Webinar Containers – Part 1

Amazon Elastic Container Service

Patrick Madec, Partner Solutions Architect
Guillaume Fedière, AWS Solutions Architect
Kun Song, AWS Solutions Architect
Roberto Migli, AWS Solutions Architect

Agenda

Time	Topic
9H00	Amazon Elastic Container Service (ECS)
10H15	Break
10H30	ECS Workshop
12H00	Wrap-up

Why are enterprises adopting containers?

- Accelerate software development
- Build modern applications
- Automate operations at web scale

Early 2014

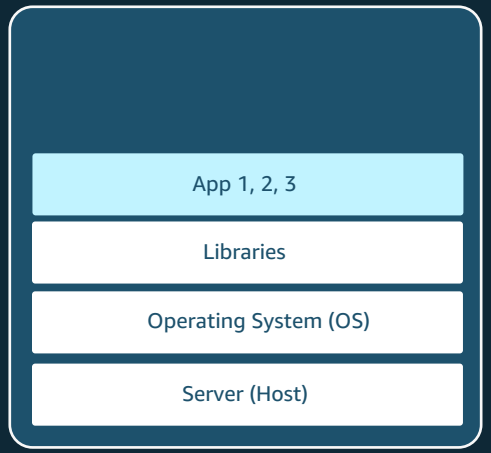
```
$ vi Dockerfile
$ docker build -t mykillerapp:0.0.1
$ docker run -it mykillerapp:0.0.1
```

aws

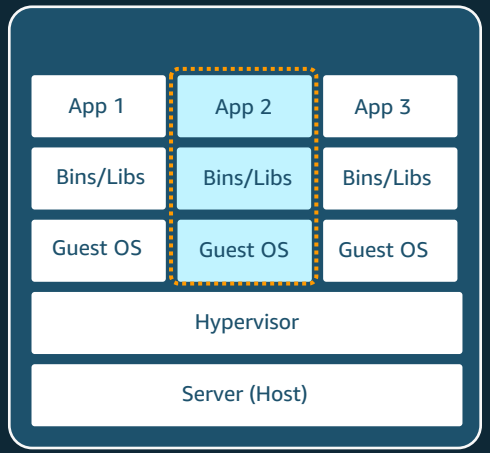
Polyglot packaging

Portable runtime

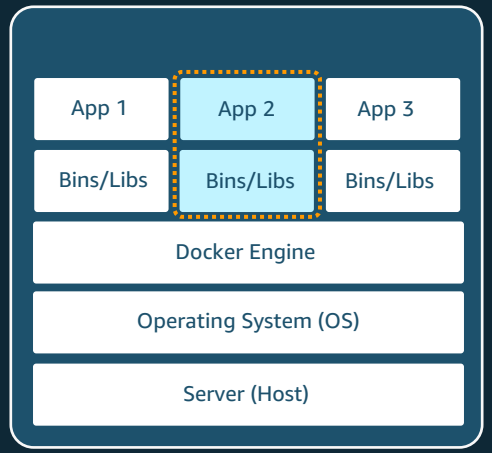
Containers vs VMs



Bare Metal



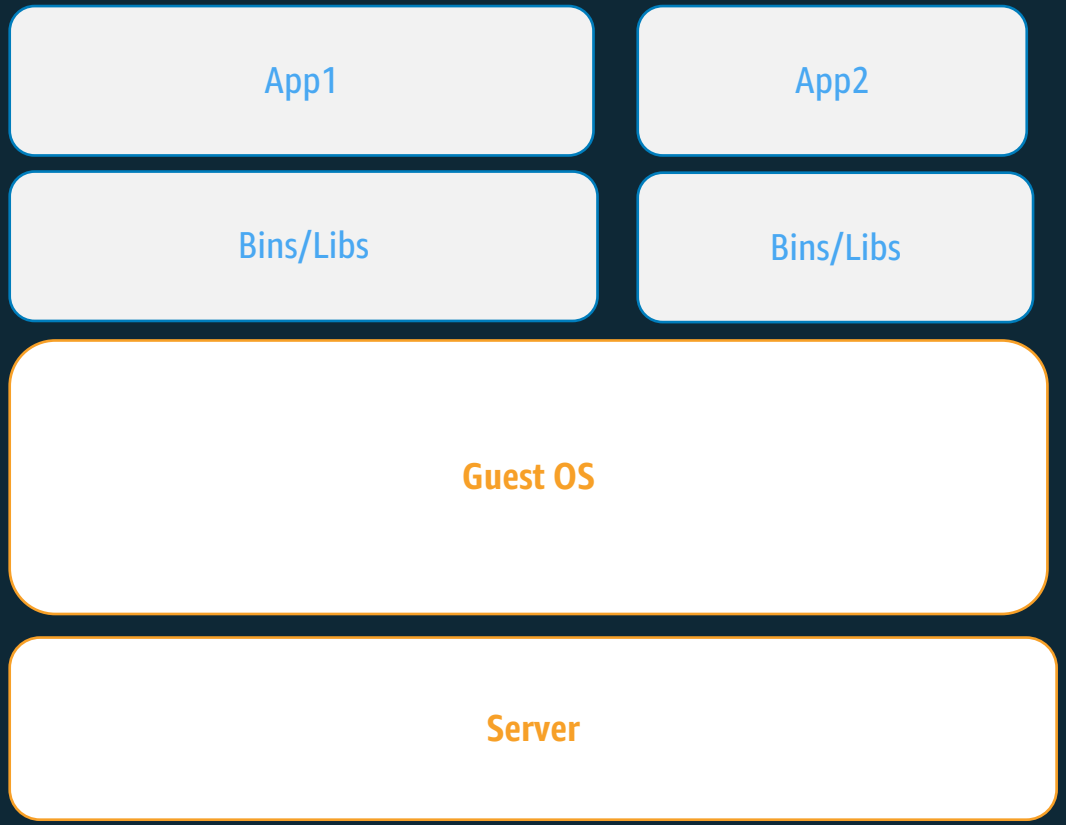
Virtual Machine



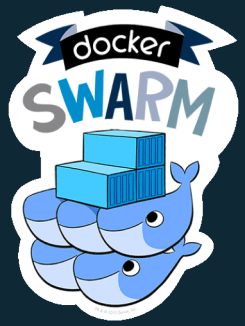
Containers

So what’s the catch?

