AWS Gyakorlat
Docker

Felde Imre

ECS – Elastic Container Servvice

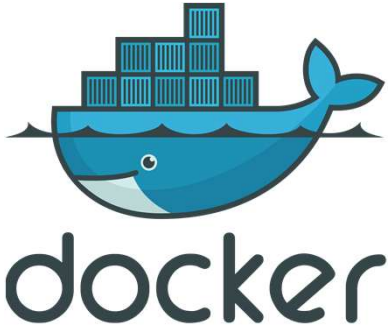
- **Amazon Elastic Container Service**
  - Egy skálázható és gyors konténerkezelő szolgáltatás
  - Orchestrálja (összehangolja) a klaszter konténereit – egyszerűen futtathatod, leállíthatod és kezelheted a konténereket

- **De miért is olyan fontos/érdekes mindez?**

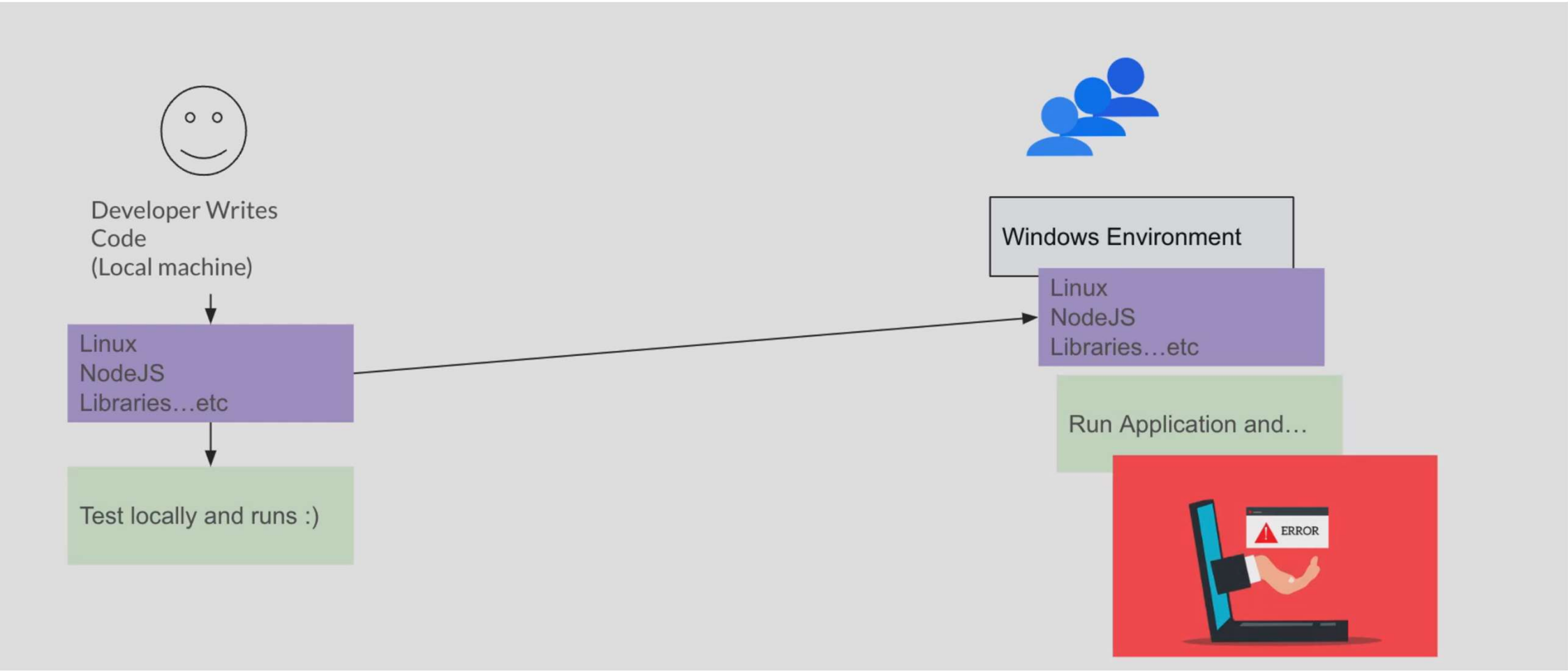
Mi a Docker?

- A Docker egy szoftverfejlesztési platform, amelyet alkalmazások telepítésére használnak.
- Az alkalmazások **konténerekbe** vannak csomagolva, amelyeket bármilyen operációs rendszeren lehet futtatni.

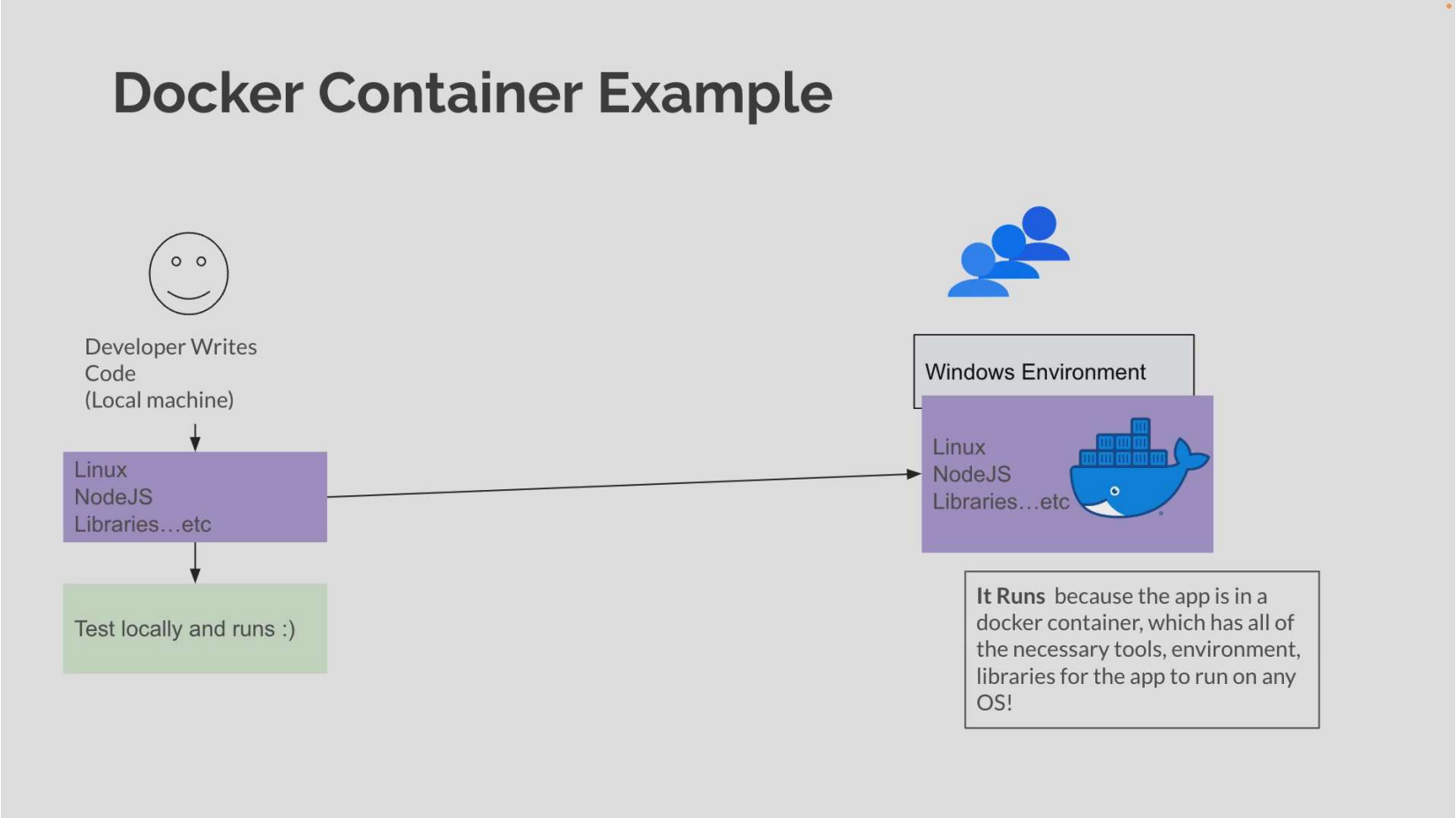
- **Az alkalmazások mindenhol ugyanúgy futnak – függetlenül attól, hol futtatják őket.**
  - Bármilyen gépen
  - Nincsenek kompatibilitási problémák
  - Előre kiszámítható működés
  - Kevesebb munka
  - Könnyebb karbantartani és telepíteni
  - Bármilyen nyelvvel, operációs rendszerrel vagy technológiával működik
  - A konténerek nagyon gyorsan skálázhatók fel és le (másodpercek alatt)



