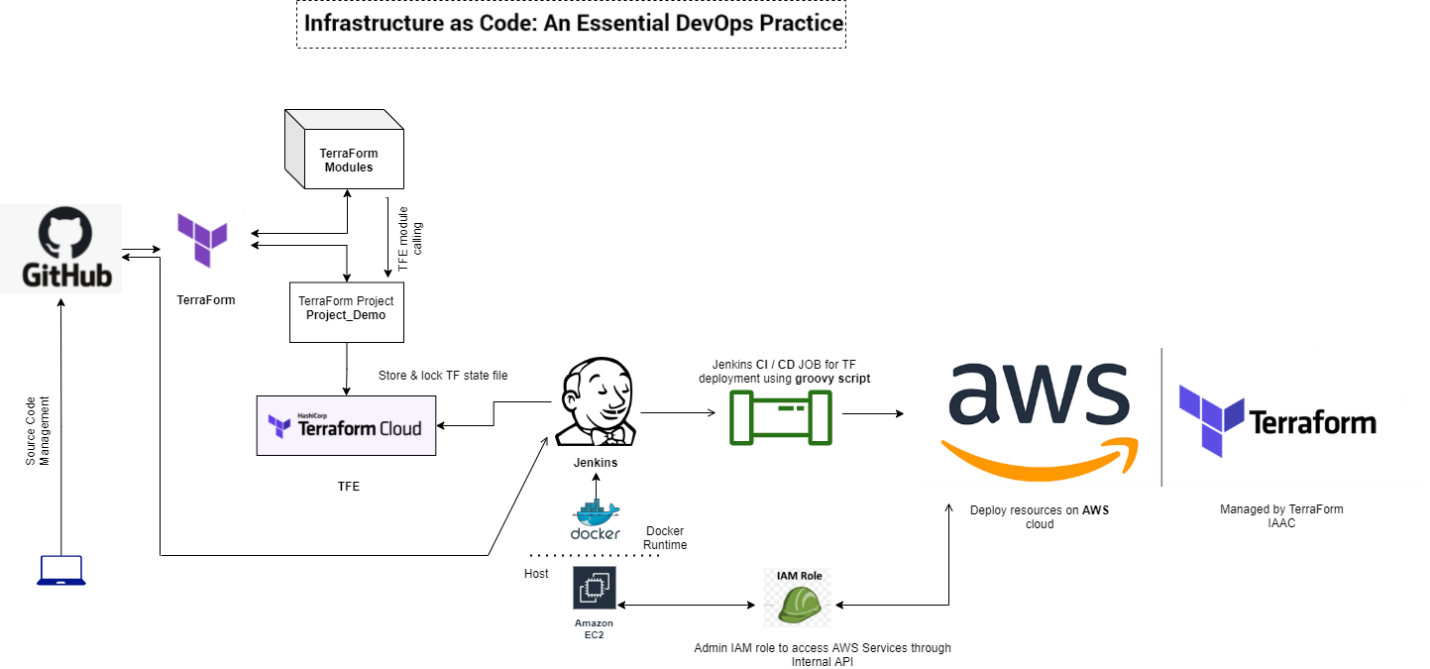
**CI-CD** project to deploy Infrastructure @ **AWS** using **Terraform** IaC.

Detailed Workflow >>>

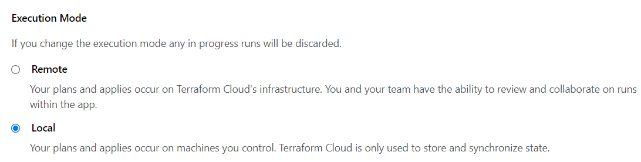


**WorkFlow Components (High level)** : GitOps , Terraform IaC , TF cloud(Free tier), TF modules, Project DIR to deploy resources to Provider: AWS, CI/CD: Jenkins(Hosting: EC2, Runtime Docker),EC2-IAM-ROLE for AWS-Services-API to deploy account Services-API access, CI-CD JOB: groovy scripted pipeline with business logic, Deploy and Destroy resources to AWS Cloud.