Managing one container is easy…



…But managing many containers is difficult

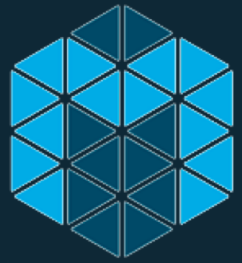


Amazon Elastic
Container Service

kubernetes

Enter containers orchestration tools

HashiCorp
Nomad



Apache
MESOS™

aws

Make AWS the **BEST PLACE** to run **ANY** containerized applications

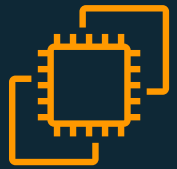
AWS container services landscape

Management
Deployment, Scheduling, Scaling & Management of containerized applications

Amazon Elastic Container Service

Amazon Elastic Container Service for Kubernetes

Hosting
Where the containers run



Amazon EC2

AWS Fargate

Image Registry
Container Image Repository

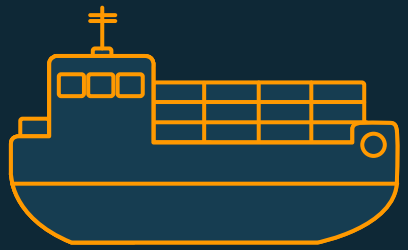
Amazon Elastic Container Registry

aws

Amazon Elastic Container Service

Helping customers scale containers

450+%
growth



Hundreds of millions
of containers started each week

of **millions**
of container instances

Customers Using Containers at Scale

aws

Why customers love AWS container services

Deeply integrated with AWS

Broad selection of compute instances and IAM security, VPC networking, load balancing, and autoscaling

DevOps Workflow

Best place to build and operate a complete DevOps workflow for containers—AWS DevTools and Cloud9

Security and Compliance

ISO, HIPPA, PCI, SOC1, SOC2, SOC3 Infocomm Media Development Auth.

Containers are a first-class citizen of the AWS Cloud


Service level agreement
99.99%
Amazon ECS

So what is ECS?

