JENKINS PIPELINE README

Tools used for the pipeline : Jenkins, Github repo to store terraform code and Jenkinsfile

Jenkinsfile: I prefer to use Jenkinsfile for pipeline builds as it gives a declarative approach to define our stages,steps and take variables from parameters, moreover it gives flexibility to edit the pipeline and store in a Source code repository, therefore it gives transparency among the team and better understanding of the pipeline stages defined.

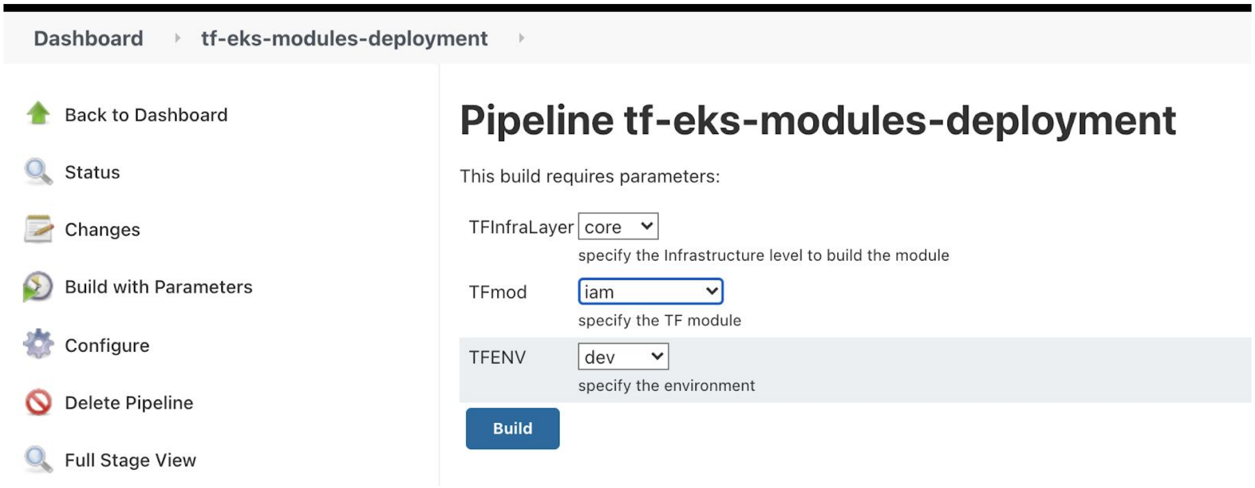
Jenkinsfile Location : Root DIrectory of the terraform repository

Prerequisites: A working jenkins server with aws access and terraform version 13 binary installed. I have used an ec2 instance for jenkins installation, so that I can easily **attach required aws IAM role with necessary policies** to the server to create different aws resources for this project.

I have taken Jenkins as the tool to build the terraform pipeline. As we need to trigger the pipeline module wise, I used the “**parameterized build**” option in jenkins.

By using “**parameterized build**”,

1. Easy to target the terraform module by selecting during the pipeline build



Pipeline tf-eks-modules-deployment

This build requires parameters:

TFInfraLayer: core
specify the Infrastructure level to build the module

TFmod: iam
specify the TF module

TFENV: dev
specify the environment

2. Add/remove parameters easily with jenkinsfile

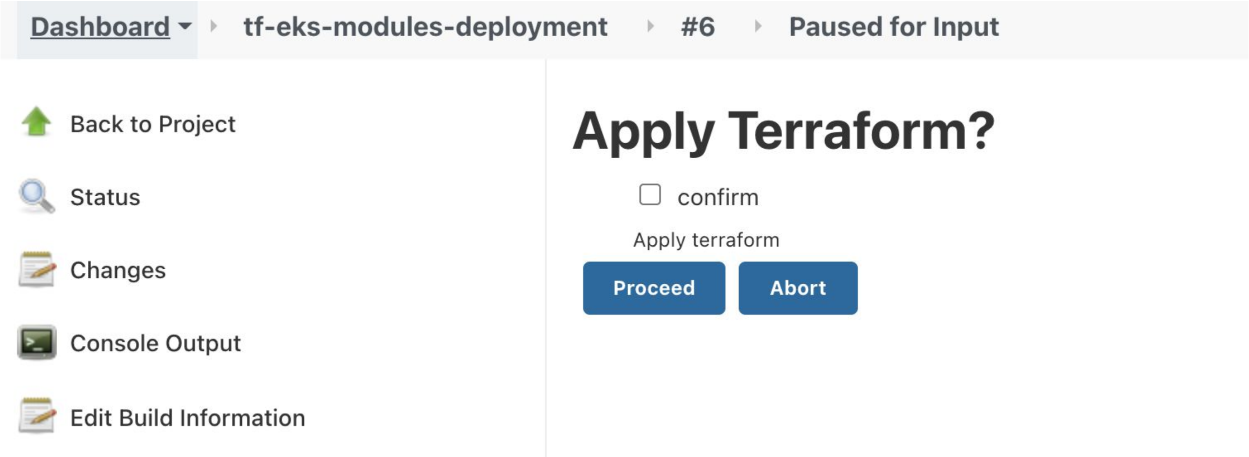
As I have divided the terraform project into multiple layers for easy understanding and layered architecture, we have options to select the infrastructure layer during the build.

Jenkinsfile have 4 stages

Stage -   Terraform init

Stage -   Terraform plan

Stage -   Approval



Apply Terraform?

□ confirm

Apply terraform

Stage - Terraform apply

Once we check the terraform plan output and confirm that everything works as expected, we can give confirmation to jenkins to apply terraform.