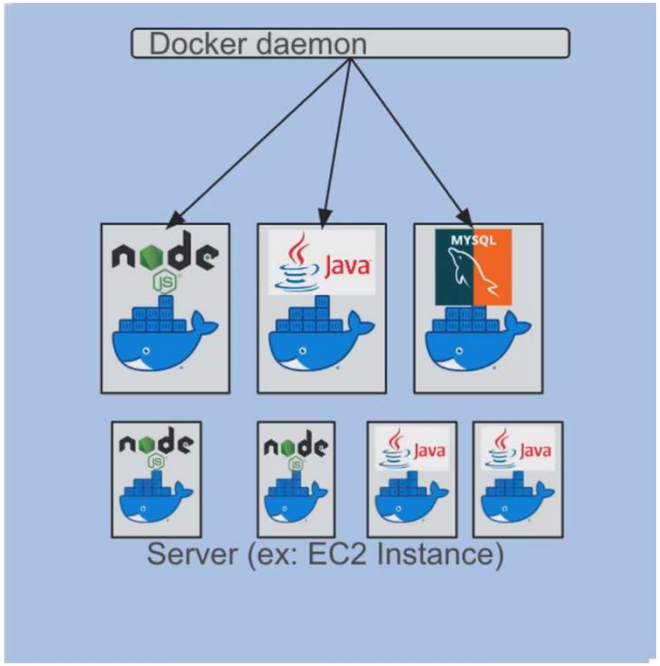
Mi a Docker?


Developer Writes
Code
(Local machine)
Linux
NodeJS
Libraries…etc
Test locally and runs :)
Windows Environment
Linux
NodeJS
Libraries…etc
Run Application and…
ERROR

Docker Container példa



Docker egy OS-n fut



Docker daemon – kezeli, építi és futtatja a Docker konténereket (a Docker ügynökkel együtt).

Docker image – Hol tárolódik?

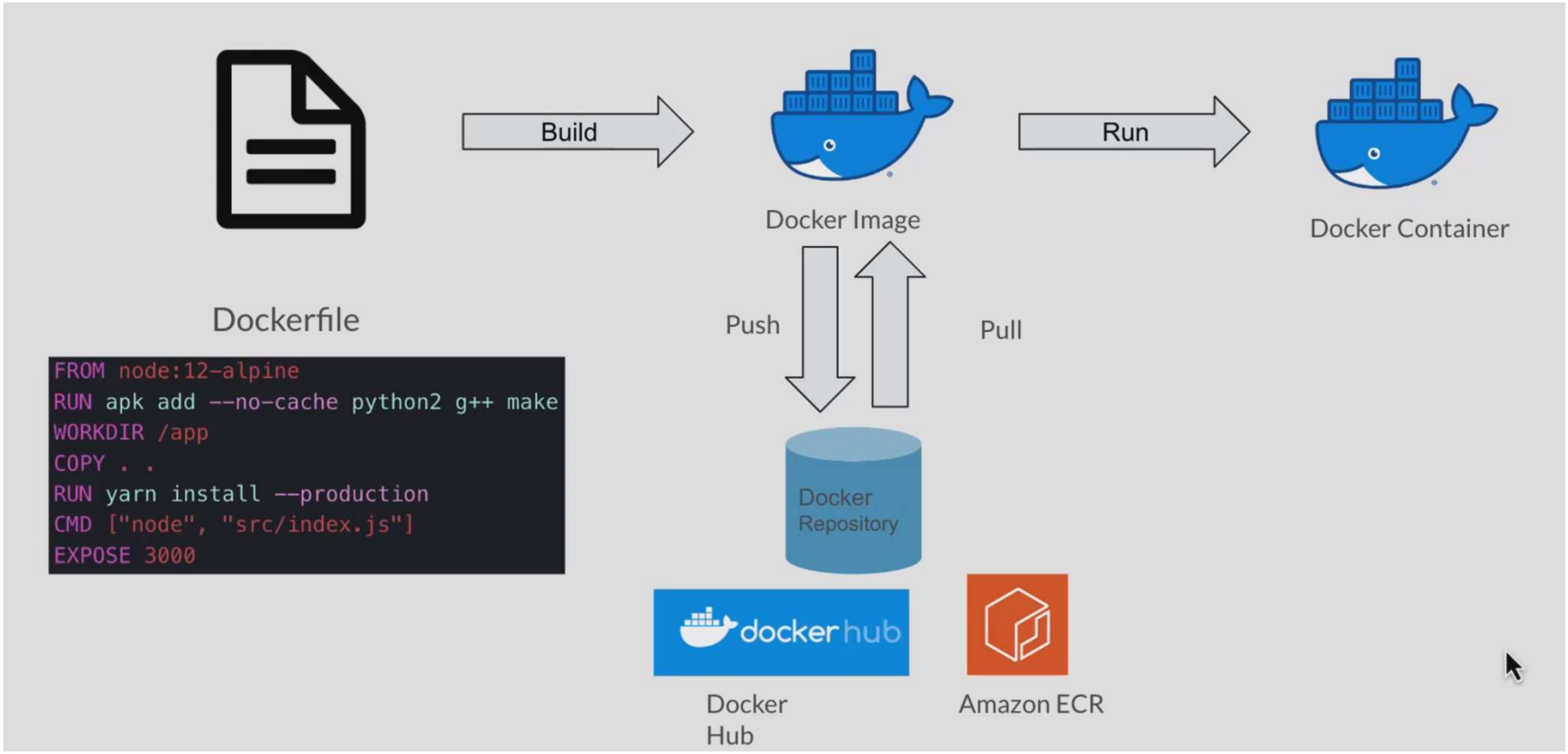
- **Docker tárolók (Repositories)**

- **Nyilvános:** Docker Hub https://hub.docker.com/ és mások

- Itt alap image-eket (képfájlokat) találhatsz sokféle technológiához vagy operációs rendszerhez, például:
  - **Ubuntu**
  - **MySQL**
  - **NodeJS, Java...**

