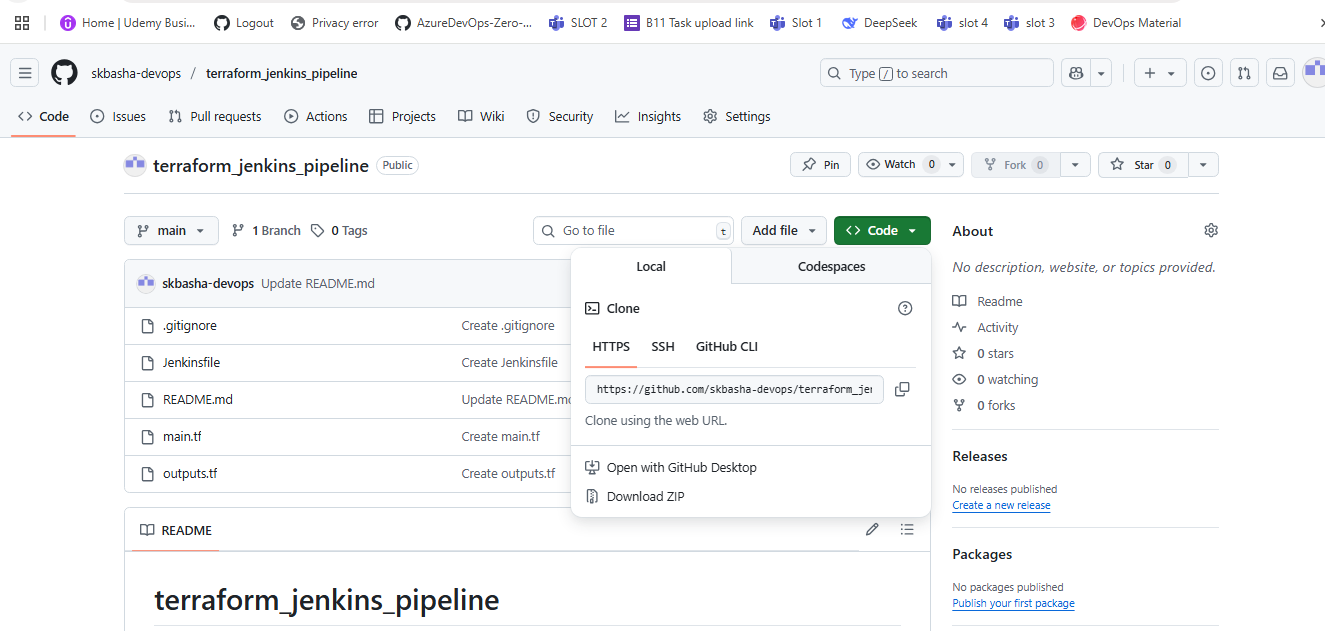
Open Google Chrome and enter 172.184.190.214:8080 , Sign-in to Jenkins using Username and Password