• Used Terraform-Module - AWS: [{SCM}/../AWS\_modules/modules](https://github.com/AnikG-Org/devops-practice/tree/main/terraform/AWS/AWS_modules/modules)/\*

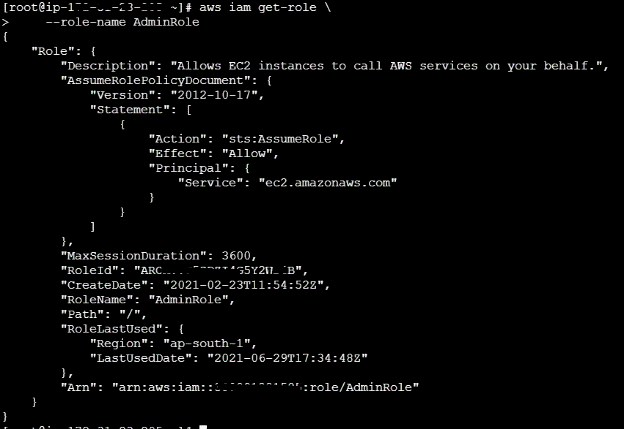
• Deployed "project\_demo" resources using TF modules: [{SCM}/../AWS\_modules/projects/project\_demo](https://github.com/AnikG-Org/devops-practice/tree/main/terraform/AWS/AWS_modules/projects/project_demo)/\*

Note: [z\_backend-remote-tfcloud-state.tf](https://github.com/AnikG-Org/devops-practice/blob/main/terraform/AWS/AWS_modules/projects/project_demo/z_backend-remote-tfcloud-state.tf) is capable to auto create TF cloud workspace based on local workspace selection.

Created workspace at TF cloud has Default **Execution mode:** Remote. Here Called Module for this project **“../{Module\_PATH}”**is **local** **source**,hence TF cloud workspace need to set > **Execution mode:** [Local](https://app.terraform.io/app/AnikG-Org/workspaces/project-demo-prod/settings/general)

