

## Deploy Containers on ECS

Start at 9:05

→ Robust deployment

→ ML model → API → Docker/Container  
Flask

→ Currently → installing and running API server on a local machine

↓  
Can we move it to a online version where the 2nd person don't have to install anything

→ Cloud Computing →

→ AWS →

→ GCP

→ AZURE

→ What type of services cloud providers provide →  
→ Storage → Compute → Network Capability

→ Key benefits of Cloud →

- 99x99% uptime
- Security
- Monitoring
- Scaling

→

⇒ Api application → Docker → how to run it on AWS  
(flask)

⇒ Cloud deployment on AWS →

- 1) ECR → Elastic Container Registry
- 2) ECS → Elastic Container Service
- 3) IAM →
- 4) Fogate →
- 5) EC2 → Elastic Compute Cloud

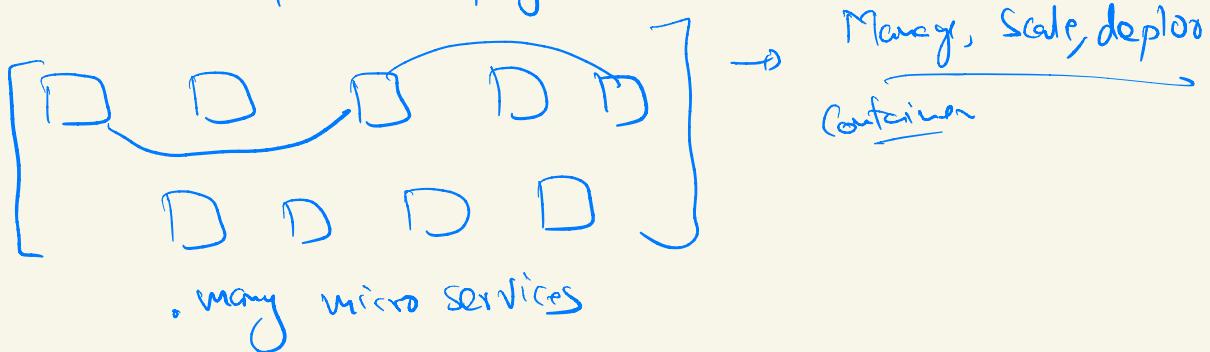
⇒ Dockerhub → was used to store / share container Image with others

①

ECR → AWS version dockerhub

② ECS → Run the containers →

- Help in deployment



. many micro services

Some other tools like → Kubernetes

ECS

Mesos

Apache Nomad

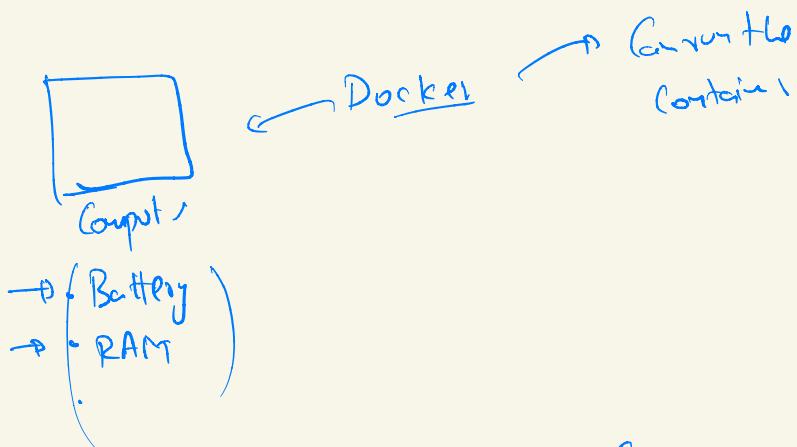
Docker Swarm

⇒ ECS → is an orchestration Service

↳ Where does it give a Container → ??

VMS → EC2 →

user / you have to ensure this up, have sufficient resources, is available to ECS



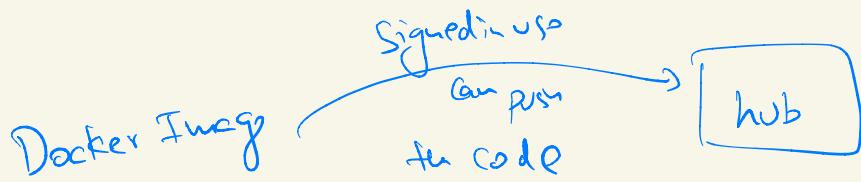
→ Amazon → Fargate → Container Orchestration  
↳ it is a compute optimised for running the Container.

Managed by Amazon → . 16 GiB  
↳ 32GiB ← auto 1 TB  
2 TB swap



Fargate which is Cheap

⇒



ECR

→ IAM → Identity & access management  
↳ which user will access to what

⇒ •

IAM → setup a secret key for your user → for connecting through CLI

↳ push a Docker Image → ECR

↓  
ECS

ECS → link → fargate → Run the Container Service  
→ how to open the access to anyone.

