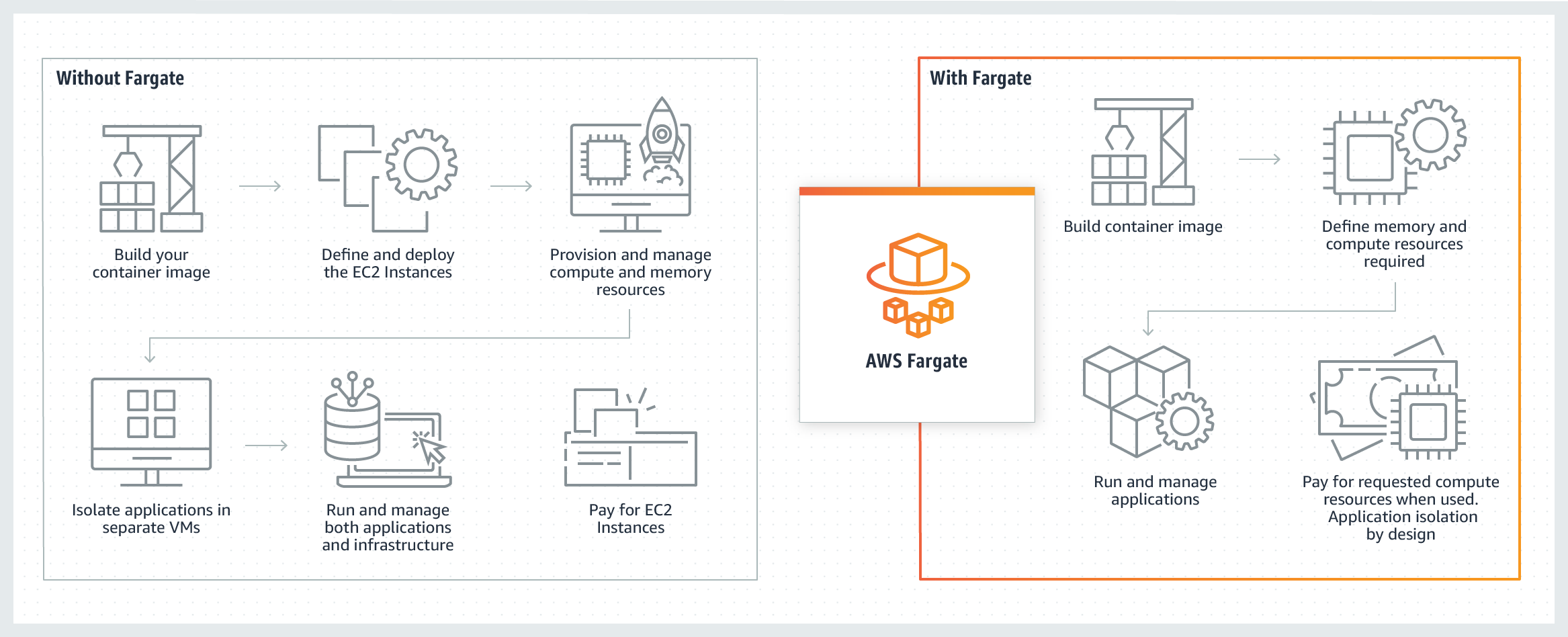
AWS Fargate is a serverless, pay-as-you-go compute engine that lets you focus on building applications without managing servers.

AWS Fargate is compatible with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS) .



**What are the limitations and challenges of Fargate ?**

By giving control of infrastructure to Fargate, users are limited in what they can customize and configure. Some of the limitations of using ECS with the Fargate launch type include:

* No option to choose specific instance types
* Running workloads like Daemon sets or privileged pods are restricted
* Many ECS task definition parameters are not valid with Fargate ([see here for a full list of unsupported task definitions)](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/AWS_Fargate.html)
* No support for GPU
* Classic load balancers are not supported, only Application and Network load balancers can be used to distribute workloads
* No support for EBS.