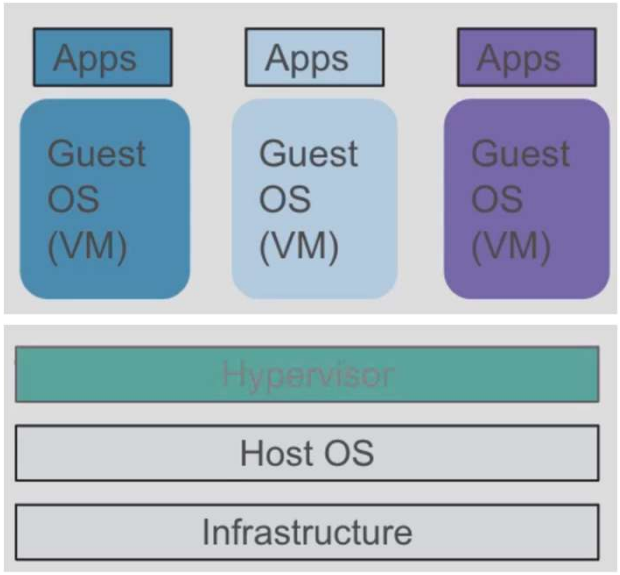
- **Privát:** Amazon ECR (Elastic Container Registry)

Docker alapok



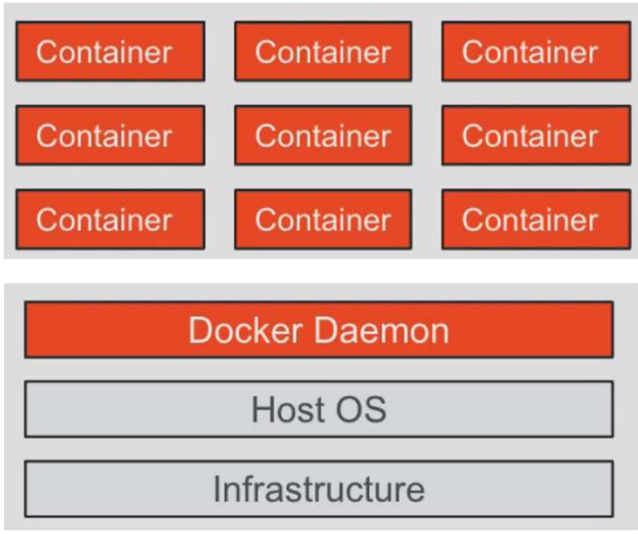
Docker vs VM

- A **Docker „félig-meddig” virtualizációs technológia**, de nem teljesen az – **konténertechnológia**.
- Az **erőforrások megosztva vannak a gazdagéppel (host)** → így **sok konténer futhat egyetlen szerveren**.



Virtual Machine

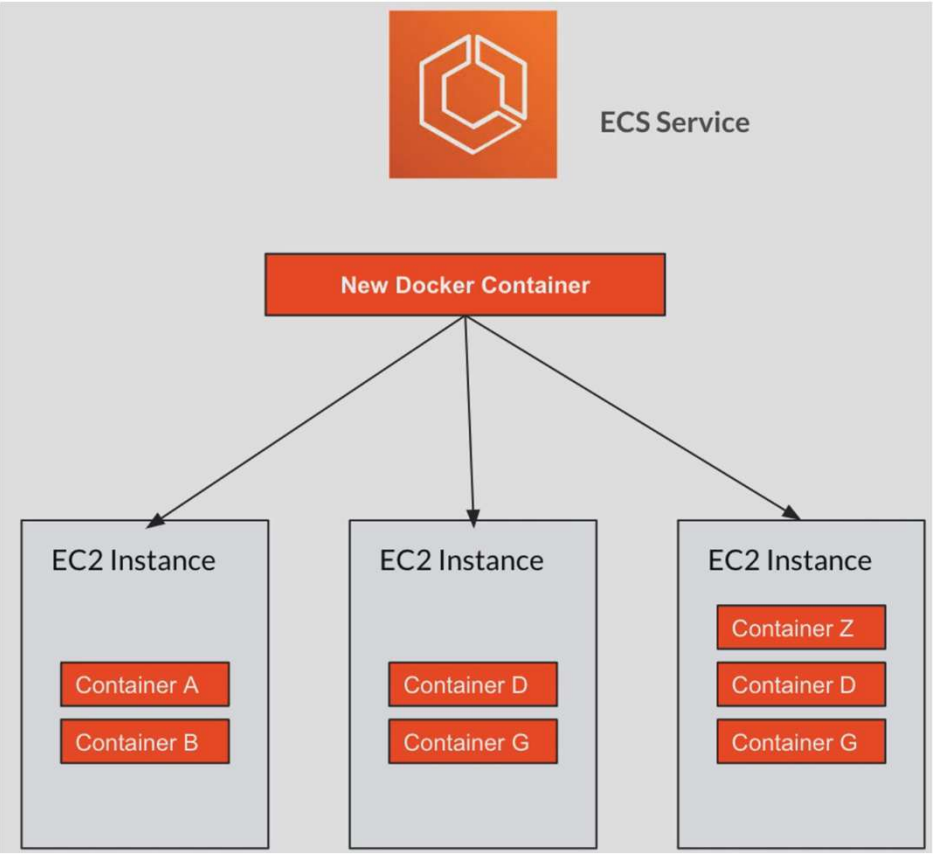
- A **Docker daemon könnyűsúlyú** (nem igényel sok erőforrást).
- **Kevesebb az elkülönítés a konténerek között** – meg tudják **osztani az erőforrásokat**, és **kommunikálni tudnak egymással**.



Docker

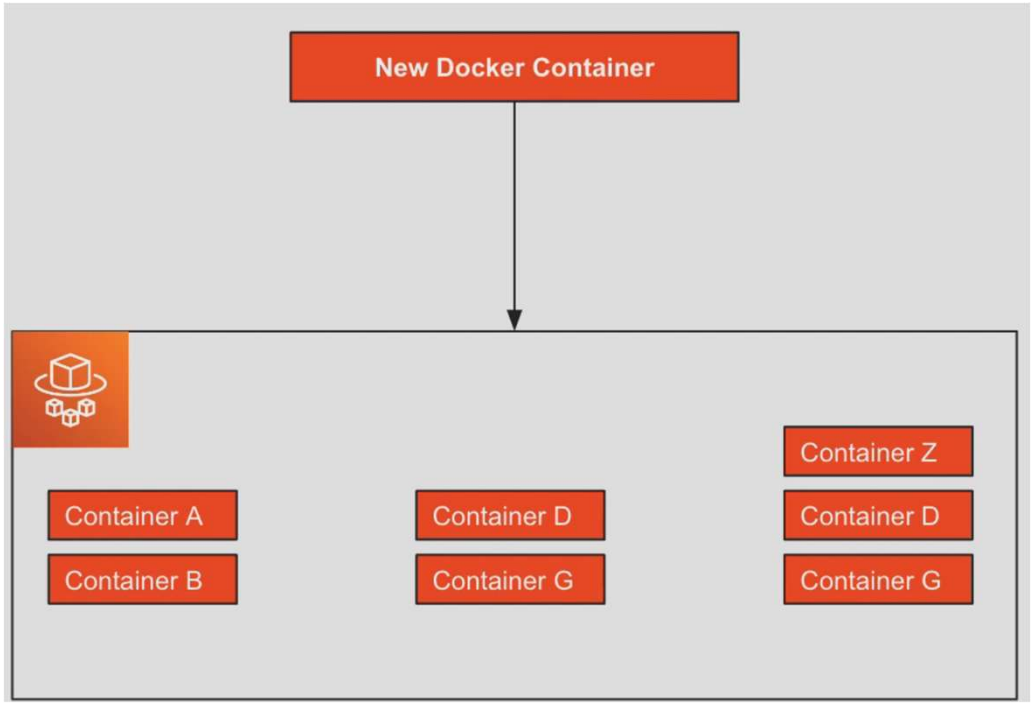
ECS - Elastic Container Servvice

- **ECS – Elastic Container Service**
- **Docker konténerek indítása az AWS-en**
- **Neked kell biztosítanod és karbantartanod az infrastruktúrát** (azaz az **EC2 példányokat**)
- Az **AWS gondoskodik a konténerek indításáról és leállításáról**
- **Össze tudod kapcsolni az Application Load Balancerrel** (Alkalmazás-terheléselosztóval)



Mi az a FARGATE?

