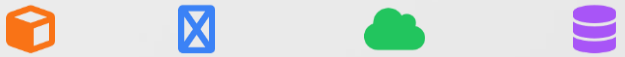
Container trên AWS

Kiến trúc, triển khai và quản lý container trên nền tảng AWS



Giới thiệu Containers on AWS

Container là gì

- Đơn vị phần mềm tiêu chuẩn hóa, đóng gói mã ứng dụng và tất cả các phụ thuộc của nó
- Đảm bảo ứng dụng chạy nhanh chóng và đáng tin cậy từ môi trường này sang môi trường khác
- Cô lập ứng dụng khỏi môi trường xung quanh, bao gồm cả các ứng dụng khác

Lợi ích khi chạy container trên AWS

- **Tính di động cao:** Ứng dụng chạy nhất quán trên mọi môi trường
- **Hiệu quả tài nguyên:** Chia sẻ tài nguyên hệ điều hành, nhẹ hơn máy ảo
- **Triển khai nhanh chóng:** Khởi động nhanh, dễ dàng mở rộng
- **Cô lập ứng dụng:** Tăng cường bảo mật và ổn định
- **Giảm chi phí vận hành:** Ít công việc bảo trì hơn, hành vi dự đoán được

Các dịch vụ chính

Amazon ECS
Nền tảng container riêng của AWS

Amazon EKS
Dịch vụ Kubernetes được quản lý

AWS Fargate
Nền tảng container serverless

Amazon ECR
Dịch vụ lưu trữ Docker images

What is Docker?

Nền tảng deploy

Docker là nền tảng phát triển phần mềm để deploy ứng dụng.

Đóng gói ứng dụng

Ứng dụng được đóng gói trong container chạy được trên mọi hệ điều hành.

Tính đồng nhất

Chạy đồng nhất trên mọi môi trường, không còn vấn đề tương thích.

Hành vi và bảo trì

Predictable behavior, ít công việc bảo trì hơn.

Hỗ trợ đa dạng

Hỗ trợ mọi ngôn ngữ, hệ điều hành, công nghệ.

Trường hợp sử dụng

- Kiến trúc microservices
- Di chuyển ứng dụng (lift-and-shift) lên AWS

"Docker hóa việc triển khai ứng dụng, đảm bảo chạy ổn định trên mọi môi trường"

Docker on an OS – Docker Repositories

Docker Images

Docker images được lưu trữ tập trung trong các Docker Repositories

- Tìm kiếm và kéo (pull) images từ repository
- Push images lên repository
- Hỗ trợ versioning cho images

Docker Hub

Là một public repository phổ biến

- Chứa nhiều base image sẵn có như Ubuntu, MySQL, Nginx, v.v.
- Cung cấp một thư viện lớn các image công khai cho cộng đồng Docker
- Một trong những repository phổ biến nhất

Amazon ECR

Là dịch vụ registry của AWS

- Hỗ trợ cả private và public repositories
- Cho phép lưu trữ, quản lý và triển khai các Docker container images một cách an toàn
- Tích hợp sâu với các dịch vụ AWS khác như Amazon ECS và Amazon EKS

ECR Public Gallery

- Kho lưu trữ công khai của Amazon ECR

Docker vs Virtual Machines

Docker Containers

- Công nghệ "gần giống" ảo hóa nhưng nhẹ hơn
- Chạy đồng nhất trên mọi môi trường
- Predictable behavior, ít công việc bảo trì hơn
- Hỗ trợ mọi ngôn ngữ, hệ điều hành, công nghệ

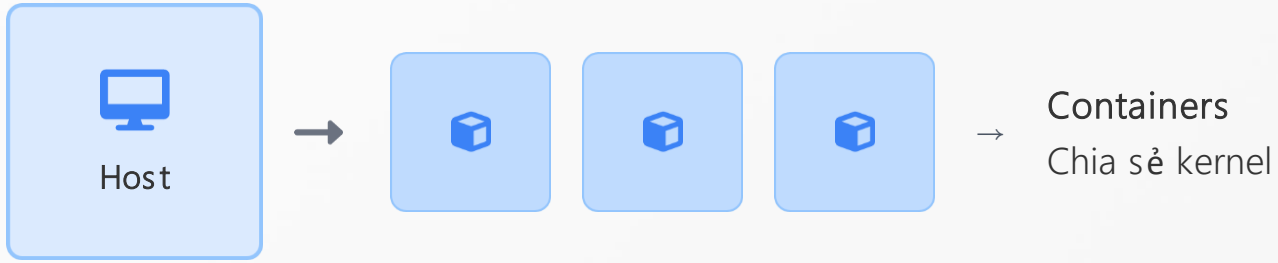
Virtual Machines

- Máy ảo đầy đủ, đóng gói toàn bộ hệ điều hành
- Cần nhiều tài nguyên hơn để chạy
- Cần cấu hình riêng biệt cho từng môi trường
- Hỗ trợ đa nền tảng nhưng phức tạp hơn

Chia sẻ tài nguyên

Containers: Chia sẻ tài nguyên với host, cho phép chạy nhiều container trên cùng một server

