

# Introduction to Amazon ECS and AWS Fargate

Containers Immersion Day: Module 3

# AWS container services landscape

## Management

Deployment, Scheduling,  
Scaling & Management of  
containerized applications



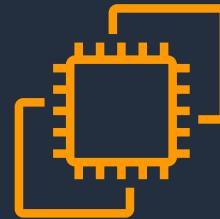
Amazon Elastic  
Container Service



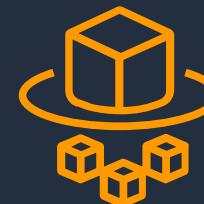
Amazon Elastic  
Kubernetes Service

## Hosting

Where the containers run



Amazon EC2



AWS Fargate

## Image Registry

Container Image Repository



Amazon Elastic  
Container Registry



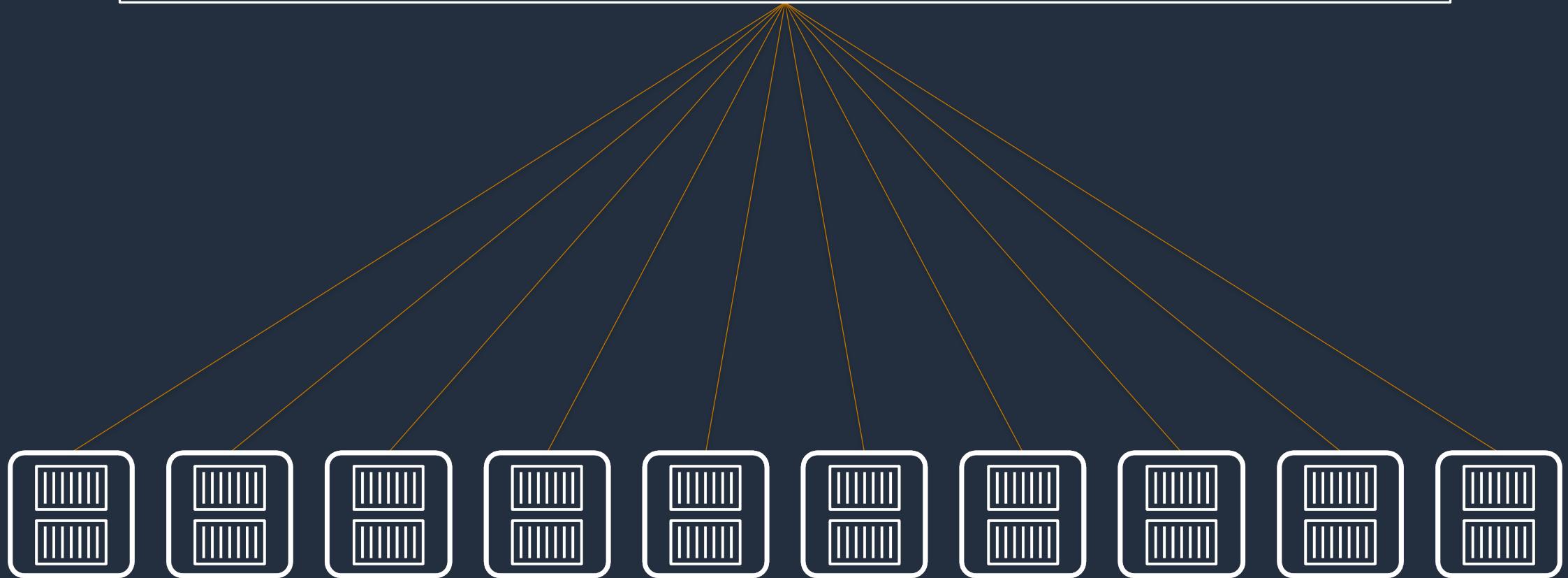
# Amazon Elastic Container Service

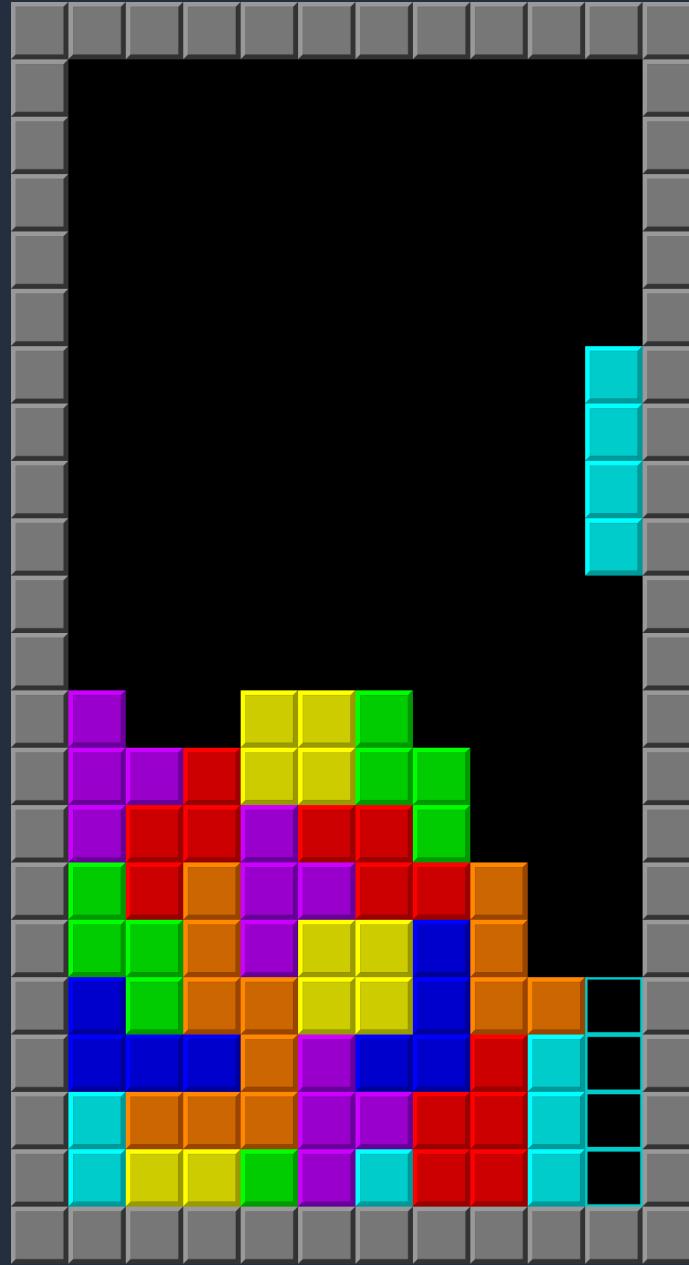


## Scheduling and Orchestration

Cluster Manager

Placement Engine







# Amazon ECS constructs

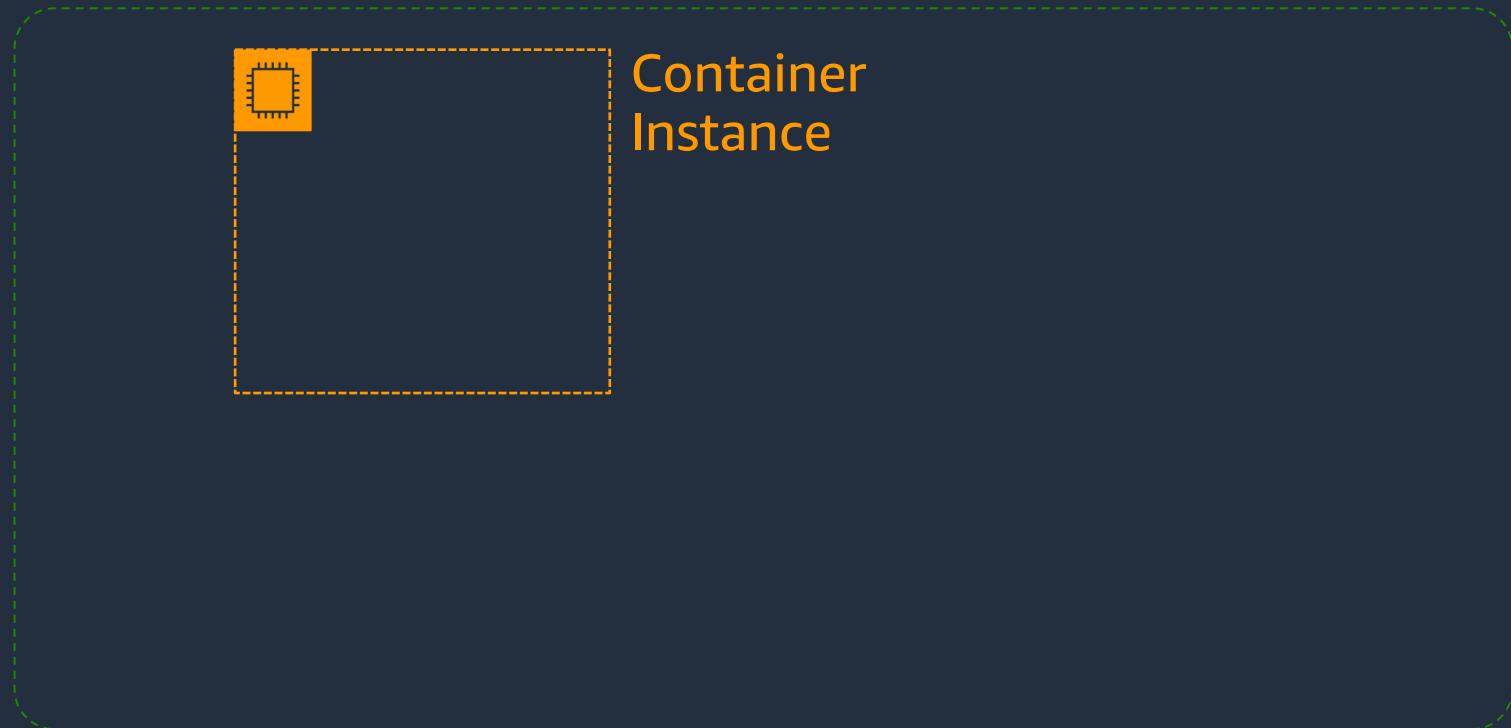
## Cluster

- Resource grouping and isolation
- IAM permissions boundary

# Amazon ECS constructs

## Cluster

- Resource grouping and isolation
- IAM permissions boundary



# Amazon ECS constructs



## Task definition

- Template used by Amazon ECS to launch tasks
- Parallels to docker run parameters
- Defines requirements:
  - CPU/Memory
  - Container image(s)
  - Logging
  - IAM role
  - Etc.

