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 according to the given requirements
- Tasks:
    - Examine the specified AWS infrastructure and CI/CD pipeline requirements
    - Development of the Terraform project structure
    - Implementation of the application deployment procedure using Terraform

Terraform Project Structure

- project
  - app
  - config
  - modules
  - provider.tf
  - terraform.tf
  - variables.tf
  - main.tf
  - outputs.tf

- app
  - web
    - index.html
    - Dockerfile
  - Makefile

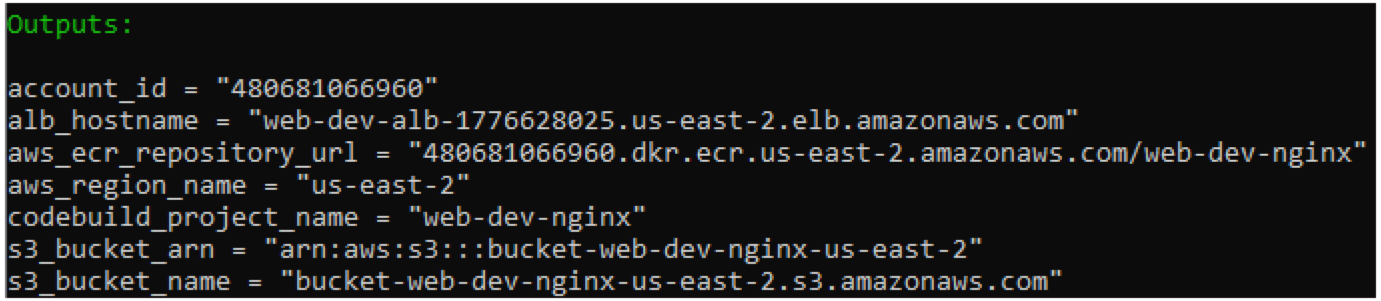
- config
  - dev.tfvars
  - prod.tfvars
  - secret.tfvars
  - buildspec.yml

- modules
  - s3
  - ecr
  - init-build
  - network
  - codebuild

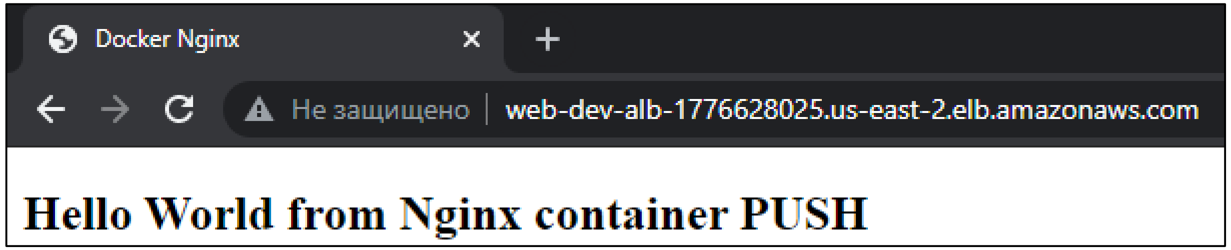
Implementation – Main Steps

- Creating an account on AWS
- Creating an user with required permissions using AWS IAM (manually)
- Installing Terraform and Atom
- Creating Terraform modules
- Creating “main” Terraform files
- Creating configuration files (Makefile, Dockerfile, buildspec.yml , *.tfvars,)
- Creating cloud infrastructure and CI/CD pipeline with the specified requirements on AWS
- Checking results