VMs: Cần tài nguyên riêng biệt cho từng máy ảo, hiệu quả kém hơn về mặt tài nguyên



Docker Containers Management on AWS

Amazon ECS

- Nền tảng container riêng của AWS
- Elastic Container Service
- Dễ dàng triển khai và quản lý các ứng dụng containerized

Amazon EKS

- Managed Kubernetes service
- Kubernetes được quản lý hoàn toàn trên AWS
- Tích hợp tốt với các dịch vụ AWS khác

AWS Fargate

- Nền tảng container serverless
- Serverless container platform
- AWS quản lý hoàn toàn hạ tầng

Amazon ECR

- Lưu trữ container images
- Elastic Container Registry
- Tích hợp chặt chẽ với các dịch vụ container khác của AWS

ECS Use Cases & Advantages

Trường hợp sử dụng

Microservices

ECS là nền tảng lý tưởng để triển khai và quản lý các ứng dụng theo kiến trúc microservices, cho phép phát triển và triển khai độc lập các thành phần nhỏ.

Batch jobs

ECS rất phù hợp để chạy các công việc xử lý theo lô (batch jobs) hoặc các tác vụ tính toán cường độ cao, tận dụng khả năng mở rộng và quản lý tài nguyên hiệu quả.

Lợi ích

Cô lập lỗi

Mỗi container chạy độc lập, giúp cô lập lỗi. Nếu một container gặp sự cố, nó sẽ không ảnh hưởng đến các container khác trong cùng một ứng dụng hoặc dịch vụ.

Tăng bảo mật

ECS cung cấp các tính năng bảo mật mạnh mẽ thông qua IAM roles, Security Groups và tích hợp với các dịch vụ bảo mật khác của AWS, giúp bảo vệ các ứng dụng container hóa.

Mở rộng độc lập

Các dịch vụ và tác vụ trong ECS có thể được mở rộng (scaling) một cách độc lập dựa trên nhu cầu tải, tối ưu hóa việc sử dụng tài nguyên và hiệu suất.

Dễ rollback

ECS cho phép dễ dàng rollback về các phiên bản trước của ứng dụng trong trường hợp có vấn đề sau khi triển khai, đảm bảo tính ổn định của hệ thống.

ECS Cluster & Task Concepts

Cluster

Cluster là một nhóm logic các services và tasks.

- Quản lý tập trung các resource(container)
- Có thể có nhiều node khác nhau

Task

Task là đơn vị công việc cơ bản trong ECS, bao gồm một hoặc nhiều containers.

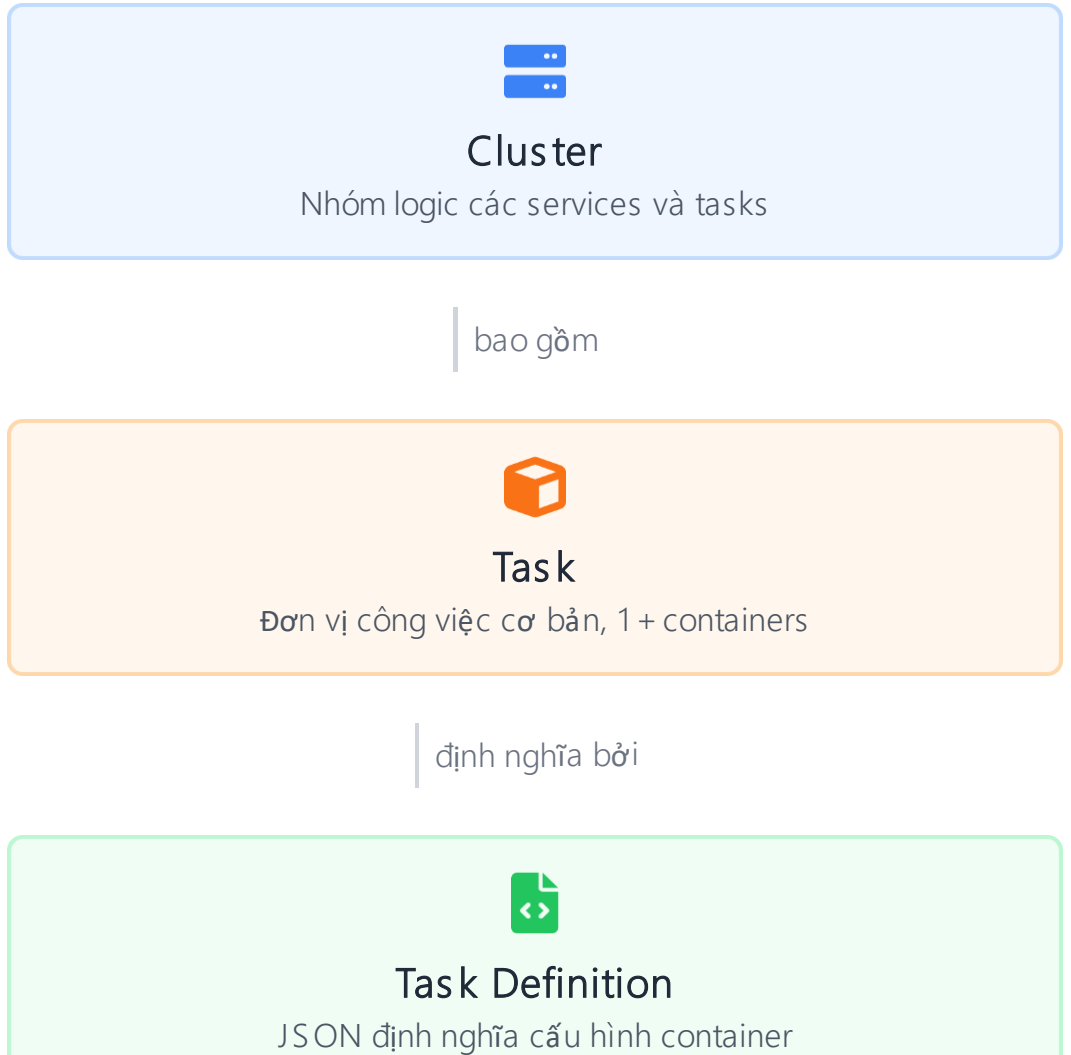
- Chạy cùng nhau trên cùng một EC2 instance
- Cùng chia sẻ network và storage

