Part 3

What have been done?

* Nginx load balancer was containerized and had network with project ASPNET container. Refer folder [nginx](https://github.com/arunnirmal/ao-docker-tech-test/tree/master/nginx)
* AWS Infrastructure resources like VPC, Subnets etc are created using Terraform (IaC automation). Refer folder [infrastructure](https://github.com/arunnirmal/ao-docker-tech-test/tree/master/infrastructure)
* As AWS Cloud is infrastructure hosting platform, CICD pipeline service called AWS Codepipeline was created using Terrafom to automate the build and deployment. Refer folder [IaC-CICDpipeline](https://github.com/arunnirmal/ao-docker-tech-test/tree/master/IaC-CICDpipeline)
* Please refer the Gitrepo <https://github.com/ao-com/ao-docker-tech-test>

Recommended Improvements:

* Please refer the Git repository and clone it to use for your project rather than create from scratch. You can customize based on your application requirements.
* If possible, try to use the same VPC network but create your own subnets resources to deploy your applications.
* Use AWS ECS Fargate service for your applications containers deployment rather than on-premises servers.
* Leverage AWS KMS for Security
* CICD automation rather than using Terraform set up the AWS Codebuild and Code Pipeline services for your containers deployments.
* Design Micro-services for your applications and use EKS for best container orchestration approach