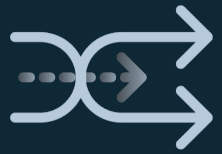
ECS

Highly scalable, high performance container management system

A managed platform

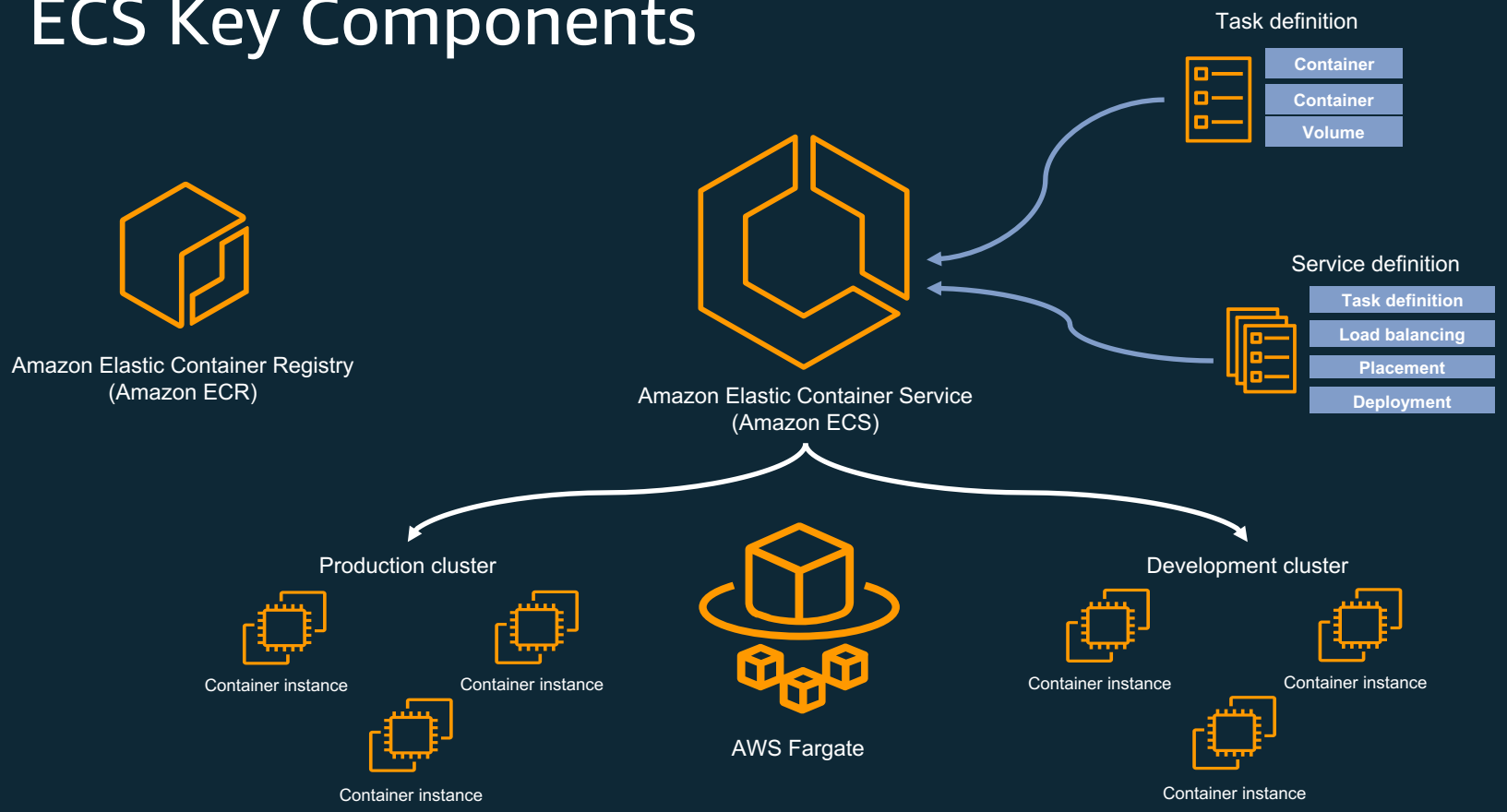
Cluster management

Container orchestration



Deep AWS integration

ECS Key Components



Tasks is a fundamental compute primitive

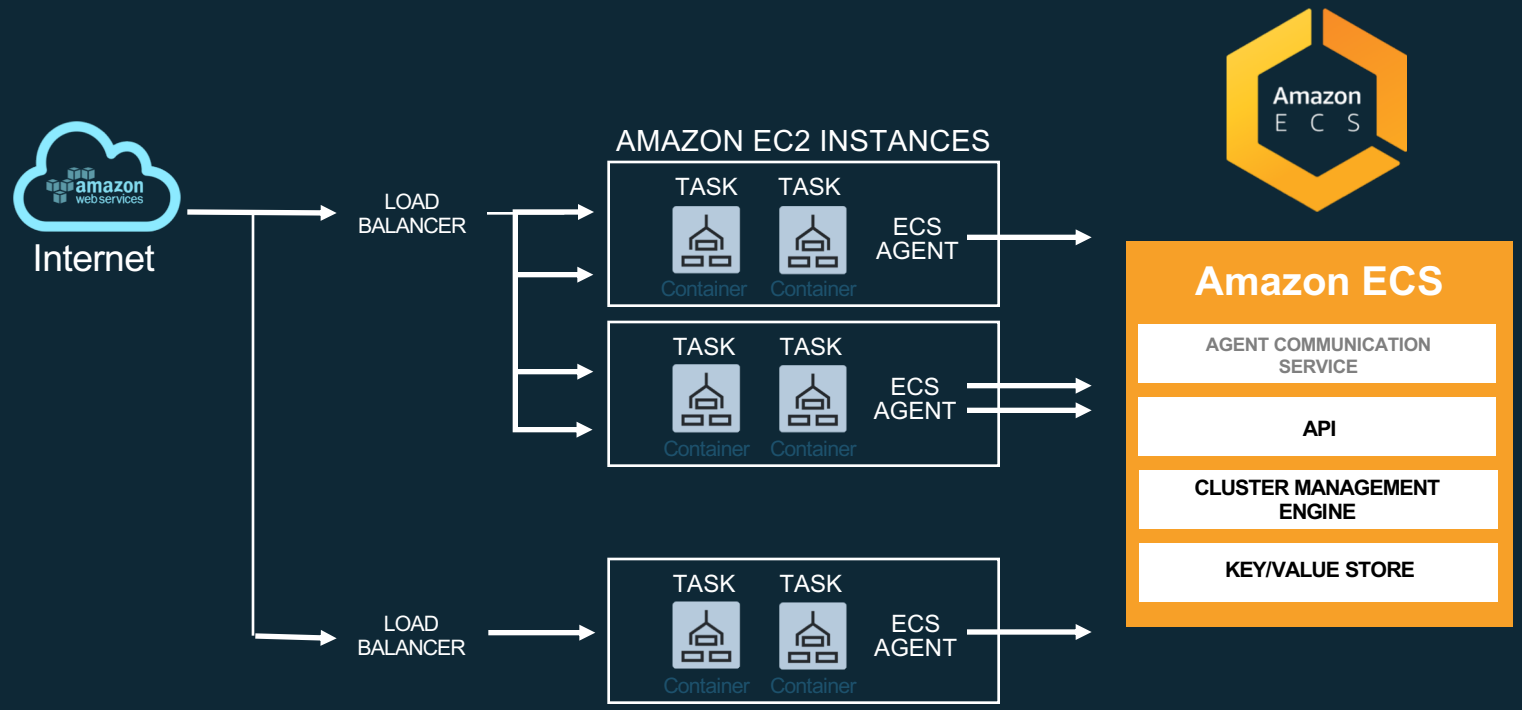
IAM roles for tasks

