# Deploy a Web Application in Docker Container on AWS Using Terraform

Mykola Shtompel

Goal, tasks

- Goal – deploy a web application in Docker container by creating AWS infrastructure and CI/CD pipeline according to the given requirements
- Tasks:
  - Examination the project requirements
  - Development of the AWS infrastructure
  - Development of the CI/CD pipeline
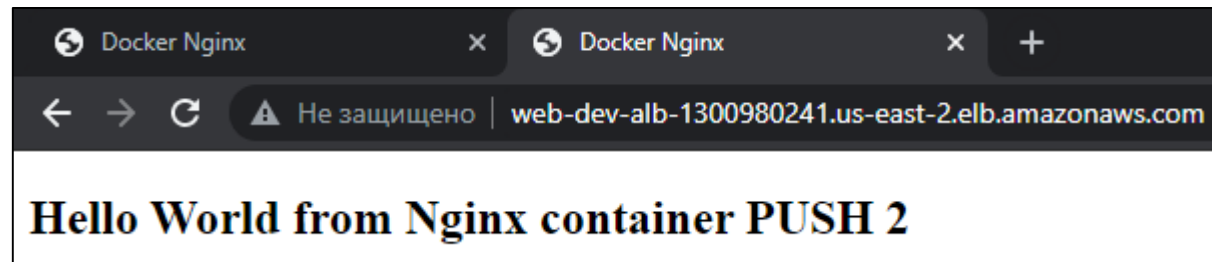 - Implementation of the application deployment procedure using Terraform
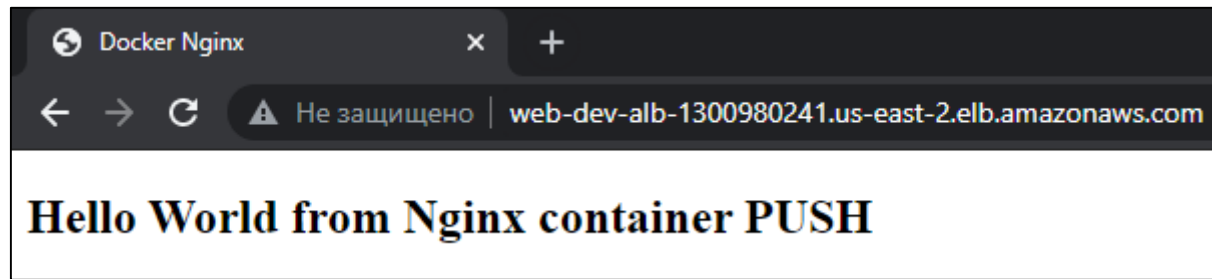
AWS Infrastructure
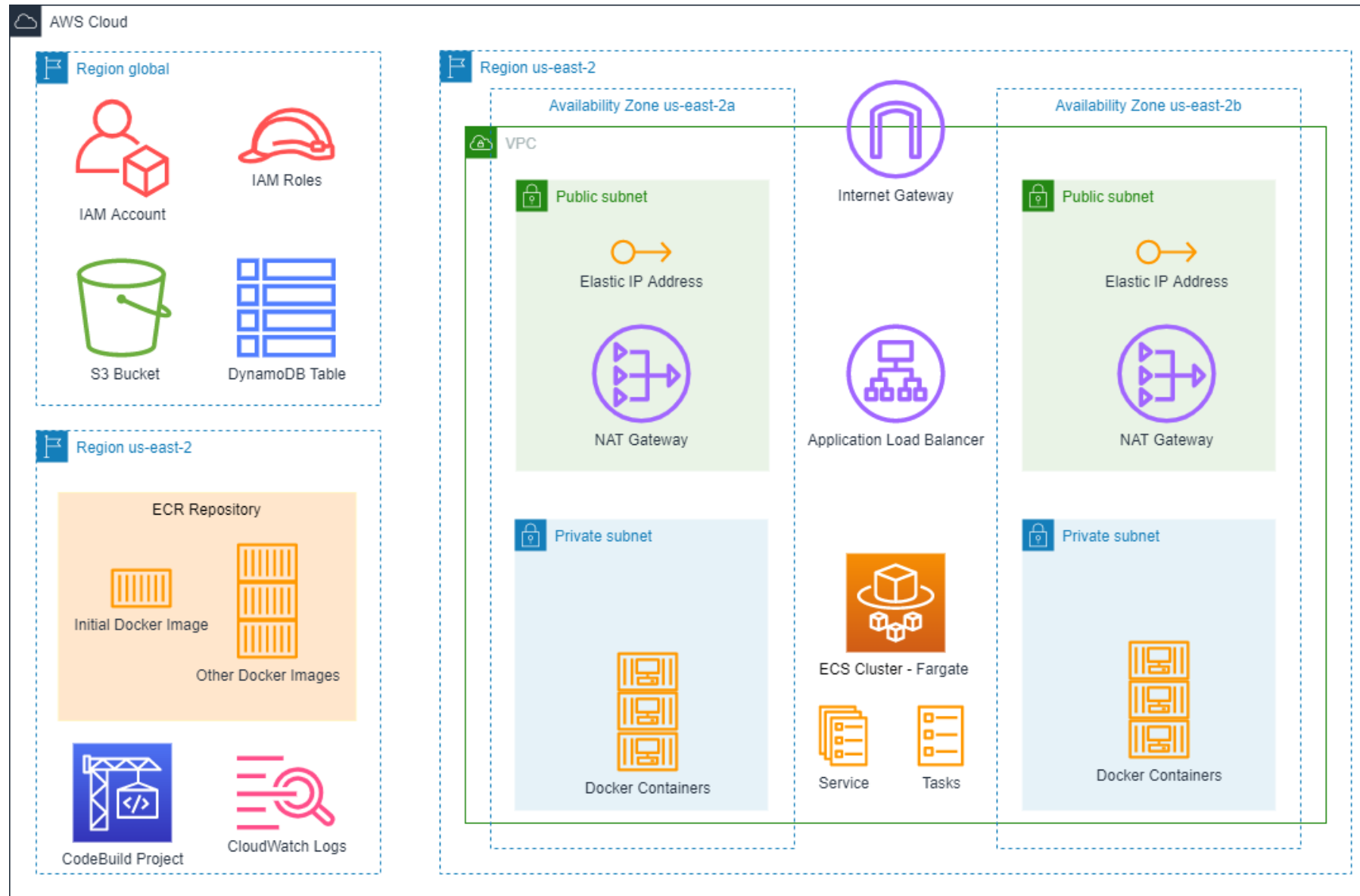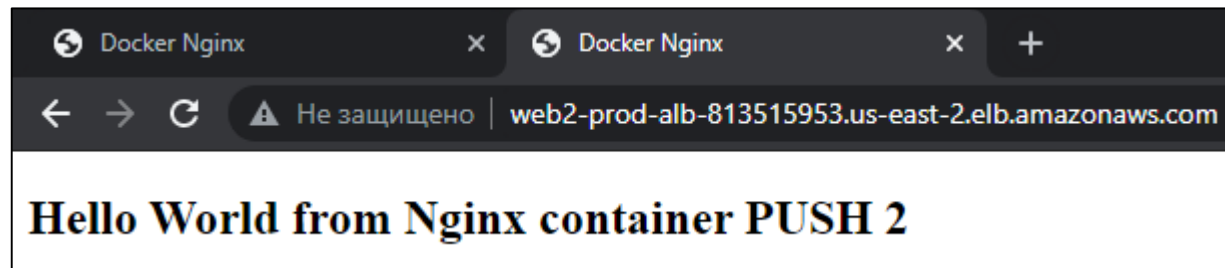
CI/CD Pipeline

