



CODESTAR

System Architecture

CodeStar Academy

Nội dung chính

Monolithic và Microservice

Beanstalk

Docker

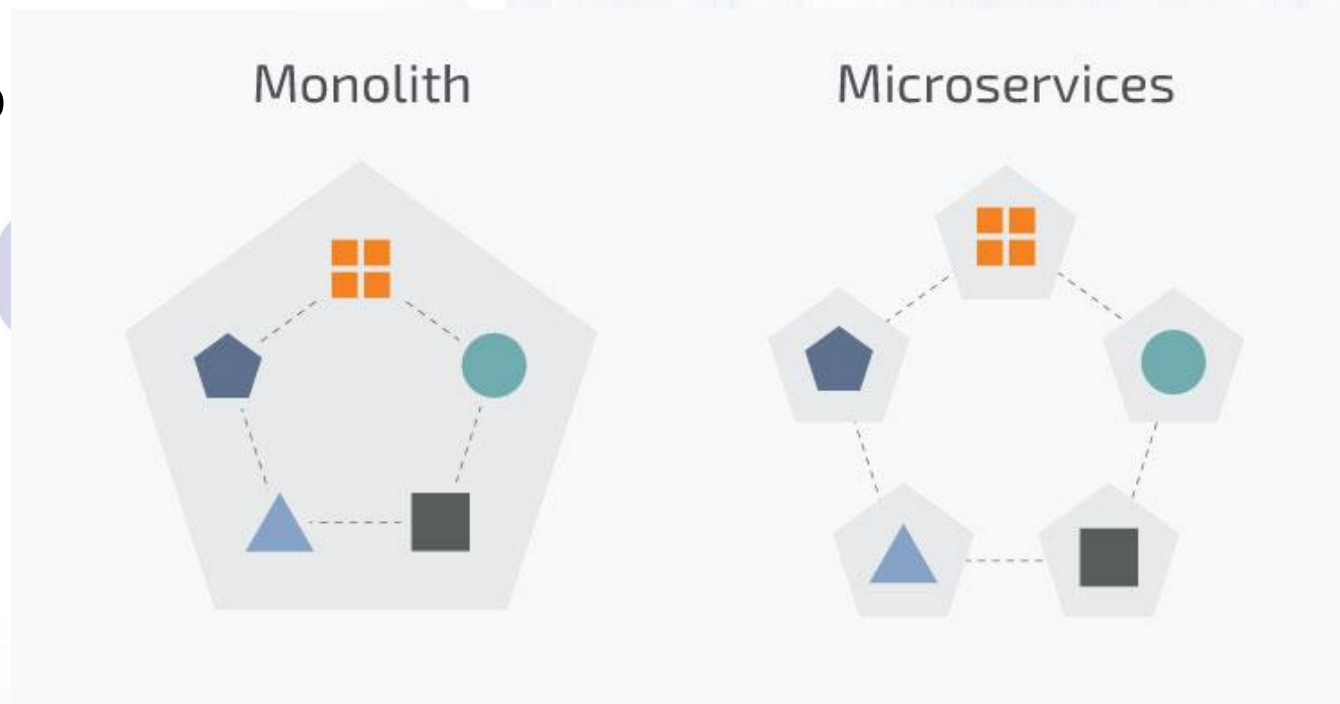
ECS



Monolithic và Microservice

Trong quá trình phát triển phần mềm và xây dựng hệ thống, có 2 hướng phát triển:

- Monolithic
- Microservice

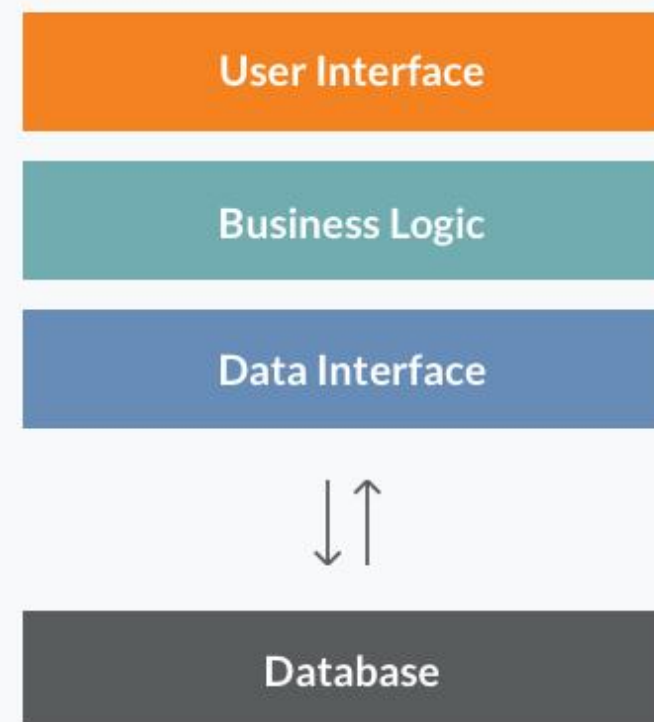


Monolithic và Microservice

Monolithic:

- Traditional way
- Các thành phần hợp nhất và không thể tách rời
- Code base thường rất lớn và nhiều chức năng.
- Deploy nguyên khối

