Project now provisions a NGINX container which will act as a load balance for a web application container. To run it on a local machine use the following command.

docker-compose up

It will provision a nginx container and a web application container. The nginx container will route traffic to the web. If you need more web container instances run the following command.

docker-compose up --scale app=4 –build

Project now contains the AWS infrastructure as a code folder which contains the terraform files to provision resources on AWS. This will provision a VPC, Load balancer, ECS Cluster ,Fargate for web app, Codebuild and CodePipline. To provision the AWS resource you need to navigate into the Infrastructure-as-Code folder and run the following commands:

To initialise terraform and state files use

terraform init

To see what resources are going to provision on AWS use the command

terraform plan

To apply the changes to AWS use

terraform apply -auto-approve

To delete the resources from AWS

terraform destroy -auto-approve

Note : After IaC deployment you will get the load balancer DNS name on the terminal output. Pipeline will automatically kick in after deployment and will take 5-10 min to fully deploy the web to fargate service.

- I have used the shared terraform module, some are open source and one I have created for this project in IaC which can be easily copied to other projects. This is much easier to manage and clean the code. I have a few recommendations on code and infrastructure which are below to follow for the future projects.
- Networking/VPC infrastructure code should have separate repository which will be only responsible for provision the shared resources for multiple projects like VPC. That will save us some cost as well and we can refer it to other projects by creating a shared module.
- Using a proper domain name for the web application.
- Use https-ssl for the web app which currently is on http which is not secure.
- For all website project I will recommend using CDN service.
- For security I will suggest using WAF and Shield to protect the web apps.
- Terraform states files should be on s3 bucket and encrypted rather than leaving on a local machine.