Task Auto Scaling

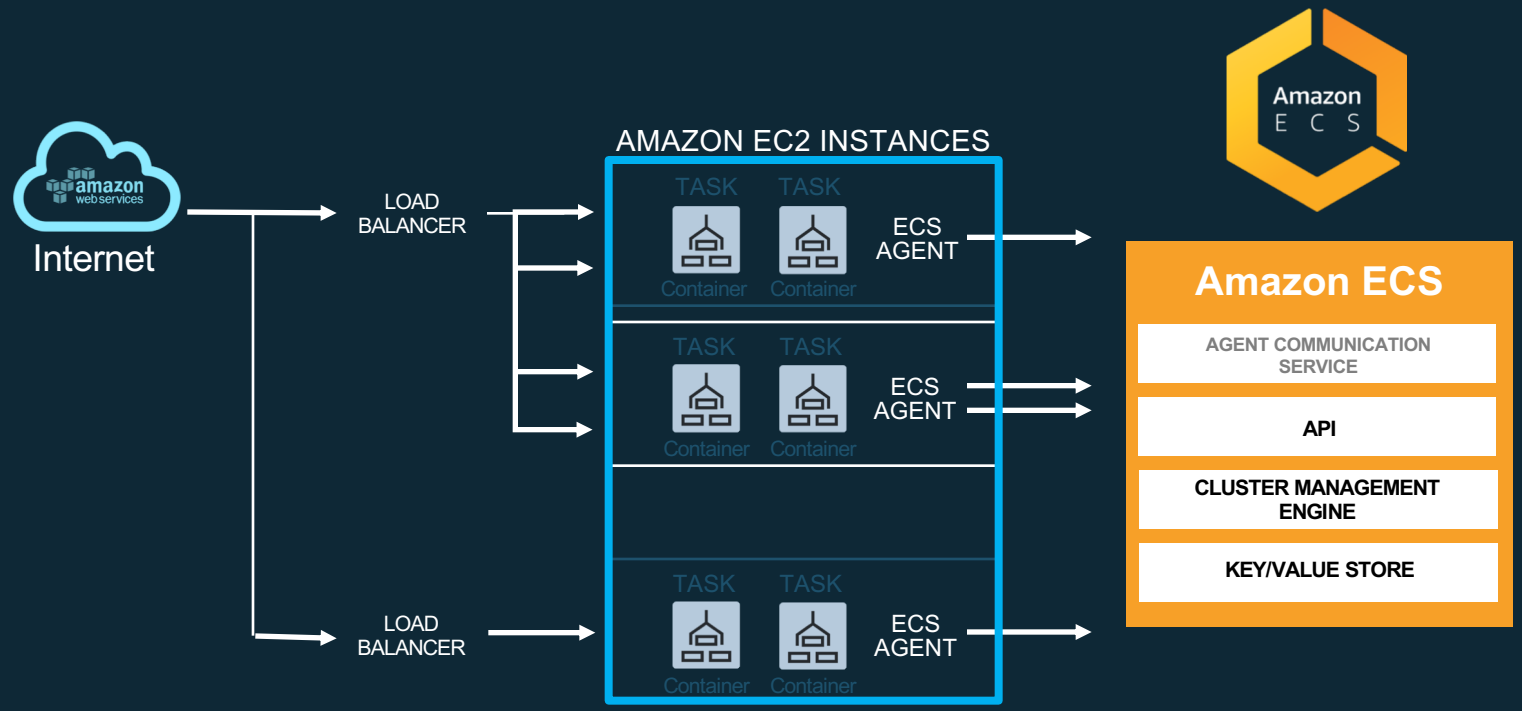
Task LBs

Task networking

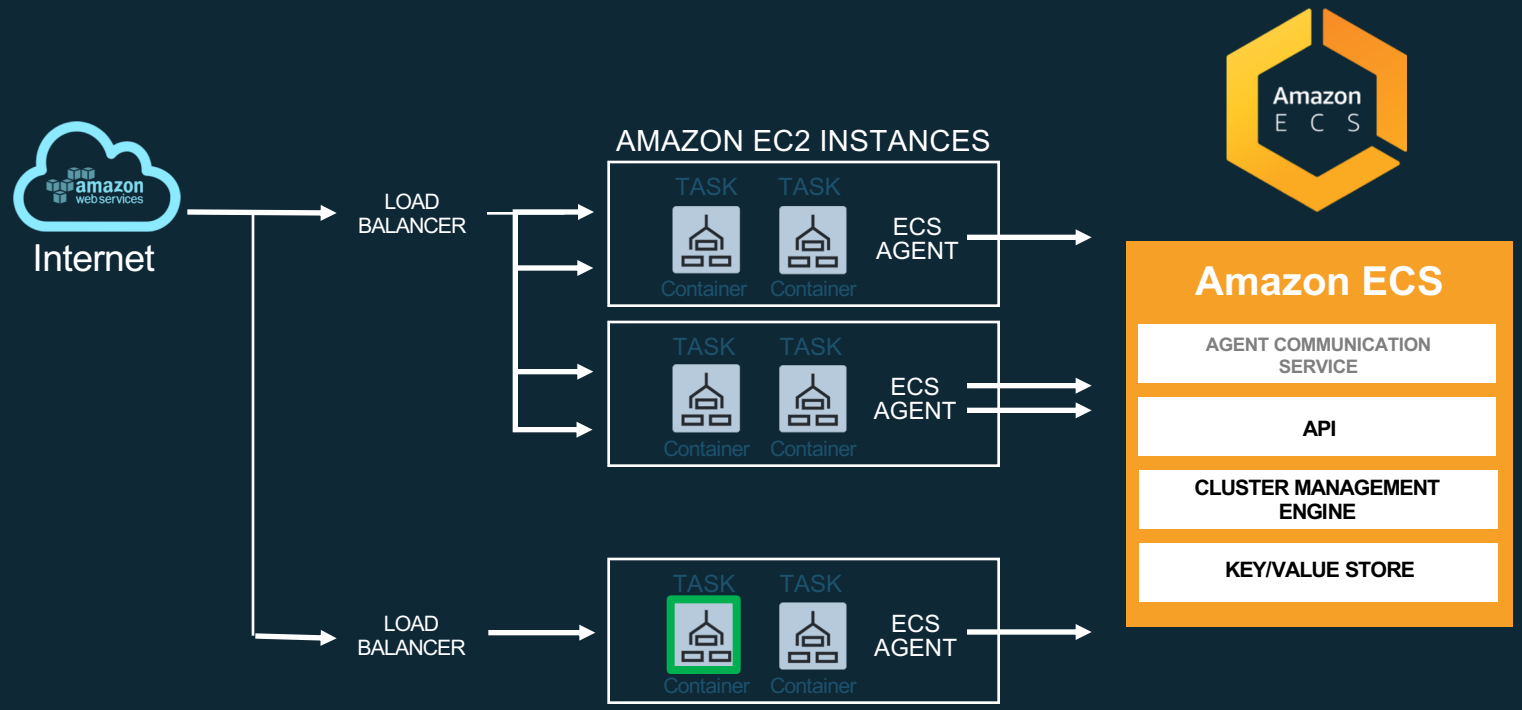
Amazon ECS



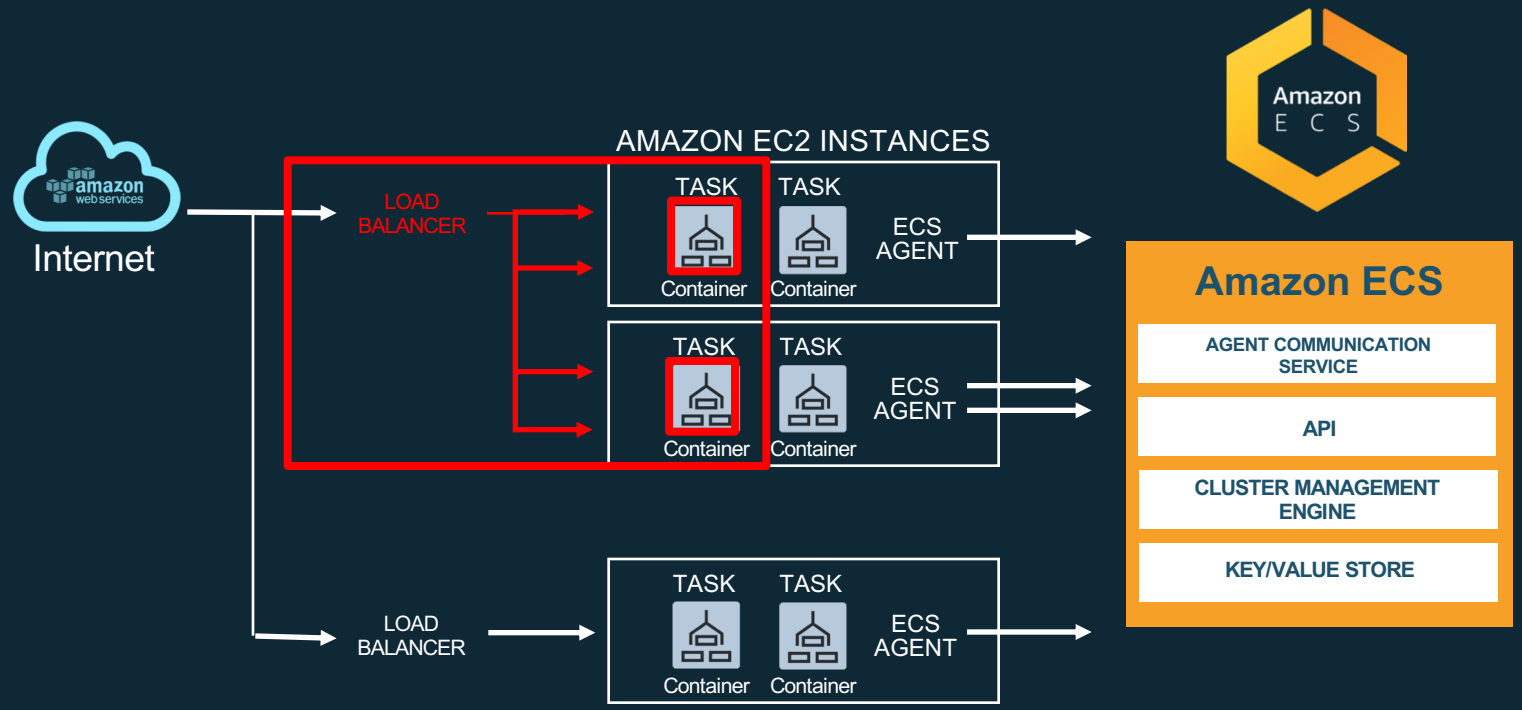
Amazon ECS - cluster



Amazon ECS - task



Amazon ECS - service



Constructs

register

Task Definition

Define application containers: Image URL, CPU & Memory requirements, etc.