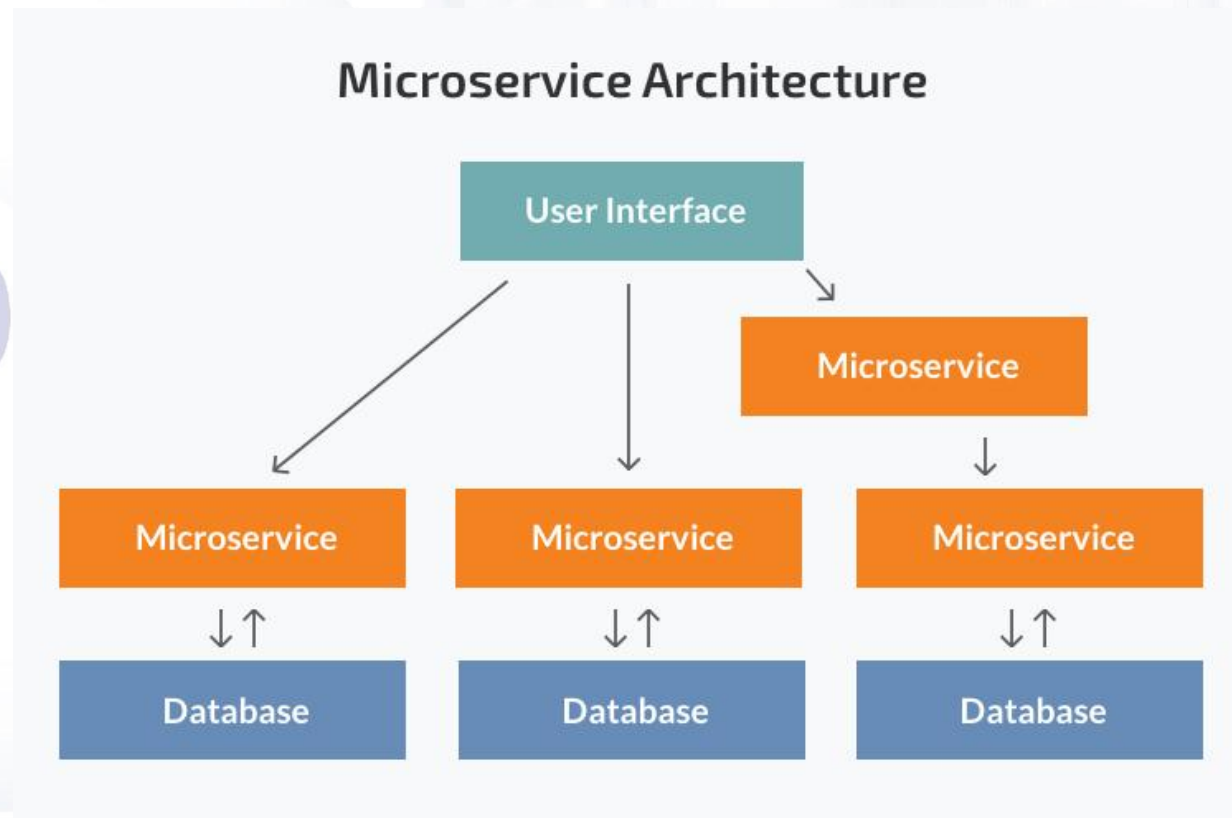
Monolithic Architecture



Monolithic và Microservice

Microservice:

- Phương án hiện đại, thích hợp cho những dự án có mức độ scale nhanh và mạnh.
- Các thành phần tách rời nhau.
- Codebase phân mảnh
- Các thành phần có thể scale, deploy, update độc lập với nhau.



Monolithic và Microservice

Monolithic

vs

Microservice

- Đơn giản
- Dễ debug, dễ test
- Dễ deploy
- Cần hiểu biết rộng cả hệ thống
- Rất khó thay đổi, update 1 phần hệ thống.
- Chỉ scale được cả hệ thống

- Tăng tính phức tạp
- Khó debug, test
- Deploy phức tạp, cần hiểu biết liên kết hệ thống.
- Dễ hiểu từng thành phần
- Dễ thay đổi, update 1 phần hệ thống.
- Scale được các thành phần riêng lẻ

Monolithic và Microservice

Monolithic:

- Team nhỏ
- Hệ thống đơn giản
- Cần deploy nhanh
- Không có áp lực phải scale nhanh.
- Không cần chuyên gia trong microservice

use-case

Microservice

- Team lớn
- Hệ thống lớn và phức tạp.
- Cần scale nhanh, và có hệ thống, quy trình deploy phức tạp.
- Team member có đủ hiểu biết về Microservice