## Cluster

- Resource grouping and isolation
- IAM permissions boundary

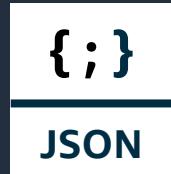


## Container Instance

## Task

- Running instance of a task definition
- One or more containers

# Amazon ECS constructs

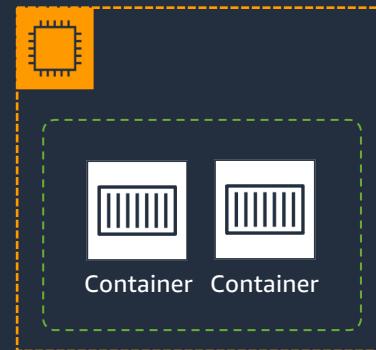


## Task definition

- Template used by Amazon ECS to launch tasks
- Parallels to docker run parameters
- Defines requirements:
  - CPU/Memory
  - Container image(s)
  - Logging
  - IAM role
  - Etc.

## Cluster

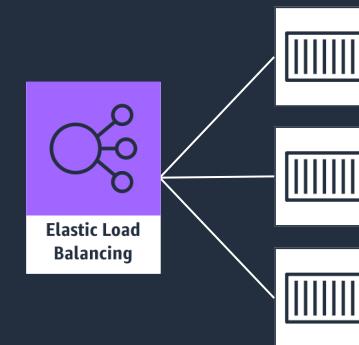
- Resource grouping and isolation
- IAM permissions boundary



## Task

- Running instance of a task definition
- One or more containers

## Container Instance



## Service

- Maintains desired # of running tasks
- Replaces unhealthy tasks
- ELB integration

# Task definition

```
{  
  "containerDefinitions": [  
    {  
      "memory": 128,  
      "portMappings": [  
        {  
          "hostPort": 80,  
          "containerPort": 80,  
          "protocol": "tcp"  
        }  
      ],  
      "essential": true,  
      "name": "nginx-container",  
      "image": "nginx",  
      "logConfiguration": {  
        "logDriver": "awslogs",  
        "options": {  
          "awslogs-group": "ecs-log-streaming",  
          "awslogs-region": "us-west-2",  
          "awslogs-stream-prefix": "fargate-task-1"  
        }  
      },  
      "cpu": 0  
    }  
  ],  
  "networkMode": "awsvpc",  
  "executionRoleArn":  
    "arn:aws:iam::123456789012:role/ecsTask  
ExecutionRole",  
  "memory": "2048",  
  "cpu": "1024",  
  "requiresCompatibilities": [  
    "FARGATE"  
  ],  
  "family": "example_task_1"  
}
```

continued...

# Deploying on ECS: Tasks vs Services

## On-Demand Workloads

ECS task scheduler

Run once or at intervals

Batch jobs

RunTask API

StartTask (custom)

## Long-Running Apps

ECS service scheduler

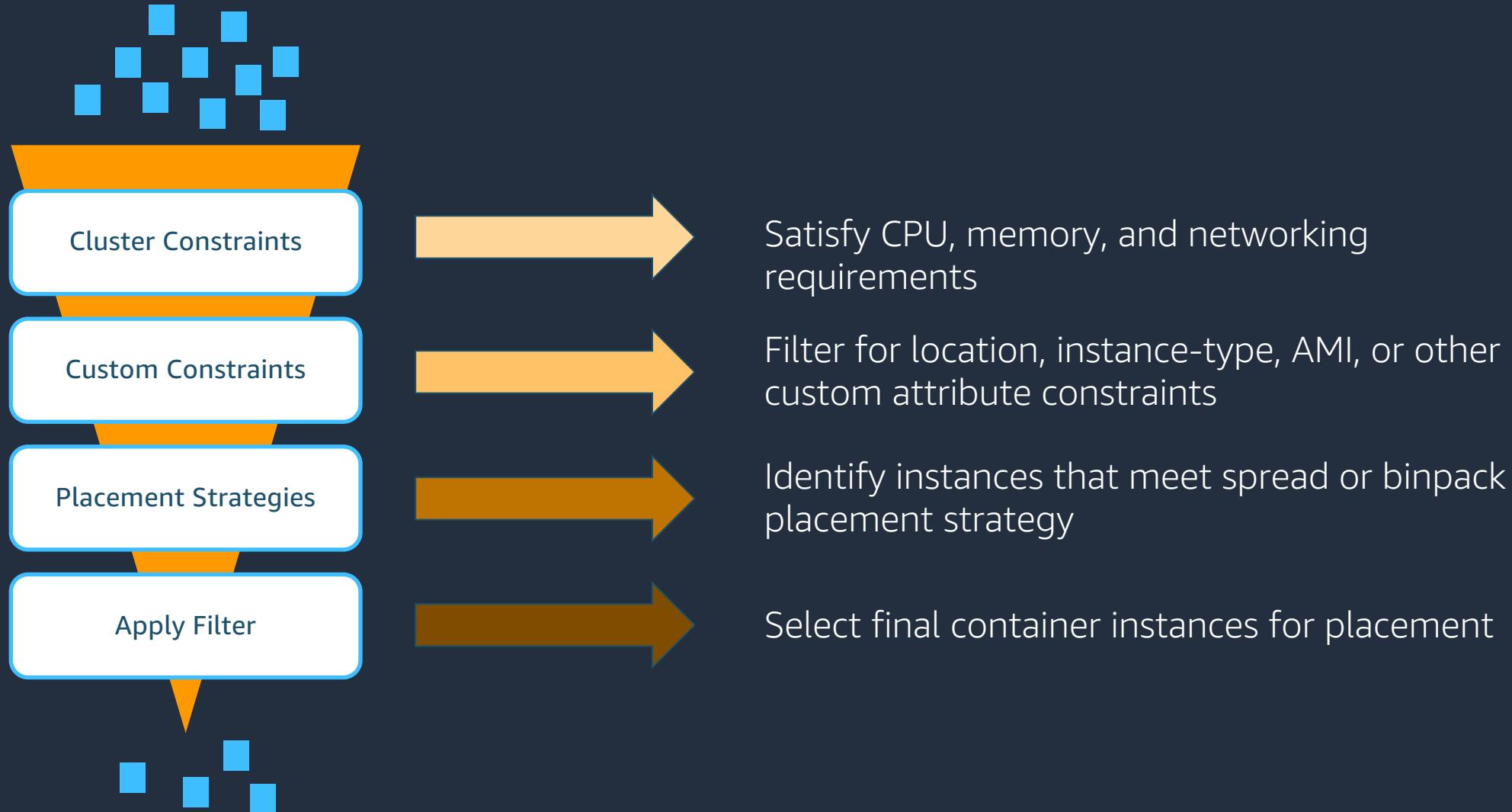
Health management

Scale-up and scale-down

AZ aware

Grouped containers

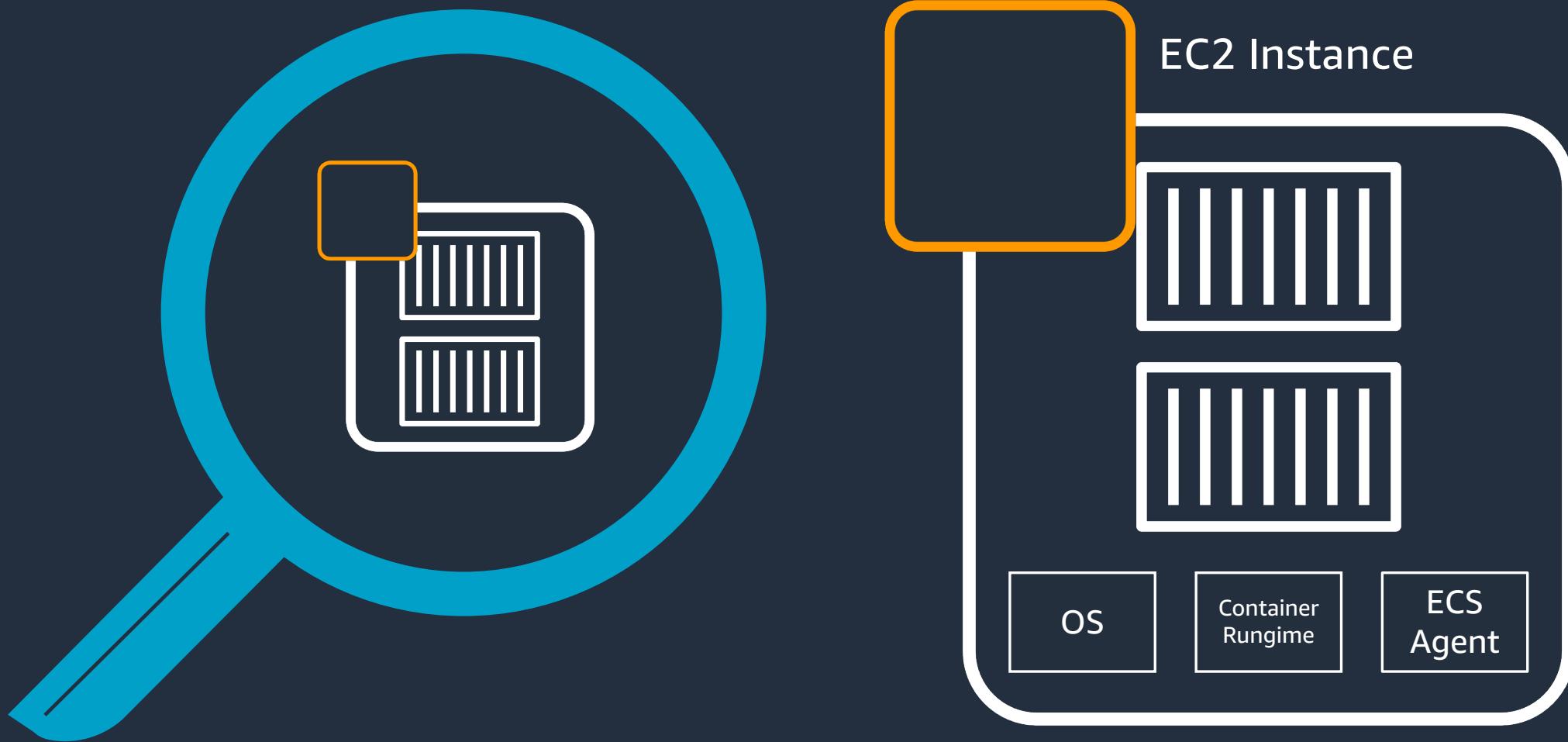
# Task placement





# AWS Fargate

# Without Fargate, you end up managing more than just containers



- Patching and Upgrading OS, agents, etc.
- Scaling the instance fleet for optimal utilization





