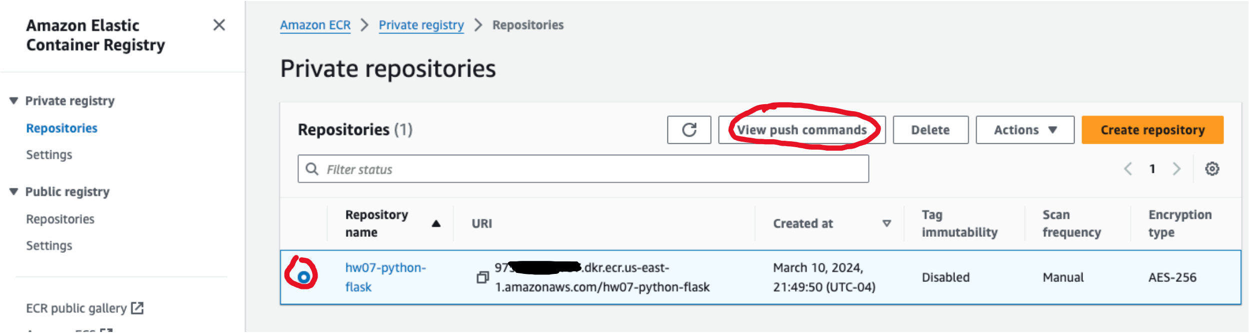
# ECR / ECS Tutorial

This tutorial assumes you have built an app (e.g. Python Flask) and created a Docker image that you want to push to ECR and access via ECS.

## AWS Elastic Container Repository (ECR)

An image repository is used to store and access container images.

- Push your docker image to ECR
  - In the AWS Management Console, select ECR and click "Create Repository" if you have not already created an ECR repo. Take most of the defaults and provide the repository name like "hw07-flask-app".
  - Once created, select the radio button for the repo and click the button "View push commands." Execute each of these CLI instructions once by one. (Note: prior to executing these commands, make sure you have an active Learner Lab session and have updated your ~/.aws/credentials file.) These CLI instructions will:
    - Retrieve an authentication token and authenticate your Docker client to your registry.
    - Tag image with required ECR tags
    - Push the image to your ECR repo



| | Repository name ▲ | URI | Created at ▽ | Tag immutability | Scan frequency | Encryption type |
|---|---|---|---|---|---|---|
| ● | hw07-python-flask | 97[redacted].dkr.ecr.us-east-1.amazonaws.com/hw07-python-flask | March 10, 2024, 21:49:50 (UTC-04) | Disabled | Manual | AES-256 |

## AWS Elastic Container Service (ECS)

ECS is a full-managed container orchestration service that helps you deploy, manage, and scale container applications.

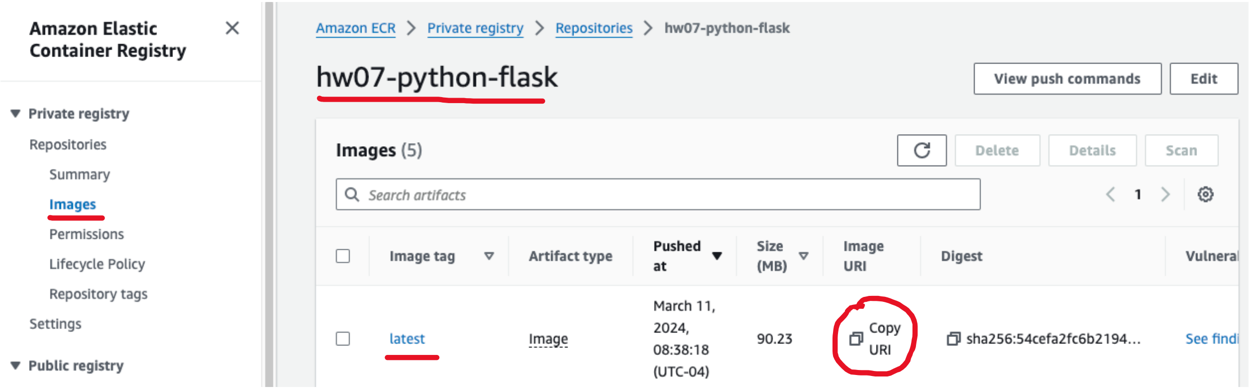
Summary: You create several things in ECS:

- "Task Definition" - your blueprint for how to run your container.
- "Cluster" - your environment to run the container. Think of this as "dev", "prod", etc.
- "Task" - running instance of your container on/in your cluster. You can EITHER create the Task directly, OR you can create a "Service" that will launch tasks automatically. The

advantage of the Service is that any new images pushed to ECR will automatically get pulled and deployed to new Tasks.