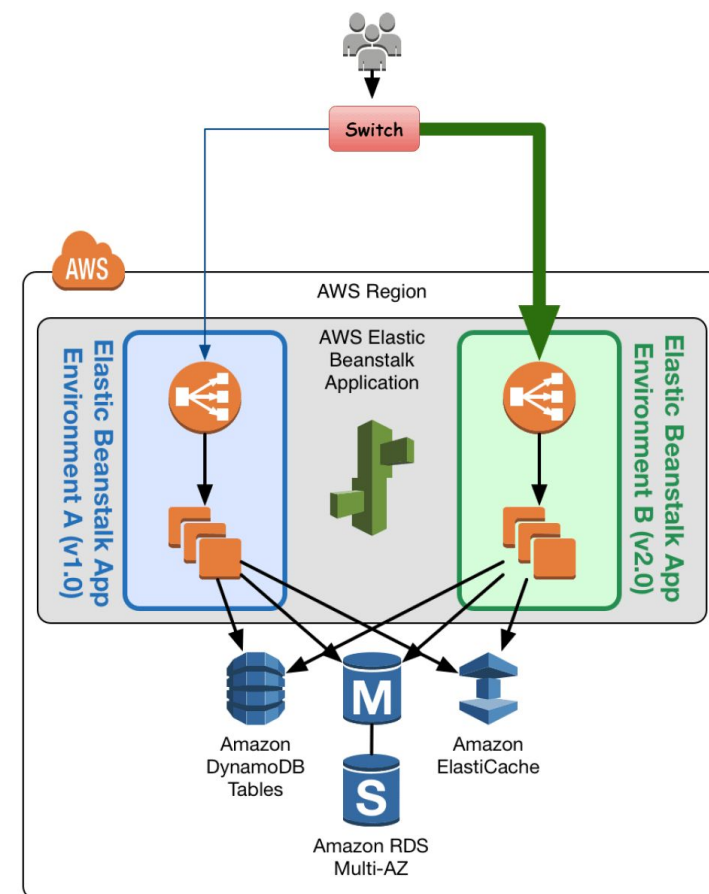
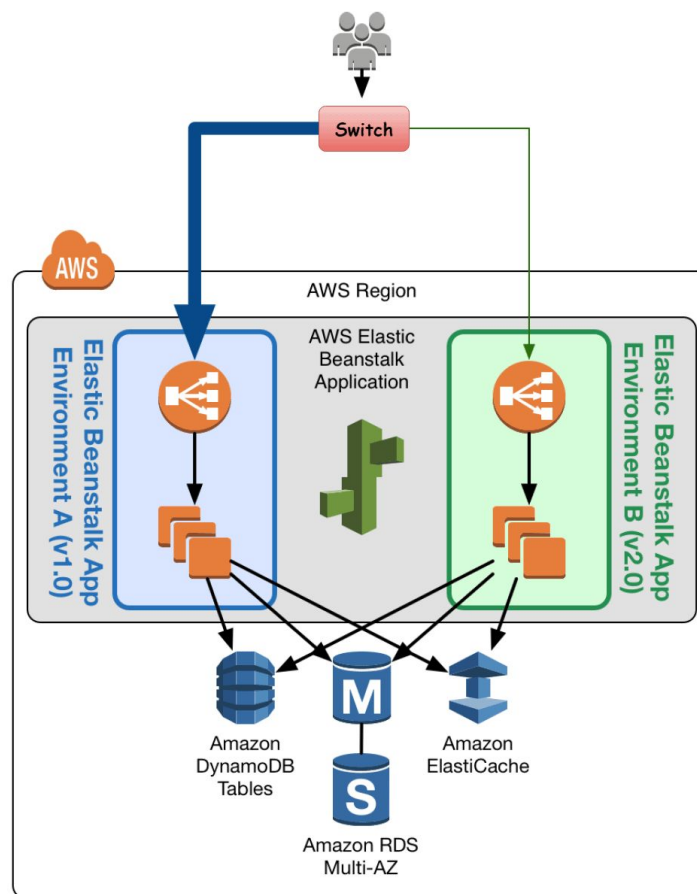
Beanstalk

- Là 1 easy-to-use service
- Dùng để deploy các hệ thống quen thuộc dễ dàng, nhanh chóng
- Cung cấp giao diện đơn giản quản lý các resource AWS
- Vẫn toàn quyền quản lý các resource AWS thủ công nếu muốn.



Beanstalk

- AWS Beanstalk hỗ trợ Blue-Green Deployment tự động.
- Không cần trả phí cho Beanstalk, chỉ phải trả phí cho các resource (tương tự dùng thủ công)



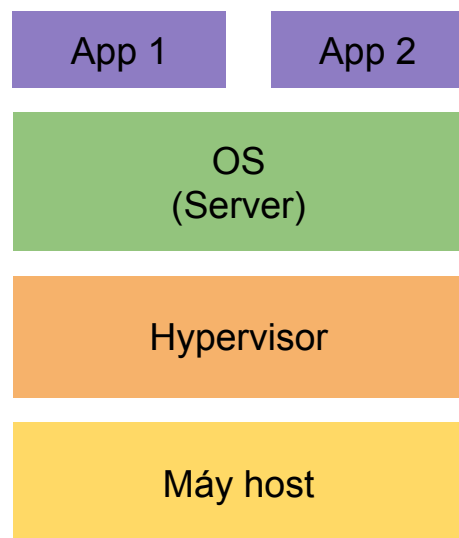
Beanstalk

Use case:

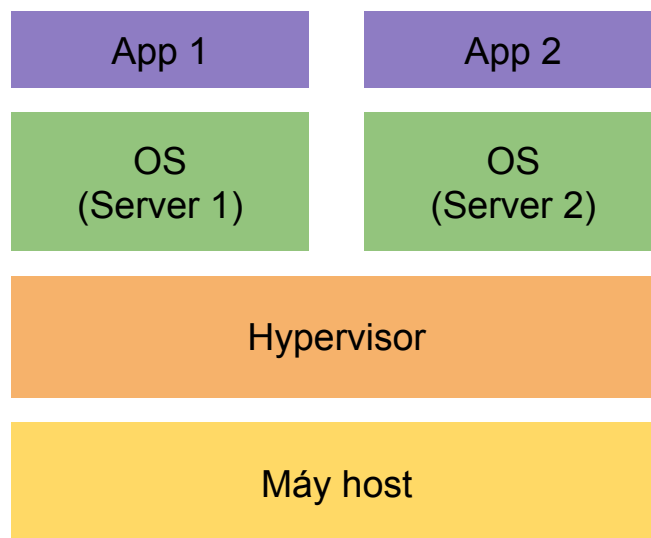
- Tích hợp với CloudWatch và X-Ray cho performance data và metrics
- Managed Platform Updates - tính năng tự động áp dụng các bản cập nhật của hệ điều hành, ngôn ngữ lập trình (Java, PHP, Node.js...)
- Thích hợp dùng cho hệ thống Monolithic và/hoặc Team muốn đơn giản hóa công việc Deploy.
- Có thể sử dụng cho một số hệ thống Container nhưng đơn giản

