**Container Project**

An organization wants a simpler, more streamlined way to deploy and manage containerized applications in AWS. However, the traditional ECS console and workflows can be complex, error-prone, and require deep understanding of multiple steps (defining clusters, tasks, services, load balancers). This complexity can slow down development teams and increase operational risk.

**Goal:** In this **project,** we are going to learn how to use the elastic container service (ECS). We will create an ECS cluster, a service and a task that uses NGINX containers. That will let us run a simple web page.

**Terminologies and concepts**

**ECS cluster**: This is a logical grouping of tasks or services that run on infrastructure that’s registered to that cluster. Infrastructure may be EC2 instances in AWS and can be registered to the clusters with the aid of Elastic Container Service (ECS) agents running on Docker.

A close-up of a computer chip

AI-generated content may be incorrect.

But how does this infrastructure get there?

Launch Types

1. **EC2**: You have to register and manage the EC2 instances yourself
2. **Fargate**: Serverless; managed and handled by AWS

These EC2 instances are containers services

A close-up of a label

AI-generated content may be incorrect.

**What are tasks or services**: A task is a running container (e.g. NGINX), whose settings are defined in a Task Definition. You can also think of task as a blueprint of an application. A task is an instantiation of Task Definition

Example of Task Definition

A screenshot of a computer

AI-generated content may be incorrect.

A service is a Long-running tasks with the same task definition. There could be one container running or multiple containers running the same task definition.

It now looks like this:

A close-up of a computer screen

AI-generated content may be incorrect.

Showing one task per instance, but there could be more than one container in one task. The ECS scheduler is responsible for placing those tasks within the cluster and there are different schedulers available also.

Now, let’s go to ECS service in AWS management console.

At the left panel, click on clusters

A screenshot of a computer

AI-generated content may be incorrect.

Then Create Cluster

A screenshot of a computer

AI-generated content may be incorrect.

Configure the following:

**Cluster name**: my-cluster (You can use your own name)

**Infrastructure**: Fargate only

Leave the others as default and click create

A screenshot of a computer

AI-generated content may be incorrect.

Click into the cluster we just created.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the infrastructure tab

A screenshot of a computer

AI-generated content may be incorrect.

You will see the two capacity providers, Fargate and Fargate Spot

A screenshot of a computer

AI-generated content may be incorrect.

We now have the cluster that we just created, but the infrastructure, the container instance (Fargate) running on it.

Next, we are going to use the NGINX container in the AWS Elastic Container Registry (ECR), the public gallery. NGINX will give us a web server so we can create a simple web page. The service we create will then and go grab that container and run it as a task. Then a user will be able to use the public IP address of the container in that task and see the web page

A diagram of a service

AI-generated content may be incorrect.

Back to ECS, on the left pane, click on task definition

A screenshot of a computer

AI-generated content may be incorrect.

Click on Create new task definition

A screenshot of a computer

AI-generated content may be incorrect.

On the task definition configuration, enter nginx-task-definition

Now, Open a new tab and go to gallery.ecr.aws. In the search bar enter nginx and press enter. On the left pane, click verified account to filter

A screenshot of a computer

AI-generated content may be incorrect.

Click on the on the verified first one

A screenshot of a computer

AI-generated content may be incorrect.

Copy the image you want and go back to ECS.

A screenshot of a computer

AI-generated content may be incorrect.

Enter nginx as the name of the container and paste the image url you copied from the ECR website into the url of container-1

A screenshot of a computer

AI-generated content may be incorrect.

Configure the following:

CPU: 0.5

Memory: 1GB

A screenshot of a computer

AI-generated content may be incorrect.

Screw to the bottom, leave everything as default and click create

Next, go back to clusters.

Click the Launch at the upper right hand of the page and click service

A screenshot of a computer

AI-generated content may be incorrect.

Leave the default VPC and subnets. Click Create security group.

Configure the security to allow inbound HTTP rule traffic and from anywhere.

A screenshot of a computer

AI-generated content may be incorrect.

Screw to the bottom and click create to launch the service.

Next is to copy the IP4 address of the service and paste in a new browser

A screenshot of a computer

AI-generated content may be incorrect.

You can see the web page

A screenshot of a computer

AI-generated content may be incorrect.

**Project Complete**