- **Docker konténerek futtatása az AWS-en**
- **Nem kell** a felhasználónak biztosítani és karbantartani az infrastruktúrát (**nincsenek EC2 példányok**)
- **Serverless (szerver nélküli) megoldás**
- Az **AWS automatikusan futtatja a konténereket**, az általad megadott **CPU- és memóriaigény** alapján



Mi az a Fragate?

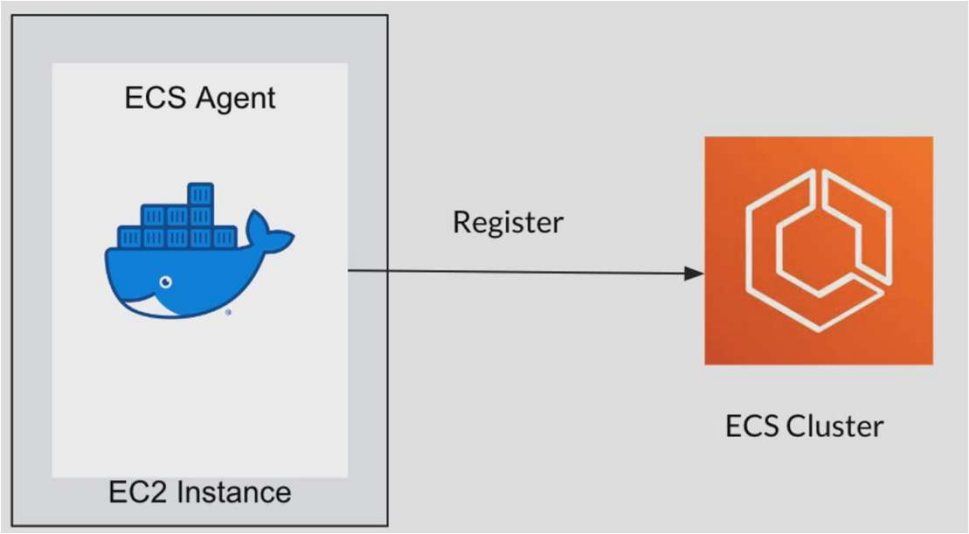
- Fargate egy serverless (szerver nélküli) konténer-futtatási szolgáltatás.
- Ez azt jelenti, hogy nem kell saját szervereket (EC2 példányokat) indítanod, beállítanod vagy karbantartanod.
- Az AWS automatikusan gondoskodik az infrastruktúráról – te csak megmondod, mennyi CPU-t és memóriát szeretnél, és milyen Docker konténert futtatsz.

Egyszerűen:

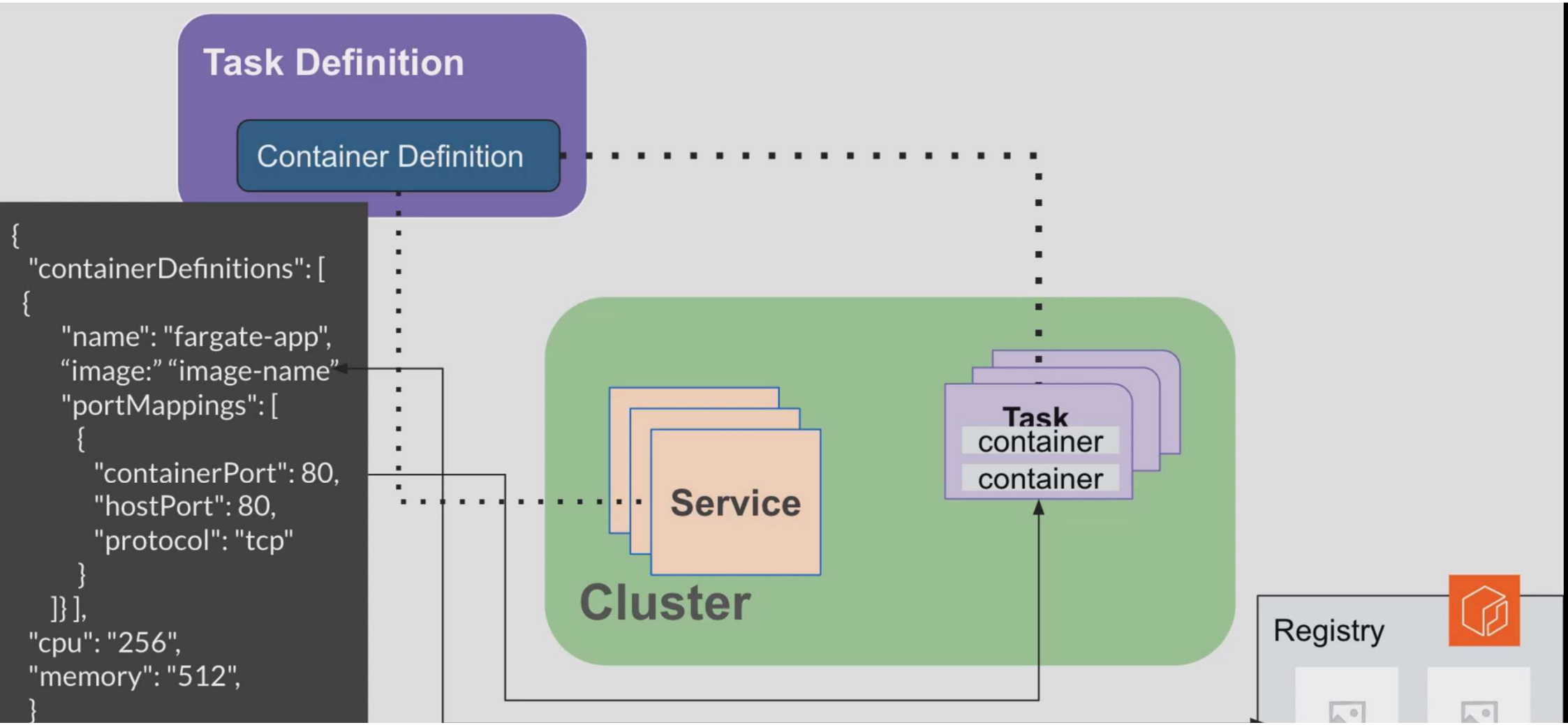
- **Csak a konténert és az erőforrásigényt kell megadni**
- **Az AWS elindítja, futtatja és leállítja a konténert, amikor kell.**
- **Nem kell szervereket kezelni (serverless).**
- **Kisebb adminisztráció, gyorsabb skálázás és kevesebb hiba.**