Docker

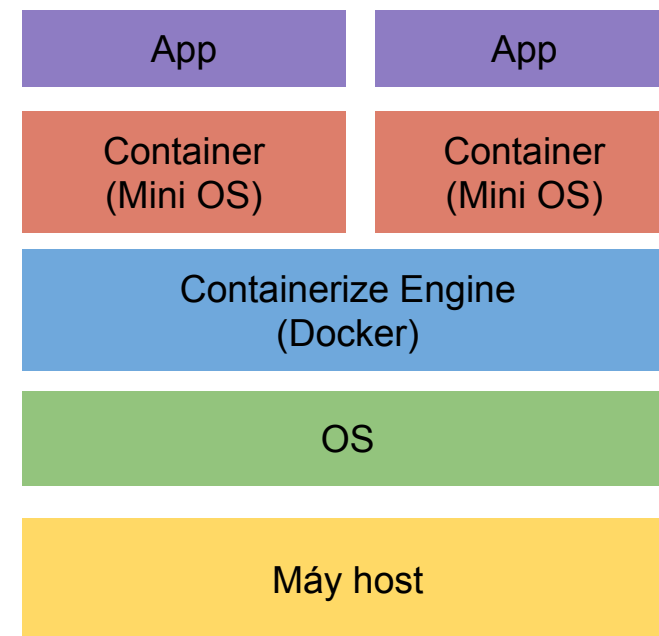
Docker: Structure



Dedicated host

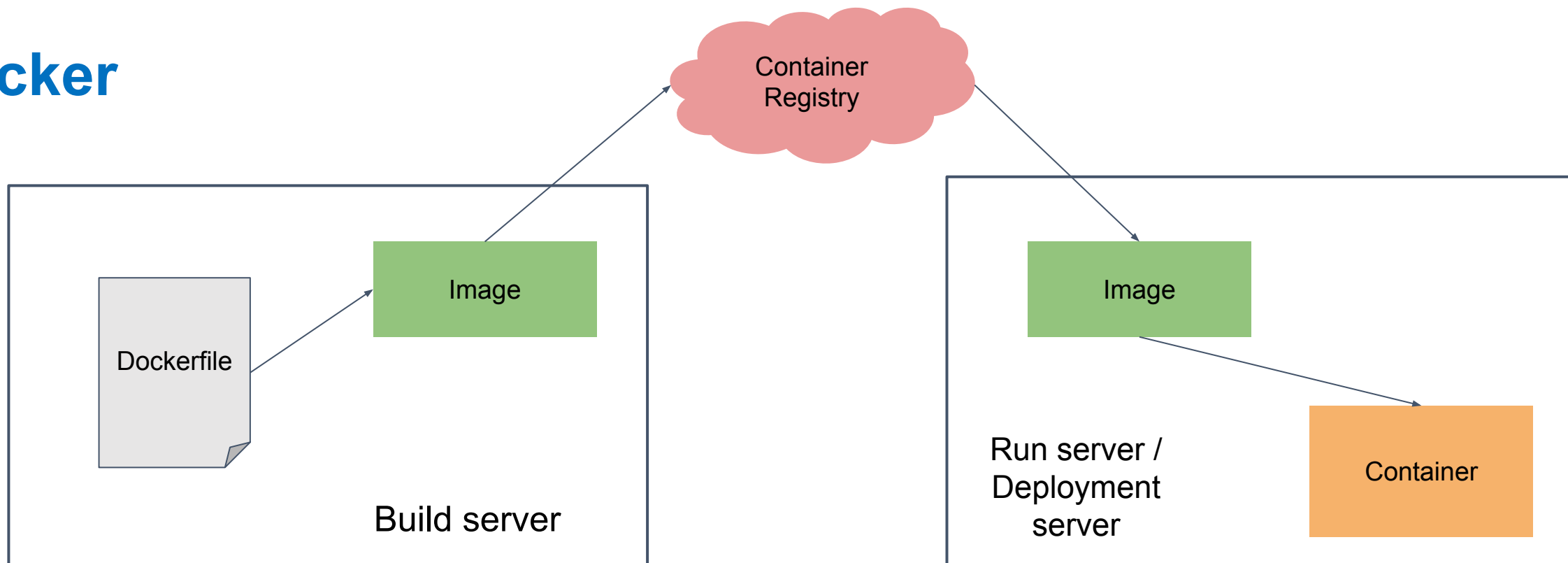


Share host



With Docker

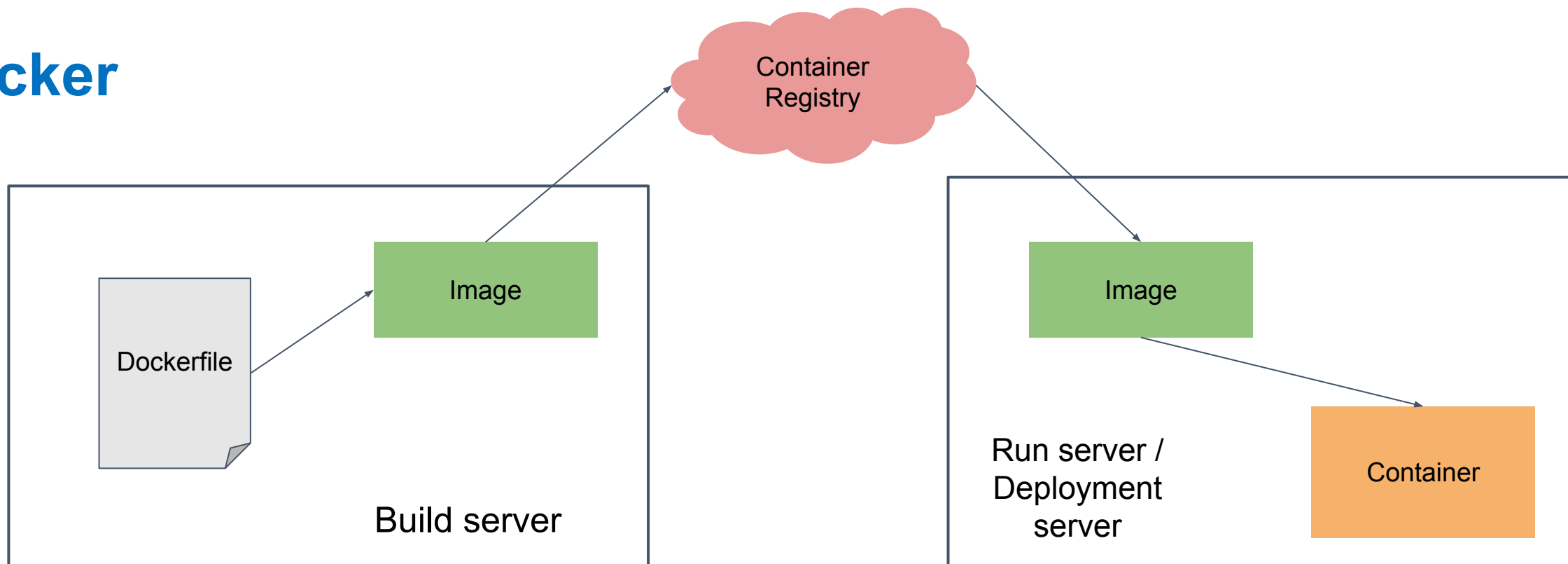
Docker



Container flow

- **Build:** Dockerfile định nghĩa các step để khởi tạo 1 image và đóng gói.