**Note :** Creation of the Aws fargate is not recommended for continuous or long run cluster the bill will be high.

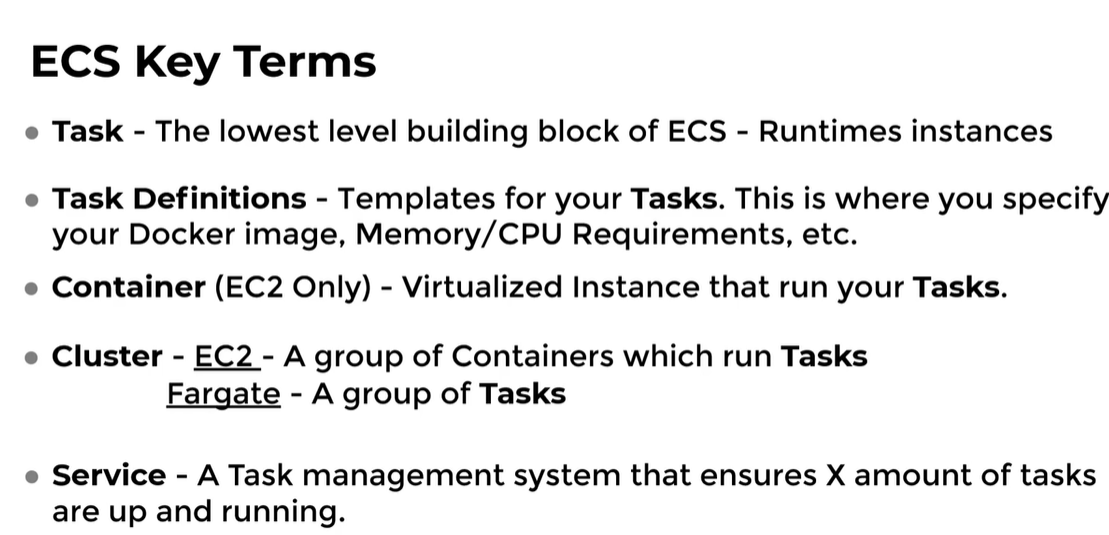
On note Aws Ec2 instance with Ecs is Recommended for long run application .

