Creating Infrastructure( You can run the terraform apply to get these deployed automatically, update test.tfvars with desired variable values)

* Creating a VPC with desired subnets, route tables, and internet gateway.
* Creating a security group allowing inbound traffic on the desired ports.
* Creating an Application Load Balancer (ALB) and configure the target group.
* Creating an ECR to save the image of application.
* Creating IAM role for ECS task definition.
* Creating ECS Task Definition.( configure container configurations and image from ECR repo)
* Creating ECS cluster.
* Creating a service within the ECS cluster and associate it with the task definition.( Desire number of tasks/instances can be set up using Ecs service)
* Configure the ALB listener rules and target group settings.
* Associate the ALB with the ECS service and target group.

Build and Push Docker Image( Run the deployment\_script.sh to automate the below process)

* Use the Docker file to create the image of the Fask Api.
* Build the Docker image locally using the Docker CLI.
* Authenticate with your ECR repository using AWS CLI.
* Tag the Docker image with the ECR repository URL.
* Push the Docker image to the ECR repository.

Execute the deployment\_scripts.sh which performs the following steps:

* Builds and pushes the Docker image.
* Updates the ECS service with the new task definition.
* Waits for the ECS service to stabilize.
* Performs a health check using the ALB DNS name.

**Steps to follow to deploy the infra and Application**

* Configure AWS CLI
* Install and setup git bash.
* Install and set up docker(set environment variable)
* Install and set up terraform.(set environment variable)
* Clone the git repository.
* Update the test.tfvars with desired values( please set container\_port , hostPort and listner-port-tg as 5000 as the container is running as the application is listening to port 5000 and I have health check command at port 5000)
* Run below commands from the path where git repo is cloned.

terraform init

terraform plan -var-file="test.tfvars"

terraform apply -var-file="test.tfvars"

Above step will deploy the required infrastructure mentioned above to AWS.

Make a note of output variables which you get from running terraform apply as this is required to run deployment script.

* Run the script deployment\_script.sh to dockerize the flask app, push to ECR, update the task definition and check the health of application using application load balancer dns and port number( 5000). I have hard coded the port 5000 in health check where the container is running.