run

Task

- A running instantiation of a task definition

Elastic Load Balancer

create

Service

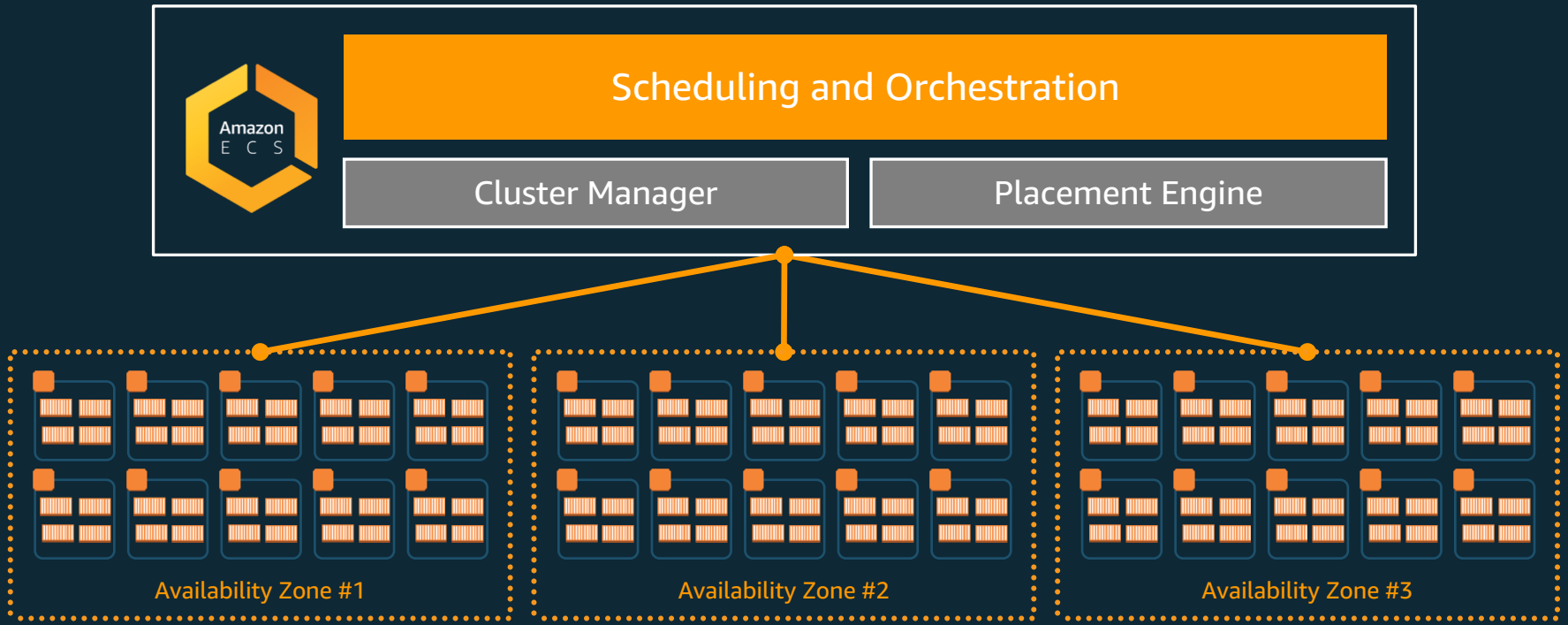
- Maintain n running copies
- Integrated with ELB
- Unhealthy tasks automatically replaced

create

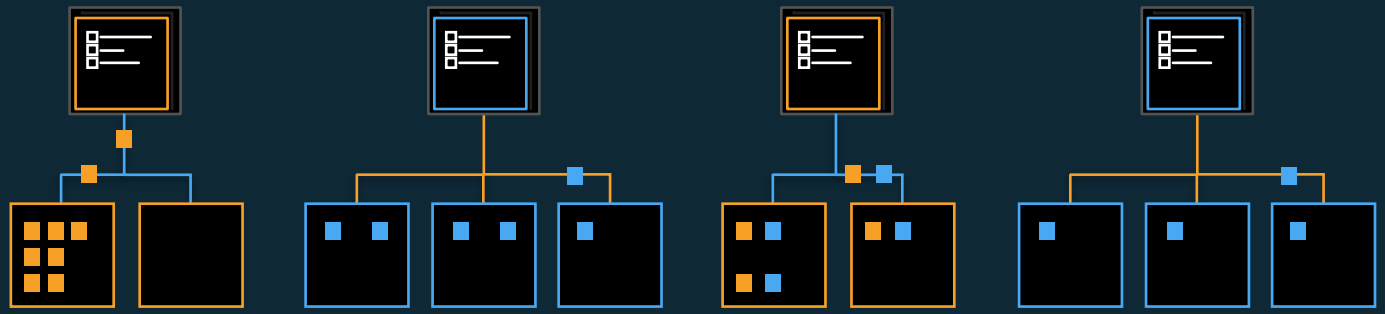
Cluster

- Infrastructure Isolation boundary
- IAM Permissions boundary

Running containers at scale with ECS



Task placement strategies



Binpacking | Spread | Affinity | Distinct instance

Custom task placement constraints

Name	Example
AMI ID	attribute:ecs.ami-id == ami-eca289fb
Availability Zone	attribute:ecs.availability-zone == us-east-1a
Instance Type	attribute:ecs.instance-type == t2.small
Distinct Instances	type="distinctInstances"
Custom	attribute:stack == prod

What we did with ECS

Target Tracking Autoscaling

CLI Supports Docker Compose V3

App Mesh Preview

Faster Launch Times

Daemon Scheduling

Docker Container Health Checks

SSM Parameter Support

GPU Pinning

Task metric & metadata endpoint

Configure shm-size and tmpfs

ECS Agent Signed for Security

Service Discovery

aws

What we did with ECS

Target Tracking Autoscaling

CLI Supports Docker Compose V3

App Mesh Preview

Faster Launch Times

Daemon Scheduling

Docker Container Health Checks

SSM Parameter Support

GPU Pinning

Task metric & metadata endpoint

Configure shm-size and tmpfs

ECS Agent Signed for Security

Service Discovery

Service Discovery

Amazon ECS updates service registry based on naming convention, task registrations, de-registrations and health

Amazon Route 53 provides Service Registry

AWS Cloud Map keeps track of all task instances

aws

Target Tracking Autoscaling

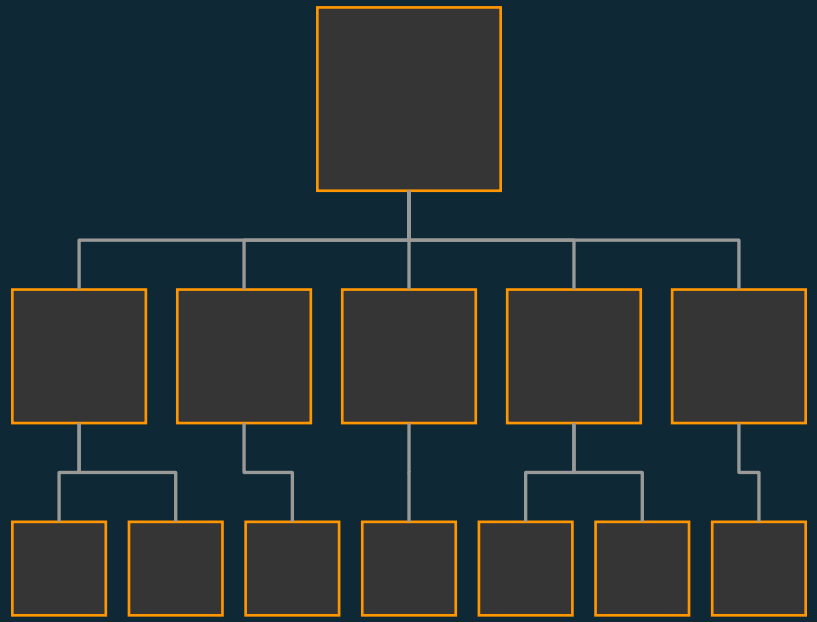
- Target tracking scaling policies, you select a **CloudWatch metric** and set a target value.
  Example : ALBRequestCountPerTarget

- Amazon ECS creates and manages the **CloudWatch alarms that trigger the scaling policy** and calculates the scaling adjustment based on the metric and the target value.