Task Definition

Task Definition là một tệp JSON định nghĩa cấu hình cho container.

- Image Docker
- Cổng (port)
- Giới hạn CPU/RAM
- Biến môi trường (env vars)

Mối quan hệ giữa các khái niệm



ECS Launch Types

ECS hỗ trợ ba loại khởi chạy khác nhau, mỗi loại có những đặc điểm và use case riêng:

EC2 Launch Type

- Chạy các container trên các phiên bản EC2 do người dùng quản lý
- Yêu cầu người dùng tự quản lý hạ tầng cơ bản (máy chủ, hệ điều hành, vá lỗi)
- Cung cấp quyền kiểm soát cao nhất đối với môi trường chạy container

Fargate Launch Type

- Nền tảng container serverless, AWS quản lý hoàn toàn hạ tầng
- Người dùng chỉ cần định nghĩa Task Definition và AWS sẽ cấp phát, quản lý tài nguyên
- Loại bỏ gánh nặng quản lý máy chủ, giúp tập trung vào phát triển ứng dụng

External Launch Type

- Cho phép chạy các tác vụ ECS trên các máy chủ bên ngoài AWS
- Có thể là các máy chủ on-premise hoặc các máy chủ trên các nhà cung cấp đám mây khác
- AWS chỉ quản lý các tasks, người dùng chịu trách nhiệm quản lý hạ tầng máy chủ

So sánh tổng thể

EC2:

- **Quản lý hạ tầng:** Người dùng
- **Quản lý tasks:** AWS
- **Phù hợp:** Kiểm soát cao

Fargate:

- **Quản lý hạ tầng:** AWS
- **Quản lý tasks:** AWS
- **Phù hợp:** Serverless

External:

- **Quản lý hạ tầng:** Người dùng
- **Quản lý tasks:** AWS
- **Phù hợp:** Hybrid

ECS Agent & IAM Roles

ECS Agent

ECS Agent được cài đặt trên các EC2 instance

Quản lý vòng đời của container trên các instance đó

EC2 Instance Profile là một IAM Role được gán cho EC2 instance

Cho phép ECS Agent gọi các API của AWS
Gửi log từ container đến CloudWatch
Kéo (pull) các Docker image từ Amazon ECR

ECS Task Role

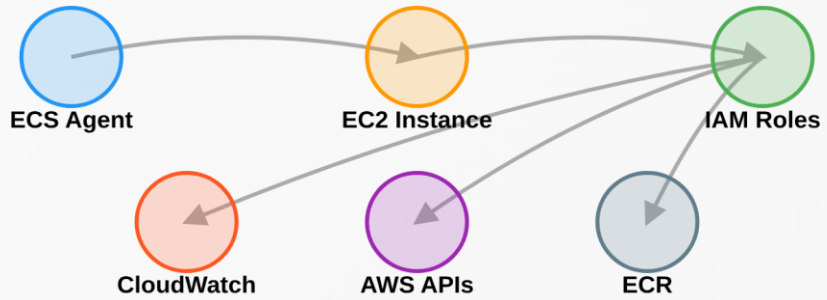
ECS Task Role là một IAM Role được gán cho từng Task cụ thể

Cho phép mỗi Task có các quyền riêng biệt để tương tác với các dịch vụ AWS khác

Ví dụ:

Task A: Quyền truy cập S3

Task B: Quyền truy cập DynamoDB



1. ECS Agent cài trên EC2

ECS Fargate & External Launch Types

Fargate Launch Type

- **Hoàn toàn serverless:** Loại bỏ nhu cầu quản lý máy chủ
- **Định nghĩa Task đơn giản:** AWS sẽ lo phần còn lại
- **Quản lý tài nguyên tự động:** Bao gồm CPU và RAM
- **Phù hợp cho:** Ứng dụng không cần kiểm soát sâu vào hạ tầng

Fargate là lựa chọn tốt nhất cho các ứng dụng microservices và không cần quyền kiểm soát cao vào hạ tầng