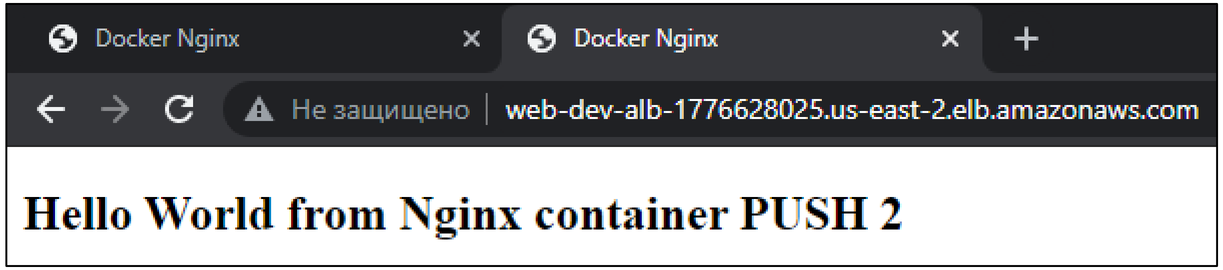
Demo - Results



```
Outputs:

account_id = "480681066960"
alb_hostname = "web-dev-alb-1776628025.us-east-2.elb.amazonaws.com"
aws_ecr_repository_url = "480681066960.dkr.ecr.us-east-2.amazonaws.com/web-dev-nginx"
aws_region_name = "us-east-2"
codebuild_project_name = "web-dev-nginx"
s3_bucket_arn = "arn:aws:s3:::bucket-web-dev-nginx-us-east-2"
s3_bucket_name = "bucket-web-dev-nginx-us-east-2.s3.amazonaws.com"
```