- The scaling policy **adds or removes service tasks** as required to keep the metric at, or close to, the specified target value

AWS Fargate

Managing Clusters Is Not Fun!



Changing Compute Consumption Model

No instances to manage

Task native API

Resource based pricing

Simple, easy to use, powerful – and new consumption model

Production Workloads on AWS

AWS VPC networking mode

Task placement strategy : Spread across AZ

Deep integration with AWS platform

ECS CLI

Global footprint

Powerful scheduling engines

Auto scaling

CloudWatch metrics

Load balancers

HOW DO I RUN CONTAINERS ON FARGATE?

Running Fargate Containers with ECS

Run Task

Select the cluster to run your task definition on and the number of copies of that task to

Launch type FARGATE EC2

Task Definition ecsdemo-frontend

Platform version LATEST

Cluster fargate-demo-ECSCluster-SZ3JMMD6KBB2

Number of tasks 1

Task Group

Running Fargate Containers with ECS

Same *Task Definition* schema

Use ECS APIs to launch Fargate Containers

Easy migration – Run *Fargate* and *EC2* launch type tasks in the same cluster

Service level agreement

99.99%

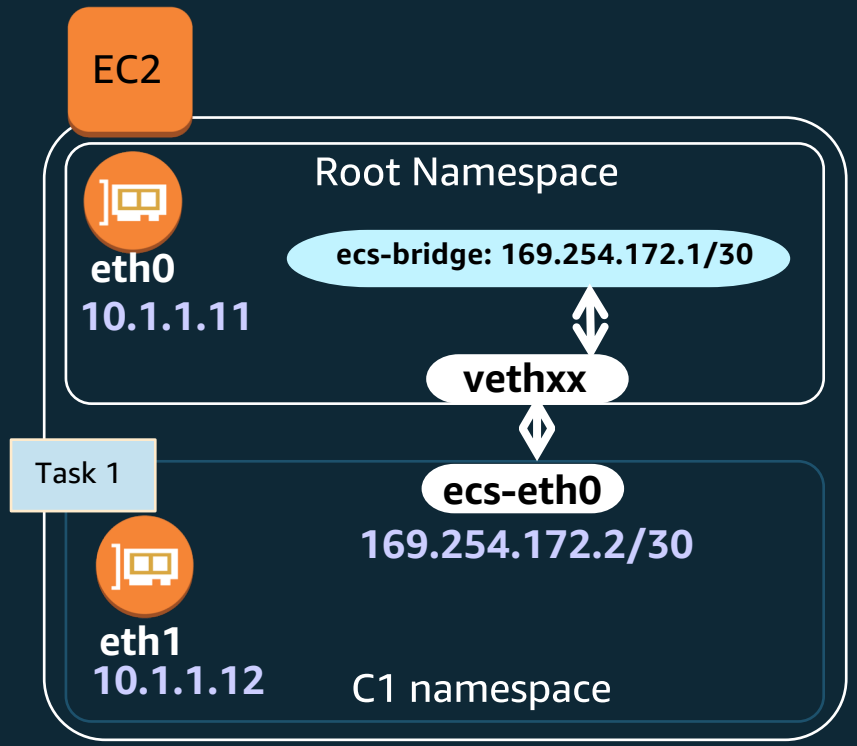
Amazon ECS

AWS Fargate

NETWORKING

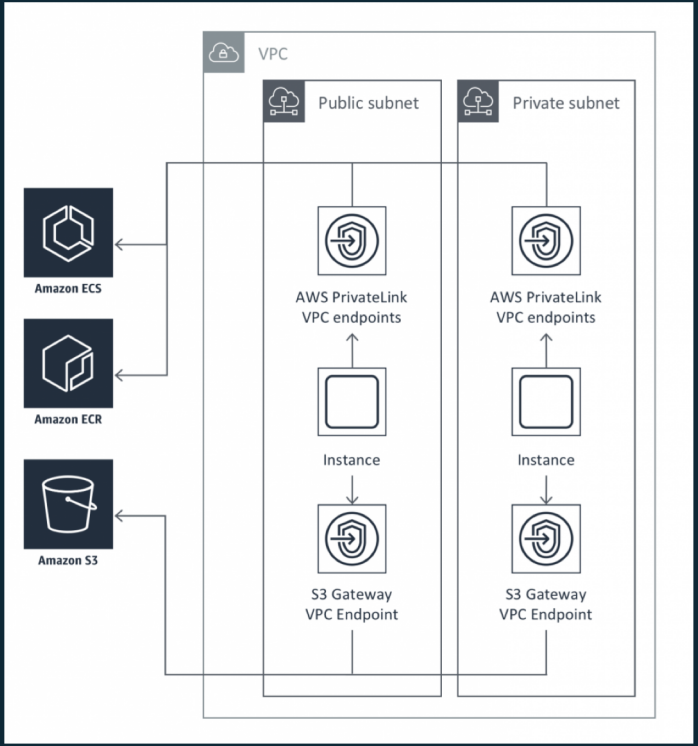
ECS networkMode

- None : Disables all networking
- Host : adds a container on the host's network stack
- Bridge : default Docker network mode
- aws-vpc (Fargate mode)