In Jenkins Dashboard click on New Item and select Pipeline job and click on OK



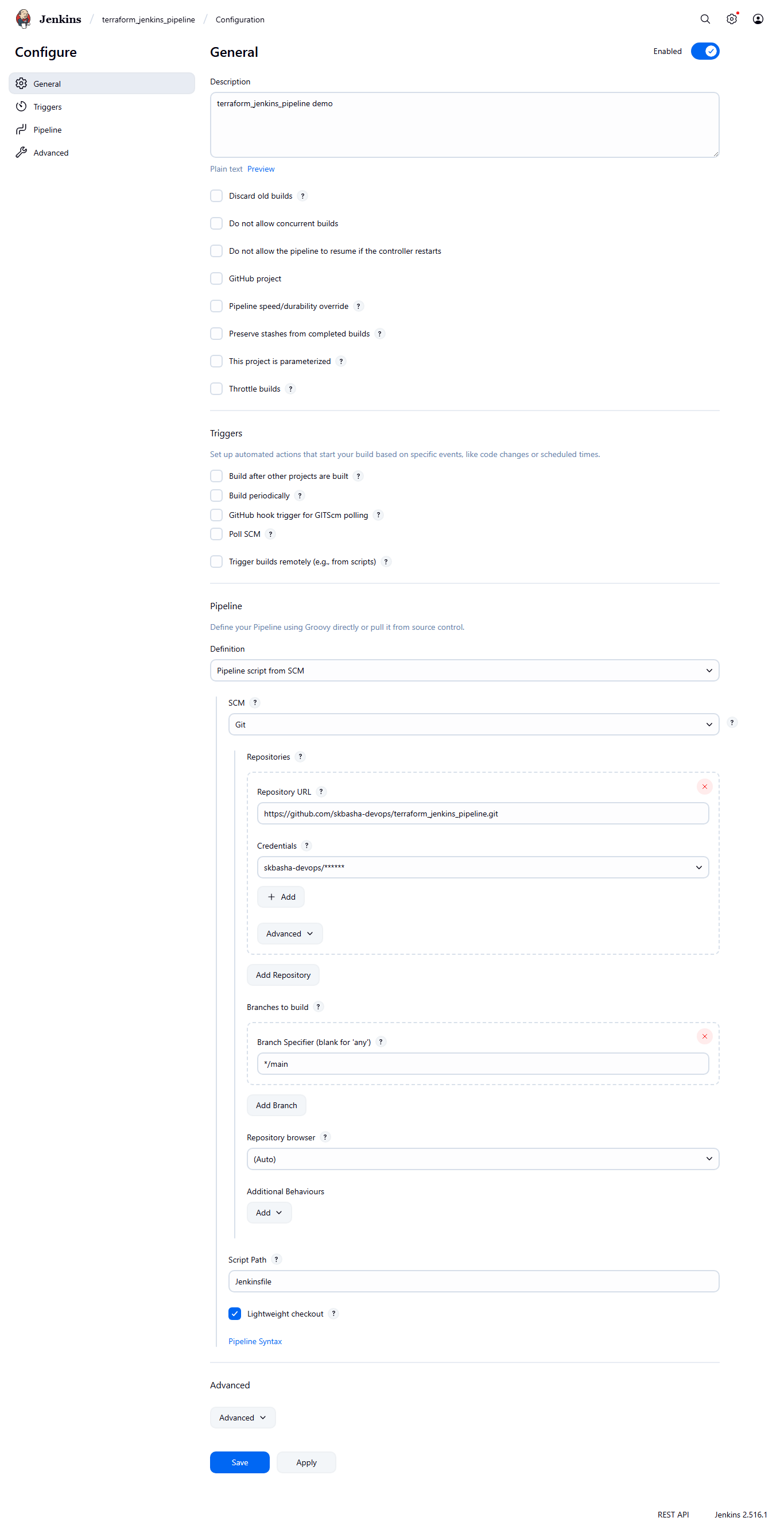
Copy the Repository URL



Enter the details as follows

Give description.

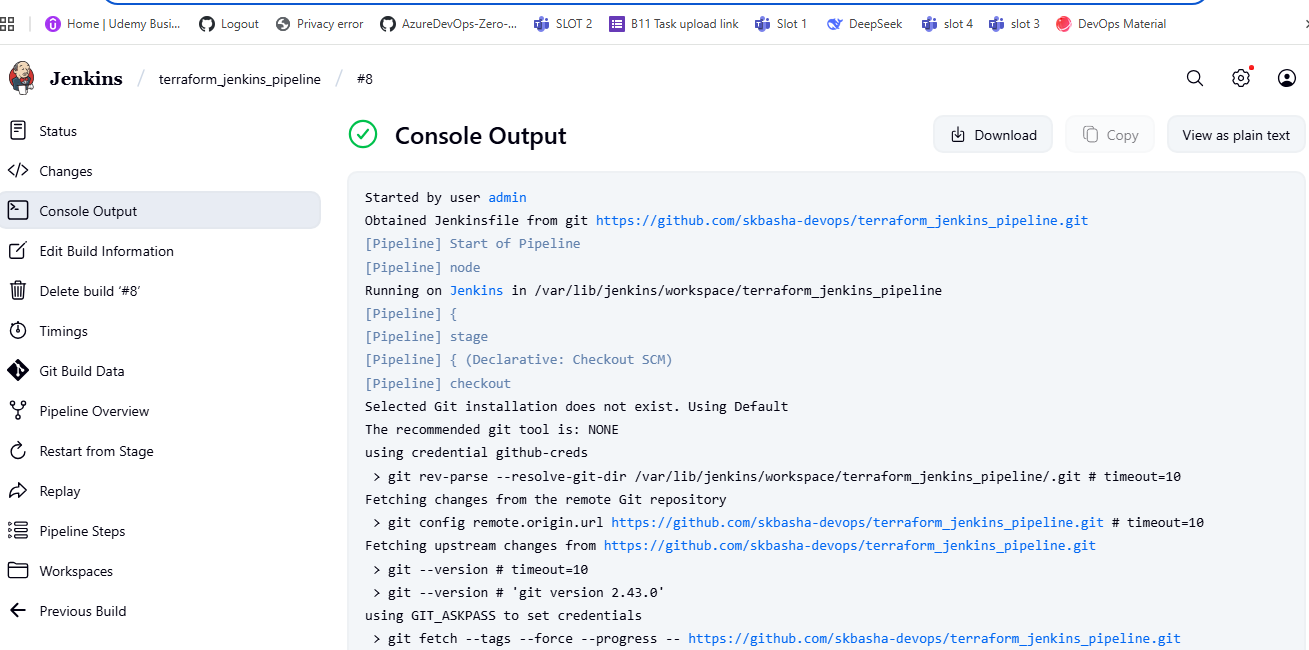
Use "Pipeline script from SCM", Paste the Repository URL, Select the credentials and give branch specifier as \*/main and click on Save