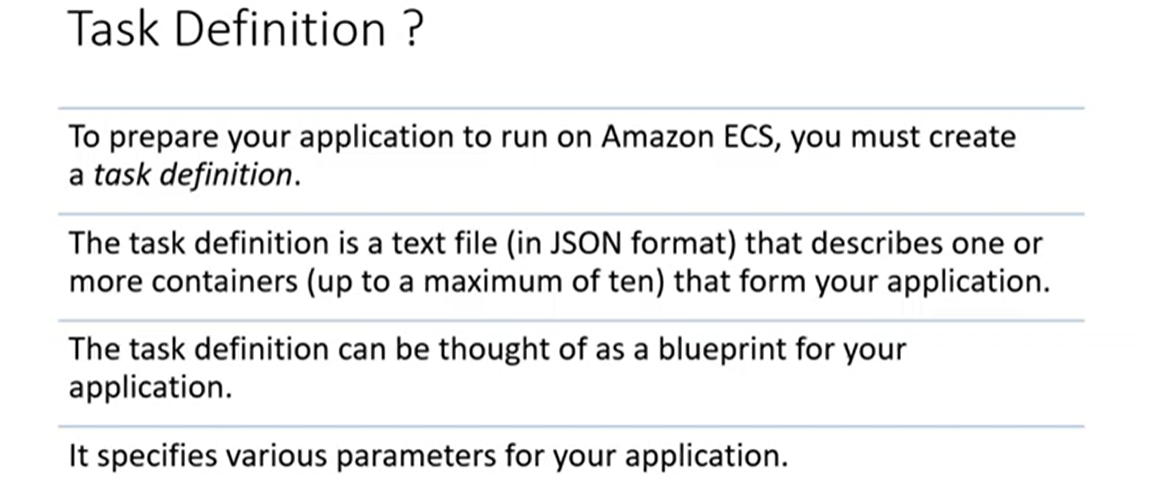
**ECS with Aws Fargate**

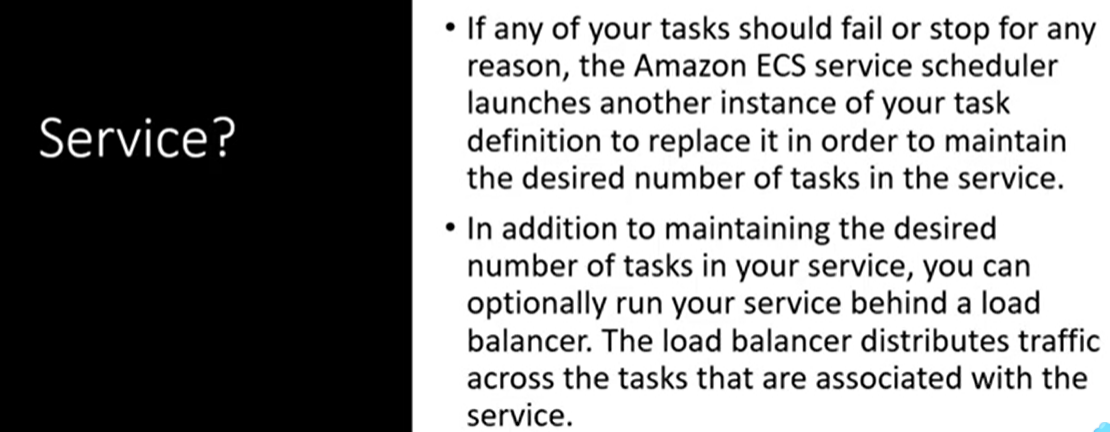
Running ECS with Fargate however, eliminates the need to manually provision, scale and manage compute instances. Users create a cluster, add workloads to it and specify resource requirements (CPU and memory), and when ECS containers are deployed, Fargate will launch, run and manage pre-configured servers that meet container requirements. These time-saving benefits eliminate the operational burden of managing compute, but the trade-off is limited features, less control and potentially higher costs.

By referring the below doc

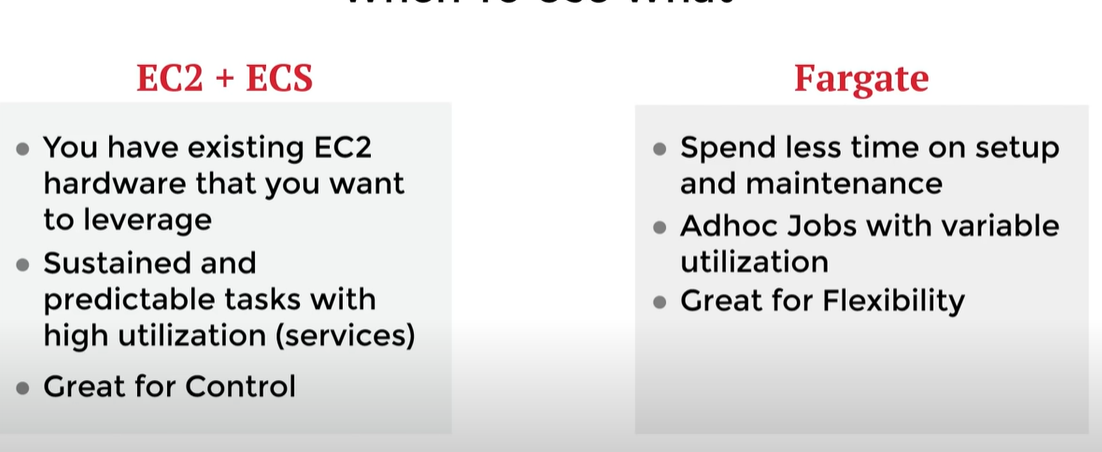
<https://spot.io/blog/fargate-vs-ecs-comparing-amazons-container-management-services/>.