## Amazon Elastic Container Service





## Amazon Elastic Container Service

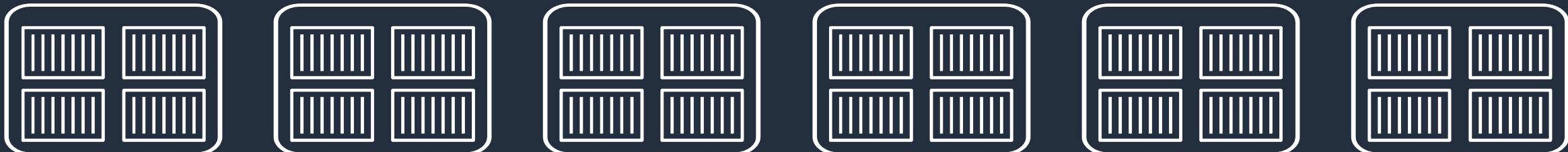


**AWS Fargate**  
run serverless containers

# AWS Fargate platform versions



**Amazon Elastic Container Service**



**AWS Fargate**  
Platform version 1.4.0



Your  
containerized  
applications

## Managed by AWS

No EC2 Instances to provision, scale or manage

## Elastic

Scale up & down seamlessly. Pay only for what you use

## Integrated

With the AWS ecosystem: VPC Networking, Elastic Load Balancing, IAM Permissions, CloudWatch and more

# Fully managed container environment with AWS ECS + Fargate



## Bring existing code

No changes required of existing code, works with existing workflows and microservices built on Amazon ECS



## Production ready

ISO, PCI, HIPAA, SOC compliant.  
Launch ten or tens of thousands of containers in seconds in 9 global regions (+7 in 2018)



## Powerful integrations

Native AWS integrations for networking, security, CI/CD, monitoring, and tracing

---

Fargate runs tens of millions of containers for AWS customers every week

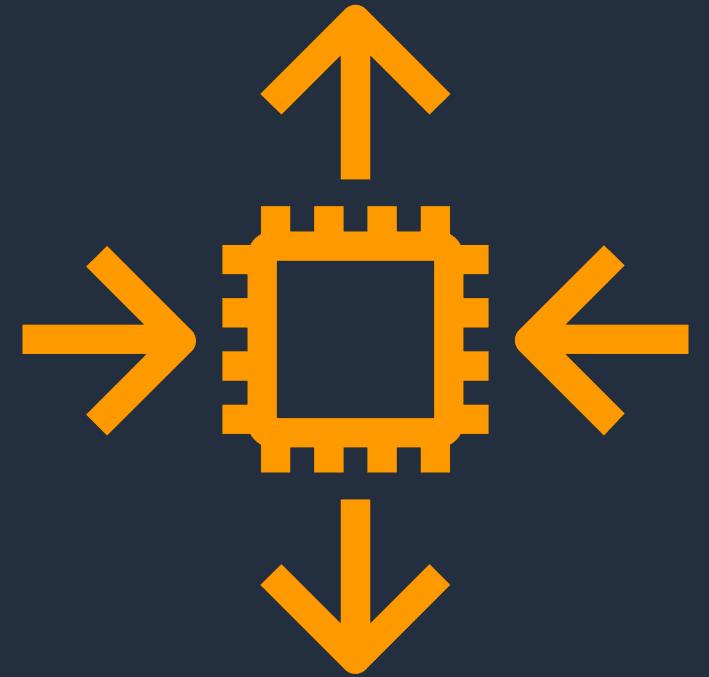
# Fargate launch type: Compute

---

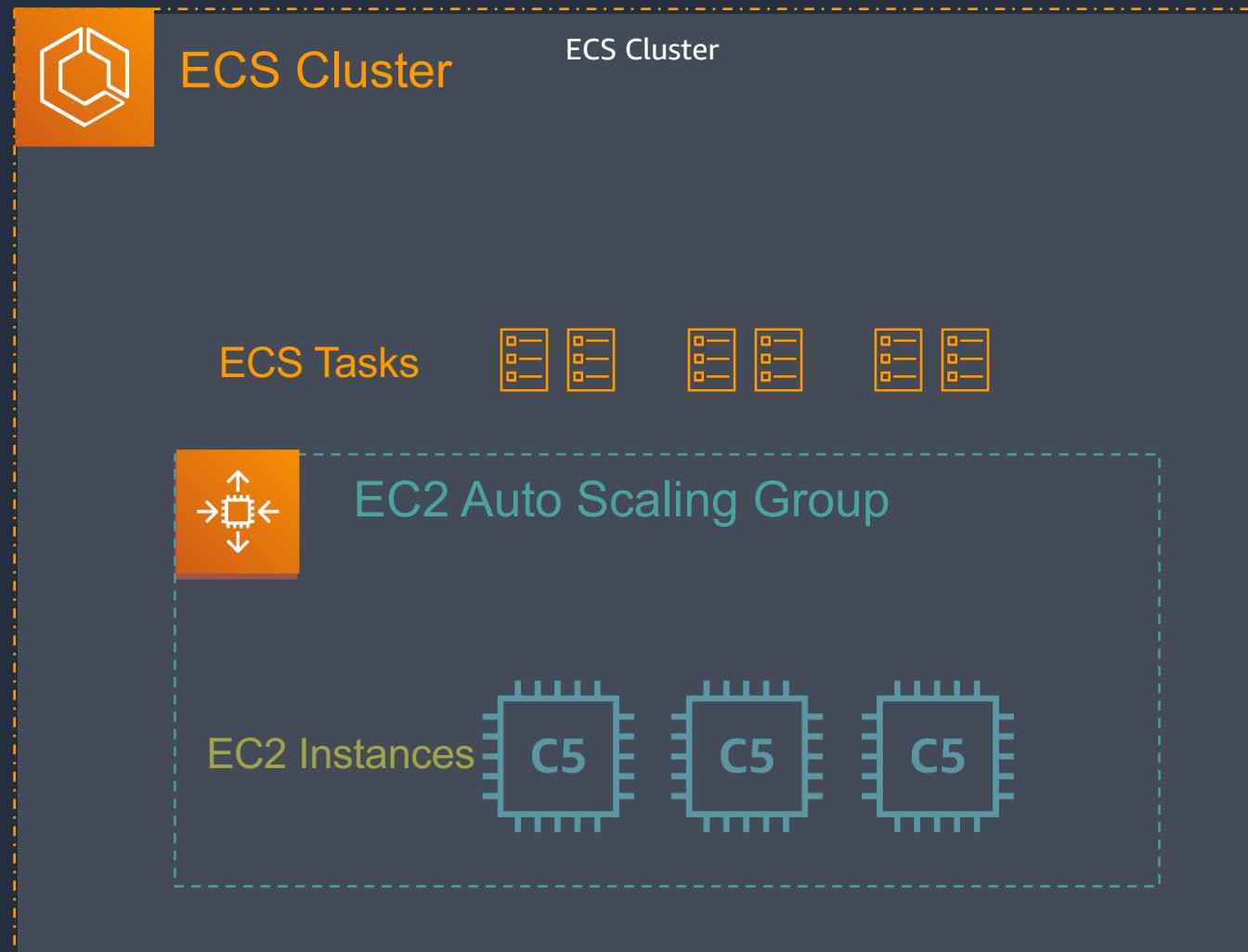
CPU	Memory
256 (.25 vCPU)	512 MB, 1 GB, 2 GB
512 (.5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB
1,024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB
2,048 (2 vCPU)	4 GB–16 GB (in 1 GB increments)
4,096 (4 vCPU)	8 GB–30 GB (in 1 GB increments)

50 different CPU/memory configurations per task to choose from

# Auto Scaling



# Amazon ECS cluster autoscaling



## Capacity provider

- Used to determine infrastructure needed to run tasks.