External Launch Type

- **Quản lý máy chủ tự do:** Bạn tự quản lý máy chủ vật lý hoặc máy ảo
- **AWS chỉ quản lý tasks:** Quản lý vòng đời container
- **Linh hoạt cao:** Lựa chọn và cấu hình hạ tầng theo nhu cầu
- **Thích hợp cho:** Môi trường hybrid hoặc khi cần tận dụng tài nguyên hiện có

External là lựa chọn tốt nhất khi bạn cần kiểm soát cao vào hạ tầng hoặc đang di chuyển ứng dụng từ môi trường khác

So sánh	Fargate	External
Quản lý hạ tầng	AWS	Người dùng
Tính linh hoạt	Thấp	Cao
Phù hợp với ứng dụng	Microservices, serverless	Môi trường hybrid, ứng dụng cần kiểm soát cao

ECS Service & Cân bằng tải

ECS Service

Giám sát & Đảm bảo

Giám sát và đảm bảo các task luôn chạy theo cấu hình mong muốn

Tự động khôi phục

Tự động thay thế các task bị lỗi hoặc dừng hoạt động

Triển khai & Cập nhật

Quản lý việc triển khai và cập nhật ứng dụng

Loại cân bằng tải

ALB (Application Load Balancer)

- Loại Load Balancer phổ biến nhất
- Hoạt động ở tầng ứng dụng (Layer 7)
- Hỗ trợ định tuyến dựa trên nội dung, SSL termination, và cân bằng tải cho các microservices

NLB (Network Load Balancer)

- Cung cấp hiệu suất cao và độ trễ thấp
- Hoạt động ở tầng mạng (Layer 4)
- Lý tưởng cho các ứng dụng yêu cầu hiệu suất cực cao và hỗ trợ PrivateLink

CLB (Classic Load Balancer)

- Loại Load Balancer cũ hơn
- Cung cấp ít tính năng hơn so với ALB và NLB
- Thường được khuyến nghị thay thế bằng ALB hoặc NLB cho các ứng dụng mới

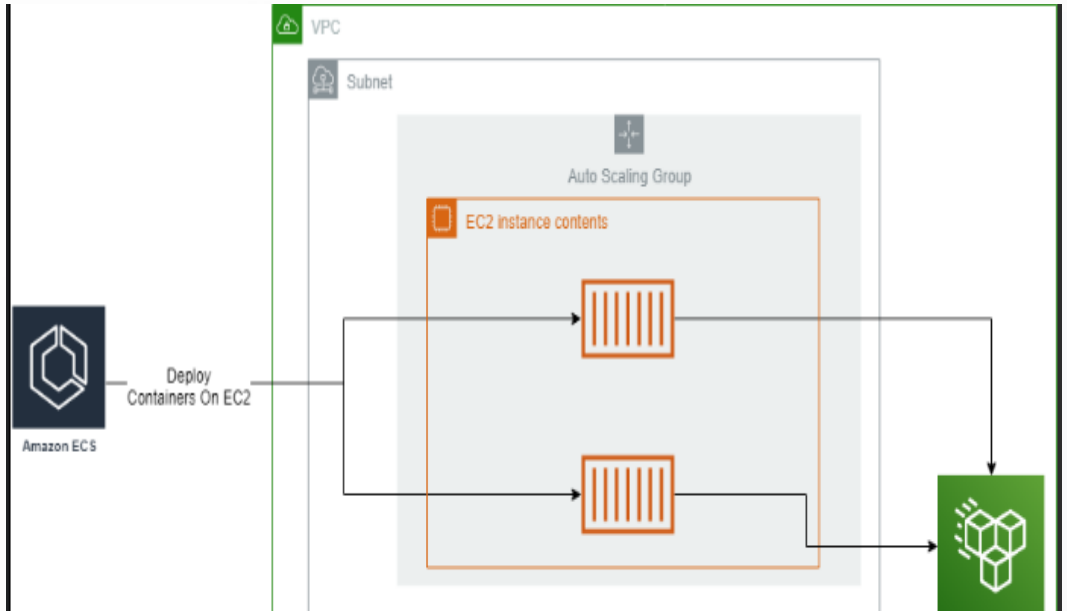
ECS Data Volumes (EFS)

Mount EFS vào ECS Tasks

Mount EFS vào ECS tasks
Cho phép chia sẻ dữ liệu giữa các container

Hỗ trợ cả EC2 và Fargate
Dễ dàng triển khai trên nhiều nền tảng

Lưu ý về S3
Không thể mount S3 như một file system trực tiếp



Lợi ích và Use Cases

Dữ liệu chia sẻ đa AZ
Dữ liệu được lưu trữ trên nhiều Availability Zones

Giải pháp Serverless Storage
Kết hợp EFS và Fargate tạo thành giải pháp Serverless

Use case
Cung cấp giải pháp lưu trữ chia sẻ, bền vững và đa AZ

Kiến trúc ứng dụng
Hỗ trợ các ứng dụng microservices cần chia sẻ dữ liệu

Ví dụ Task Definition:

```
{
  "family": "efs-task",
  "containerDefinitions": [
    {
      "name": "app",
      "image": "nginx",
      "mountPoints": [
        {
```