Task Definition

- Create a new "Task Definition".
  - Task definition family - name it something meaningful
  - Infrastructure requirements - keep most of the defaults but be sure to select "LabRole" both as the task role and the task execution role.
  - Container - 1
    - Container - name it like "hw07-python-flask"
    - Image URI - get the uri from the repo you deployed on ECS that contains the Docker image you pushed.



| | Image tag | Artifact type | Pushed at | Size (MB) | Image URI | Digest | Vulnera |
|---|---|---|---|---|---|---|---|
| □ | latest | Image | March 11, 2024, 08:38:18 (UTC-04) | 90.23 | Copy URI | sha256:54cefa2fc6b2194... | See find |

  - Add a port mapping for port 5000 (Protocol is TCP, App protocol is HTTP, Port name can be something like "flask")
  - Optional: Select "Read Only" in the Read only root file system check box
  - Resource allocation limits - be sure to type in values for these! e.g. CPU = 1, GPU = 1, Memory hard limit = 3, memory soft limit = 1

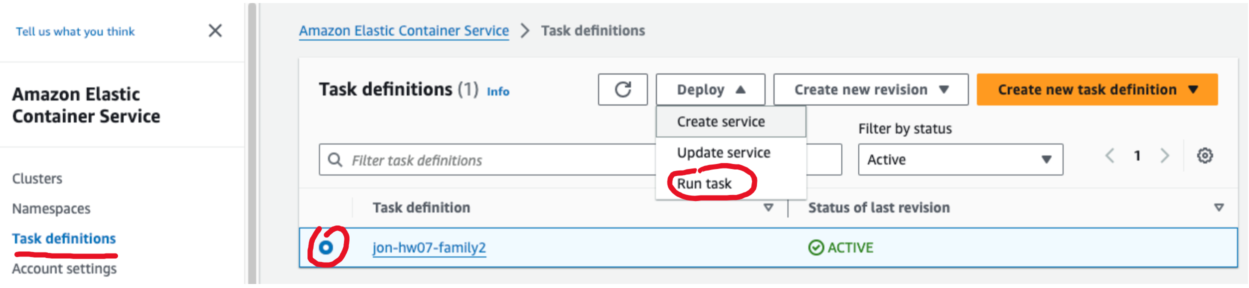
Cluster

- Go to ECS and select "Create Cluster". Name it like "hw07-cluster..." and select "AWS Fargate (serverless)". (Fargate eliminates the need for creating EC2 auto-scaling groups to manage your ECS cluster.)

Lastly, create either a Task or a Service.