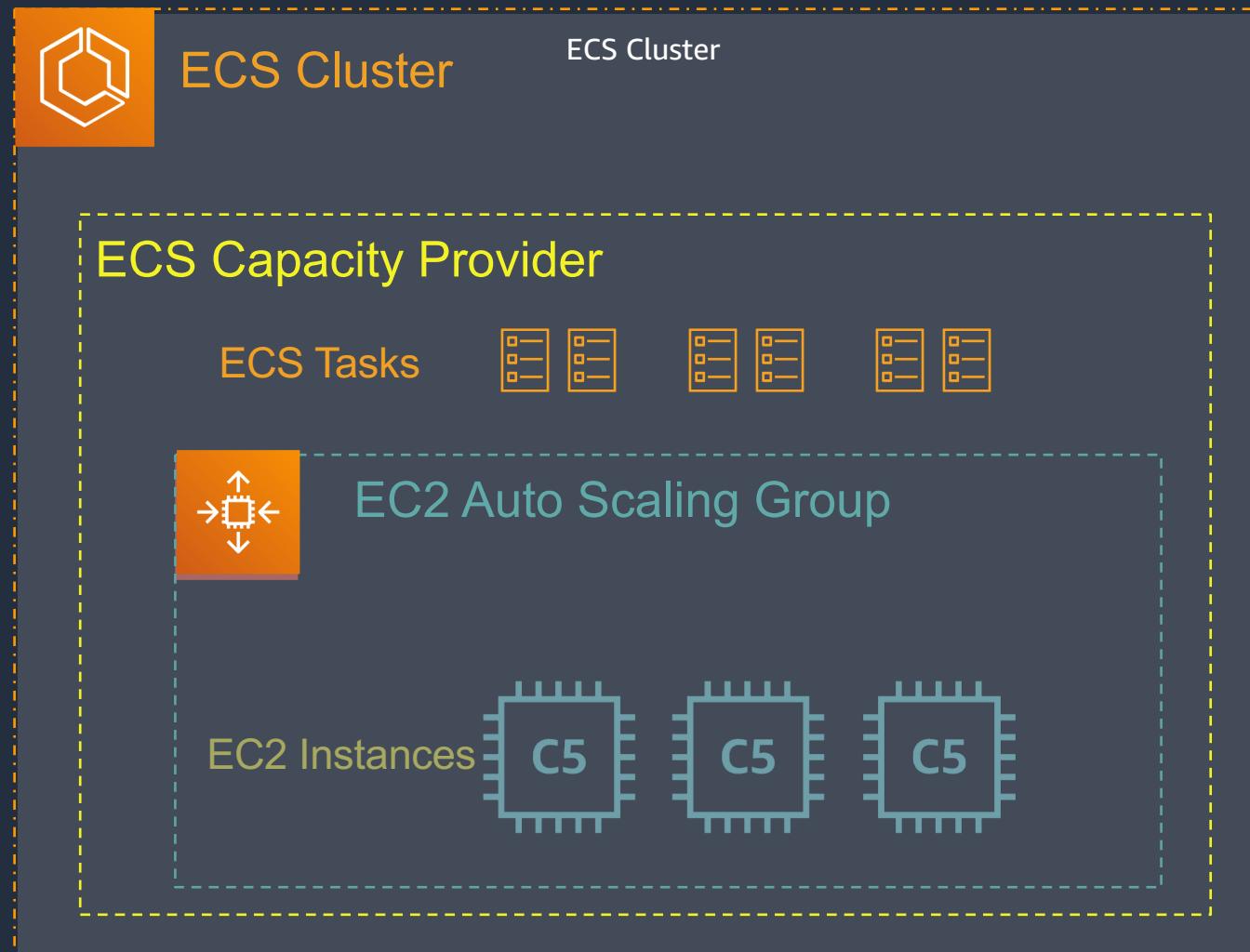
## Capacity provider strategy

- Gives you control over how your tasks use one or more capacity providers

## Default capacity provider strategy

- Determines capacity provider strategy used if not other capacity provider or launch type is specified.

# Amazon ECS capacity providers



## Capacity provider

- Used to determine infrastructure needed to run tasks.

## Capacity provider strategy

- Gives you control over how your tasks use one or more capacity providers

## Default capacity provider strategy

- Determines capacity provider strategy used if not other capacity provider or launch type is specified.

# Three type of scaling policies

## Target Tracking

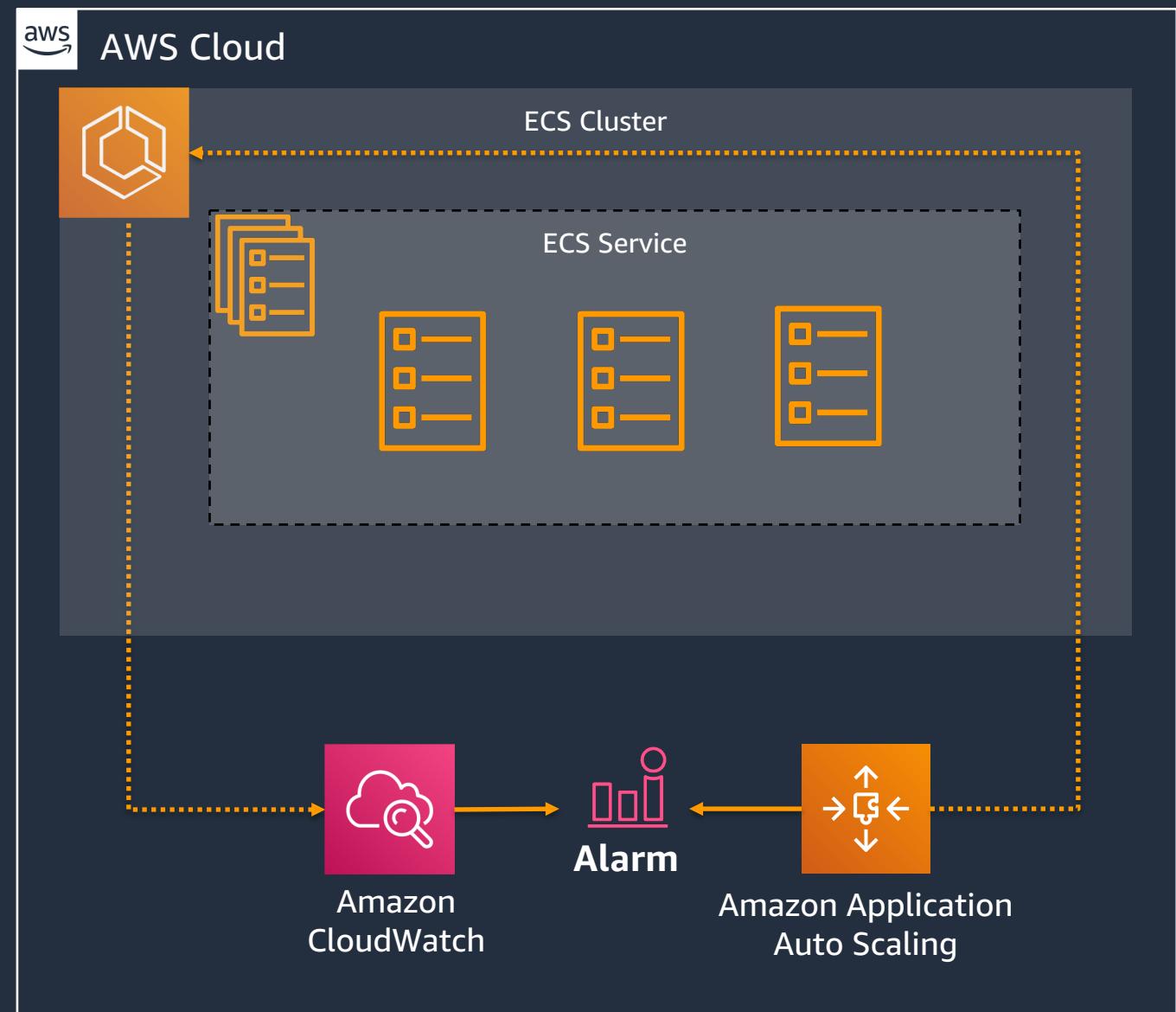
- Scale based on a target value for a specific metric

## Step Scaling

- Scale based on a set of scaling adjustments, or steps, that vary based on the size of the alarm breach