ECS Cluster

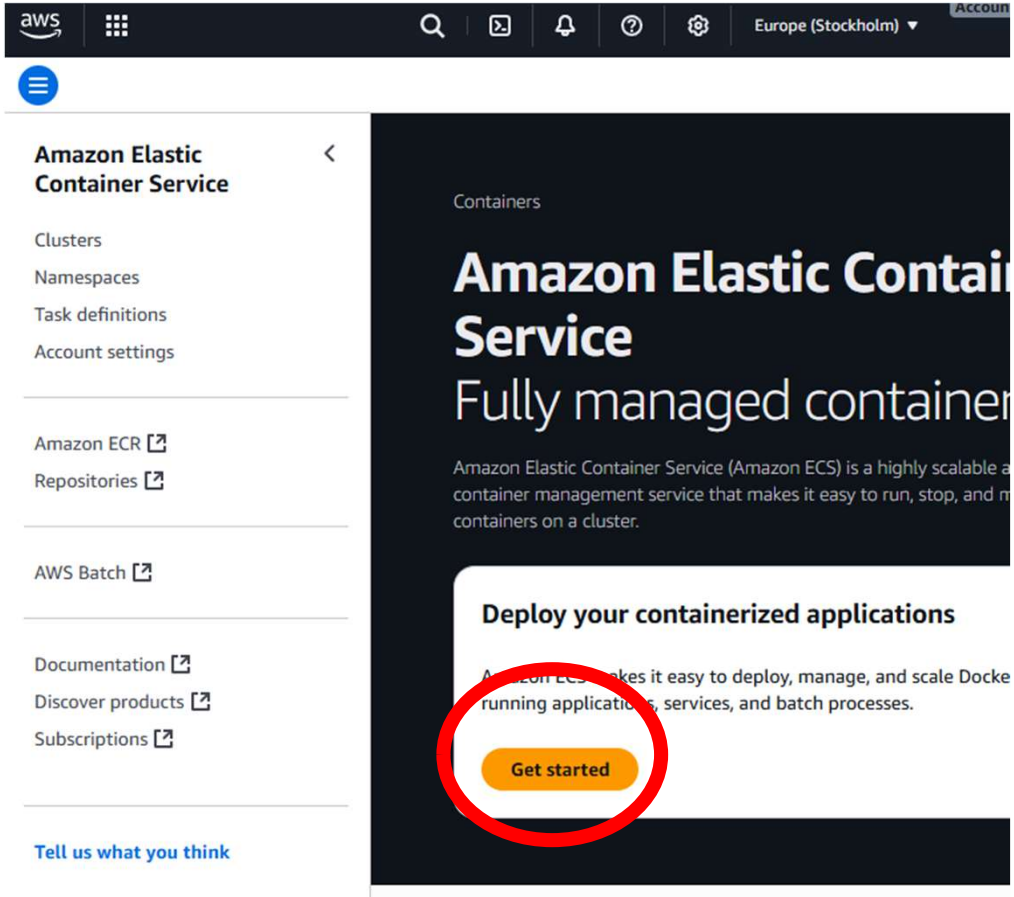
- Az **ECS klaszterek** az **EC2 példányok logikai csoportosításai**.
- Az **EC2 példányok** futtatják az **ECS Agentet** (ami egy Docker konténer).
- Az **ECS Agent** regisztrálja a példányt az **ECS klaszterhez**.
- Az **EC2 példányok** egy **speciális AMI-t (Amazon Machine Image)** futtatnak, amit **kifejezetten az ECS-hez** készítettek

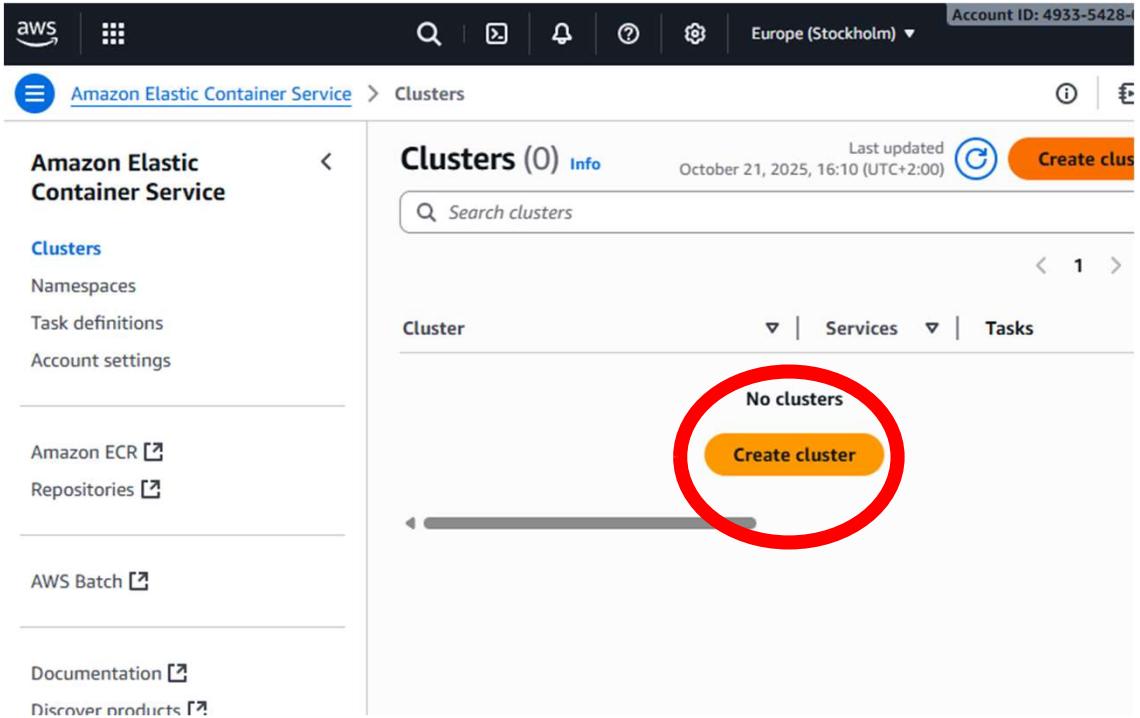


Amazon ECS

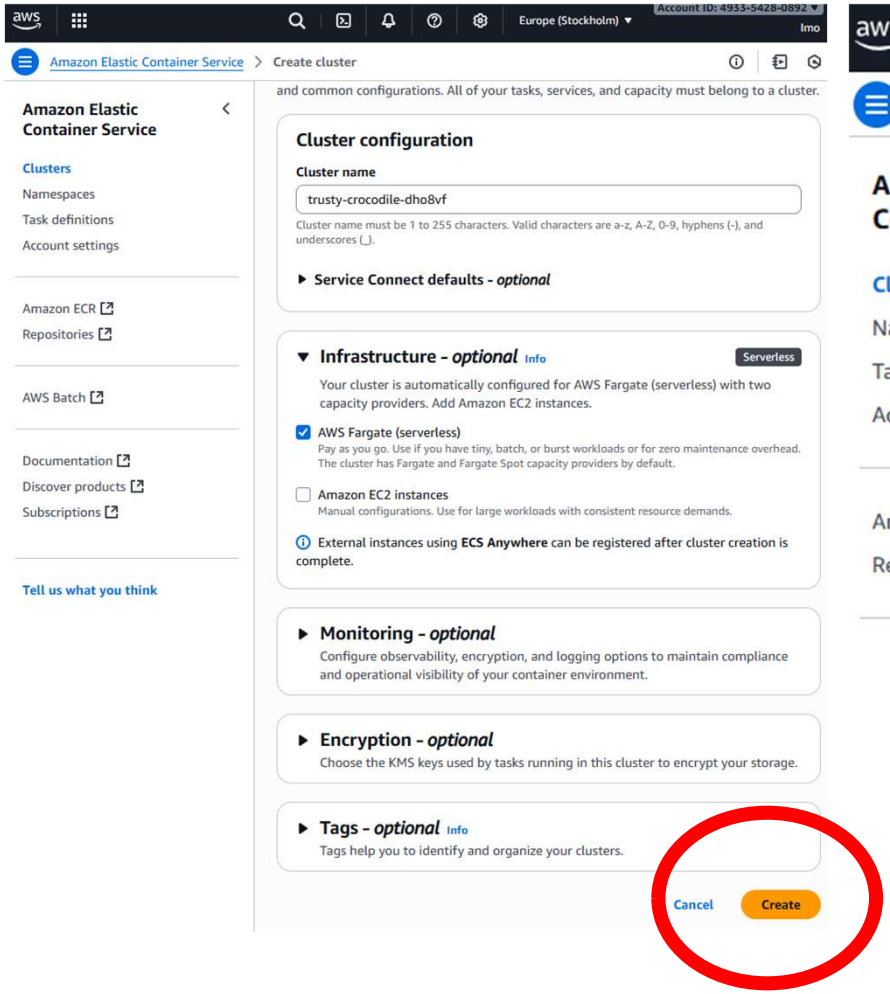

Task Definition
Container Definition
{
"containerDefinitions": [
{
"name": "fargate-app",
“image:” “image-name”
"portMappings": [
{
"containerPort": 80,
"hostPort": 80,
"protocol": "tcp"
}
]}],
"cpu": "256",
"memory": "512",
}
Service
Cluster
Task
container
container
Registry