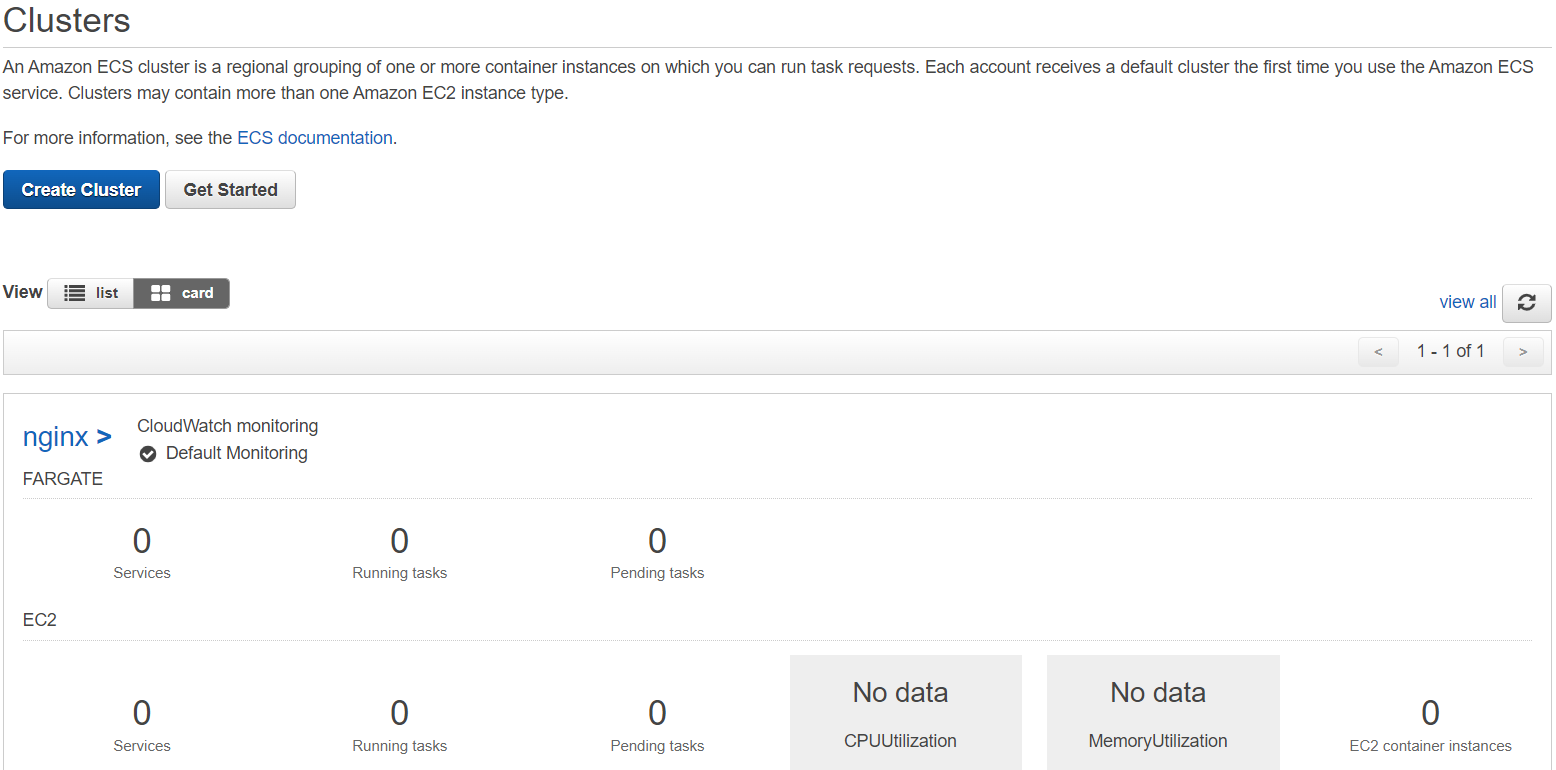


Difference between Ecs with Ec2 Instance & Ecs with Fargate.