## Scheduled Scaling

- Scale based on the date and time

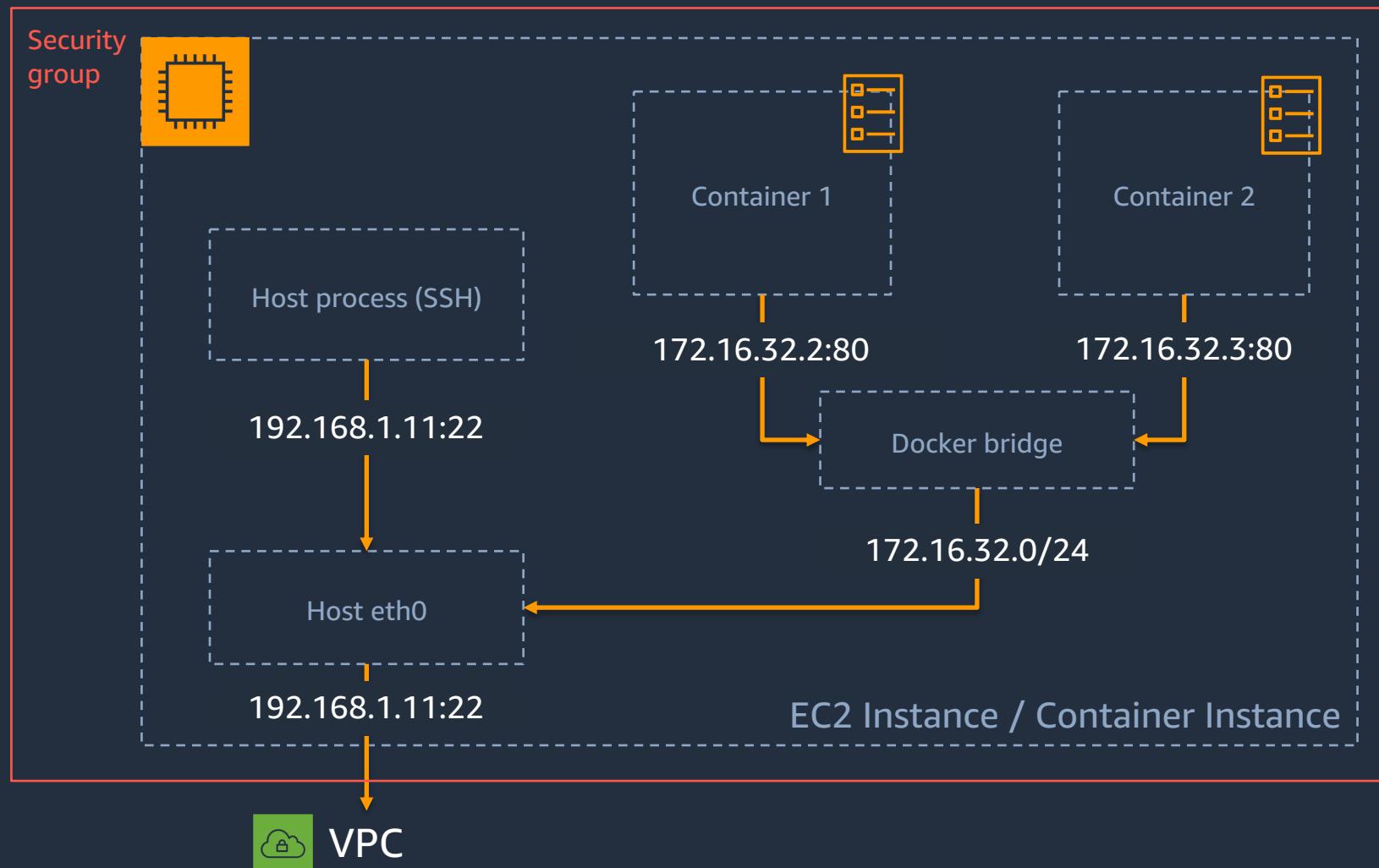


# Networking

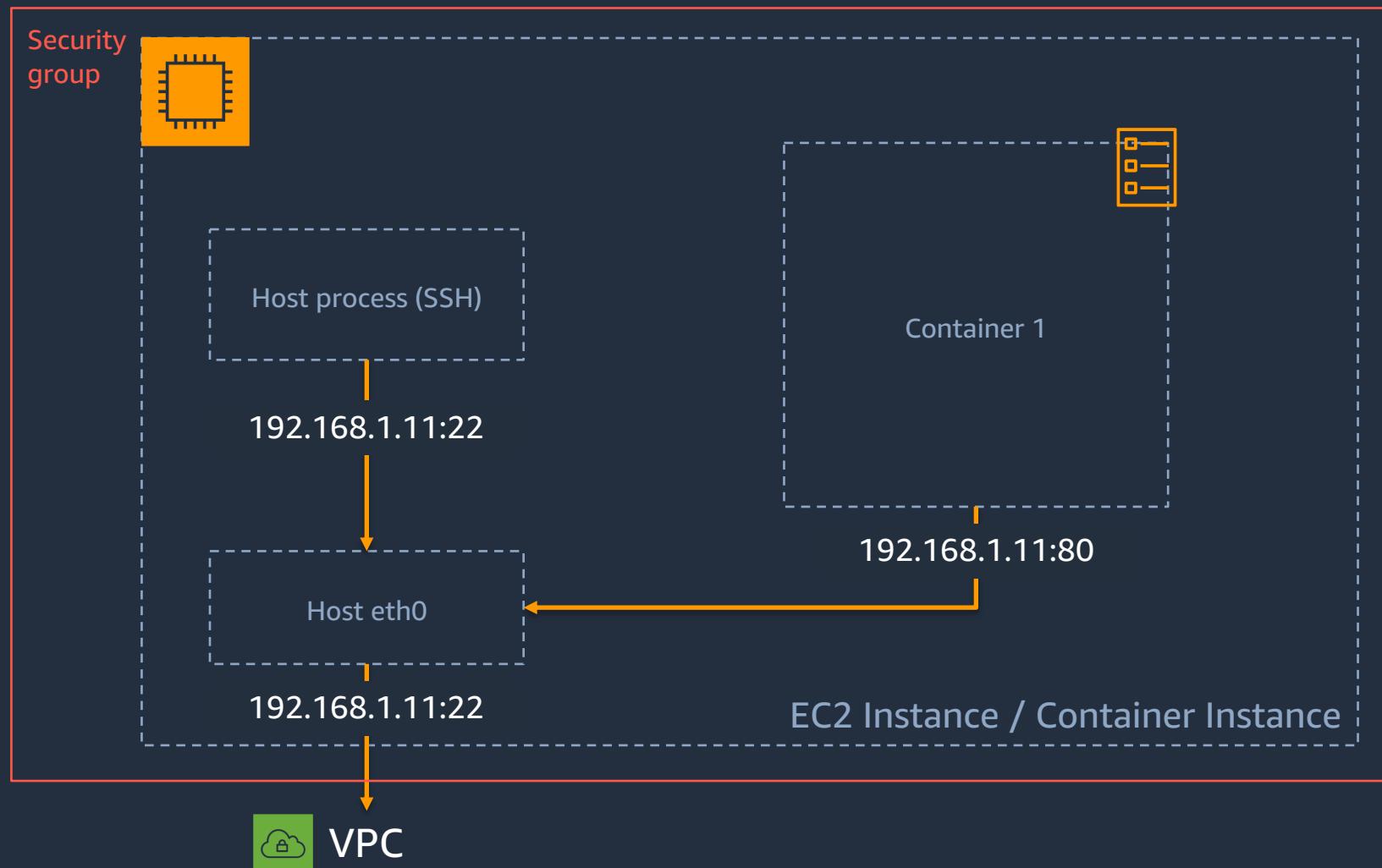
# ECS and Fargate networking modes

Mode		
Bridge	YES	NO
Host	YES	NO
awsvpc	YES	YES

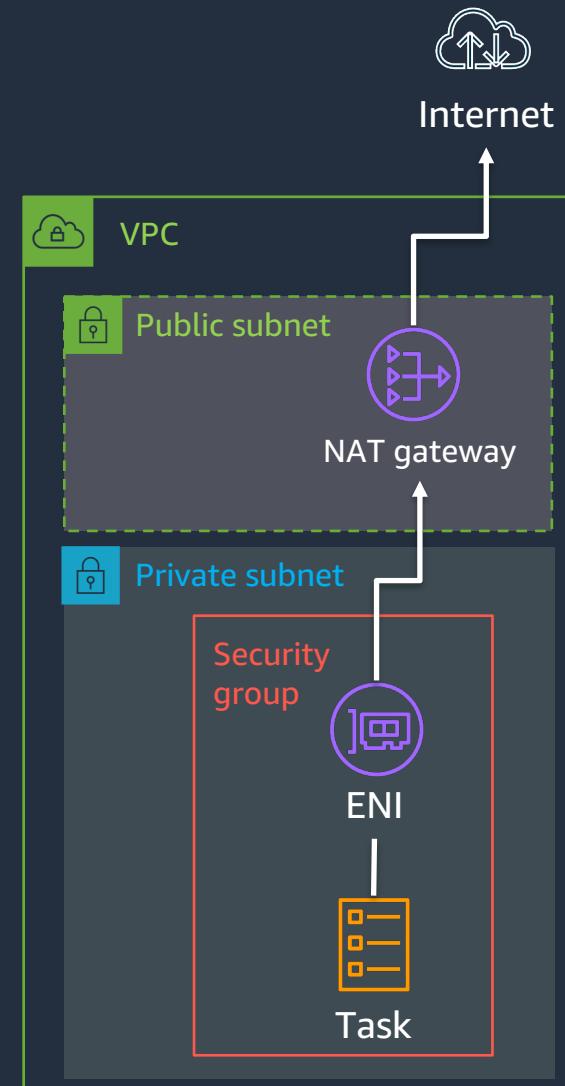
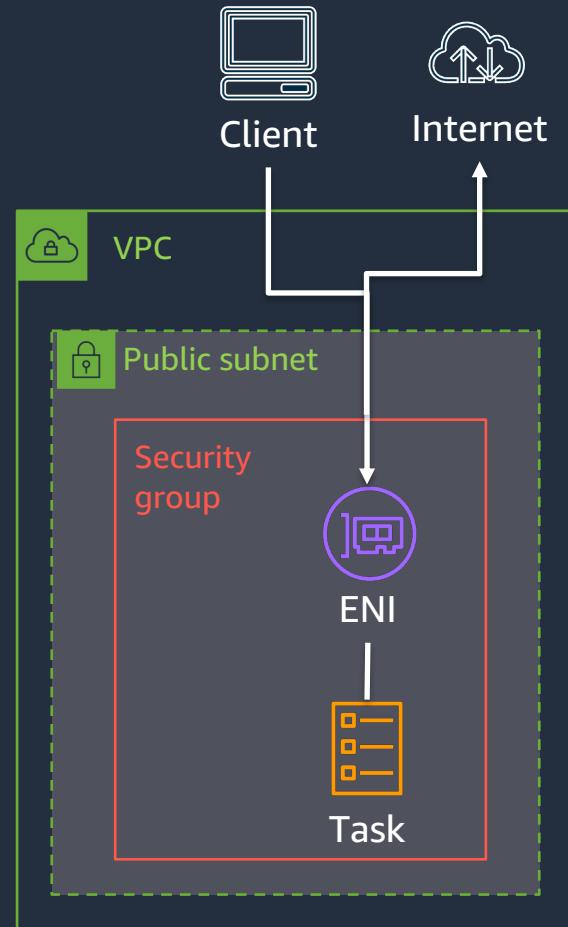
# Networking modes: Bridge