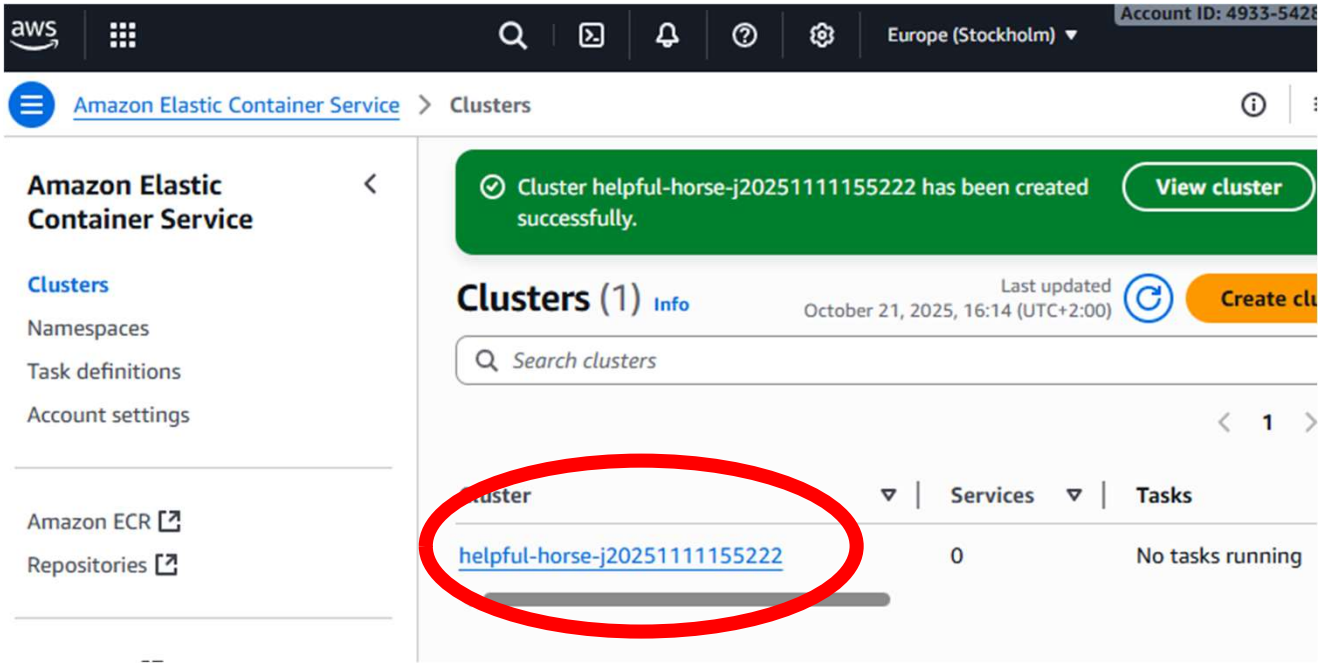
ECS


aws
Europe (Stockholm)
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
Containers
Amazon Elastic Contai
Service
Fully managed containe
Amazon Elastic Container Service (Amazon ECS) is a highly scalable
container management service that makes it easy to run, stop, and
containers on a cluster.
Deploy your containerized applications
kes it easy to deploy, manage, and scale Docke
running applicati s, services, and batch processes.
Get started

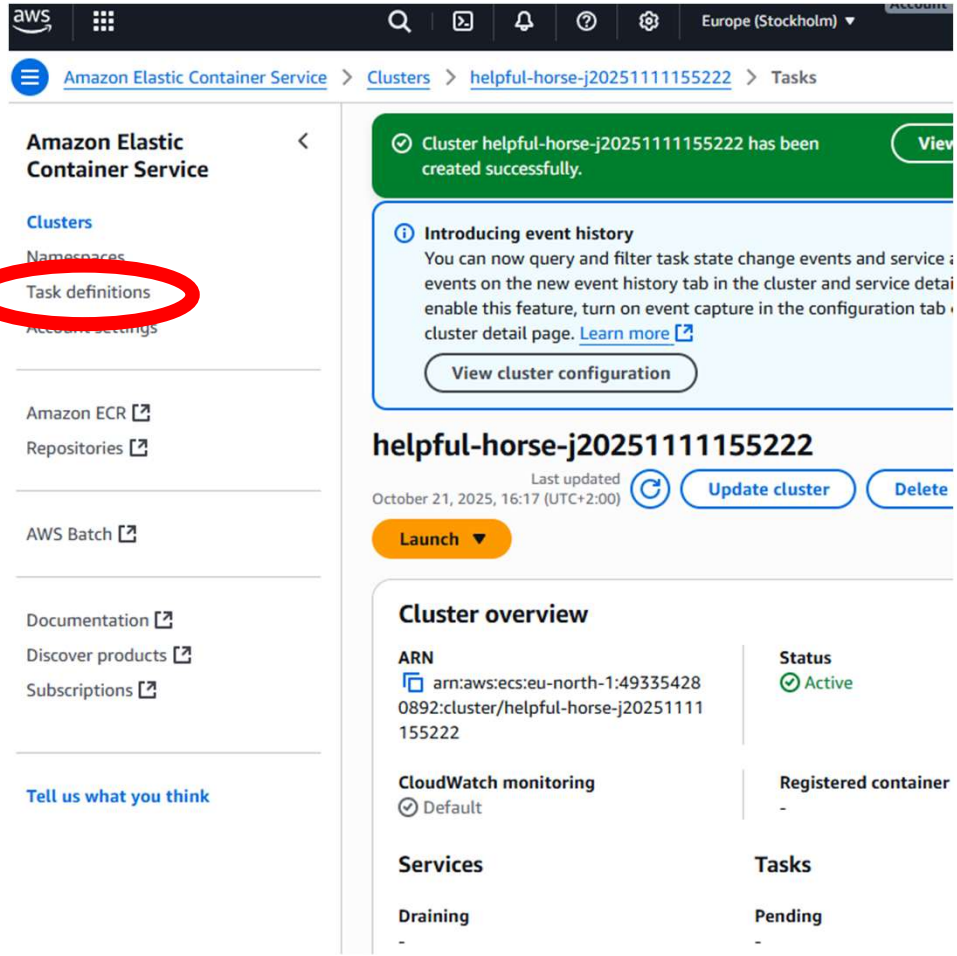

aws
Europe (Stockholm)
Account ID: 4933-5428-
Amazon Elastic Container Service
Clusters
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Clusters (0) Info
Last updated
October 21, 2025, 16:10 (UTC+2:00)
Create clus
Search clusters
1
Cluster
Services
Tasks
No clusters
Create cluster