ECS Auto Scaling

Chức năng

Tự động điều chỉnh số lượng task

- Tăng số lượng task khi nhu cầu tải tăng
- Giảm số lượng task khi nhu cầu tải giảm
- Đáp ứng tự động nhu cầu lưu lượng
- Tối ưu hóa việc sử dụng tài nguyên

Metric

CPU

Sử dụng mức độ sử dụng CPU của các task

RAM

Dựa trên mức độ sử dụng bộ nhớ RAM

ALB Request Count

Số lượng yêu cầu đến từ Application Load Balancer

Loại scaling

Target Tracking Scaling

Duy trì một metric ở mức mục tiêu cụ thể (ví dụ: giữ CPU ở 70%)

Step Scaling

Điều chỉnh số lượng task theo các bước được định nghĩa dựa trên ngưỡng của metric

Scheduled Scaling

Tăng hoặc giảm số lượng task theo lịch trình đã định trước

Ghi chú về Fargate: Fargate dễ dàng thiết lập Auto Scaling hơn do bản chất serverless, AWS quản lý hoàn toàn hạ tầng bên dưới.

EC2 Launch Type Auto Scaling

Cơ chế hoạt động

Auto Scaling Group (ASG)
Dựa trên ASG để tự động điều chỉnh số lượng instance EC2

Metric theo dõi
CPU, RAM, mạng, và các metric tùy chỉnh khác

Scale in/out
Tự động thêm hoặc bớt instance dựa trên nhu cầu

ECS Cluster Capacity Provider

Vai trò
Hỗ trợ thêm các instance EC2 vào cluster khi phát hiện thiếu tài nguyên

Capacity Provider
Quản lý tài nguyên cluster một cách linh hoạt

Cân bằng tải
Đảm bảo phân phối đều workload trên các instance

Auto Scaling Group
Tự động điều chỉnh số lượng instance

ECS Cluster
Nhiệm vụ: Quản lý container

EC2 Instances
Máy chủ ảo chạy container

ECS Rolling Updates

Kiểm soát quá trình cập nhật

Kiểm soát số lượng task chạy song song

ECS cho phép bạn kiểm soát số lượng task cũ và mới chạy cùng lúc trong quá trình cập nhật

Cấu hình Min/Max

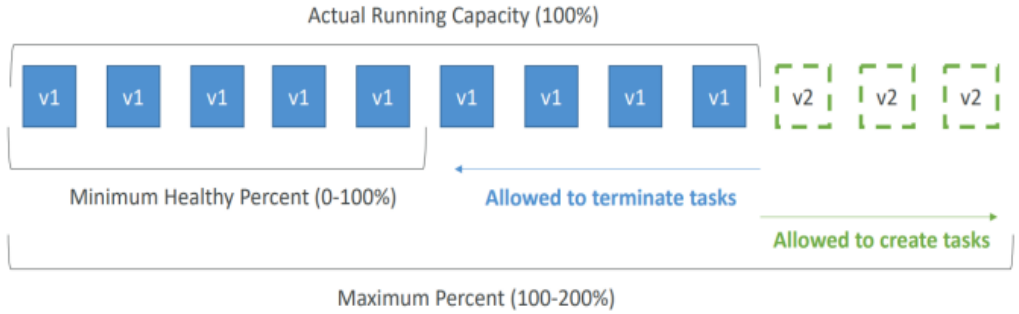
Bạn có thể chỉ định:

- Min: Tỷ lệ phần trăm task cũ vẫn chạy trong khi cập nhật
- Max: Tỷ lệ phần trăm task mới được triển khai so với tổng số task

Ví dụ 1: Min 50% - Max 100%

Đảm bảo ít nhất 50% task cũ vẫn chạy trong khi tối đa 100% task mới được triển khai