# Networking modes: Host



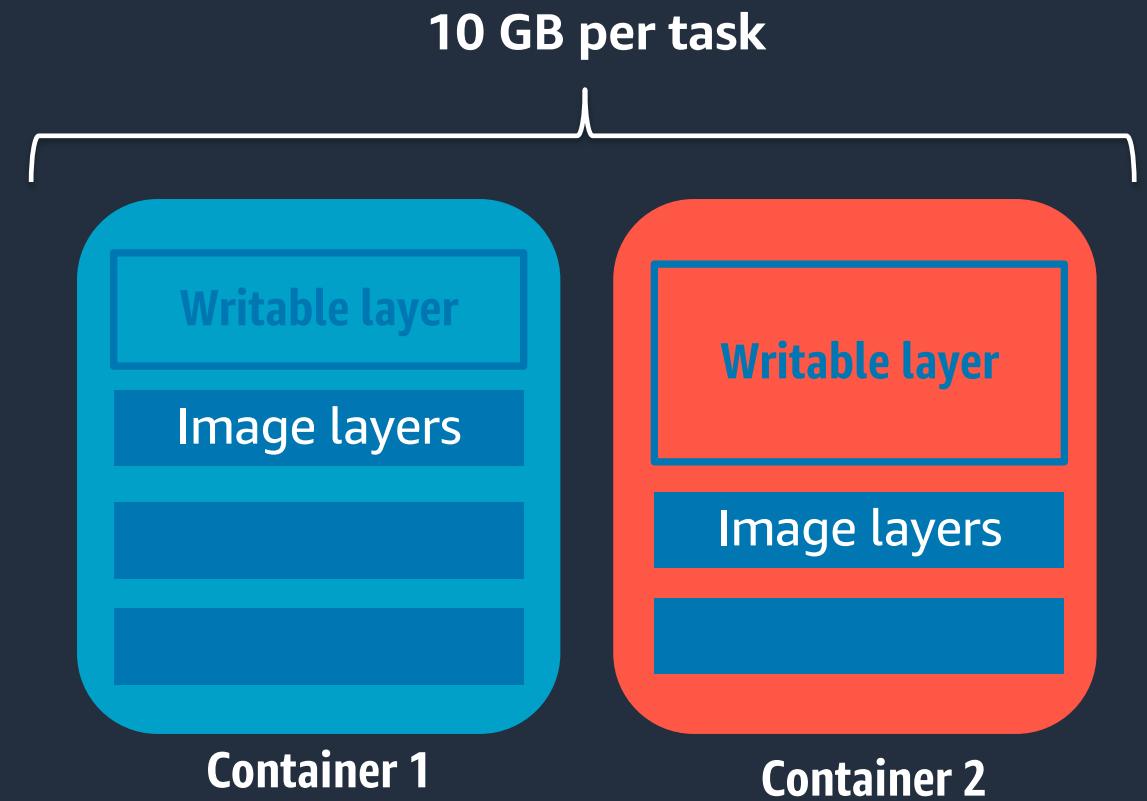
# Networking modes: awsvpc



# Storage

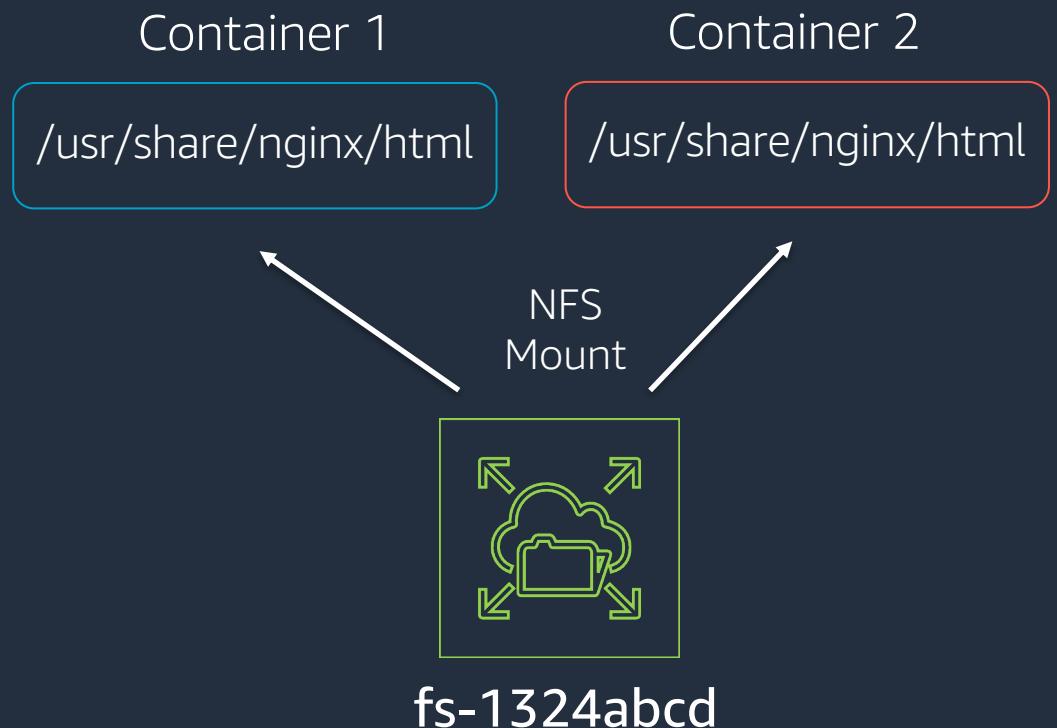
# Layer storage - ephemeral

- Container images are composed of layers - topmost layer is the writable layer to capture file changes made by the running container
- 20 GB layer storage available per task across all containers, including image layers
- Writes are not visible across containers
- Ephemeral storage is not available after the task stops



# EFS storage

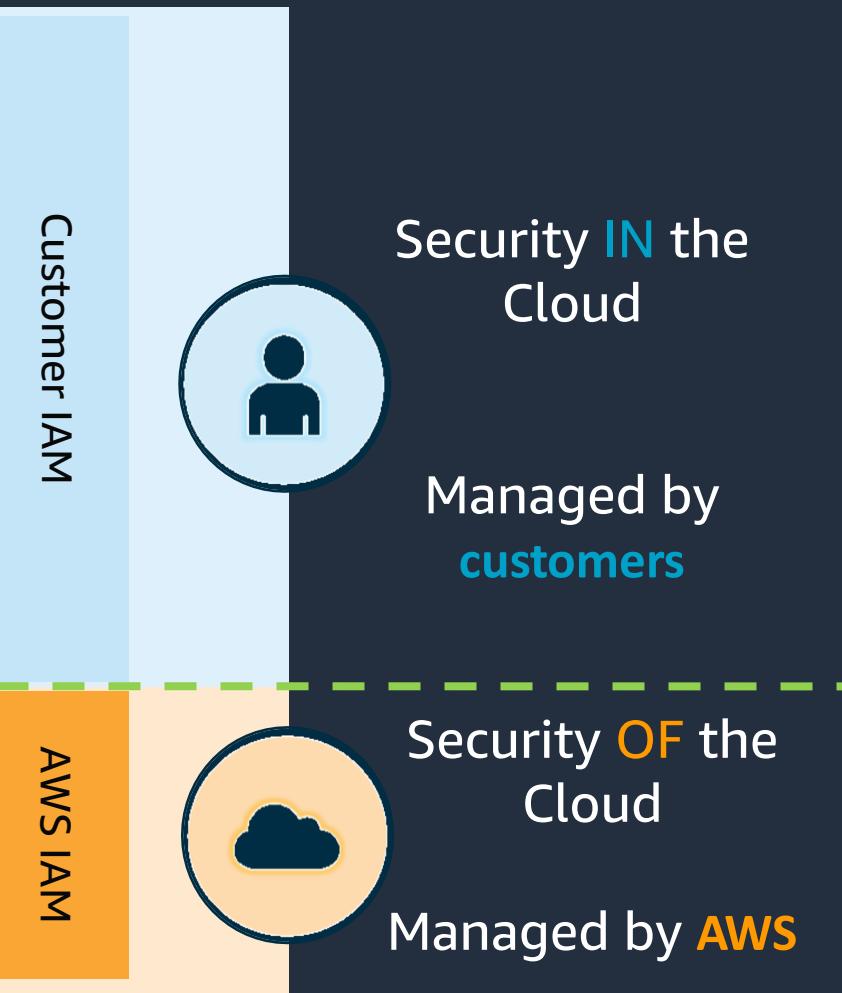
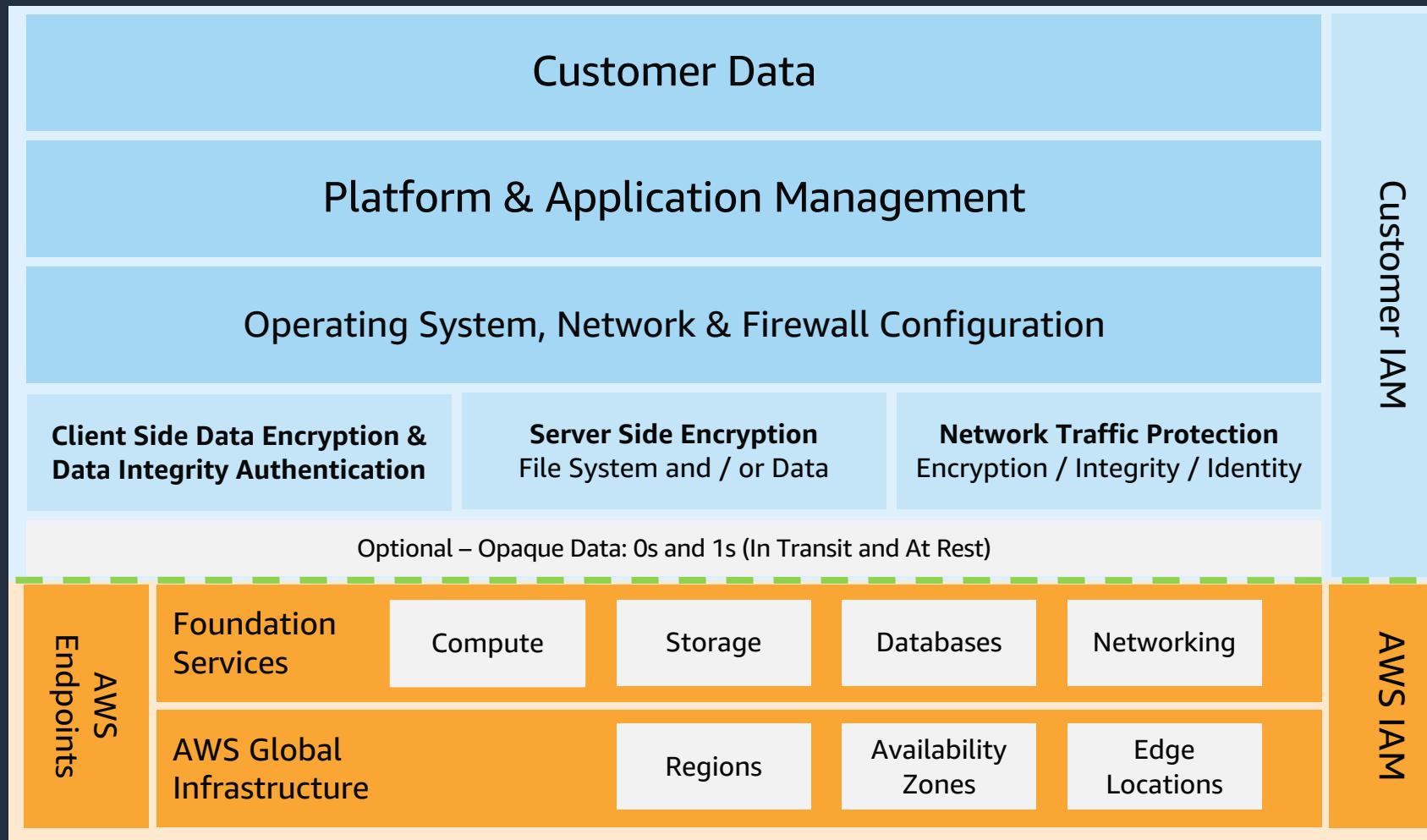
- Need persistence beyond the task lifecycle?
- Fargate platform version 1.4 supports mounting EFS file systems to containers in your task.
- Configure via NFS mounts in task definition
  - Can mount at different container paths