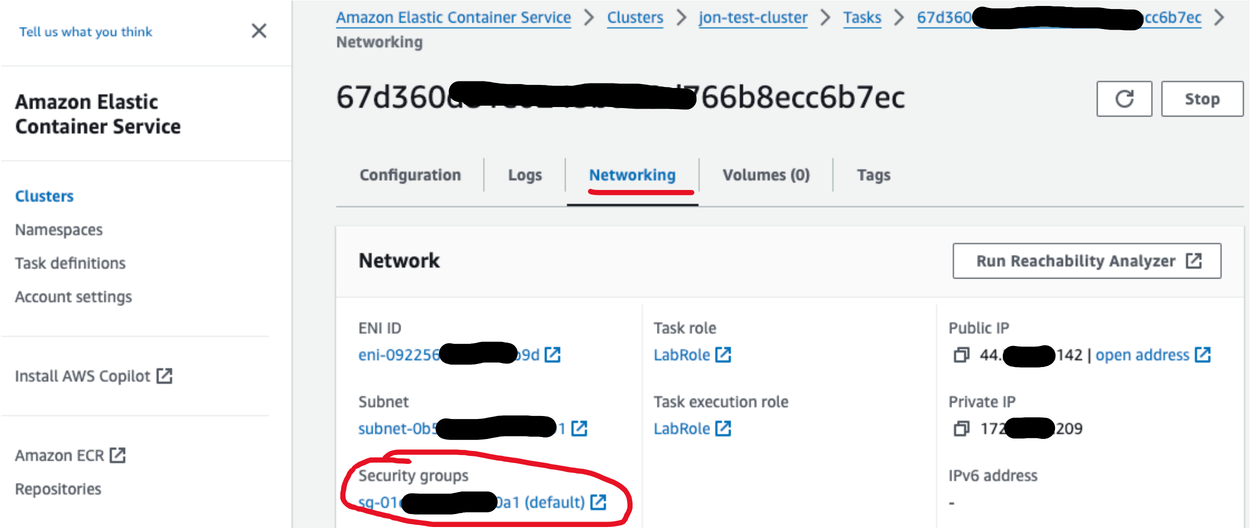
Task

- From the left-hand menu of ECS, select "Task Definitions". Click the radio button for your task definition and select "Run task" from the Deploy menu. By launching your Task this way, all the settings are defaulted for you.



| Task definition | Status of last revision |
|---|---|
| jon-hw07-family2 | ACTIVE |

- When your task shows as "Running" in the status column, click on it to view the settings. Select the "Networking" tab.
  - This is very important! You need to allow inbound traffic on port 5000 in your Security Group. To do that, click on the security group link within your Task.


Tell us what you think
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Install AWS Copilot
Amazon ECR
Repositories
Amazon Elastic Container Service > Clusters > jon-test-cluster > Tasks > 67d360[redacted]cc6b7ec >
Networking
67d360[redacted]66b8ecc6b7ec
Stop
Configuration
Logs
Networking
Volumes (0)
Tags
Network
Run Reachability Analyzer

ENI ID: eni-092256[redacted]9d

Subnet: subnet-0b5[redacted]1

Security groups: sg-01[redacted]a1 (default)