- **Run:** Container Orchestration sử dụng Image để tạo ra các Container và chạy

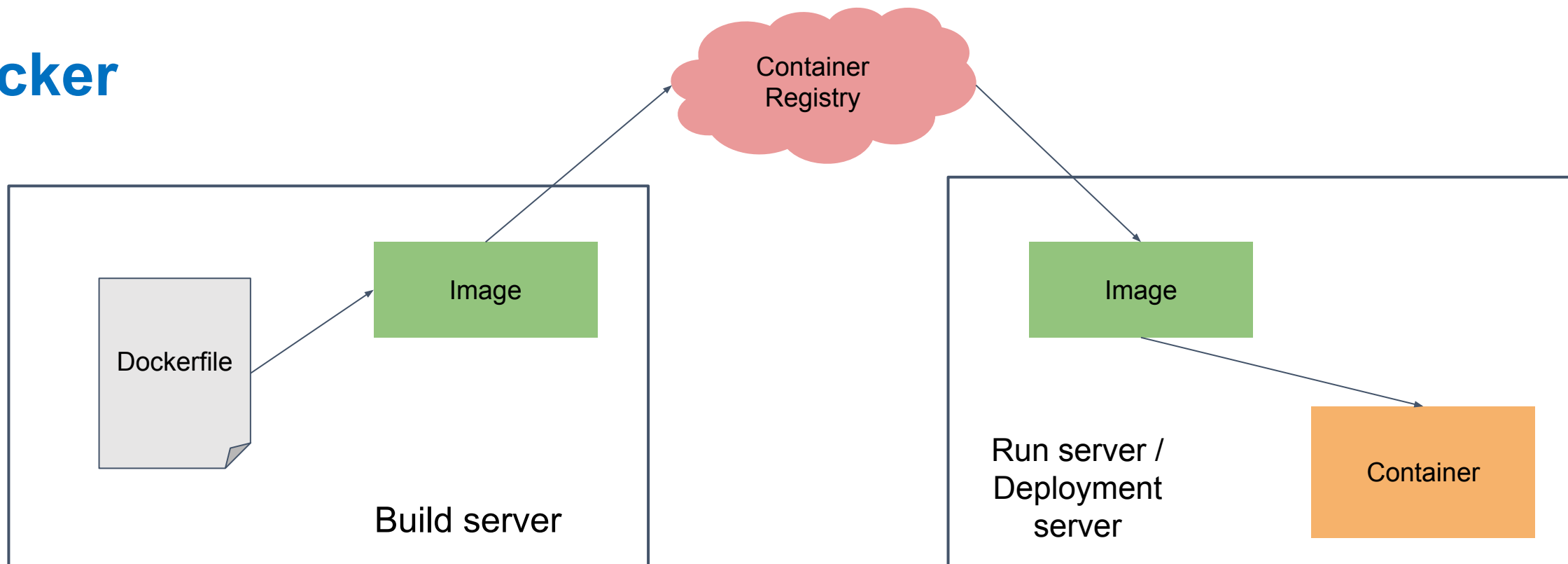
Docker



Docker Image

- Là file đóng gói hệ thống, sử dụng Dockerfile làm hướng dẫn khởi tạo.
- Là môi trường build trong dự án