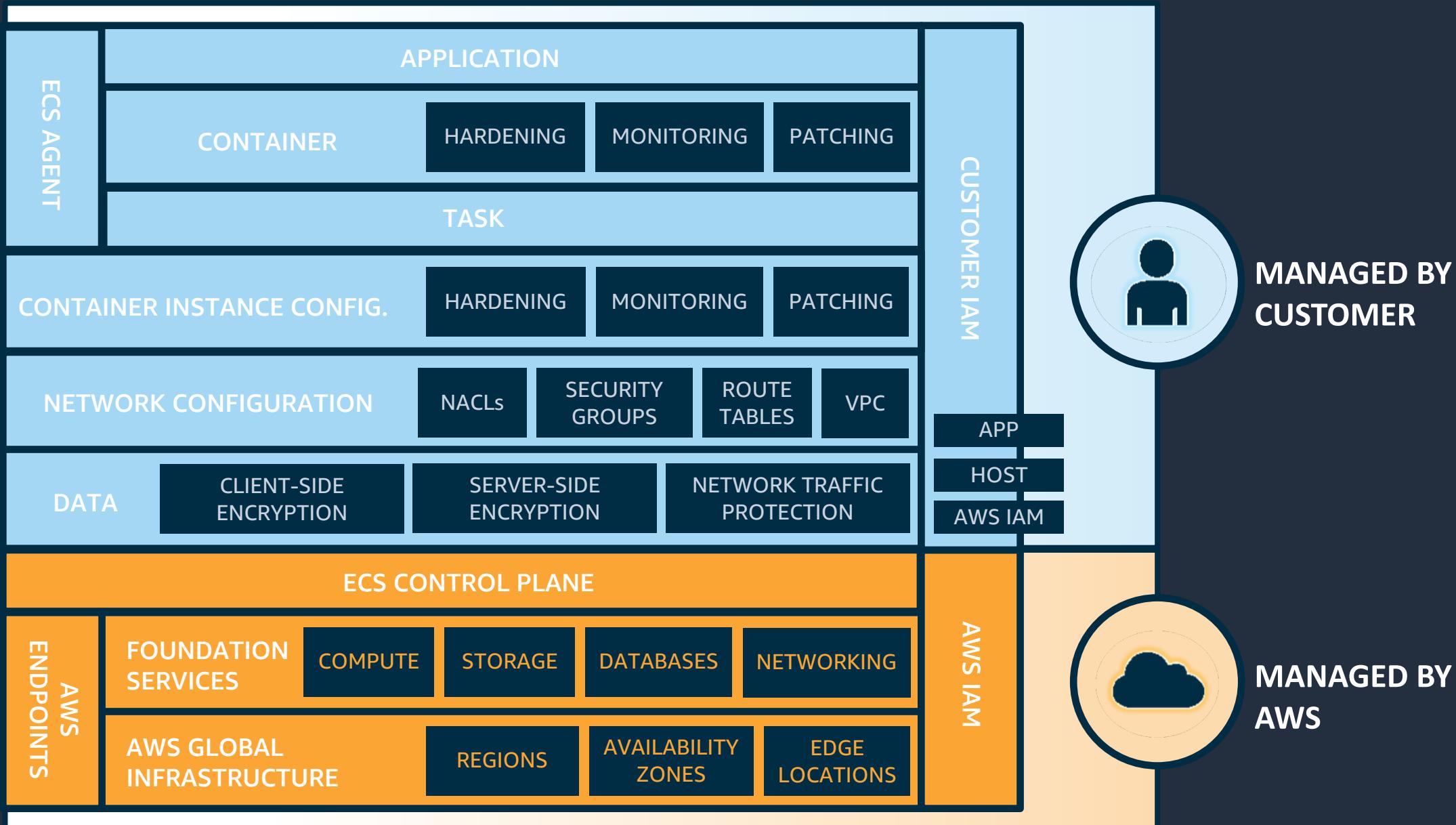
# Security

# Working together

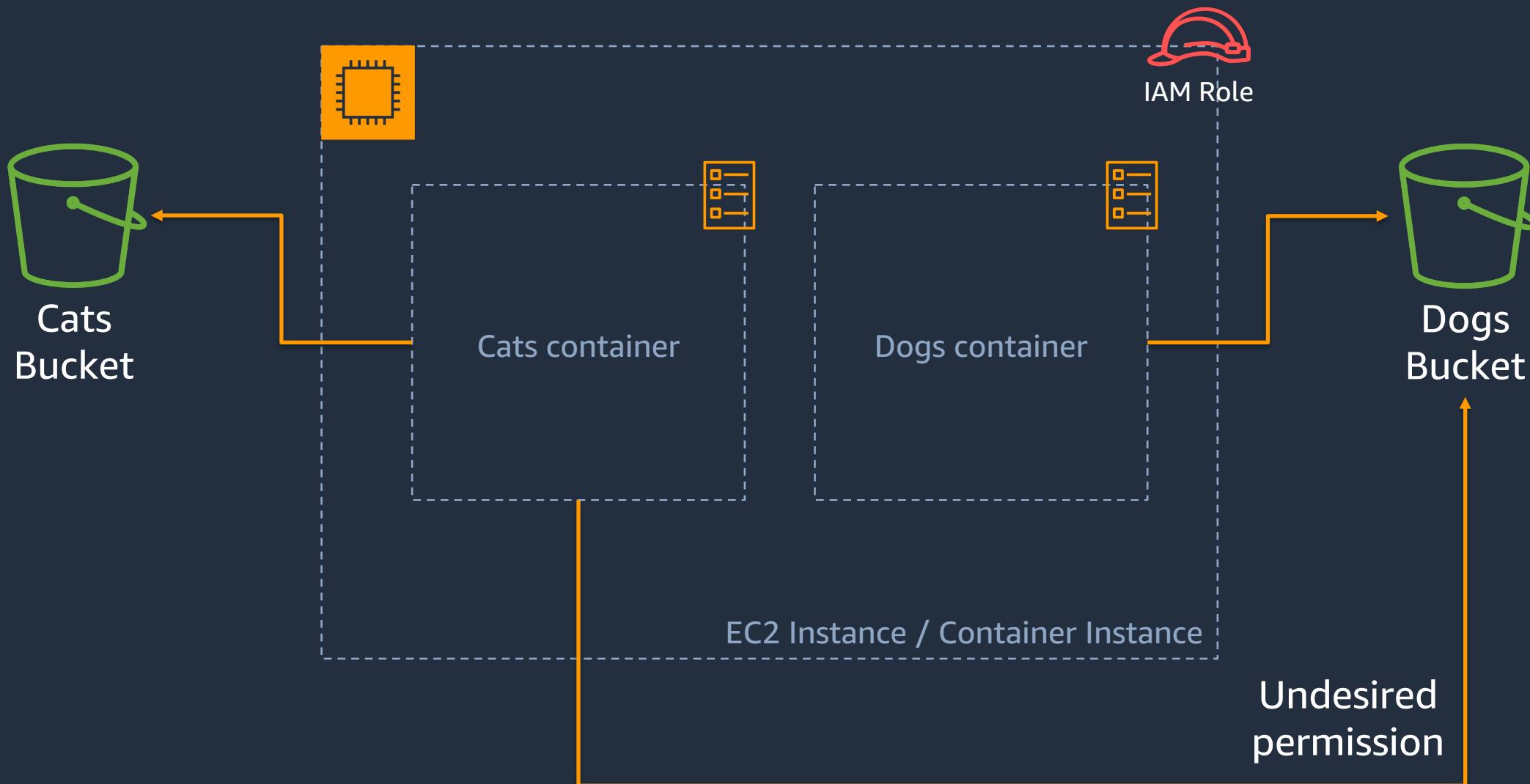
Security in the Cloud is a Shared Responsibility



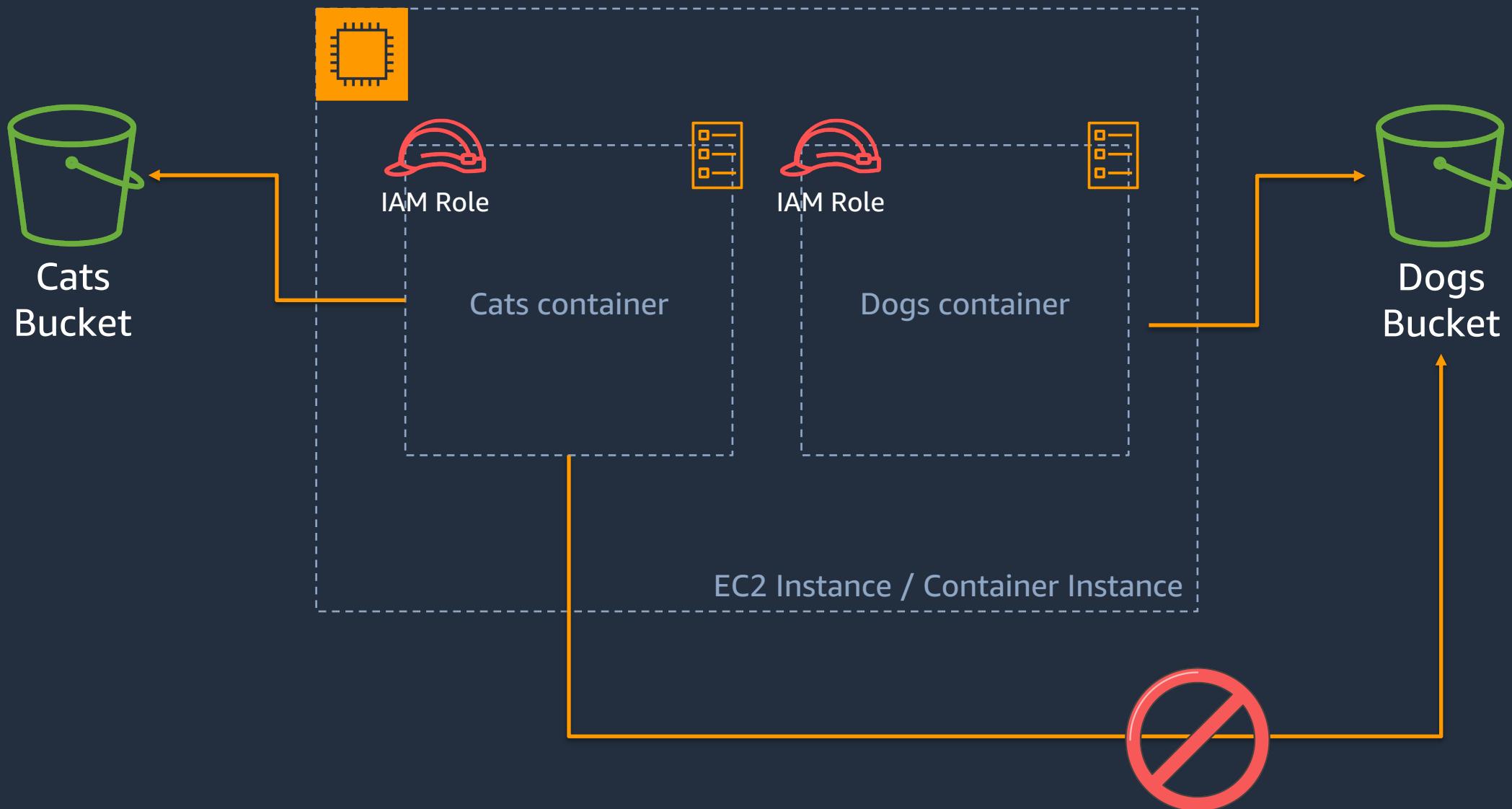
# Shared responsibility model: Amazon ECS for EC2