**Jenkins** (CI CD Tool) on Docker:

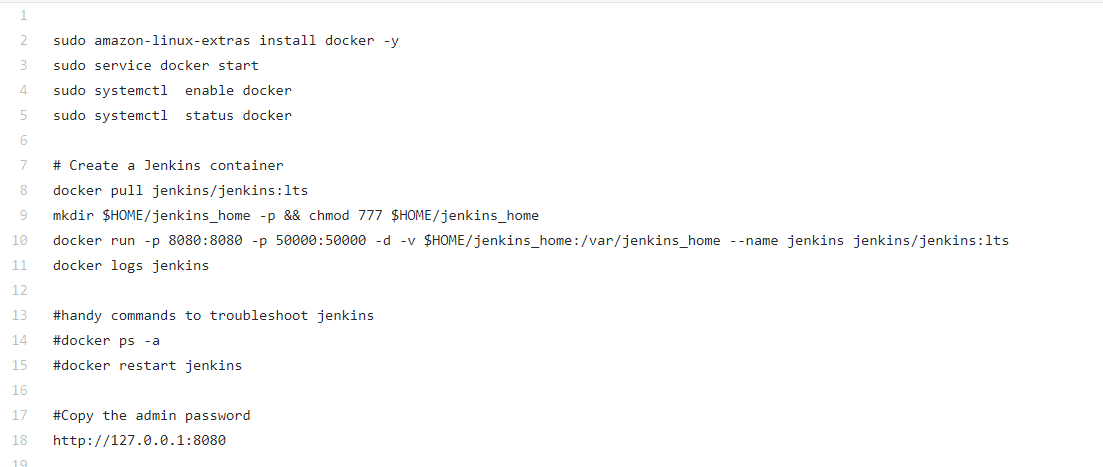
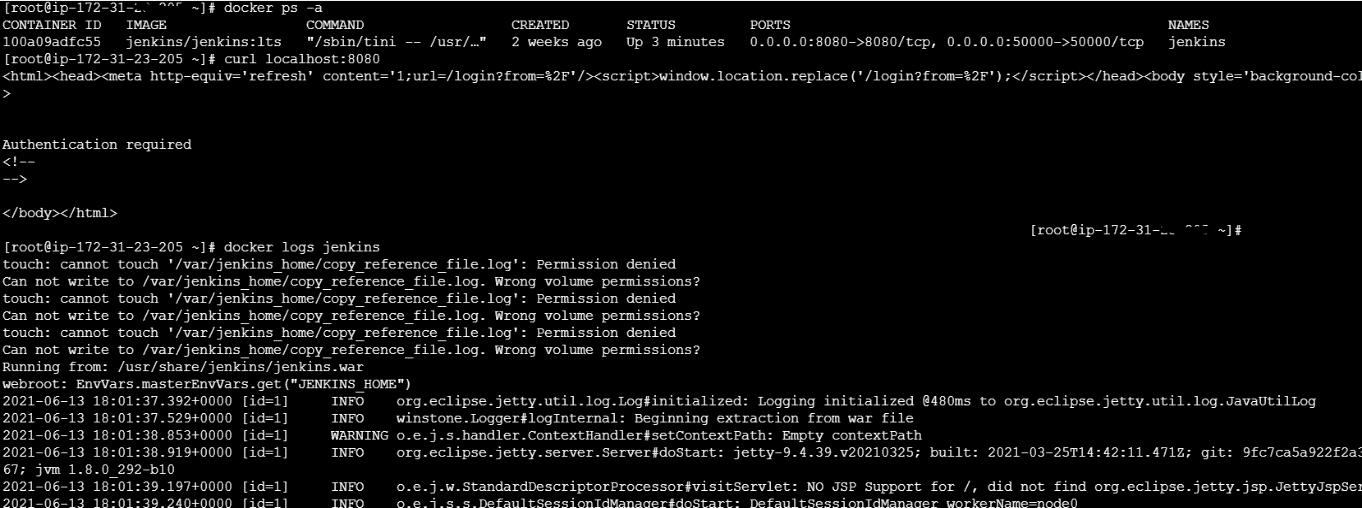
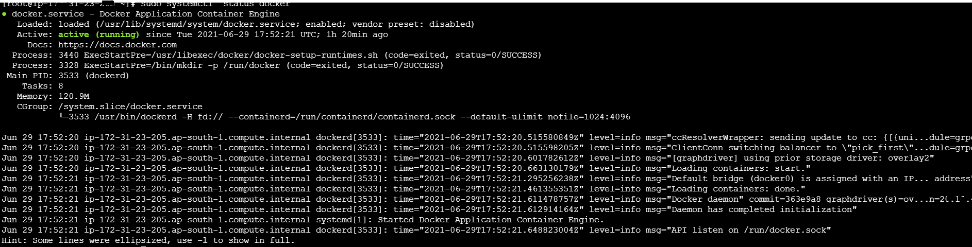
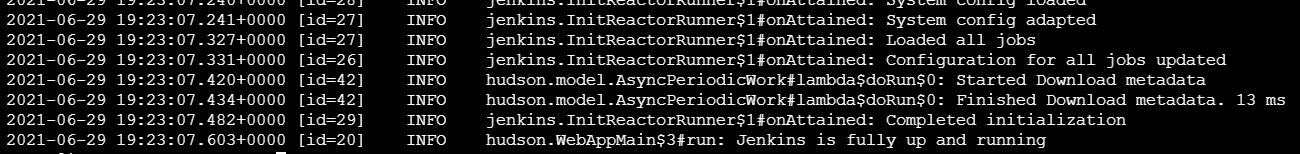
**EC2**: >>