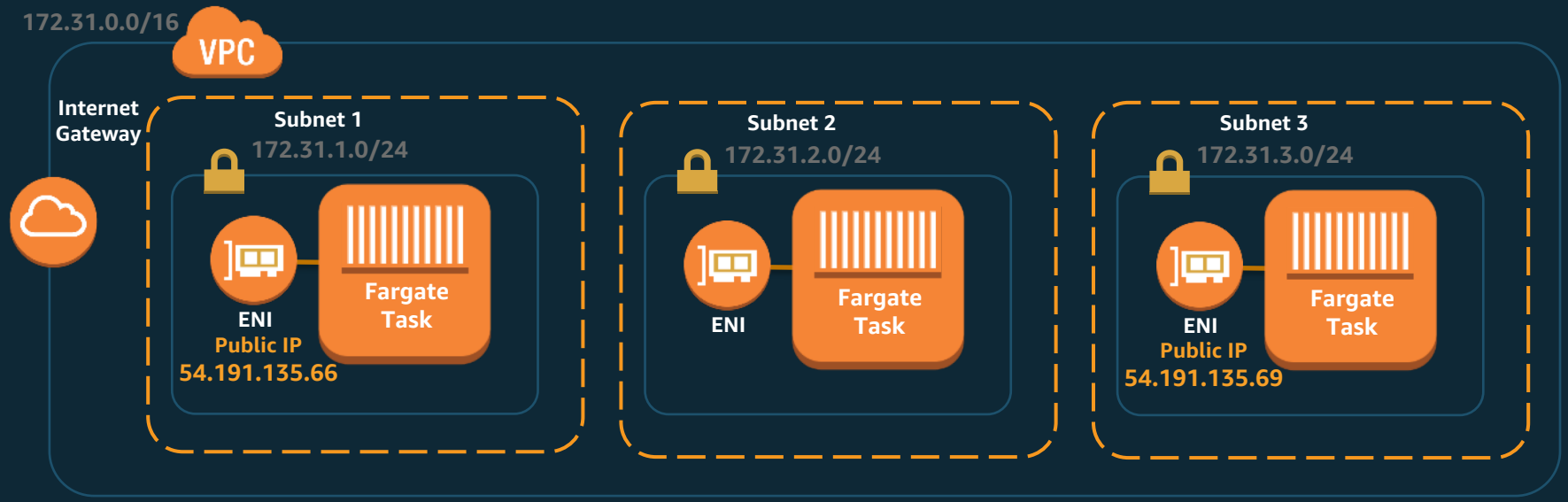


Amazon ECS Interface VPC Endpoints

- Use AWS PrivateLink
- Access Amazon ECS APIs using private IP
- Service endpoints as elastic network interfaces (ENI)
- Private connectivity to download images from Amazon ECR

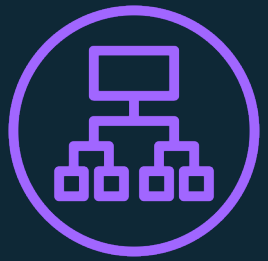


Networking with FARGATE in ECS



- *AWS VPC* Networking Mode – each task gets its own interface CNI
- Full control of network access via Security Groups and Network ACLs
- Public IP support

Routing via Application Load Balancer



Path-based routing

Allows you to define rules that route traffic to different target groups based on the path of a URL.

e.g. example.com/test , example.com/test/test1

Dynamic Port Mapping

Provides the ability to load-balance across multiple ports on the same Amazon EC2 instance. This functionality specifically targets the use of containers and is integrated into Amazon ECS.

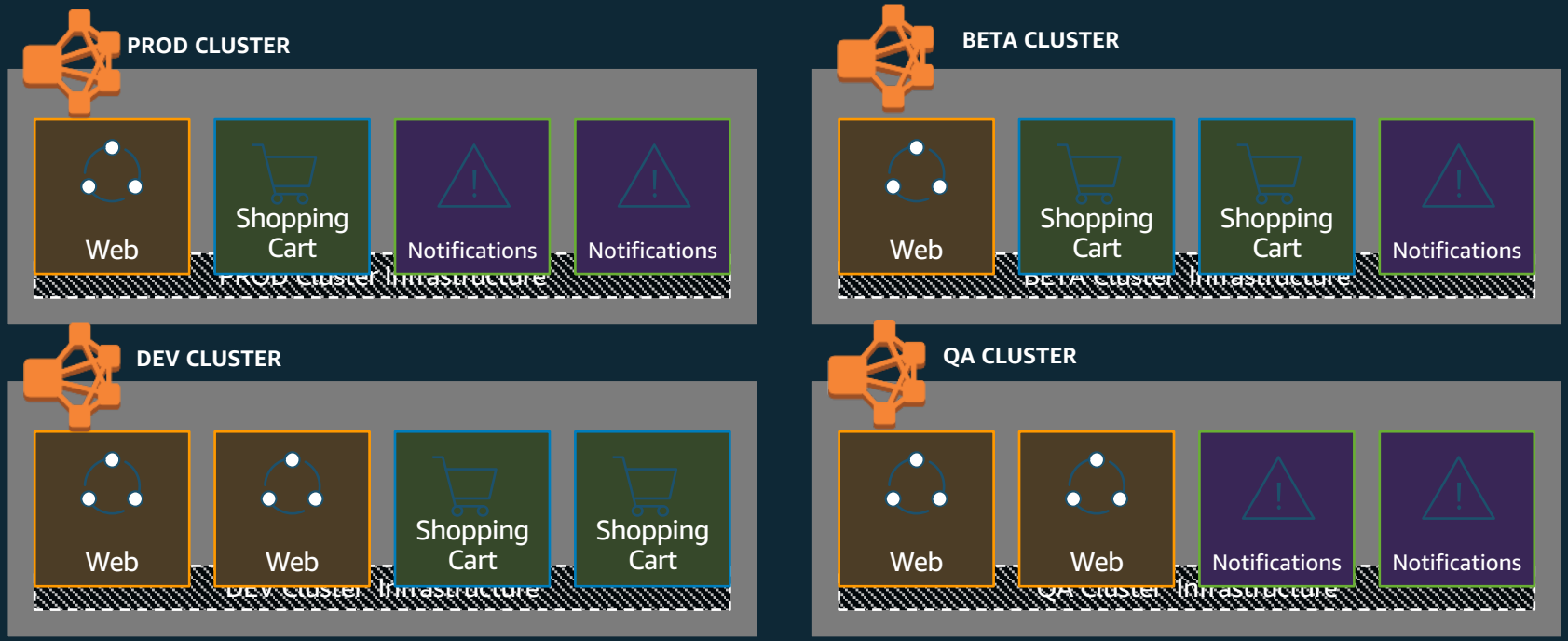
HTTP/2

WebSockets

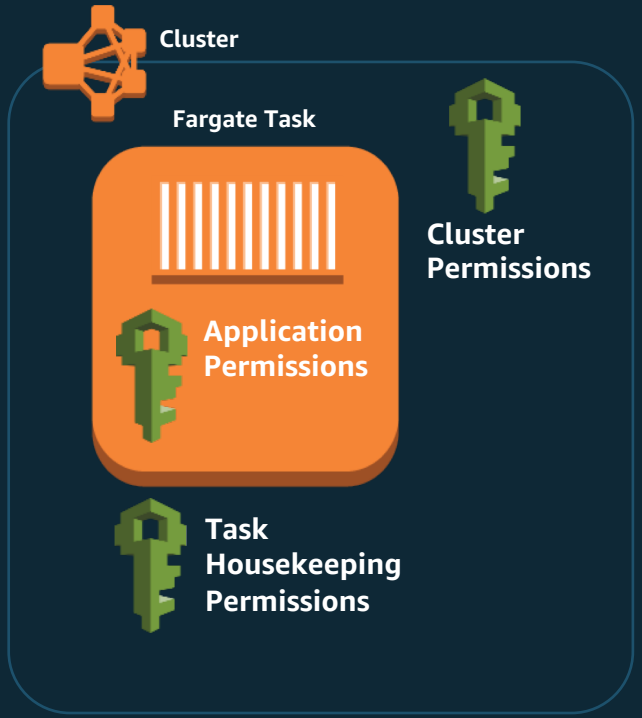
Detailed Logging

SECURITY

Cluster level isolation



Permission tiers



Cluster Permissions:

Who can run/see tasks in the cluster?