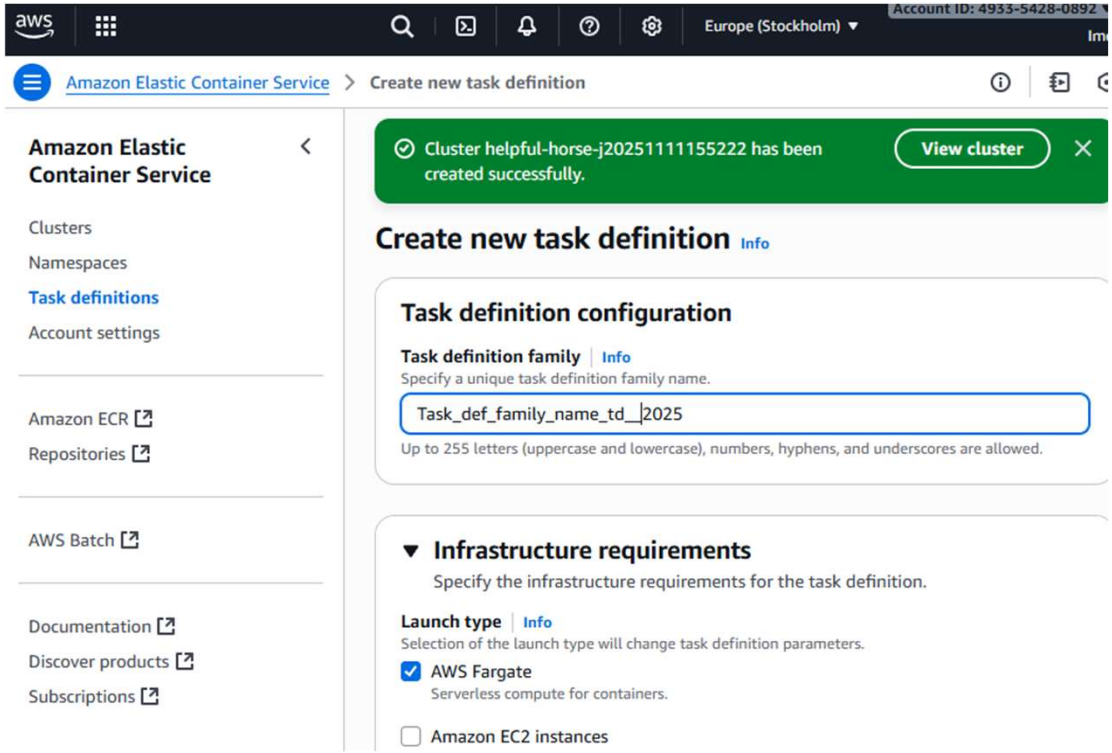
ECS


Europe (Stockholm)
Amazon Elastic Container Service
Create cluster
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
and common configurations. All of your tasks, services, and capacity must belong to a cluster.
Cluster configuration
Cluster name
trusty-crocodile-dho8vf
Cluster name must be 1 to 255 characters. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).
Service Connect defaults - optional
Infrastructure - optional Info
Serverless
Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances.
AWS Fargate (serverless)
Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.
Amazon EC2 instances
Manual configurations. Use for large workloads with consistent resource demands.
External instances using ECS Anywhere can be registered after cluster creation is complete.
Monitoring - optional
Configure observability, encryption, and logging options to maintain compliance and operational visibility of your container environment.
Encryption - optional
Choose the KMS keys used by tasks running in this cluster to encrypt your storage.
Tags - optional Info
Tags help you to identify and organize your clusters.
Cancel
Create

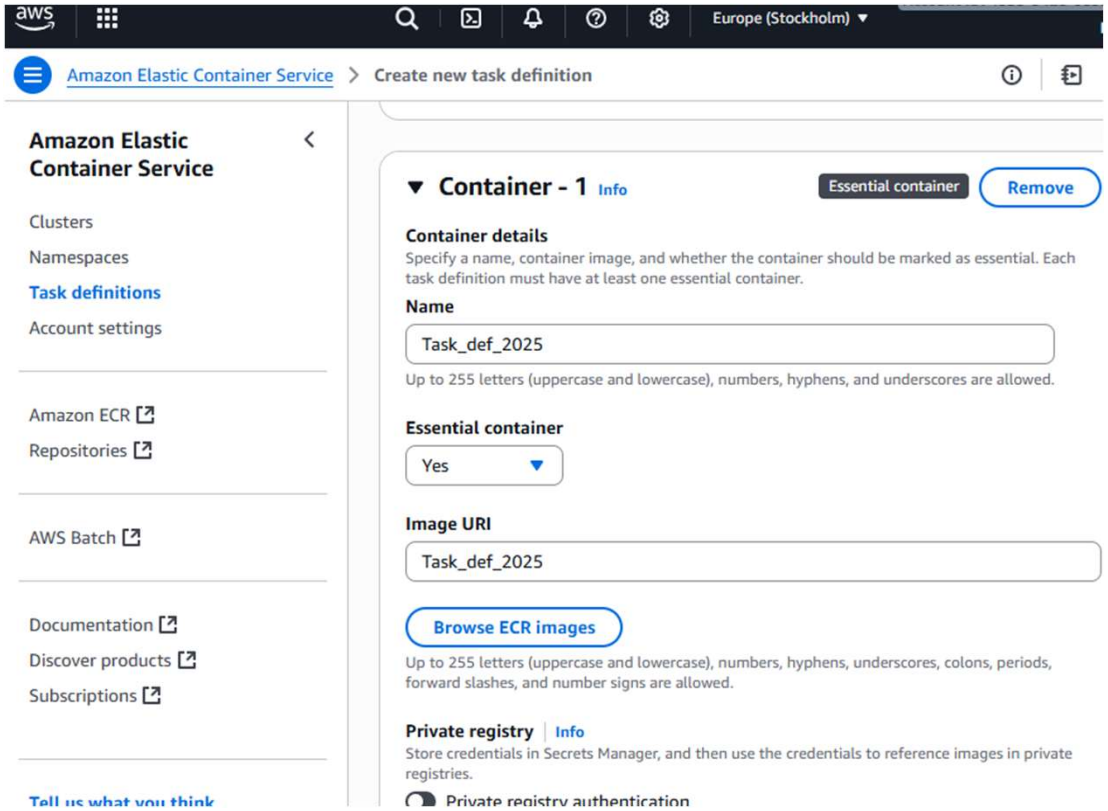

Europe (Stockholm)
Amazon Elastic Container Service
Clusters
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
Cluster helpful-horse-j20251111155222 has been created successfully.
View cluster
Clusters (1) Info
Last updated
October 21, 2025, 16:14 (UTC+2:00)
Search clusters
1
Services
Tasks
helpful-horse-j20251111155222
0
No tasks running

