

# AWS ECS Fargate Automation using Terraform

AWS Fargate is a compute engine for Amazon ECS that allows you to run containers without having to manage servers or clusters. With AWS Fargate, you no longer have to provision, configure, and scale clusters of virtual machines to run containers. This removes the need to choose server types, decide when to scale your clusters or optimize cluster packing. AWS Fargate removes the need for you to interact with or think about servers or clusters. Fargate lets you focus on designing and building your applications instead of managing the infrastructure that runs them.

This project does the following

1. Creates and configure ECS cluster.
2. Creates an AWS ECS fargate service.
3. Creates an AWS ECS fargate task definition.
4. A VPC with a private and a public subnet.
5. An ECS cluster in the private subnet.
6. Runs Docker container with stuartshay/microservice-api:2.2.1-build image.
7. Maps Docker container port 5000 with host port 5000
8. An ALB load-balancing requests to the ECS cluster.
9. Creates 2 public subnets in different AZs required by the load balancer.

This project consists of terraform configuration script which will be launch from Jenkins.

Jenkins: <https://jenkins.navigatorglass.com/job/Terraform%20AWS%20Fargate/>

Github ( terra scripts ):

<https://github.com/stuartshay/MicroService/tree/develop/terraform>

## 1) Fargate Cluster plan using terraform

It generates a Terraform plan which will use to check what are all the resources create and their configuration when you apply to build, choose option Plan\_only and then click build.

### Pipeline Microservice-api-fargate

This build requires parameters:

Plan\_Only



Uncheck to allow Terraform Apply

Destroy\_the\_Environment ☐

Select the Checkbox only when you need to destroy the stack

Build

## Expected Plan Output

```
20:44:14 [0m
20:44:14 [0m
20:44:14 [0m[1mPlan:[0m 26 to add, 0 to change, 0 to destroy.[0m
20:44:14
20:44:14 -----
20:44:14
20:44:14 Note: You didn't specify an "-out" parameter to save this plan, so Terraform
20:44:14 can't guarantee that exactly these actions will be performed if
20:44:14 "terraform apply" is subsequently run.
20:44:14
```

## 2) Create the AWS ECS Fargate environment

For creating resources click on Build only (make both the other option remain to unchecked)  
This will create AWS ECS Cluster, custom VPC, AWS ECS service, AWS ECS task definition, containers, load balancer and all the configure resources

### Pipeline Microservice-api-fargate

This build requires parameters:

Plan\_Only

☐

Uncheck to allow Terraform Apply

Destroy\_the\_Environment

☐

Select the Checkbox only when you need to destroy the stack

Build

Then it will give the following message

18:02:26 WARNING: Applying Terraform updates can result in unrecoverable destruction.  
Click "Proceed" to confirm Terraform update plan was reviewed and to authorize the  
updates to be applied, else click "Abort".

18:02:26 Proceed or Abort

Click on **Proceed**

Expected Apply Output

```

20:48:22 platform_version: "" => "<computed>"
20:48:22 scheduling_strategy: "" => "REPLICA"
20:48:22 task_definition: "" => "arn:aws:ecs:us-east-2:816939196156:task-definition/coredata-test"
20:48:22 [0m[1maws_ecs_service.main: Creation complete after 1s (ID: arn:aws:ecs:us-east-2:816939196156:service/coredata-test)
20:48:22 [0m[1m[32m
20:48:22 Apply complete! Resources: 26 added, 0 changed, 0 destroyed.[0m
20:48:22 [0m[1m[32m
20:48:22 Outputs:
20:48:22
20:48:22 alb_hostname = You can access the swagger at coredata-test-alb-2079124871.us-east-2.elb.amazonaws.com:5000/swagger
[Pipeline] }
[Pipeline] // wrap
[Pipeline]

```

Wait 4 - 5 minutes for container becoming ready to use  
Then access container by using alb\_hostname output URL

```

20:48:22 alb_hostname = You can access the swagger at coredata-test-alb-2079124871.us-east-2.elb.amazonaws.com:5000/swagger

```

### 3) Destroy all the resources created

For destroying all the running resources, click on Destroy\_the\_Environment and then click Build

## Pipeline Microservice-api-fargate

This build requires parameters:

Plan\_Only

☐

Uncheck to allow Terraform Apply

Destroy\_the\_Environment ☒

Select the Checkbox only when you need to destroy the stack

Build

Then it will give the Following message

18:02:26 WARNING: Applying Terraform updates can result in unrecoverable destruction.  
Click "Proceed" to confirm Terraform update plan was reviewed and to authorize the updates to be applied, else click "Abort".  
18:02:26 Proceed or Abort

Click on **Proceed**

## Expected Output

```
14:58:22 [0m[1maws_vpc.main: Destruction complete after 1s[0m[0m
14:58:22 [0m[1m[32m
14:58:22 Destroy complete! Resources: 26 destroyed.[0m
[Pipeline] }
[Pipeline] // wrap
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // timestamps
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## 4) To Update Docker Image to Existing Environment

For updating the Docker image on the existing environment.

Enter and save the required Docker image value in “**terraform.tfvars**” file.

### **terraform.tfvars**

<https://github.com/stuartshay/MicroService/blob/develop/terraform/terraform.tfvars>

```
# Docker image
```

```
docker_image = "stuartshay/microservice-api:2.2.1-build"
```

After updating the docker image you can perform the steps in section 2 mentioned above. it will update the task definition and deploy the latest docker image.