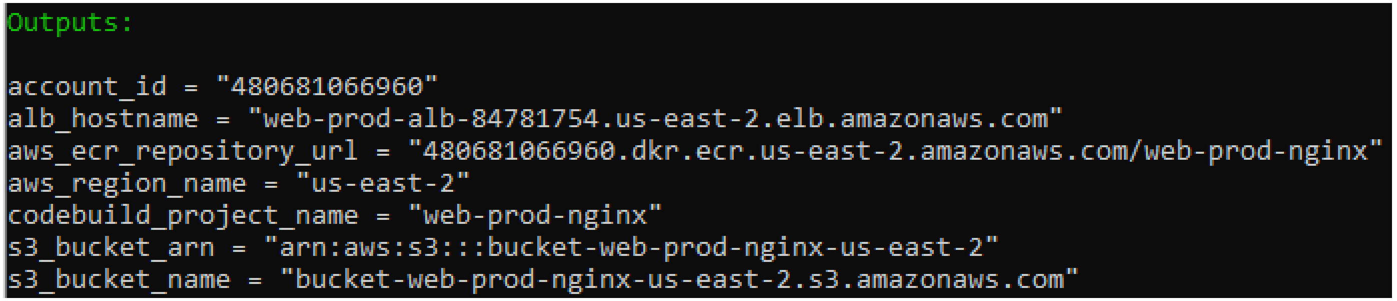


Hello World from Nginx container PUSH

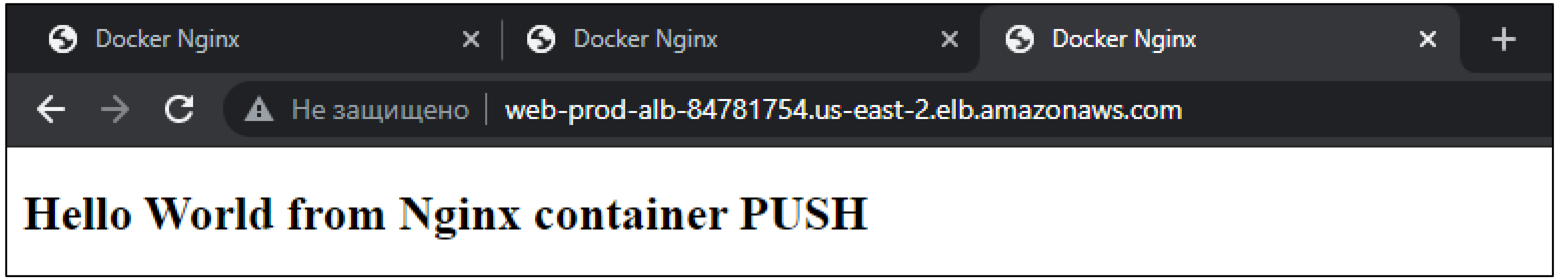


Hello World from Nginx container PUSH 2

Demo - Results



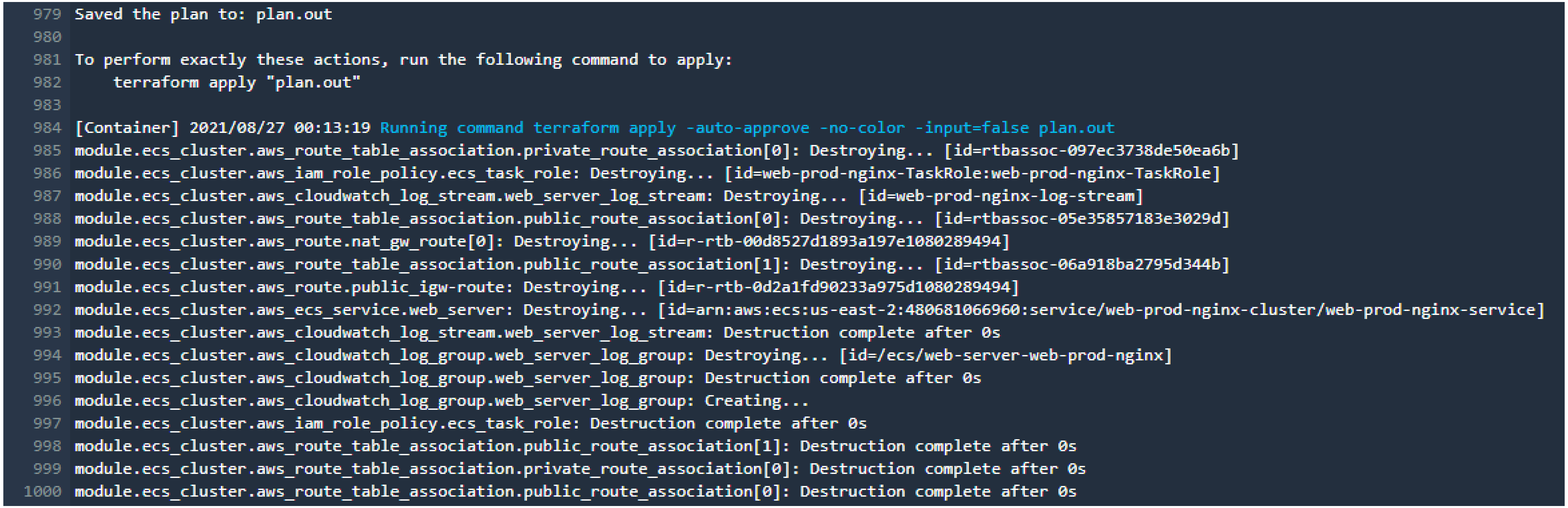
```
Outputs:

account_id = "480681066960"
alb_hostname = "web-prod-alb-84781754.us-east-2.elb.amazonaws.com"
aws_ecr_repository_url = "480681066960.dkr.ecr.us-east-2.amazonaws.com/web-prod-nginx"
aws_region_name = "us-east-2"
codebuild_project_name = "web-prod-nginx"
s3_bucket_arn = "arn:aws:s3:::bucket-web-prod-nginx-us-east-2"
s3_bucket_name = "bucket-web-prod-nginx-us-east-2.s3.amazonaws.com"
```



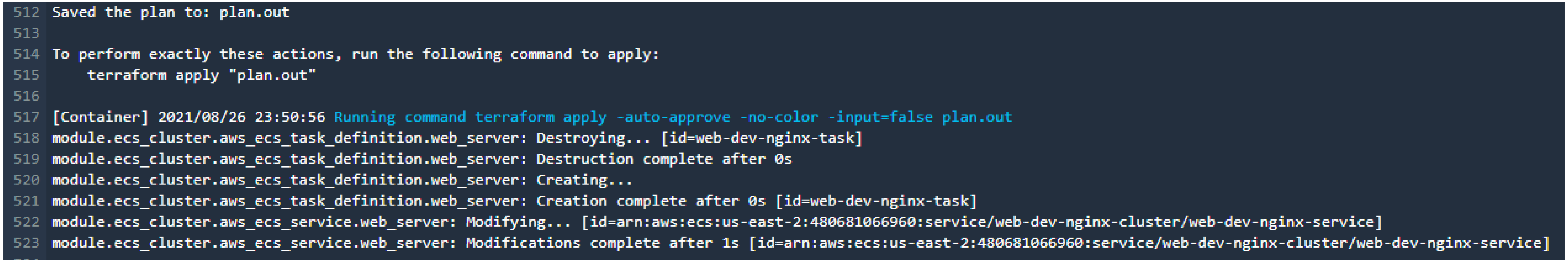
web-prod-alb-84781754.us-east-2.elb.amazonaws.com