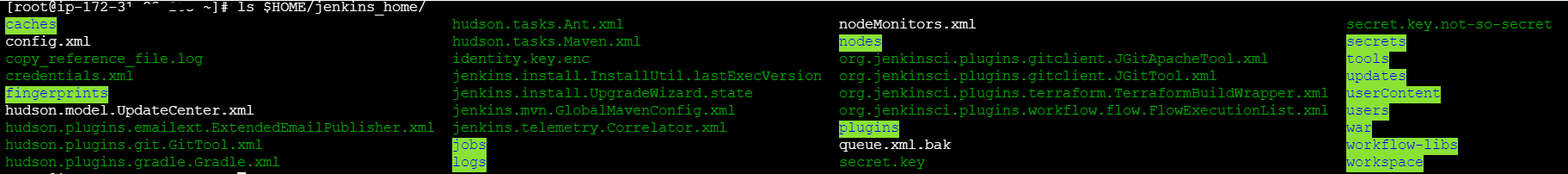
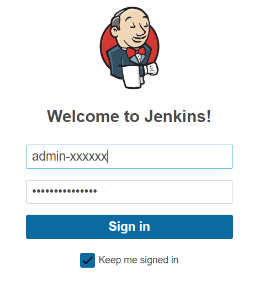
Step 1: Done Setup Ec2 instance with AL2 OS flavor and dependent configs. Attached Preconfigured AdminRole for EC2.

EC2-IAM-ROLE to get privilege on AWS Services-API to deploy AWS account Services-API access (Admin Role)

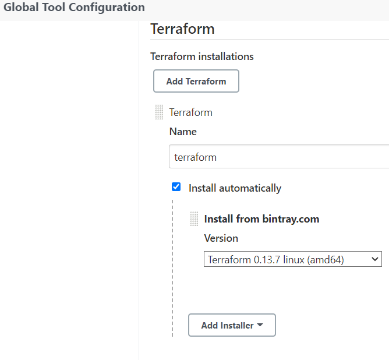
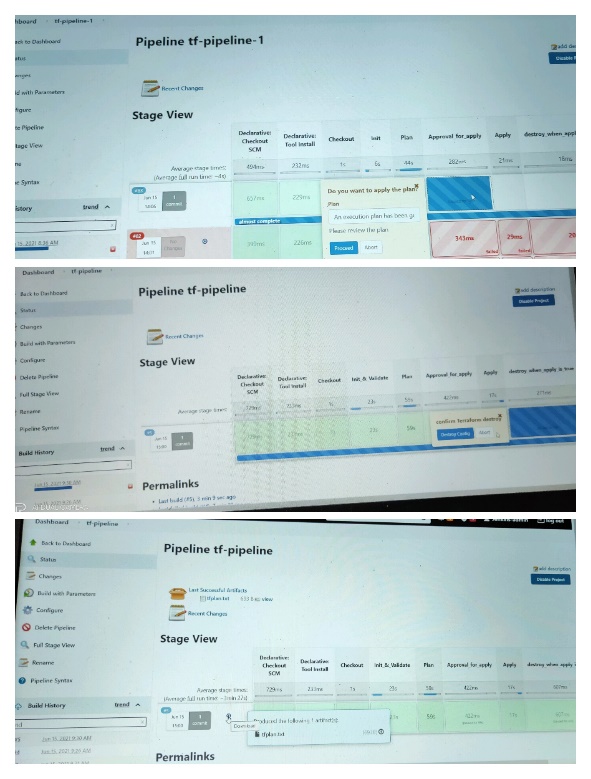
Step 2:

