

This document will guide you through the setup and usage of Terraform from your system/machine.

For Terraform scripts first of all you need access to this repo.

[https://gitlab.com/yudiz\\_devops/sport-gully/terraform.git](https://gitlab.com/yudiz_devops/sport-gully/terraform.git)

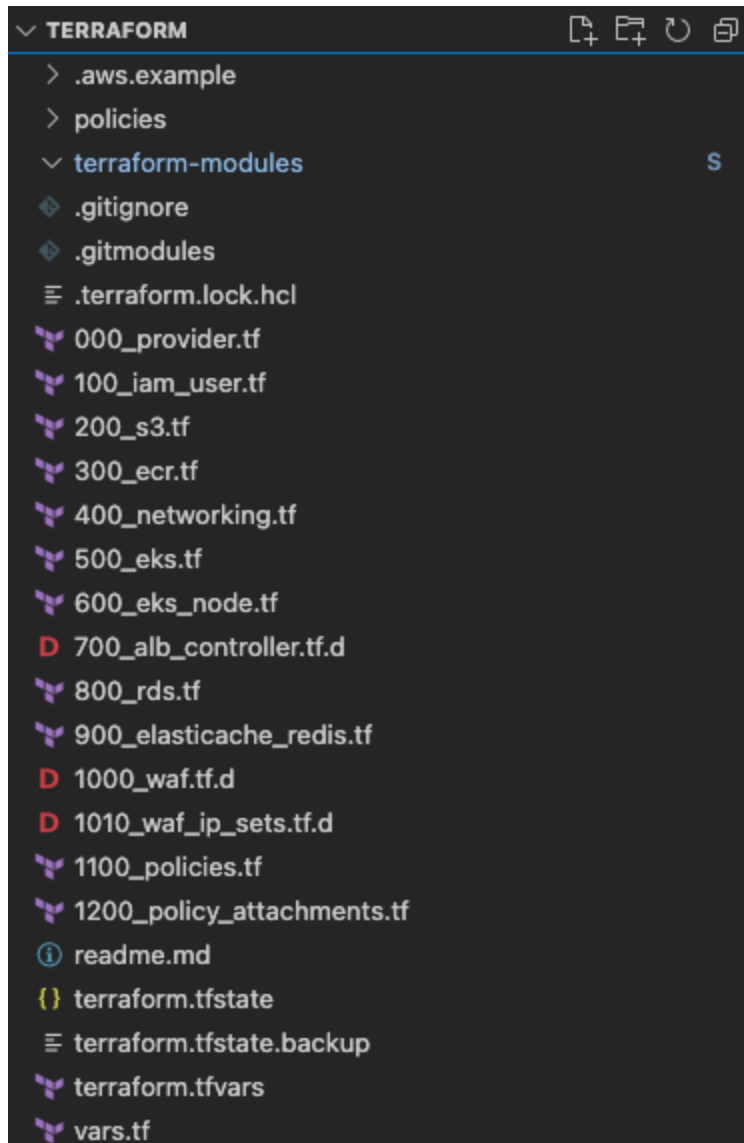
We have created our own terraform modules and the repo link for that is.

[https://gitlab.com/yudiz\\_devops/terraform-modules.git](https://gitlab.com/yudiz_devops/terraform-modules.git)

## **Installing Terraform on various platforms**

[Install Terraform](#)

**Note: we have used these Custom generated modules as Git Submodules in Terraform repository**



When you clone this repository its directory structure will look as in the above image.

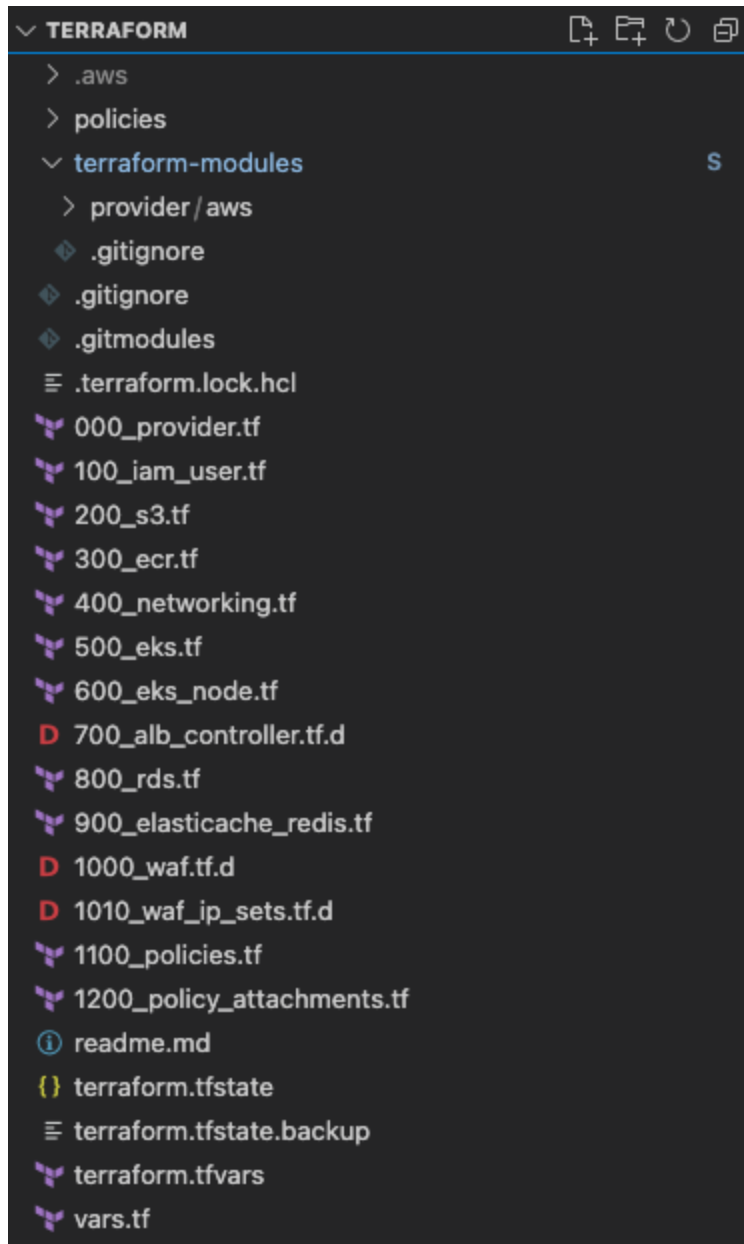
Now in order to use this repository, you need to perform two steps:-