Task role: LabRole

Task execution role: LabRole

Public IP: 44.[redacted]142 | open address

Private IP: 172[redacted]209

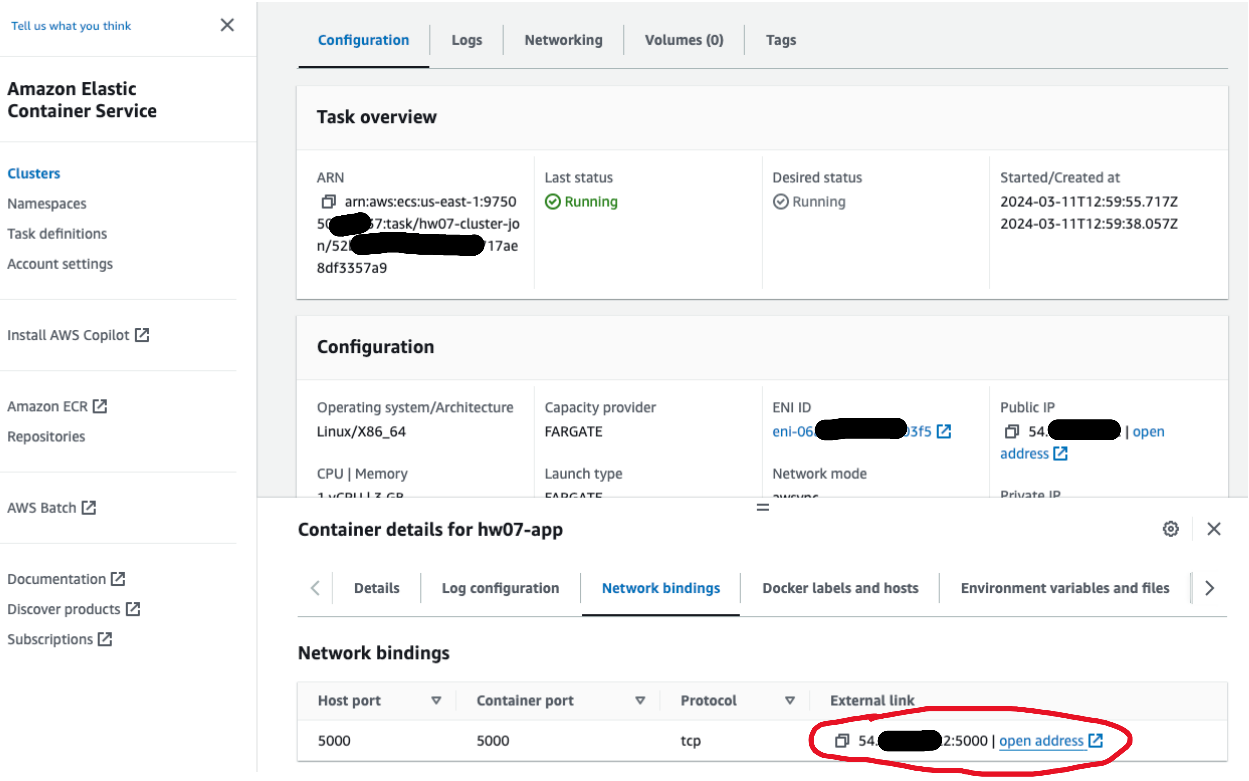
IPv6 address: -

  - Edit the Inbound Rules. Add Rule. Select "Custom TCP" and enter "5000" as the port. Set the source to "Anywhere-IPv4" which will display as "0.0.0.0/0". If you are having trouble connecting to your app in the browser, you probably forgot this step.
- Now click on the "Configuration" tab and use the Public IP Address to test that your application is running and responsive.

Service

- From the left-hand menu of ECS, select "Task Definitions". Click the radio button for your task definition and select "Create Task" from the Deploy menu.
- Select your cluster
- Select a Compute option. e.g. "Launch Type".
- Specify a Service Name like "hw07-service"
- On the "Cluster" left-hand menu, select the "Services" tab and verify that your service is running. Now select the "Tasks" tab and verify that your Task has been created by the service and the task is running.
- Click on the Task name and you will see a screen like the one below. Click on the "Network bindings" tab and copy the External link IP Address. Paste that URL into a

browser and test your application. (Note: make sure your service is fully deployed before testing the URL.)



Task overview

ARN: arn:aws:ecs:us-east-1:9750 5[redacted]7:task/hw07-cluster-jo n/52[redacted]17ae 8df3357a9

Last status: Running

Desired status: Running

Started/Created at: 2024-03-11T12:59:55.717Z 2024-03-11T12:59:38.057Z

Configuration

Operating system/Architecture: Linux/X86_64

Capacity provider: FARGATE

ENI ID: eni-06[redacted]3f5

Public IP: 54.[redacted] | open address

CPU | Memory: 1 vCPU | 3 GB

Launch type: FARGATE

Network mode: awsvpc

Private IP

Container details for hw07-app

Network bindings

| Host port | Container port | Protocol | External link |
|---|---|---|---|
| 5000 | 5000 | tcp | 54.[redacted]2:5000 \| open address |

## Costs

ECR is very cheap, so don't worry about that. Storage is $0.10 per GB / month

But be careful! You will be charged per hour for ECS regardless if you are logged into your Learner Lab account. For a lab like the one above, that could be about $1 - 2 per day of your total $100 credits for the semester. After your assignment is graded, you should stop your ECS containers.

To stop your ECS running tasks during development time, go into your cluster and select the checkbox next to the service and press "Update". Set the "desired tasks" to 0 and press "Update." Now go back to your lists of Clusters and make sure in the Tasks column there are no running tasks. If you have no running tasks, you should not incur AWS fees for ECS.

Please Note: if you are handing in an assignment for ECS, you should leave your tasks RUNNING. Your TA will be instructed to stop your running tasks once they finish grading your assignment.

This tutorial was based on this tutorial.