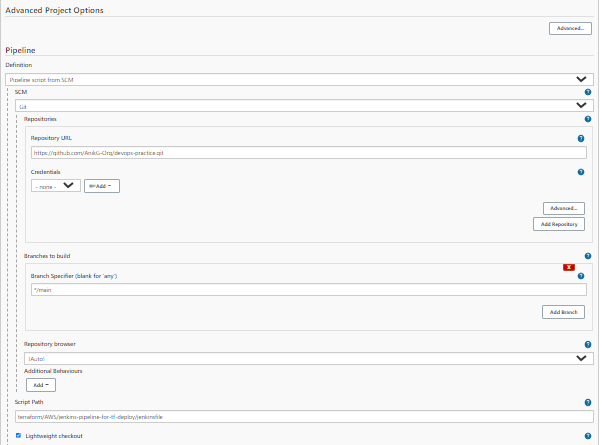
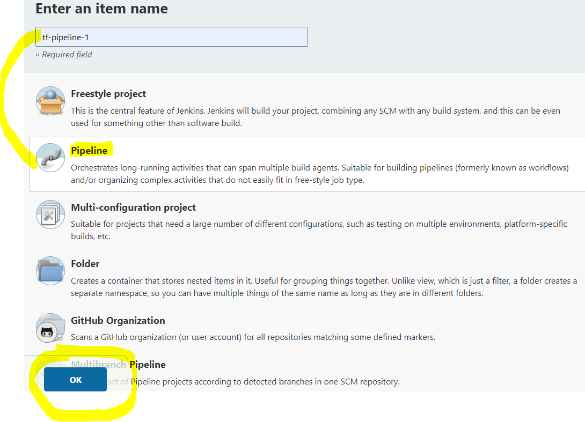
Install & configure Docker runtime & Docker PV & Jenkins setup on Docker.

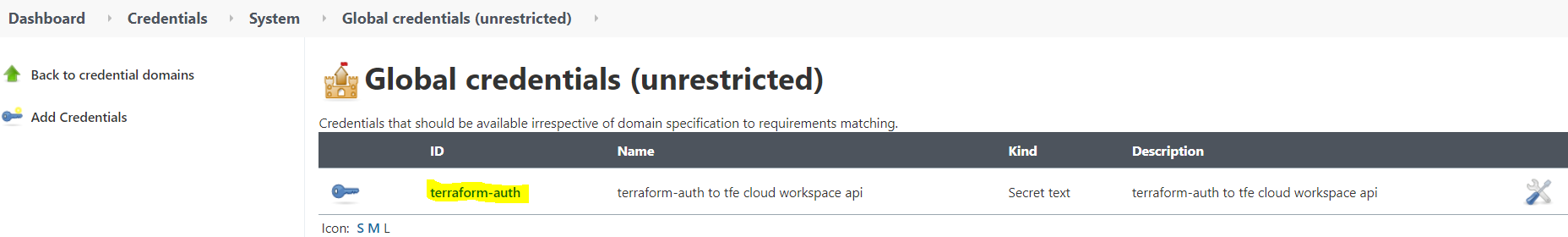
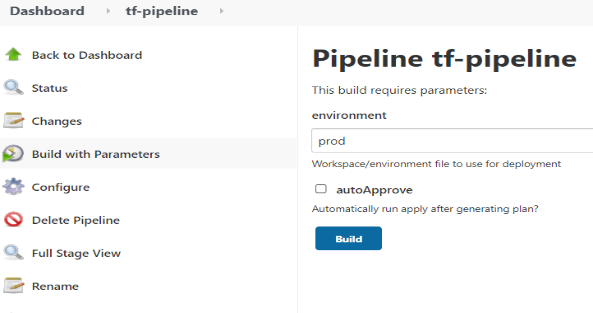


Step 3: Verify Jenkins & Docker PV path Up and running.

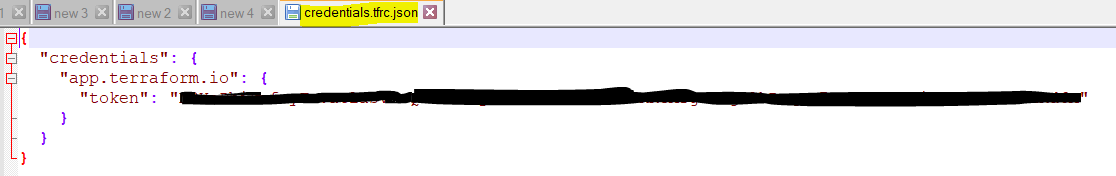


Step 4: Setup Jenkins & GIT, terraform & AWS & Global creds based on this project requirement >>>>>