Quá trình cập nhật cuốn chiếu



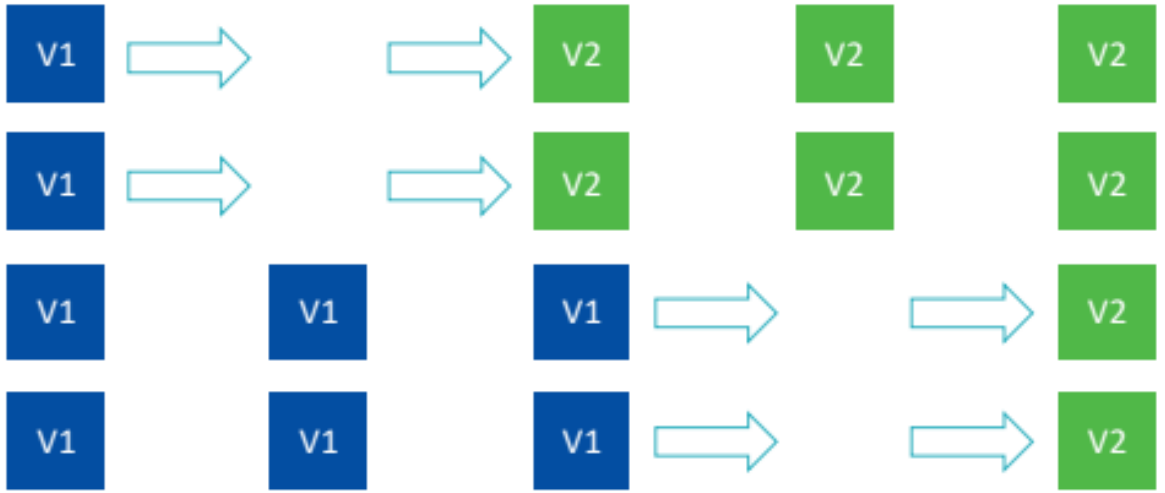
Ví dụ 2: Min 100% - Max 150%

Đảm bảo tất cả task cũ vẫn chạy cho đến khi task mới sẵn sàng, sau đó mở rộng lên đến 150% tổng số task trong quá trình chuyển đổi

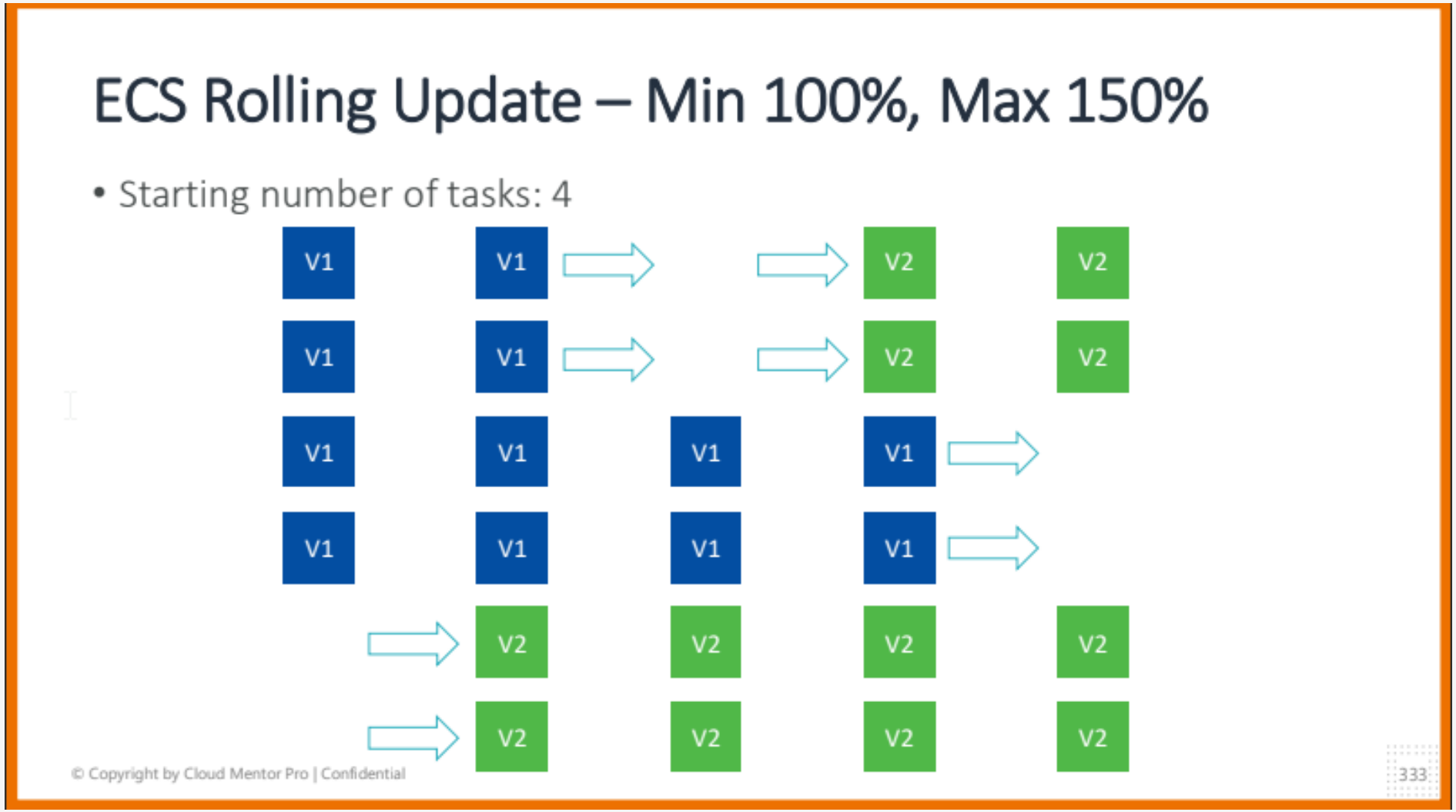
ECS Rolling Updates

ECS Rolling Update – Min 50%, Max 100%

- Starting number of tasks: 4



ECS Rolling Updates



ECS Integration Examples

ECS có thể tích hợp với nhiều dịch vụ AWS khác nhau để tạo thành giải pháp toàn diện cho việc xử lý ứng dụng containerized.