Application (Task) Permissions:

Which of my AWS resources can this application access?

Housekeeping Permissions:

What permissions do I want to grant ECS to perform?

e.g.

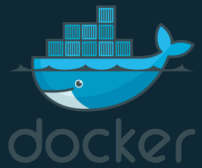
- ECR Image Pull
- CloudWatch logs pushing
- ENI creation
- Register/Deregister targets into ELB

CONTAINER REGISTRIES

Containers registery

Amazon Elastic Container Registry (ECR)

Managed AWS Docker registry service



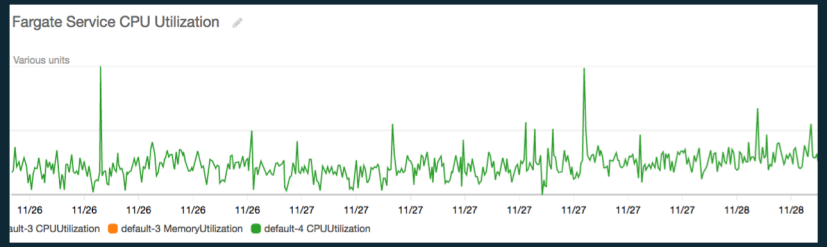
Public Repositories supported
Allow IAM users, roles, other AWS accounts

aws

Visibility and monitoring

CloudWatch Logs
CloudWatch Events supported

Service-level metrics available



Container Insights (preview)
CPU, memory, disk, and network, tasks metrics

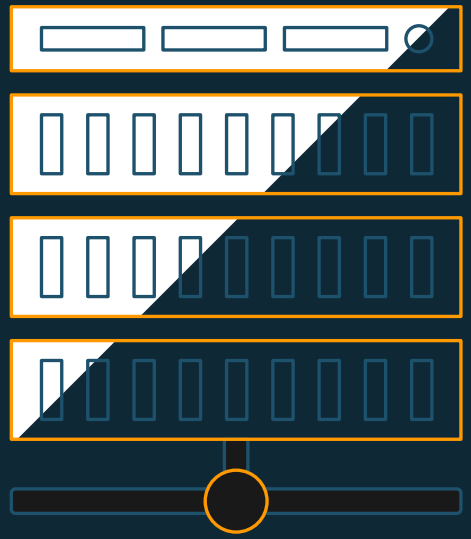
Storage

EC2 tasks

- Docker Volumes (local instance storage, EBS, EFS)
- Bind mounts

Fargate tasks

- Task storage is ephemeral !
- Container Storage Space – 10GB
- Shared volume space for containers within the task – 4GB
- Bind mounts



CONFIGURATIONS & PRICING

EC2 Launch Type Model

No additional charge

Pay for AWS resources (e.g. EC2 instances or EBS volumes) created to store and run your application

Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price

Fargate Launch Type Model

Dimensions: Task level CPU and memory

Per-second billing

Task level resources

- Configurable independently (within a range)

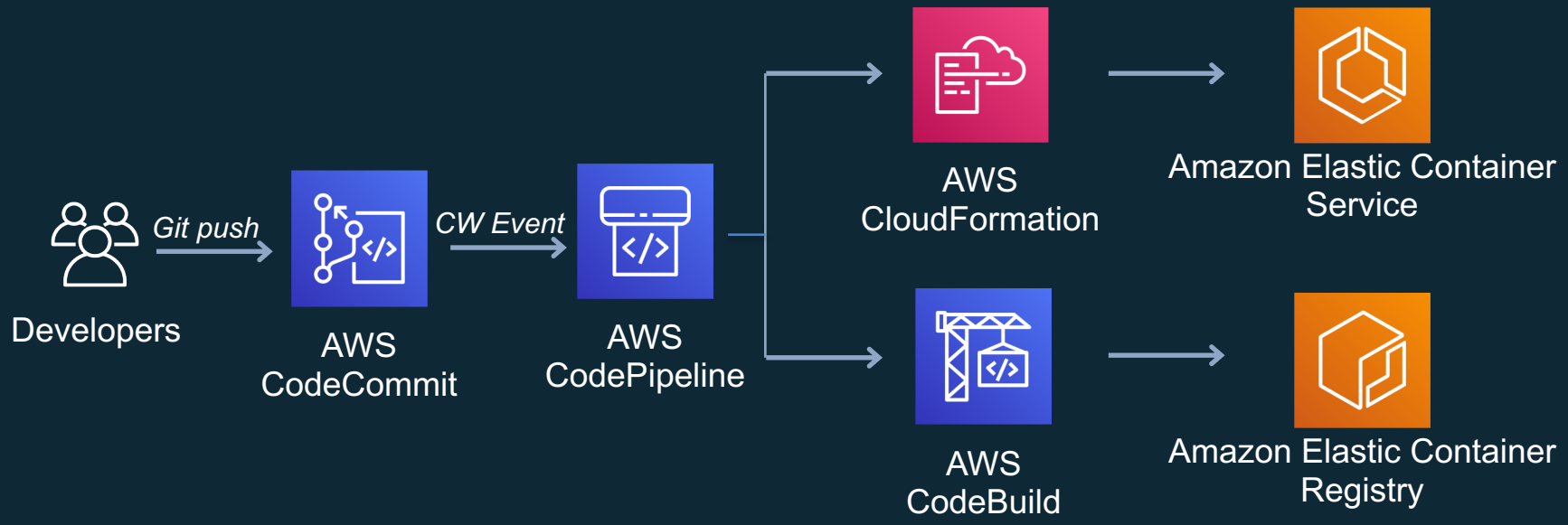
```
{
"memory": “3GB ”,
"cpu": “1 vCPU”,
"networkMode": ”AWSVPC",
"compatibilities": [”FARGATE", ”EC2"],
"placementConstraints": [],
  "containerDefinitions": [
    {

<snip>…....
```

Fargate task configuration

Flexible configuration options –
50 CPU/memory configurations

CPU	Memory
256 (.25 vCPU)	512MB, 1GB, 2GB
512 (.5 vCPU)	1GB, 2GB, 3GB, 4GB
1024 (1 vCPU)	2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB
2048 (2 vCPU)	Between 4GB and 16GB in 1GB increments
4096 (4 vCPU)	Between 8GB and 30GB in 1GB increments

CI/CD to Amazon ECS



ECS Workshop : Objectives

- Creation of 3 microservices
- Creation of 2 environments : Acceptance and Production
- Continuous Deployment of these 3 microservices

- One more thing
  - What does MU hide ? (Check the AWS Console)
  - Git commit a change in a microservice. What Happens ?

Info : Do not clean your Cloud9 Workspace at the end of the workshop

aws

https://ecsworkshop.com

Workshop EKS – 18 octobre !

https://bit.ly/2lxZAFU

Faites nous vos retours

https://bit.ly/33f83xV

Thank you