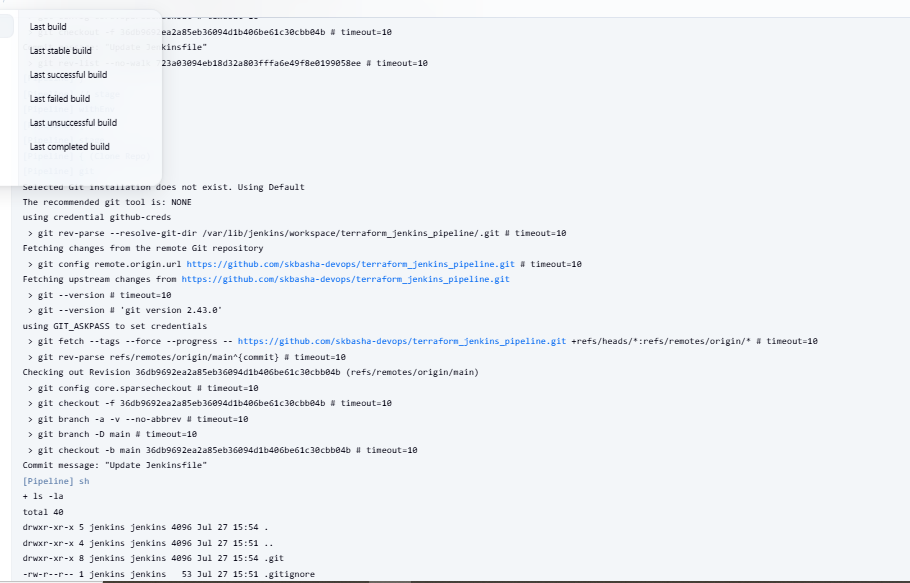


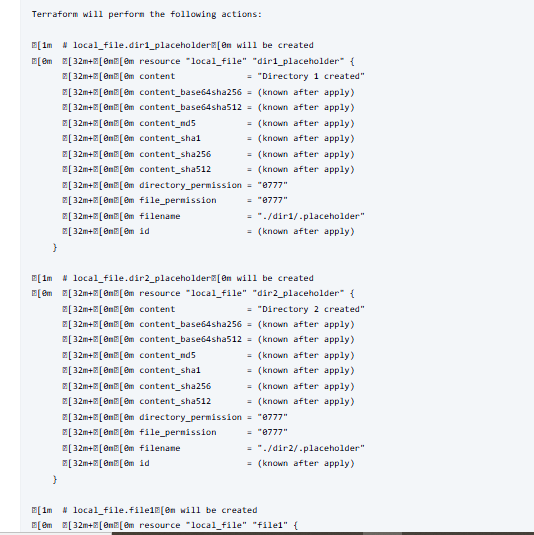
Click on Build Now



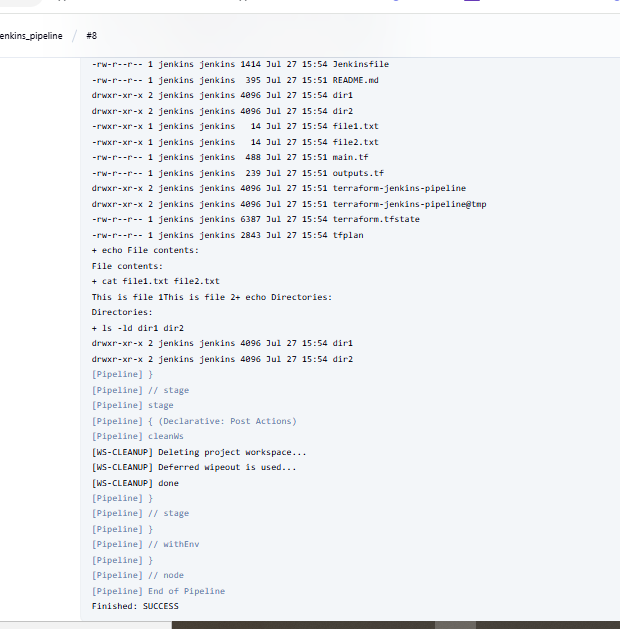
Build is successful, click on #8



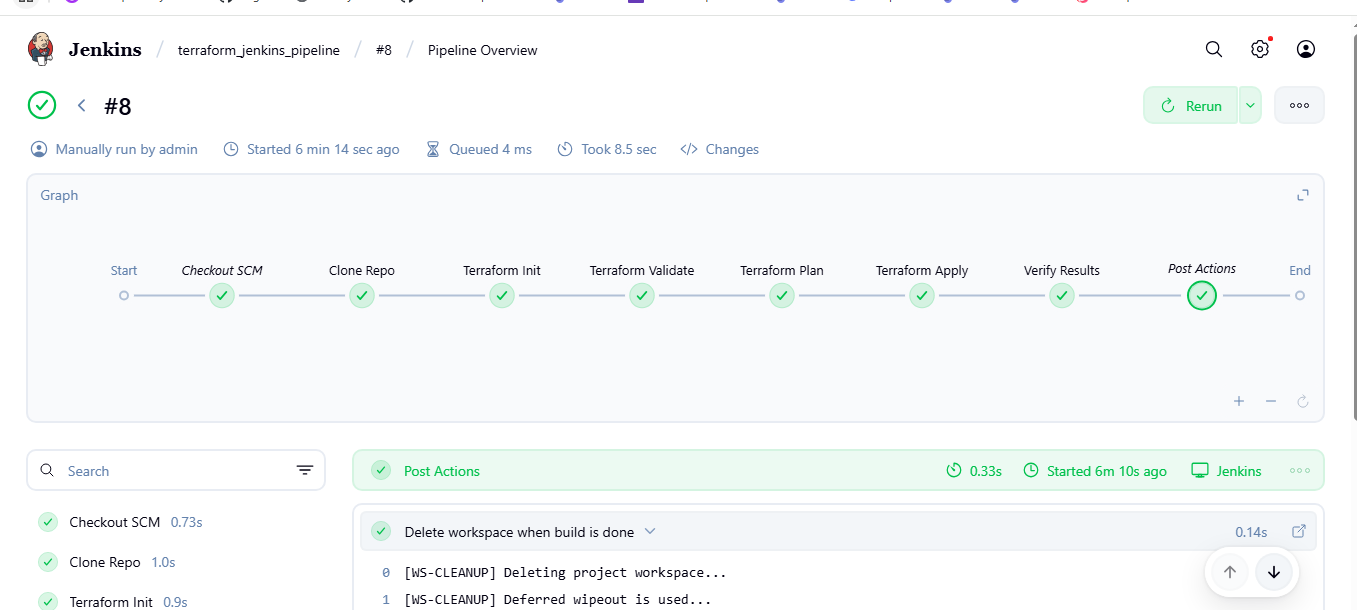