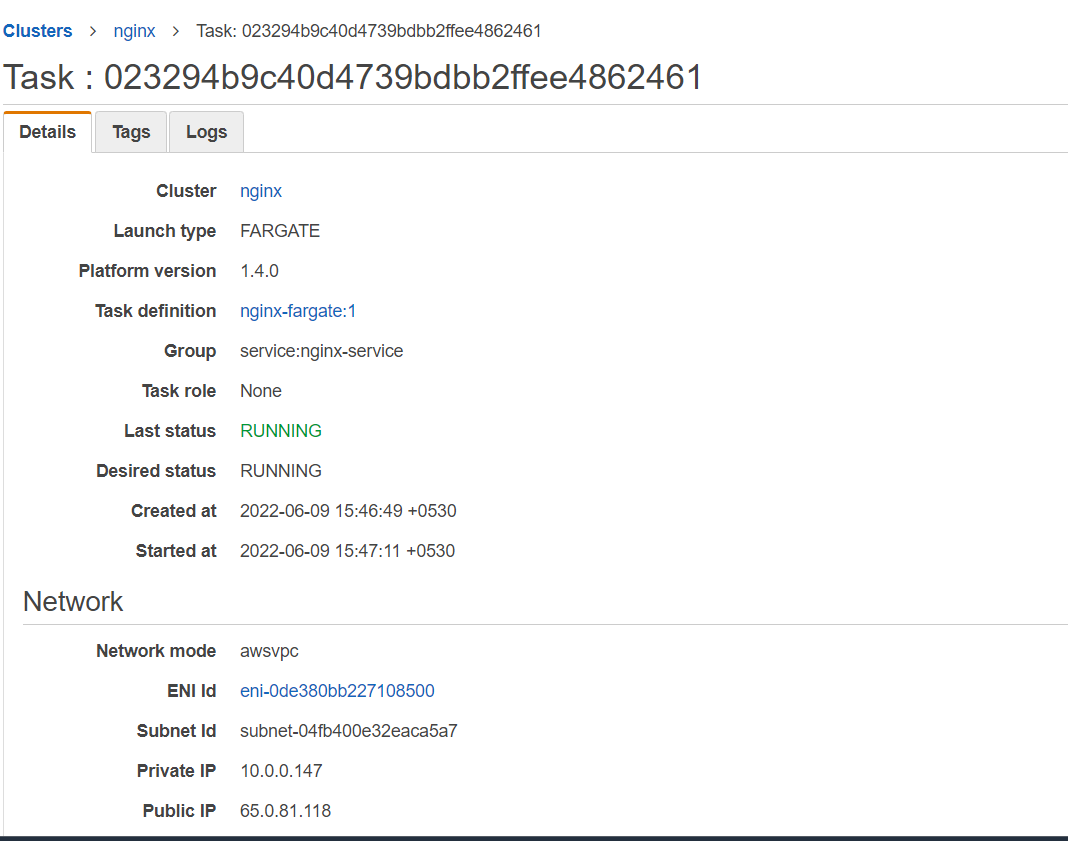


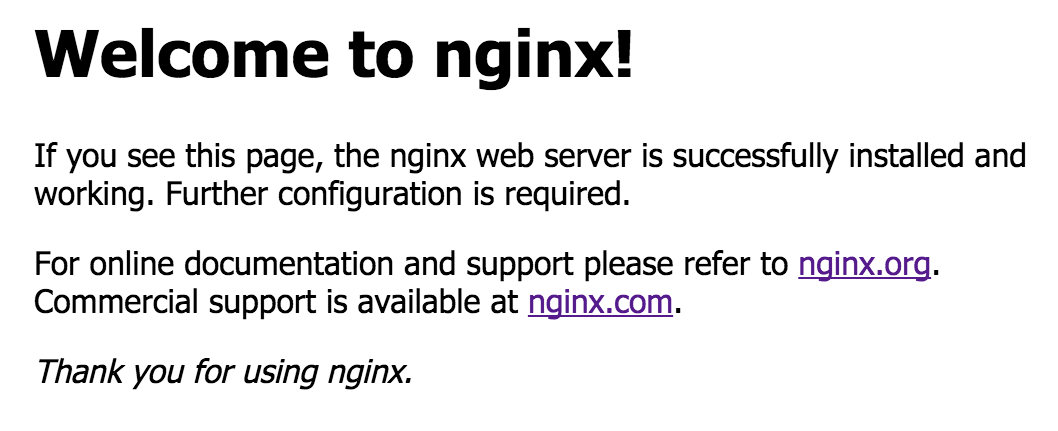
**Creation of the Aws Fargate with Ecs Manually**

Below image is the creation of the cluster of nginx



Creation of the Task Defination of the Nginx Application .



By taken the public Ip of the cluster view in the web browser we can view the web page like this 

By referring the doc creation of cluster

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/AWS_Fargate.html>

**Terraform for AWS ECS**

Creation of the Terraform script for Aws fargate with ecs.

Creation of Vpc with private and public subnets.

Creation of Multiple Running container instances.

A load Balancer Distributing traffic between the containers

Auto scaling of your resources.

Create the following

* Provider.tf we have created the provider information like cloud aws
* vpc.tf creation of the vpc,subnets,nat gateways,routetables
* Security.tf Creation of the security group and give inbound and outboundrules
* Ecs.tf creation of the task defination,service files,connect load balancers and auto scaling create a autoscaling group
* Alb.tf create a application load balancer
* Auto\_scaling.tf create a autoscaling group
* Variables.tf create a variable files
* Outputs.tf , Iam.t

These are the files creation in the terraform script

By Using doc creation of terraform script

<https://gmusumeci.medium.com/how-to-deploy-aws-ecs-fargate-containers-step-by-step-using-terraform-545eeac743be>.

<https://medium.com/@bradford_hamilton/deploying-containers-on-amazons-ecs-using-fargate-and-terraform-part-2-2e6f6a3a957f>.

By using the creation of the git repo for creation of the terraform code

<https://github.com/jayachandrareddym/terraform/tree/master/ecs>

