

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.

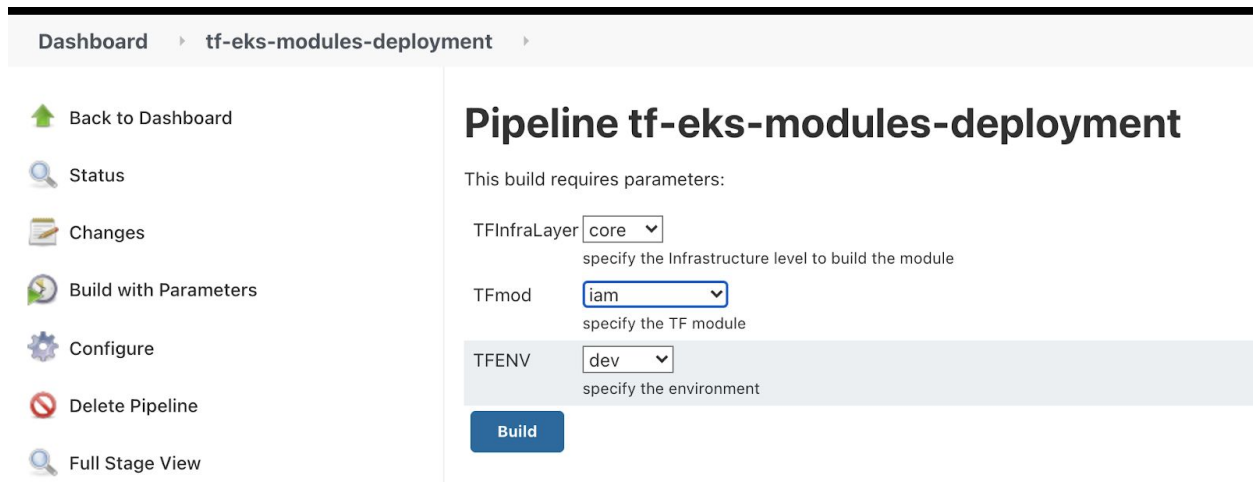
Jenkinsfile Location : Root Directory of the terraform repository

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



The screenshot shows the Jenkins web interface for configuring a pipeline. The breadcrumb navigation at the top reads 'Dashboard > tf-eks-modules-deployment >'. On the left sidebar, there are links for 'Back to Dashboard', 'Status', 'Changes', 'Build with Parameters' (which is highlighted), 'Configure', 'Delete Pipeline', and 'Full Stage View'. The main content area is titled 'Pipeline tf-eks-modules-deployment'. Below the title, it states 'This build requires parameters:'. There are three parameter fields: 'TFInfraLayer' with a dropdown menu showing 'core' and a description 'specify the Infrastructure level to build the module'; 'TFmod' with a dropdown menu showing 'iam' and a description 'specify the TF module'; and 'TFENV' with a dropdown menu showing 'dev' and a description 'specify the environment'. At the bottom of the configuration area is a blue 'Build' button.

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


[Dashboard](#) ▾ ▸ [tf-eks-modules-deployment](#) ▸ [#6](#) ▸ [Paused for Input](#)

 [Back to Project](#)

 [Status](#)

 [Changes](#)

 [Console Output](#)

 [Edit Build Information](#)

Apply Terraform?

☐ confirm

Apply terraform

[Proceed](#) [Abort](#)

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.
