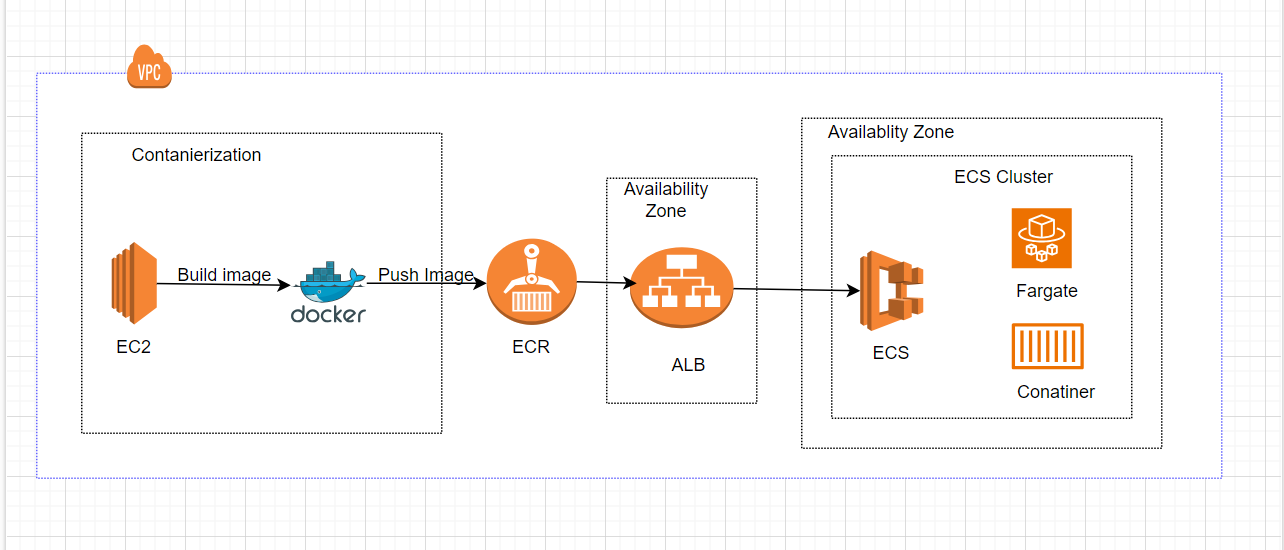
**Deploy a Web Application to Amazon ECS with EC2 | Docker | ECR |Fargate| Load balancer.**

**Design Diagram:-**



**Brief Description:-**

In this project I am deploying Container web application to Amazon ECS (Elastic Container Service) with EC2 instances, Docker, ECR (Elastic Container Registry), and a load balancer which involves several steps.

We will start by launching an EC2 instance and installing docker on it to build a docker image which will be pushed to ECR afterwards. Following that, by using AWS ECS (Elastic Container services) we will create an ECS cluster, ECS Service, and Task Definition, which will describe how the container should execute.

Finally we will using an application load balancer who will be responsible for evenly distributing traffic among the ECS cluster’s instances. This includes instructions on configuring the load balancer to employ the ECS task definition and implementing health checks to guarantee that traffic exclusively reaches functional instances.

**Steps:-**

1. **Launch EC2 instances**

Launched EC2 using Ubuntu 22.04 base image named as (Docker-instance) with security group Ec2-instance-sg and allowed SSH and http.

1. **Install Docker**

To install docker SSH into EC2 instance (“Docker-instance”) by using git bash or AWS SSH client and run the following commands:-

* *chmod 400 Project-key.pem*
* *ssh -i Project-key.pem"* [*ubuntu@ec2-18-206-206-130.us-east-2.compute.amazonaws.com*](mailto:ubuntu@ec2-18-206-206-130.us-east-2.compute.amazonaws.com)
* *sudo apt upgrade*
* *sudo apt install docker.io*
* *docker –version*
* *systemctl start docker*
* *systemctl status docker*
* *systemctl enable docker*

**Build a Docker file**

* *FROM ubuntu:22.04*
* *RUN apt update*
* *Run apt install -y apache2*
* *COPY index.html /var/www/html/*
* *EXPOSE 80*
* *CMD ["apache2ctl","-D","FOREGROUND"]*

**Build Docker image by using following commands**

* *docker build –t project-image .*
* *docker images*
* *docker run -d -p 80:80 --name (Container Name)project-container (Image name)project image*
* *docker ps to check if the container has created*

1. **Create a Repository using Elastic Container Registry (ECR) in AWS | Create Role | Attach |Permission**

* Created “projectecr-repo” as private repo.
* Created IAM role named as “ecr-role1”
* Attached “Administrator access” permissions to the ecr-role1.

**Create Access Key in AWS IAM and configure your credentials. These two keys “access key” and “secret access key” will be configuring our user profile. Check if aws-cli has installed or not if not installed aws-cli first using following commands**

* *sudo snap install aws-cli --classic*
* *aws --version*
* *aws-cli/1.16.266 Python/3.5.2 Linux/5.0.0-37-generic botocore/1.13.2*
* *in the terminal run following commands to configure access*
* *aws configure*
* *enter the access key*
* *enter the secret access*
* *default region us-east-1*

**Login to Elastic Container Registry from AWS CLI, Tag the Image and Push your Image to Amazon ECR.**

* *-Logged into our ECR “projectecr-repo” and will use the following commands to push our docker image to AWS ECR*
* *-aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 041880346382.dkr.ecr.us-east-1.amazonaws.com*
* *-docker tag project-image:latest 041880346382.dkr.ecr.us-east-1.amazonaws.com/projectecr-repo:latest*
* *-docker push 041880346382.dkr.ecr.us-east-1.amazonaws.com/projectecr-repo:latest*

1. **AWS(Application Load Balancer)**

* Created Loadbalancer named as “project-internet-facing-lb” with three availability zone to accomplish high availability
* Created target group named as “project-targetgroup”

1. **Create Task Definition, Cluster and Service in Amazon ECS | Using Load Balancer in ECS & Validation**

-Created Task definition by using launch type FARGATE.

-Assigned execution “AmazonECSTaskExecutionRolePolicy” role as required for this task

**Create an ECS cluster**

-Created ECS cluster named as “ECS-Project-Cluster1”

**Create ECS service**

-Created ECS-Project-Service to run two tasks and added to the ECS-Project-Cluster1

1. **Testing the Deployed Container Web App**