1. Rename .aws.example folder to .aws
2. Pull our Terraform Modules. ( git submodules )

## 2nd Step: Update submodules

```
git submodule update --init --recursive
```

After doing the above 2 steps your repository will look like this.



Now you have to enter your credentials in order to provision the Infrastructure as scripted in terraform files in .aws -> credentials file.

**IMP NOTE:** Infrastructure not generated with terraform ( manually generated because of some issue or not compatible with terraform configuration )

```
ALB Controller  
WAF  
WAF IP sets  
Elasticache Redis
```

## Overall Infrastructure

- **000\_provider** - Initializing the providers such as Helm, Kubernetes, AWS
- **100\_iam\_user** - Generating 2 IAM users - 1 - KYC User, 2 - S3\_full\_access

- **200\_s3** - 6 Buckets are getting created

```
fantsportiz-kyc-1638949667
fantsportiz-media-1638949732
fantsportiz-cdn-logs-1638949745
fantsportiz-mongodbdumps-1638949745
fantsportiz-mysqldumps-1638949745
fantsportiz-firebase-sdk-1638949745
```

- **300\_ecr** - 12 repositories are getting created.

```
Mongo-backup
Mongo-backup-staging
Mysql-backup
Mysql-backup-staging
fantsportiz-admin-panel
fantsportiz-admin-panel-stag
fantsportiz-app
fantsportiz-app-stag
fantsportiz-node
fantsportiz-node-backend
fantsportiz-node-backend-stag
fantsportiz-node-stag
```

- **400\_networking** - 1 VPC, 1 Internet Gateway, 1 Elastic IP,  
3 Public\_Subnets - in each AZ ( availability zones )  
3 Private Subnets - in each AZ ( availability zones )  
1 NAT Gateway - in one AZ ( you can create in all three AZ )  
1 Public Route Table  
3 Private Route Table  
2 Hosted Zones - Private and Public.
- **500\_eks** - EKS cluster IAM role, EKS cluster with private endpoint ( worker nodes will reside in the private subnet ).

- **600\_eks\_node** - EKS worker node role, 3 Node groups are created
  - 1 - Default node group - targets all 3 AZs with all the node Add-ons like VPC\_CNI, Kube Proxy, CoreDNS.
  - 2 - Node group of AZ A ( workload based because of mongo, MySQL, Redis and monitoring stack )
  - 3 - Node group AZ C ( workload based because of mongo, MySQL, Redis and monitoring stack )
- **800\_rds** - 1 Subnet Group, 1 Parameter Group, 1 RDS Security Group, 1 RDS monitoring role, 1 RDS ( as of now t2.micro you can modify all the parameters through this file ), 1 Custom Parameter Group.
- **1100\_policies** - 4 Policies are getting created
  - 1 - kyc\_restriction - IAM policy
  - 2 - Bucket policy of KYC - Bucket Policy
  - 3 - S3\_full\_access - IAM policy
  - 4 - bucket policy for media bucket
- **1200\_policy\_attachments** - this file attaches the policy to necessary components.
  - Attaching Kyc policy to KYC user.
  - Attach IAM policy to S3 full access user.

## Global Variables ( terraform.tfvars and vars.tf )

Some of the variables are declared globally because they are used in each and every file and some configurations are important like in **terraform.tfvars** file.

```
Global Variables Sensitive - account number, terraform user name, AWS
region.
Tags Information - Tags to assign to the all resource.
Domain names for Route53 - names of public and private zones
RDS Params - rds_deletion_protection = true
```

## After doing the above configurations

You can fire the commands of the terraform and launch the infrastructure.

1. terraform init - initialize the modules you have created.
2. terraform plan - it will show you the bunch of resources to be created by terraform.
3. terraform apply - it will create the infrastructure by once taking your acknowledgment.
4. terraform refresh - to refresh the infrastructure created by terraform.