Docker

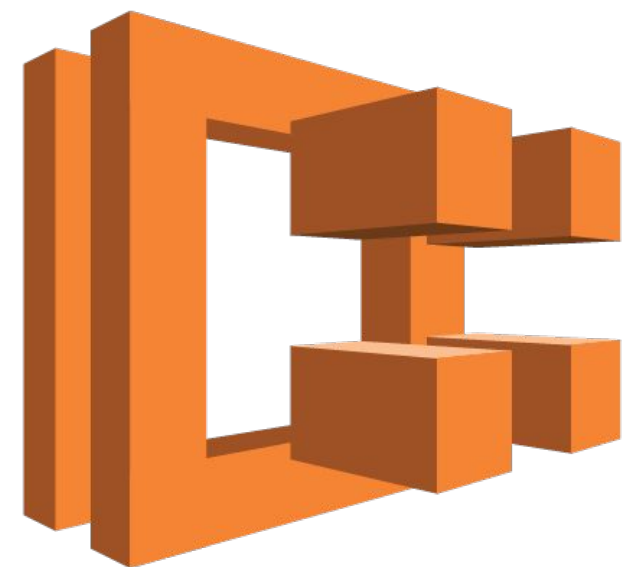


Docker Container

- Là phiên bản chạy của một image, chứa các resource của riêng nó
- Là môi trường run-time.

ECS

- Là Elastic Container Service, hoạt động như một fully managed Container Orchestration Tool
- Cho phép Launch/stop container-based App bằng API
- ECS khởi chạy các Tasks từ Container Registry

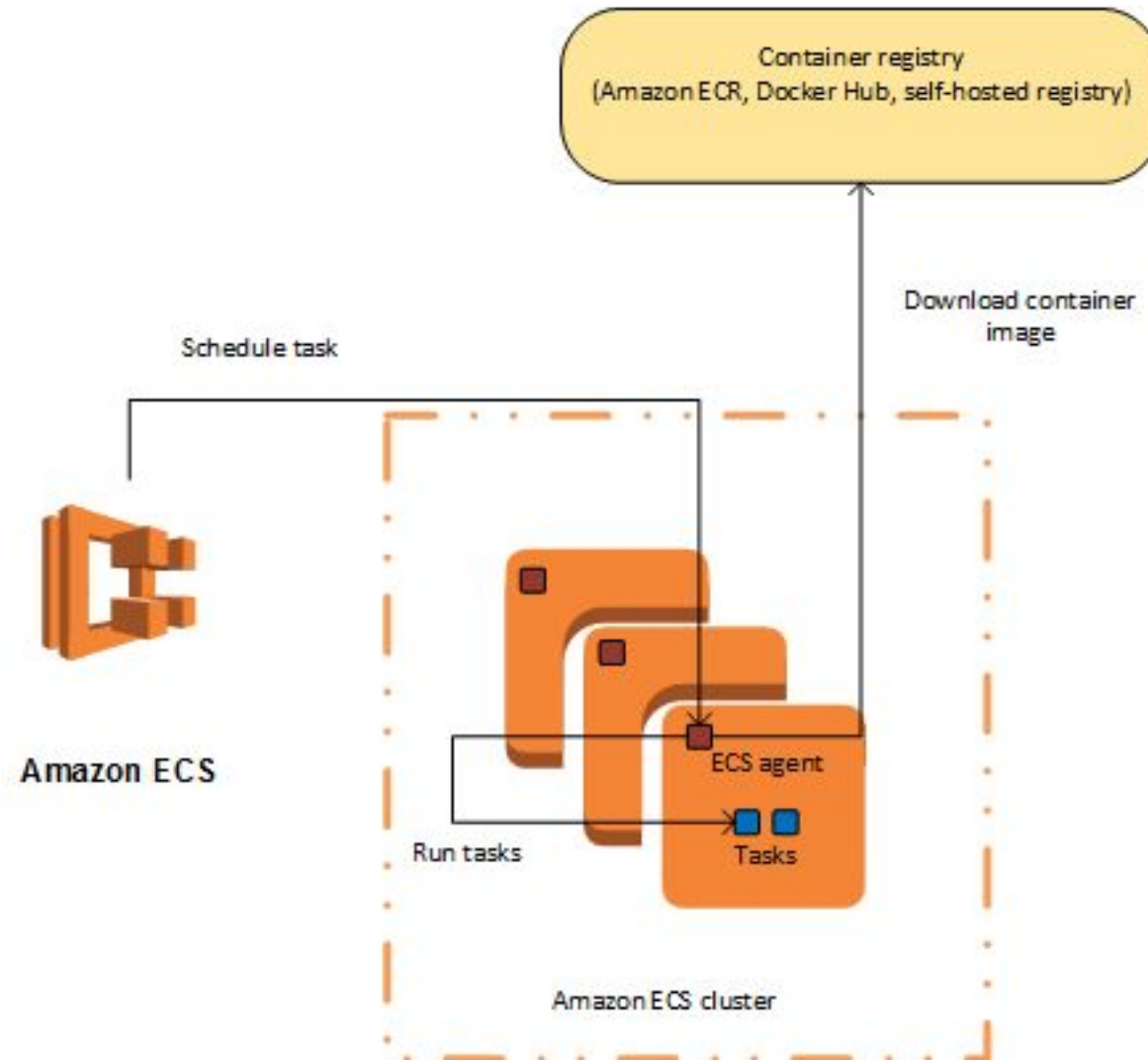


AWS ECS

ECS

Các thành phần cơ bản cho 1 hệ thống gồm:

- Task
- Task definition
- Cluster
- Container Registry
- Service



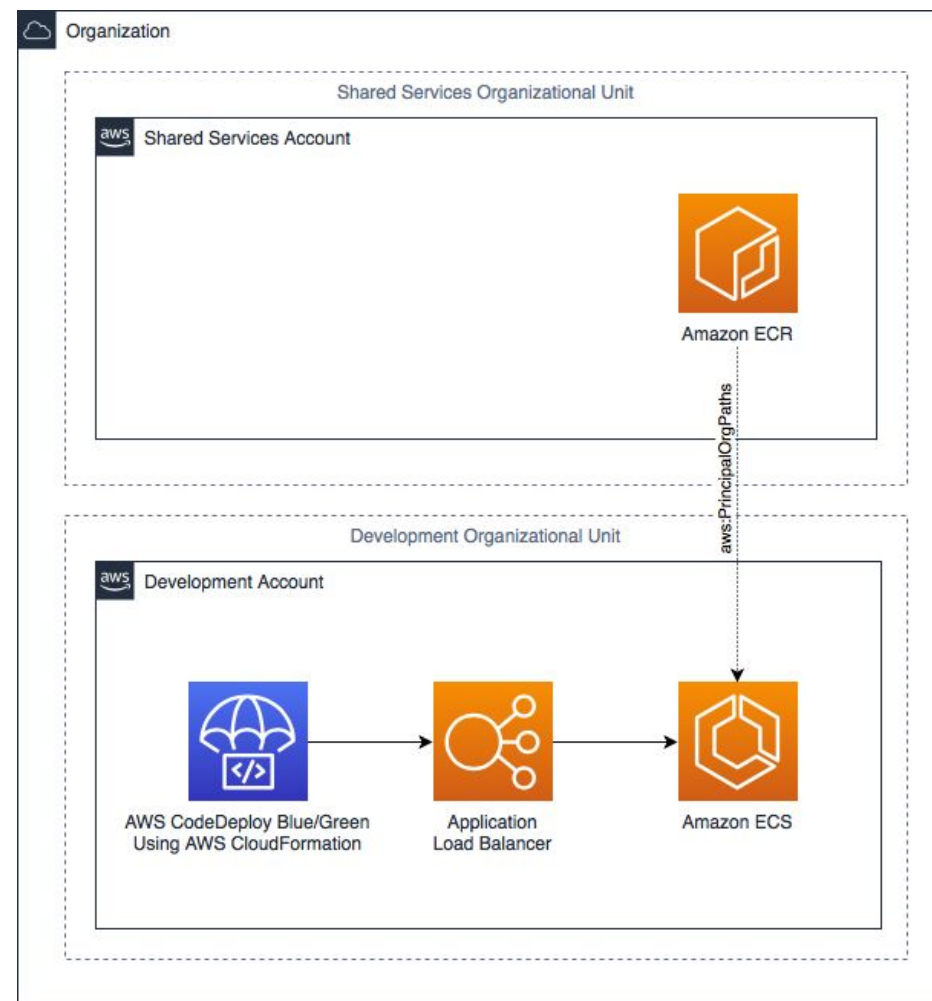
ECS

- **Image** là đơn vị chứa thông tin về source code, các môi trường thiết lập/cài đặt/thư viện cần thiết cho việc khởi chạy, được đóng gói lại.
- **Container** là 1 phiên bản đang chạy của 1 image.