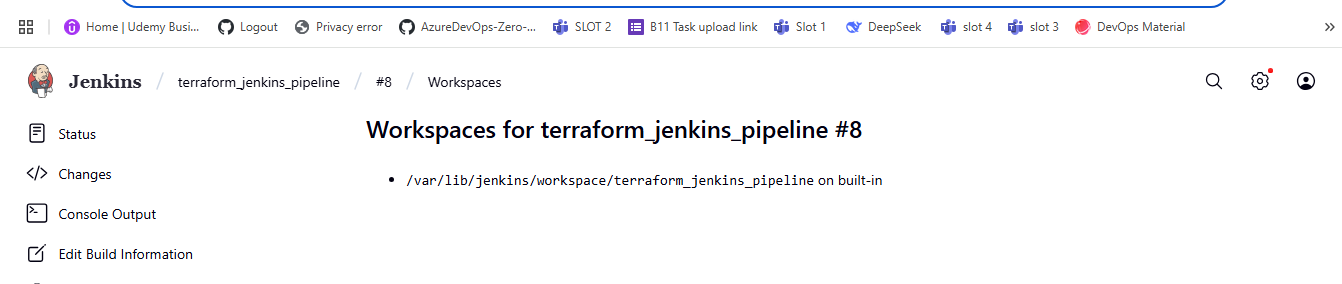




Click on pipeline overview , here we can see the flow



Click on Workspaces where the file are created



In CLI check if the files and directories are added