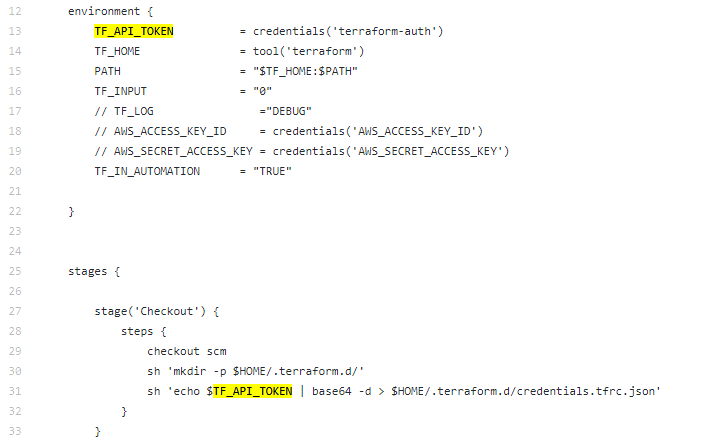
Step 5: Jenkins Pipeline setup >>>



**Key Note**: Here I Configured **terraform-auth** to tf cloud resources from Jenkins pipeline via TF **API** token.

credentials.tfrc.json template with API Token >>

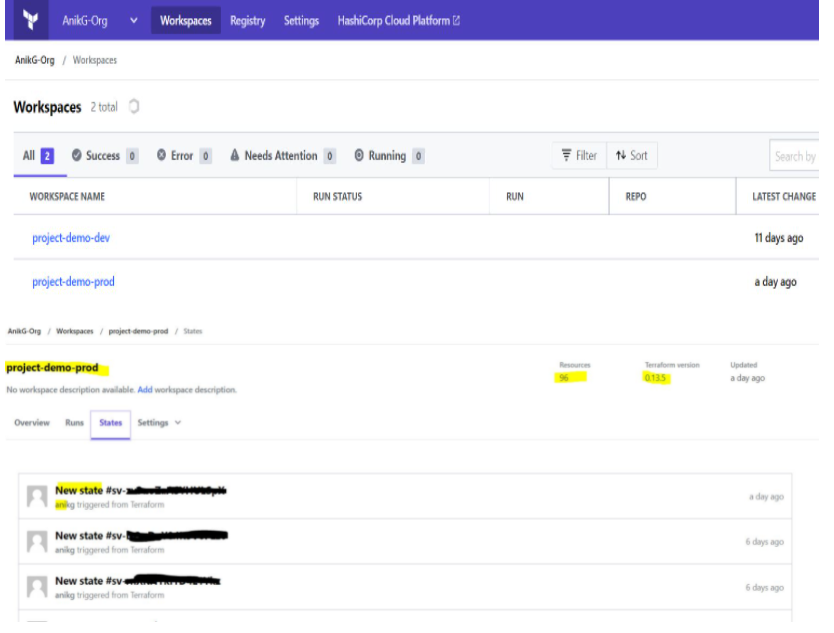
Encrypt credentials.tfrc.json

* base64 -w0 credentials.tfrc.json > secret\_file.txt
* Will get a single sting secure bas64 output.
* That will output store as **terraform-auth** as secret text @ Jenkins.
* environment TF\_API\_TOKEN used to will get **terraform-auth** cred apply usage at stage Checkout with decryption method.
* Now Pipeline is capable to communicate to TF cloud backend.
* Jenkins\_Pipeline: [{SCM}/.../jenkins-pipeline-for-tf-deploy/jenkinsfile](https://github.com/AnikG-Org/devops-practice/blob/main/terraform/AWS/jenkins-pipeline-for-tf-deploy/jenkinsfile)

This Jenkinsfile(Groovy script) use custom installed plugin: "org.jenkinsci.plugins.terraform.TerraformInstallation" "terraform" \*\*custom name

This pipeline has validations, multiple conditions and approval stages , destroy condition, and artifact output.

Step 5: Verify your state file and status at TF CLOUD: <https://app.terraform.io/>



& Verify AWS deployments.