Hello World from Nginx container PUSH

Demo - Results



```
Saved the plan to: plan.out

To perform exactly these actions, run the following command to apply:
    terraform apply "plan.out"

[Container] 2021/08/27 00:13:19 Running command terraform apply -auto-approve -no-color -input=false plan.out
module.ecs_cluster.aws_route_table_association.private_route_association[0]: Destroying... [id=rtbassoc-097ec3738de50ea6b]
module.ecs_cluster.aws_iam_role_policy.ecs_task_role: Destroying... [id=web-prod-nginx-TaskRole:web-prod-nginx-TaskRole]
module.ecs_cluster.aws_cloudwatch_log_stream.web_server_log_stream: Destroying... [id=web-prod-nginx-log-stream]
module.ecs_cluster.aws_route_table_association.public_route_association[0]: Destroying... [id=rtbassoc-05e35857183e3029d]
module.ecs_cluster.aws_route.nat_gw_route[0]: Destroying... [id=r-rtb-00d8527d1893a197e1080289494]
module.ecs_cluster.aws_route_table_association.public_route_association[1]: Destroying... [id=rtbassoc-06a918ba2795d344b]
module.ecs_cluster.aws_route.public_igw-route: Destroying... [id=r-rtb-0d2a1fd90233a975d1080289494]
module.ecs_cluster.aws_ecs_service.web_server: Destroying... [id=arn:aws:ecs:us-east-2:480681066960:service/web-prod-nginx-cluster/web-prod-nginx-service]
module.ecs_cluster.aws_cloudwatch_log_stream.web_server_log_stream: Destruction complete after 0s
module.ecs_cluster.aws_cloudwatch_log_group.web_server_log_group: Destroying... [id=/ecs/web-server-web-prod-nginx]
module.ecs_cluster.aws_cloudwatch_log_group.web_server_log_group: Destruction complete after 0s
module.ecs_cluster.aws_cloudwatch_log_group.web_server_log_group: Creating...
module.ecs_cluster.aws_iam_role_policy.ecs_task_role: Destruction complete after 0s
module.ecs_cluster.aws_route_table_association.public_route_association[1]: Destruction complete after 0s
module.ecs_cluster.aws_route_table_association.private_route_association[0]: Destruction complete after 0s
module.ecs_cluster.aws_route_table_association.public_route_association[0]: Destruction complete after 0s
```



```
Saved the plan to: plan.out

To perform exactly these actions, run the following command to apply:
    terraform apply "plan.out"

[Container] 2021/08/26 23:50:56 Running command terraform apply -auto-approve -no-color -input=false plan.out
module.ecs_cluster.aws_ecs_task_definition.web_server: Destroying... [id=web-dev-nginx-task]
module.ecs_cluster.aws_ecs_task_definition.web_server: Destruction complete after 0s
module.ecs_cluster.aws_ecs_task_definition.web_server: Creating...
module.ecs_cluster.aws_ecs_task_definition.web_server: Creation complete after 0s [id=web-dev-nginx-task]
module.ecs_cluster.aws_ecs_service.web_server: Modifying... [id=arn:aws:ecs:us-east-2:480681066960:service/web-dev-nginx-cluster/web-dev-nginx-service]
module.ecs_cluster.aws_ecs_service.web_server: Modifications complete after 1s [id=arn:aws:ecs:us-east-2:480681066960:service/web-dev-nginx-cluster/web-dev-nginx-service]
```

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

Thank You

Github repository:

https://github.com/mykshtompel/demo3