EventBridge (Invoke)

→ ECS task có thể được kích hoạt bởi EventBridge

- Kích hoạt dựa trên các sự kiện cụ thể
- Hỗ trợ cấu hình theo lịch trình định sẵn

Use case: Triển khai tự động ứng dụng khi có file mới上传 lên S3

SQS Queue

→ ECS kết hợp với SQS Queue để xử lý batch message

- Hỗ trợ xử lý các thông điệp theo lô
- Giúp quản lý và xử lý các tác vụ không đồng bộ

Use case: Xử lý hàng loạt đơn hàng trong hệ thống thương mại điện tử

EventBridge (Detect)

→ Dùng EventBridge để phát hiện khi ECS task bị dừng

- Cho phép tự động hóa các hành động phản ứng
- Hỗ trợ gửi thông báo khi có sự cố

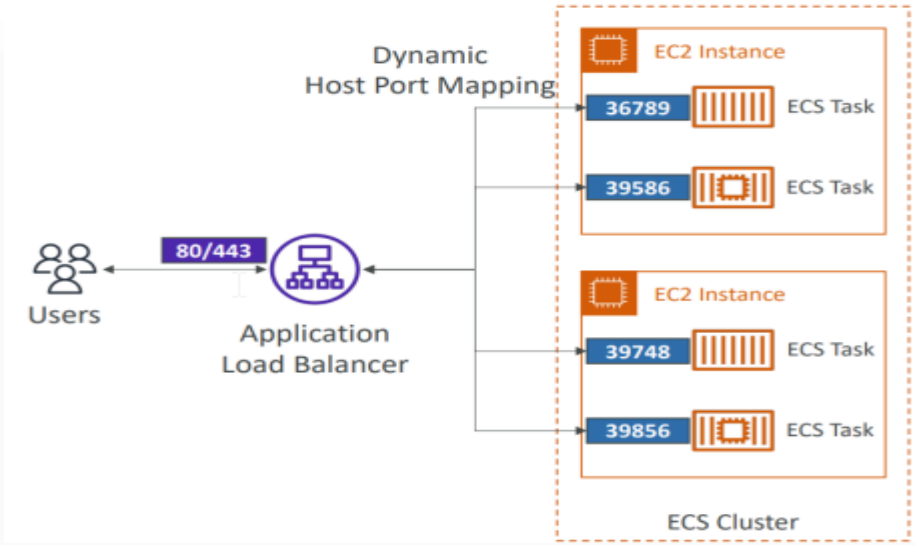
Use case: Tự động khôi phục hoặc cảnh báo khi task thất bại

ECS Load Balancing Details

EC2 Launch Type

Dynamic Host Port Mapping
Cho phép nhiều bản sao của cùng một service chạy trên cùng một EC2 instance

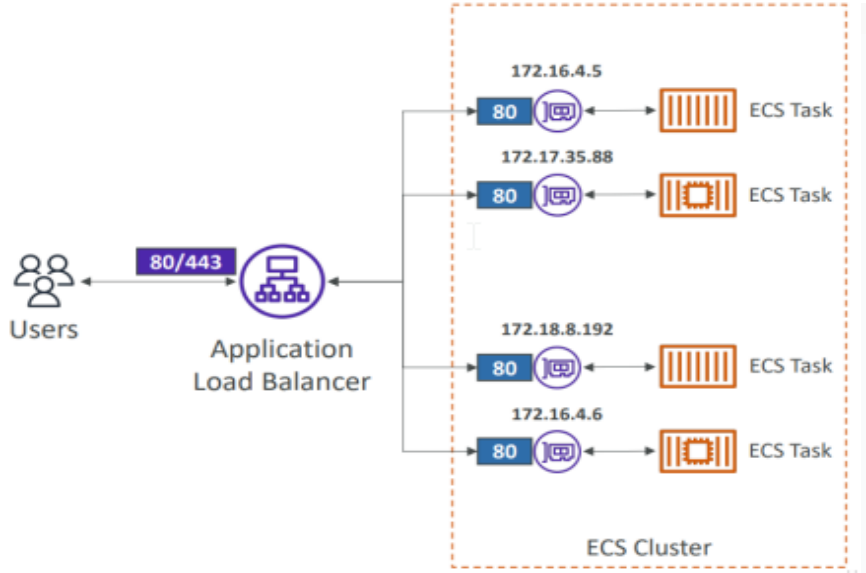
Port Mapping
Load Balancer sẽ ánh xạ lưu lượng truy cập đến các cổng động này



Fargate

Private IP Address
Mỗi task có private IP riêng

Simplified Network Configuration
Không cần quản lý cổng trên host



Cấu hình Security Group

ECS Environment Variables

Hardcoded

- Sử dụng cho các biến môi trường **không nhạy cảm** và **ít thay đổi**
- Ví dụ: URL của một dịch vụ công cộng
- Định nghĩa trực tiếp trong Task Definition