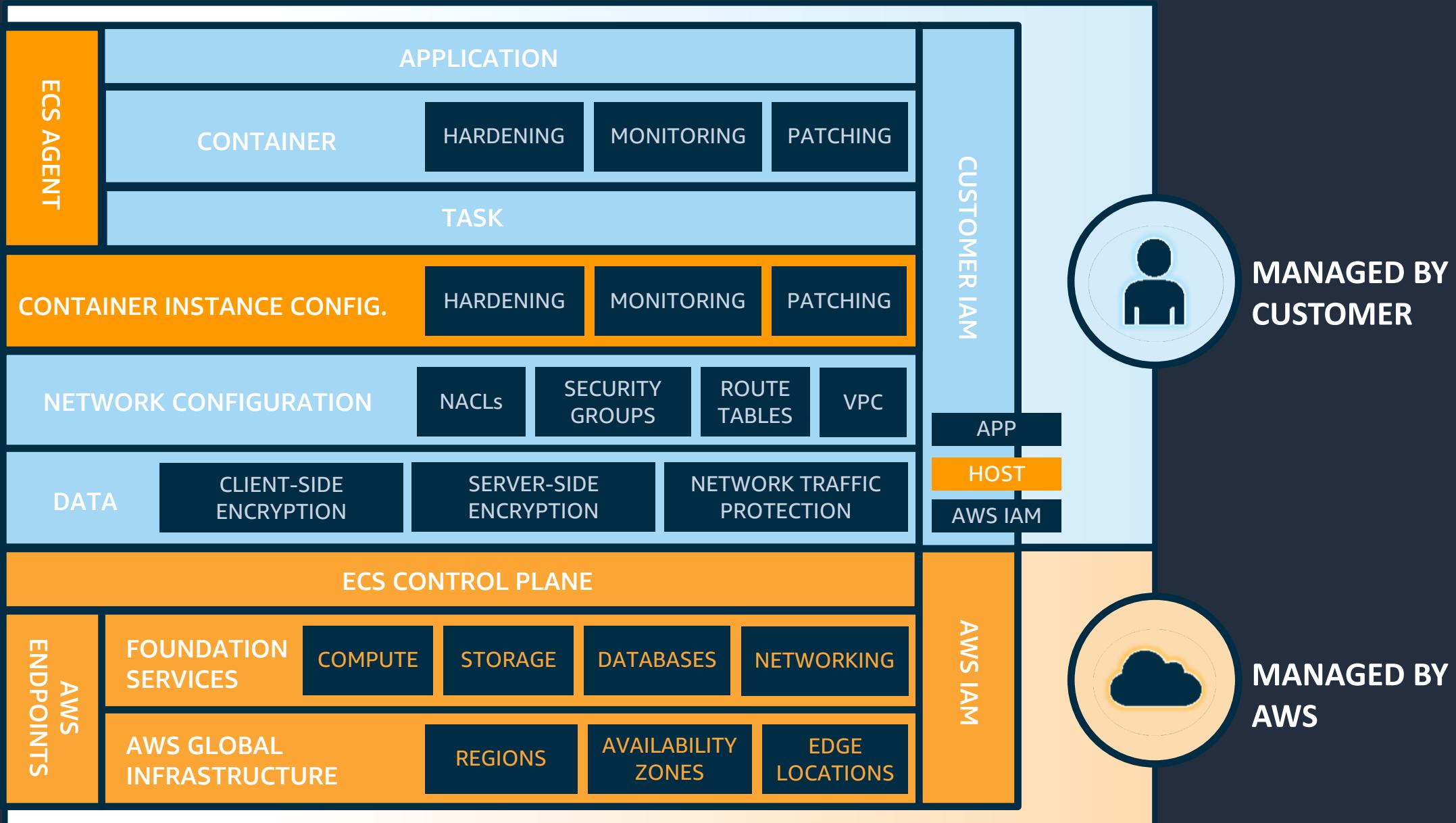
# Security: IAM Roles for Tasks



# Security: IAM Roles for Tasks



# Shared responsibility model: Amazon ECS for AWS Fargate



# Security: Benefits of Fargate

We do more, you do less.

- Patching (OS, Docker, ECS Agent, etc.)
- Task isolation (via Clusters)
- No --privileged mode for containers
- Requires awsvpc network mode so there is an ENI and SG per Task
- Ecs-exec required for runtime access (ssh or interactive commands)

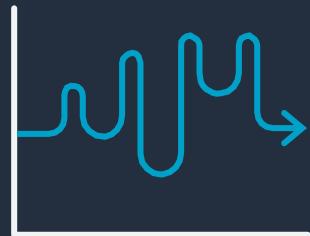


# Cost optimisation

# Fargate Purchase Options

## Fargate

Pay for containers **per-second** with no long-term commitment



Capacity needs can change rapidly

## Compute Savings Plan New

Make a 1 or 3-year commitment and receive a **significant discount**



Baseline compute needs known in advance

## Fargate Spot New

Spare capacity with **savings up to 70%** off Fargate standard pricing



Fault-tolerant, flexible workloads



*Spare compute* Capacity

Save up to **70%** over standard Fargate

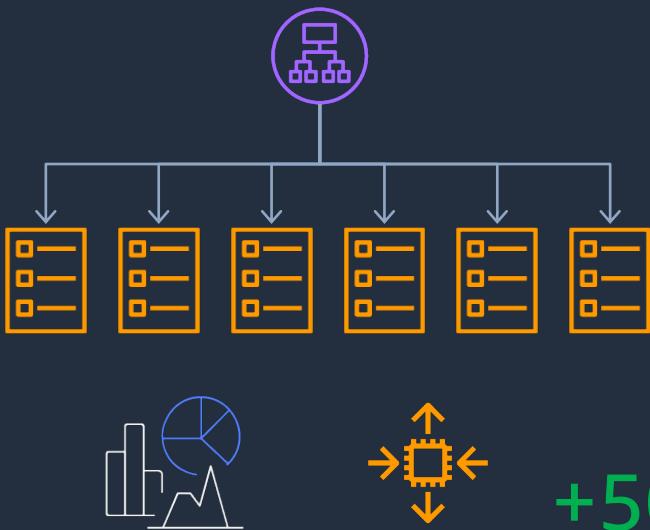
Can be *reclaimed*  
(with two minute warning)

Automatic diversification

# Fargate and Fargate Spot Capacity Provider Mix

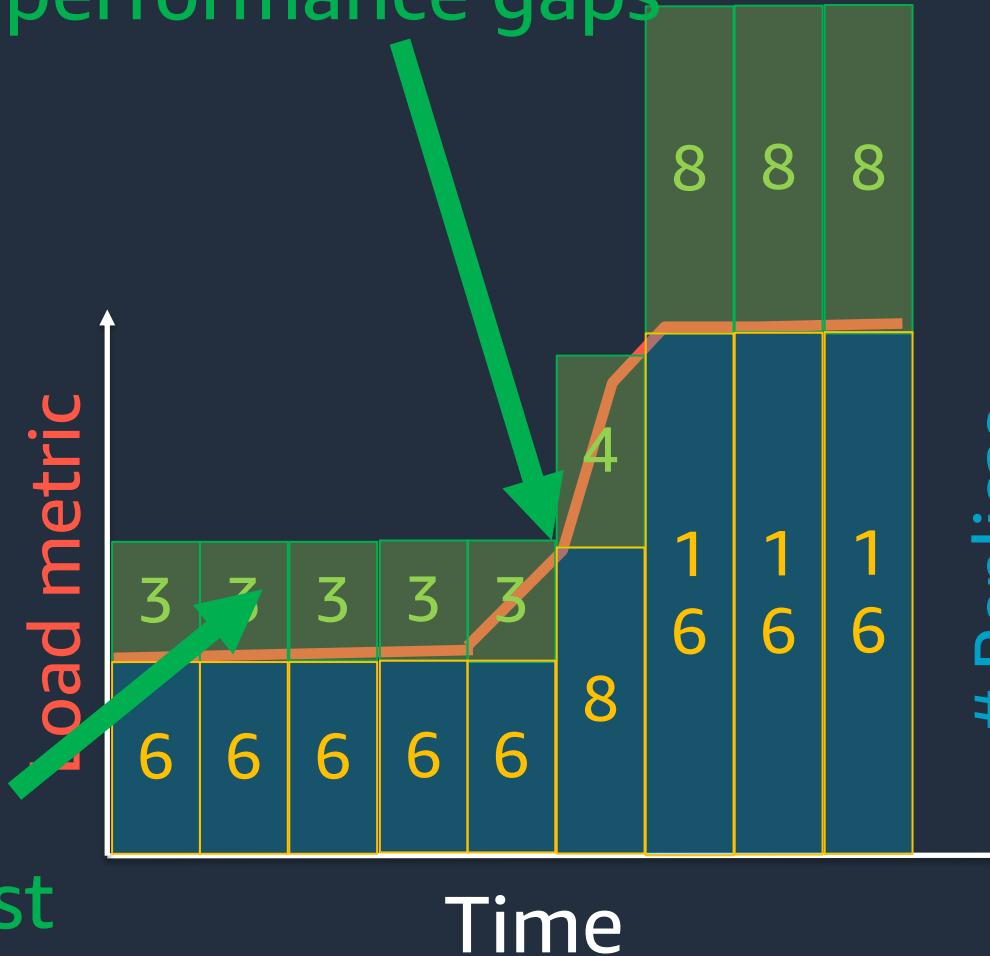
Overprovision by 50%:  
Reduce metric target value by 1/3

Run 2/3 On-Demand, 1/3 on Spot



+50% capacity  
for +5-10% cost

No performance gaps



# Questions?

Introduction to Amazon ECS and AWS Fargate