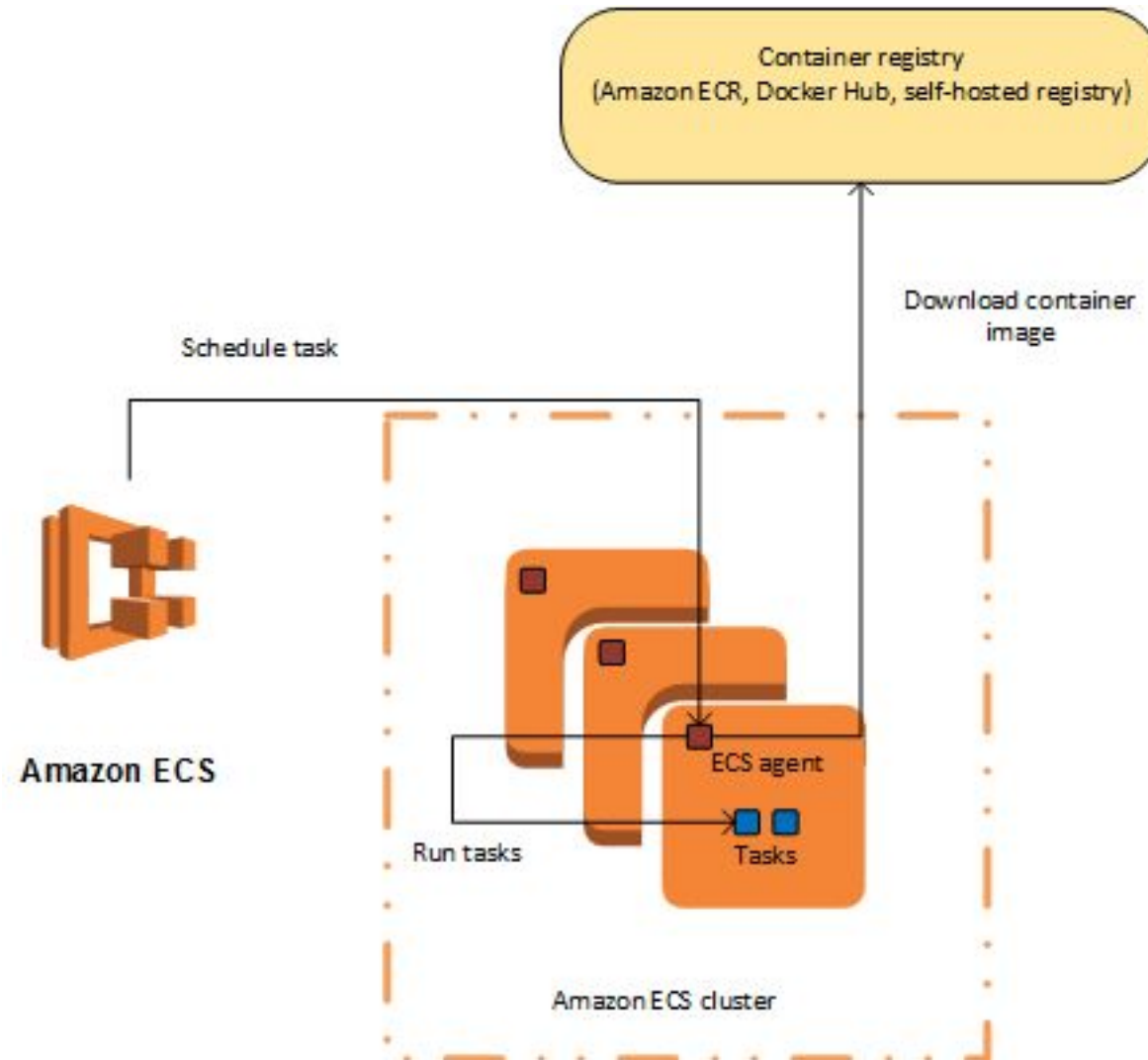
ECS: ECR

- Là nơi lưu trữ các container image (Tương tự Docker hub, Github Container Registry)
- Có thể share container nội bộ, hoặc public
- Tích hợp với các dịch vụ của AWS như IAM, ...



ECS: Service

- Là những thiết lập giúp duy trì lượng instance nhất định cho các tasks
- Vai trò giống như Auto Scaling groups nhưng dành cho các Container là các Tasks.



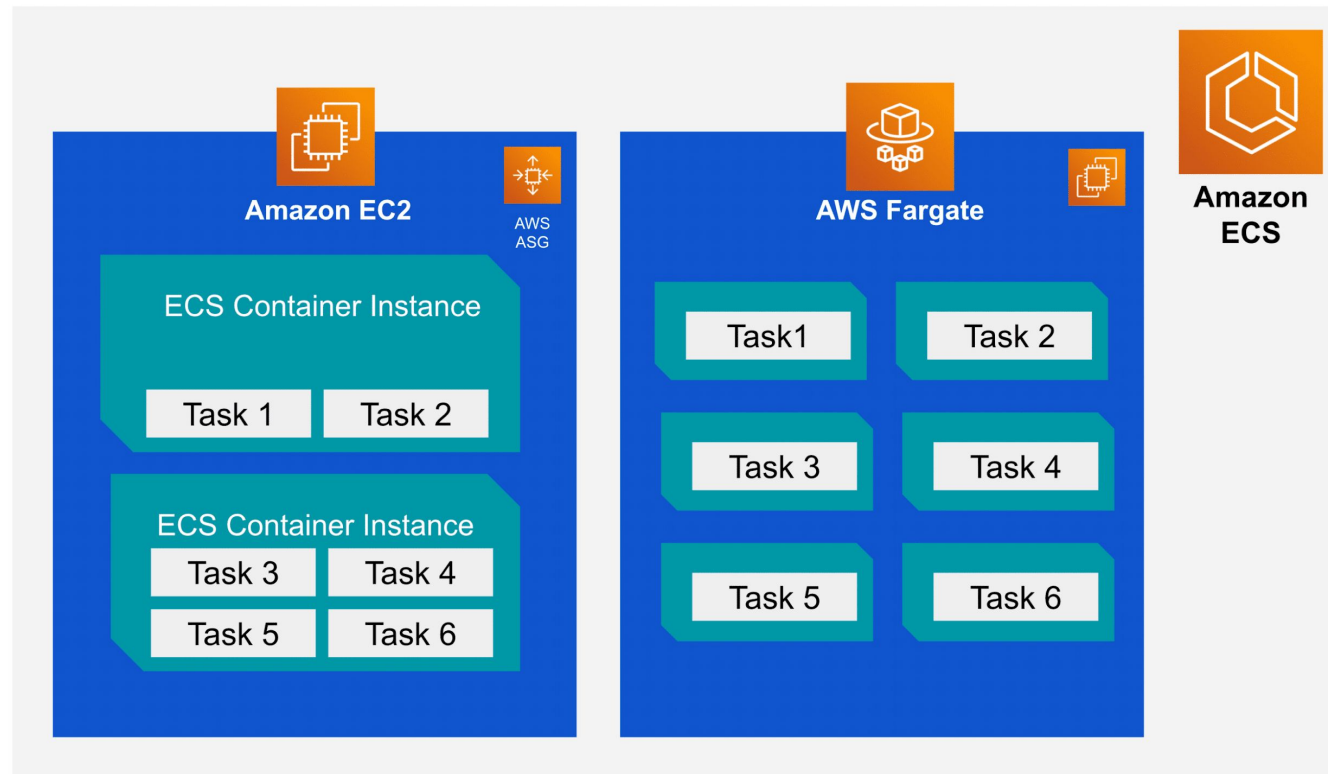
ECS: Task definitions

- Là những thiết lập cho các container
- Dạng text (JSON format)
- Mô tả từ 1 đến 10 (max) container cho 1 tác vụ cụ thể.

```
{
  "containerDefinitions": [
    {
      ...
    }
  ],
  "cpu": "256",
  "executionRoleArn":
    "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
  "family": "fargate-task-definition",
  "memory": "512",
  "networkMode": "awsvpc",
  "requiresCompatibilities": [
    "FARGATE"
  ]
}
```

ECS: Model

- ECS Có 2 launch type chính là Fargate và EC2





THANK YOU