Create IAM tokens  
Login to local machine  
Send Image to ECR  
ECS Setup with ECR  
Running the service

IAM Token Generation:

1. Go to IAM > User > Username > Security credentials > Access keys
2. Choose cli and click next to get your tokens

Login to local machine:

1. "aws configure" in terminal
2. Provide the required details

Create a repo in ECR

1. Give repo some name. Follow the push commands listed inside the repo to send local image to ECR

If you are using Mac OS please ensure to also add `--platform=linux/amd64` in the build image step

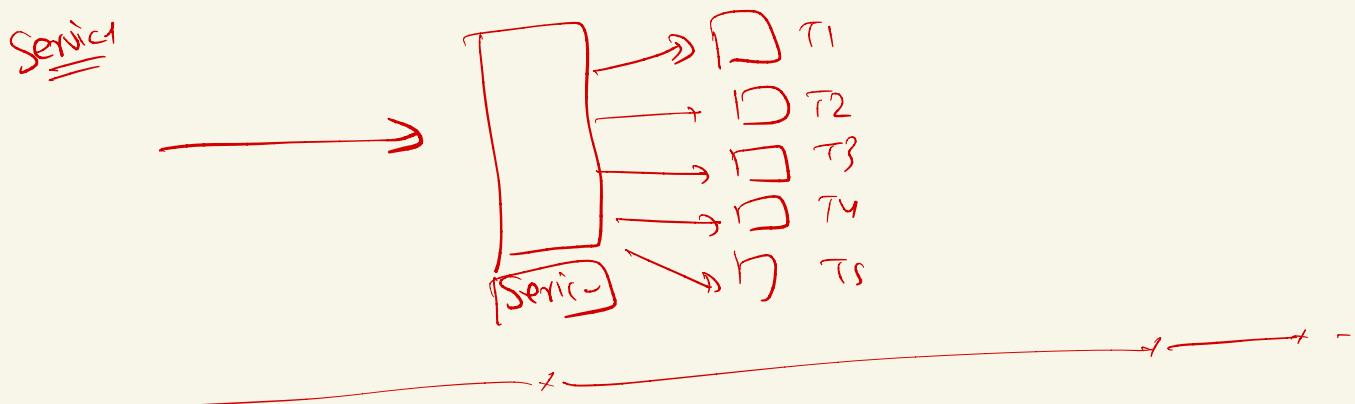
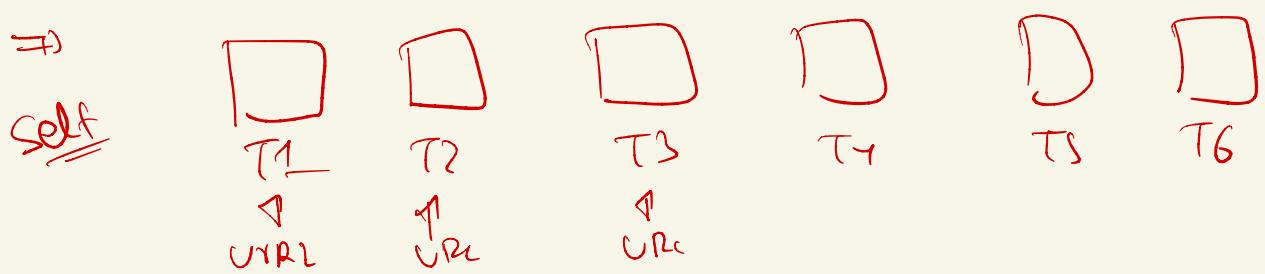
⇒ Security groups → ensure internet user only see limited features.

→ 3000      5000      7000

⇒ If you are using Mac OS (M1, M2)  
Docker build cmd    --platform =

⇒ Service      ↗ task 1  
                    ↗ task 2

Same task 10 times →



→ IAM

→ ECR

→ ECS → Cluster  
→ Task Defin

→ Task Run

→ Service

### ECS Steps :

1. Go to ECS , choose the old layout from the left panel
2. GO to Clusters from the left Panel, then Click on create cluster, choose networking only/fargate.
3. Give name to the cluster (You can also choose to create VPC) and submit.
4. From left panel got to task definitions and create a new task definitions.
5. Provide name , Default Memory and CPU , Click on Add container.
6. Provide name of container and image path for the ECR repo. Also open the needed port in this dialog.
7. Create the task definition

8. Click on the newly created task definition and choose the name.  
Click on the actions menu and choose Run Task.
9. Provide details (fargate, number of tasks ) and run the task.
10. On successful execution you will get a Public IP which can be used to access your own Api server.