ECS

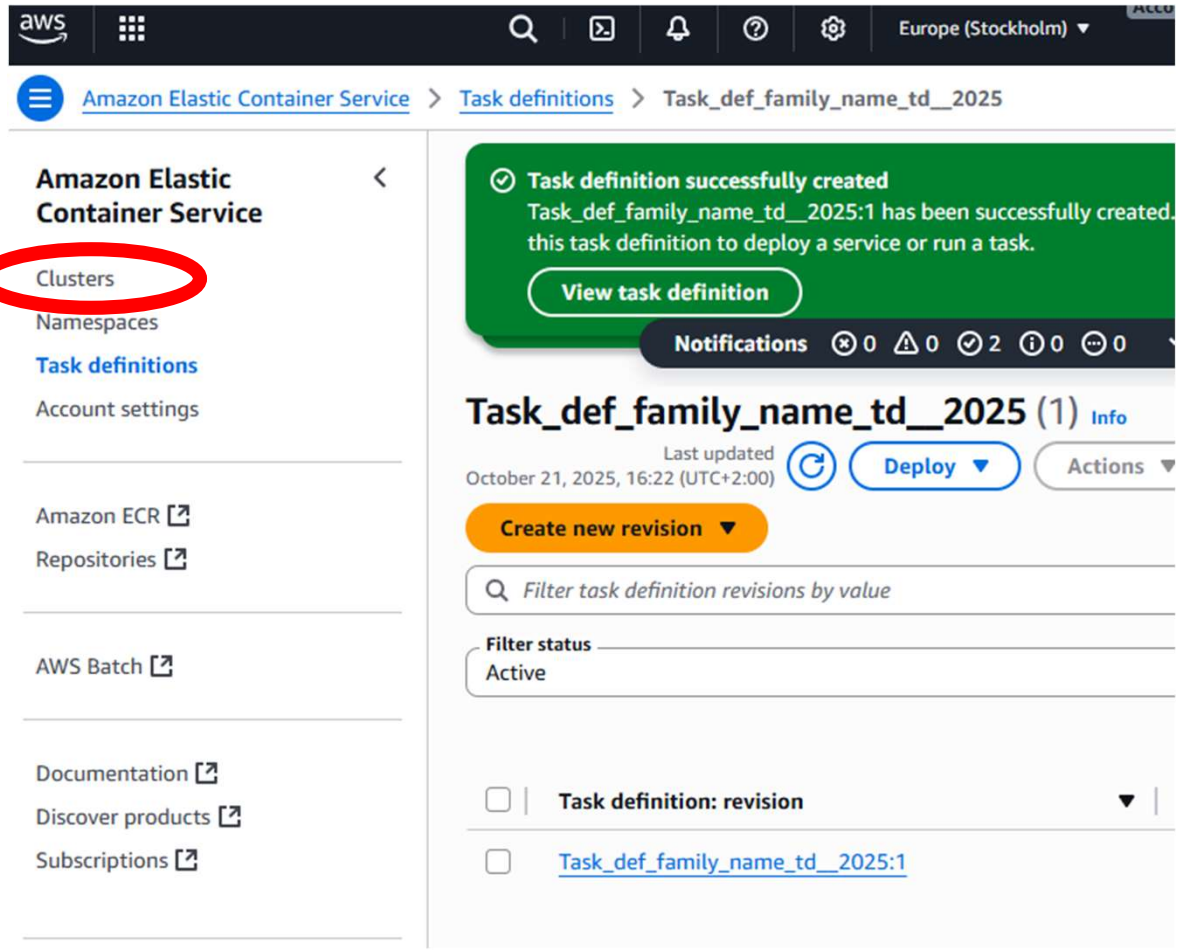

Europe (Stockholm)
Amazon Elastic Container Service > Clusters > helpful-horse-j20251111155222 > Tasks
Amazon Elastic Container Service
Clusters
Task definitions
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
Cluster helpful-horse-j20251111155222 has been created successfully.
Introducing event history
You can now query and filter task state change events and service
events on the new event history tab in the cluster and service detai
enable this feature, turn on event capture in the configuration tab
cluster detail page. Learn more
View cluster configuration
helpful-horse-j20251111155222
Last updated
October 21, 2025, 16:17 (UTC+2:00)
Update cluster
Delete
Launch
Cluster overview
ARN
arn:aws:ecs:eu-north-1:493354280892:cluster/helpful-horse-j20251111155222
Status
Active
CloudWatch monitoring
Default
Registered container
-
Services
Draining
-
Tasks
Pending
-

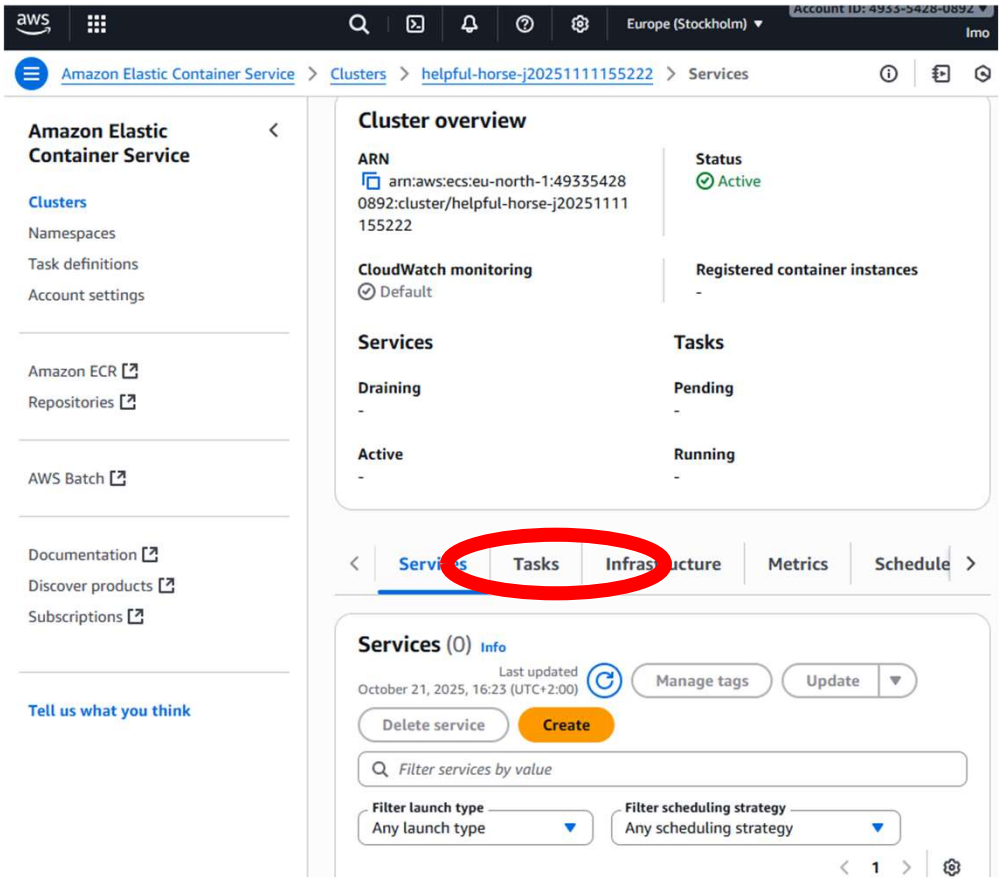

Europe (Stockholm)
Account ID: 4933-5428-0892
Amazon Elastic Container Service > Create new task definition
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Cluster helpful-horse-j20251111155222 has been created successfully.
View cluster
Create new task definition
Info
Task definition configuration
Task definition family
Info
Specify a unique task definition family name.
Task_def_family_name_td_2025
Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
Infrastructure requirements
Specify the infrastructure requirements for the task definition.
Launch type
Info
Selection of the launch type will change task definition parameters.
AWS Fargate
Serverless compute for containers.
Amazon EC2 instances

ECS


aws
Europe (Stockholm)
Amazon Elastic Container Service
Create new task definition
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
Container - 1 Info
Essential container
Remove
Container details
Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.
Name
Task_def_2025
Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
Essential container
Yes
Image URI
Task_def_2025
Browse ECR images
Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed.
Private registry Info
Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.
Private registry authentication

ECS – Run a task





ECS - Run a task