Demo – Results – Environment "dev"
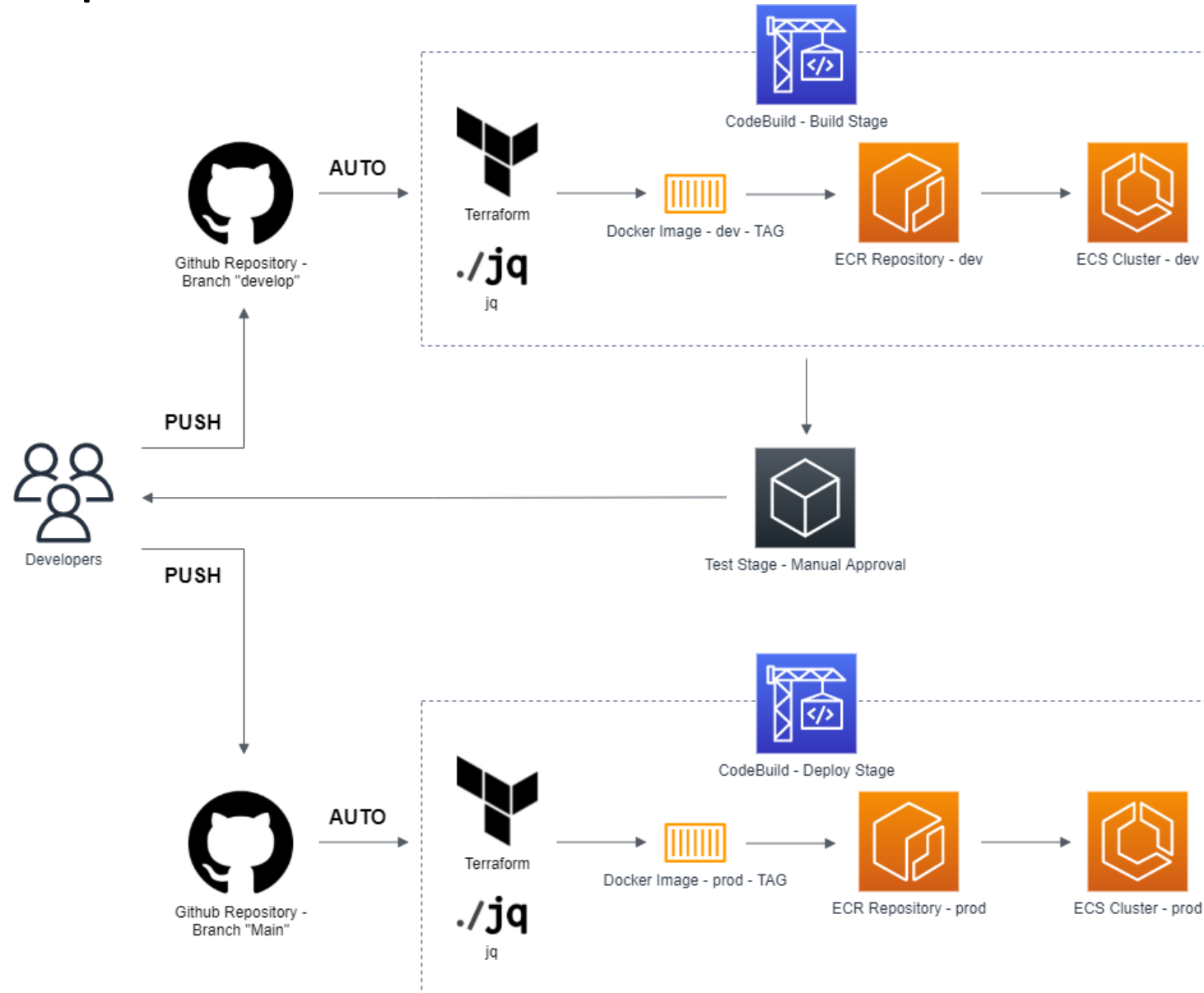
Demo – Results – Environment "prod"

```
Outputs:

account_id = "447854022972"
alb_hostname = "web2-prod-alb-813515953.us-east-2.elb.amazonaws.com"
aws_ecr_repository_url = "447854022972.dkr.ecr.us-east-2.amazonaws.com/web2-prod-nginx2"
aws_region_name = "us-east-2"
codebuild_project_name = "web2-prod-nginx2"
dynamodb_table_name = "table-web2-prod-nginx2-us-east-2"
s3_bucket_arn = "arn:aws:s3:::bucket-web2-prod-nginx2-us-east-2"
s3_bucket_name = "bucket-web2-prod-nginx2-us-east-2.s3.amazonaws.com"
```