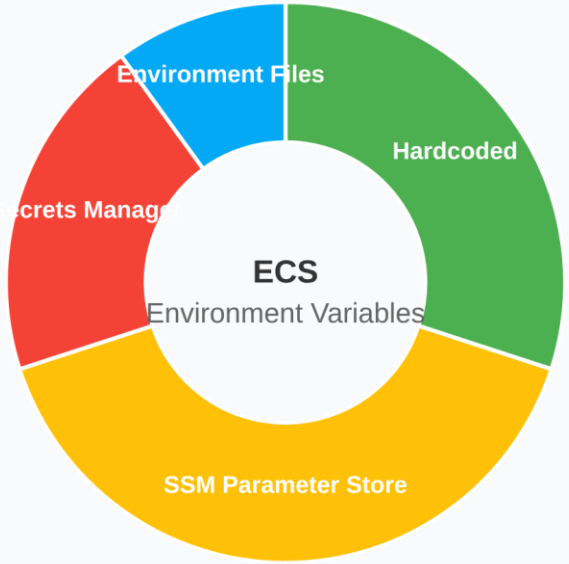
Biến nhạy cảm

- Sử dụng **SSM Parameter Store** để lưu trữ các tham số cấu hình
- Sử dụng **Secrets Manager** để quản lý các thông tin nhạy cảm như khóa API, mật khẩu
- Đảm bảo bảo mật và kiểm soát truy cập chặt chẽ

Tệp môi trường

- Cho phép tải hàng loạt biến môi trường từ một tệp
- Tệp môi trường có thể được lưu trữ trên S3
- Hữu ích khi có nhiều biến cần cấu hình cho một task

Mối quan hệ giữa các phương pháp



- Hardcoded: Dễ dàng nhưng kém bảo mật
- SSM Parameter Store: Cấu hình linh hoạt
- Secrets Manager: Bảo mật cao cho biến nhạy cảm

Amazon ECR

Định nghĩa & Mục đích

Elastic Container Registry

Dịch vụ lưu trữ và quản lý các Docker image một cách an toàn và có khả năng mở rộng

Kiểu repository

Public repo

Kho lưu trữ công khai, hỗ trợ các image phổ biến

Private repo

Kho lưu trữ riêng tư, bảo mật các image của bạn

Tích hợp & Lưu trữ

Tích hợp chặt chẽ với AWS

Tight integration với Amazon ECS và các dịch vụ container khác của AWS

Lưu trữ trên S3

Images được lưu trữ trên Amazon S3, đảm bảo độ bền và tính sẵn sàng cao

Kiểm soát truy cập

Kiểm soát quyền truy cập thông qua AWS Identity and Access Management (IAM)

Tính năng khác

- Quét lỗ hổng bảo mật
- Versioning
- Lifecycle policy
- Public Gallery

Amazon EKS Overview

Elastic Kubernetes Service

Định nghĩa

Dịch vụ Kubernetes được quản lý hoàn toàn trên AWS

Kubernetes

Hệ thống mã nguồn mở dùng để tự động hóa việc triển khai, mở rộng và quản lý các ứng dụng được đóng gói (containerized applications)

Worker Node

EKS cho phép sử dụng các instance EC2 hoặc AWS Fargate làm worker node để chạy các container của bạn

Trường hợp sử dụng

Môi trường đa đám mây

Lý tưởng cho các công ty đã sử dụng Kubernetes tại chỗ (on-premise) hoặc trong môi trường đa đám mây (multi-cloud)

Mở rộng sang AWS

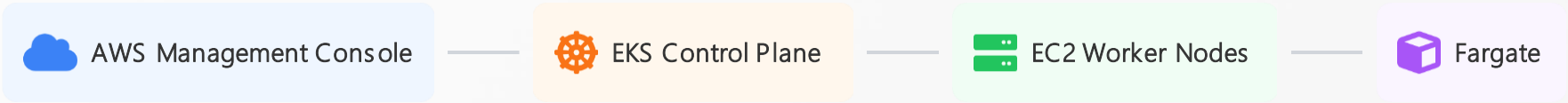
Các đội ngũ muốn di chuyển hoặc mở rộng Kubernetes sang AWS

Cloud-agnostic

Nền tảng Cloud-agnostic, cho phép triển khai nhiều cluster theo từng khu vực (region)

Giám sát

Việc giám sát hiệu suất và tình trạng của các container trên EKS được thực hiện thông qua CloudWatch Container Insights



Tổng kết

Docker

Giúp đóng gói và triển khai ứng dụng một cách đồng nhất, đảm bảo ứng dụng chạy ổn định trên mọi môi trường

ECS & Fargate

Giải pháp container hóa đơn giản, dễ sử dụng và tích hợp sâu với hệ sinh thái AWS

EKS

Lựa chọn lý tưởng cho các đội ngũ đã quen thuộc với Kubernetes, mang lại khả năng quản lý Kubernetes trên AWS

ECR

Kho lưu trữ image an toàn, đáng tin cậy và tích hợp chặt chẽ với các dịch vụ container khác của AWS

Hệ sinh thái AWS Container

- AWS cung cấp một hệ sinh thái toàn diện, hỗ trợ mọi giai đoạn trong vòng đời của container
- Từ phát triển, triển khai đến quản lý và giám sát
- Tích hợp sâu giữa các dịch vụ container với nhau
- Hỗ trợ đa dạng ngôn ngữ, framework và kiến trúc
- Cung cấp giải pháp linh hoạt cho mọi nhu cầu container hóa

Cảm ơn bạn đã xem!