Hello World from Nginx container PUSH

Hello World from Nginx container PUSH 2
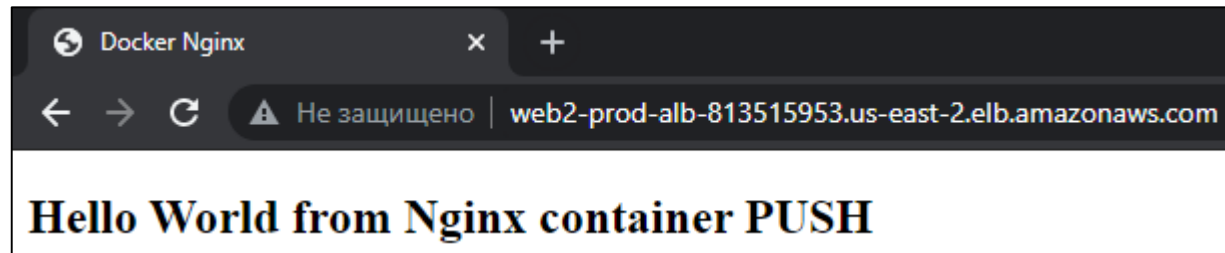
Conclusions

- To deploy a web application in Docker container by creating a cloud infrastructure and CI/CD pipeline on AWS, it is advisable to use Terraform

- To implement different environments, you can use .tfvars files with specified variable values and store the .tfstate files in different S3 buckets

Changes from previous demo

- The Terraform project structure is modified
- The DynamoDB Tables are added to lock Terraform state
- The code is cleared
- The new github repo is created
- The README file is modified
- The presentation is adapted

Thank You

Github repository:

https://github.com/mykshtompel/demo_test