Introducing AWS Fargate

Running Containers without Infrastructure

Rohini Gaonkar | Solutions Architect

August 14, 2018

aws

MOTIVATION

At first there was

Amazon EC2

Then Docker!

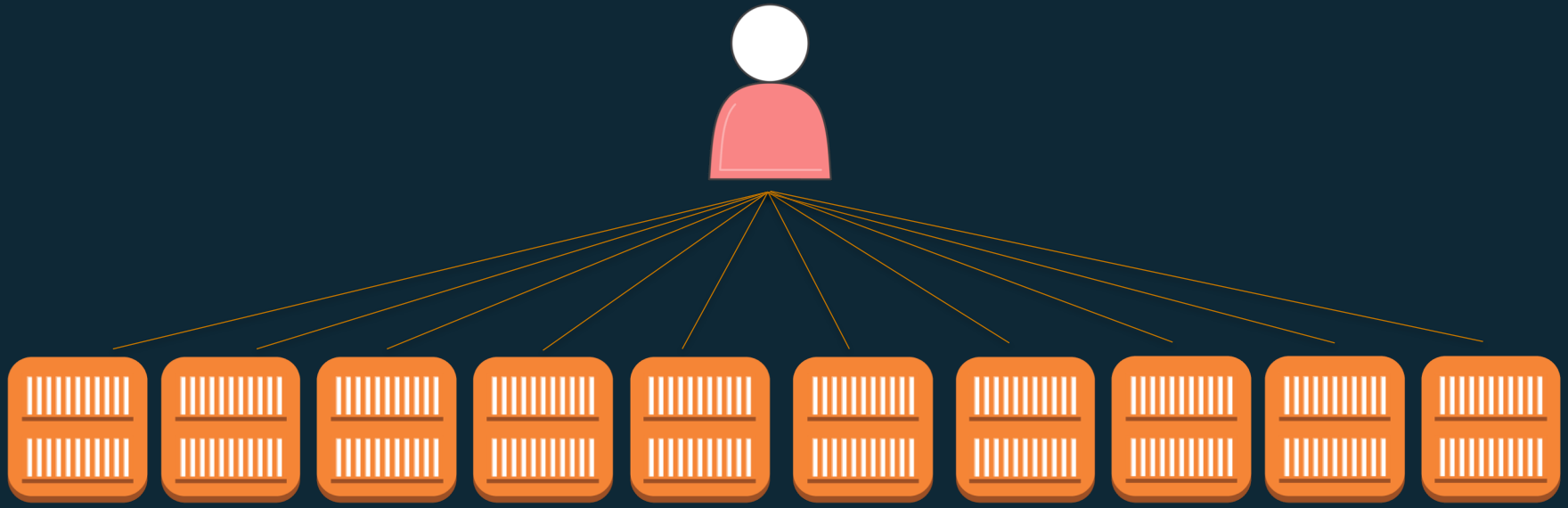
Customers started containerizing applications within EC2 instances

aws

Containers made it easy to build and scale cloud-native applications

aws

Customers needed an easier way to manage large clusters of instances and containers



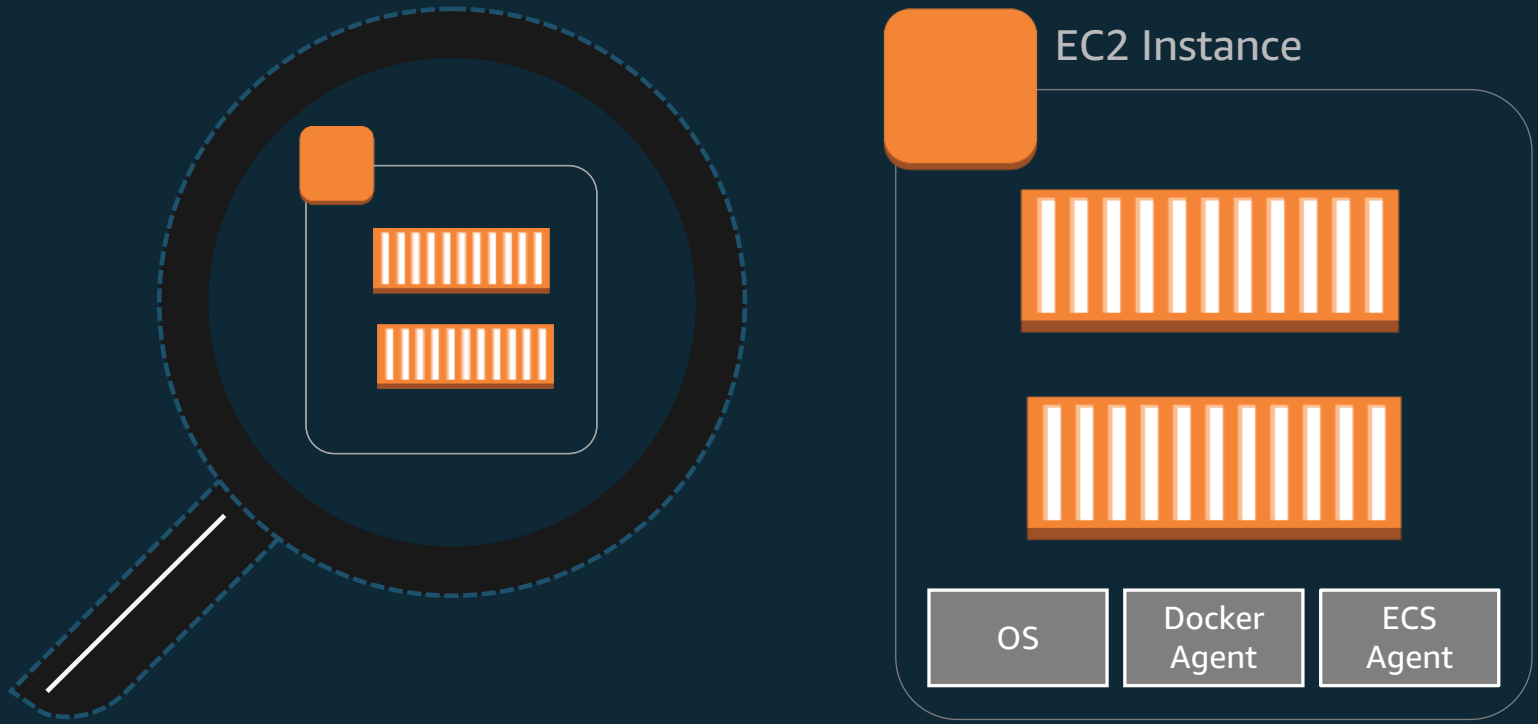
aws

AMAZON ELASTIC CONTAINER SERVICE

Cluster Management as a hosted service

aws

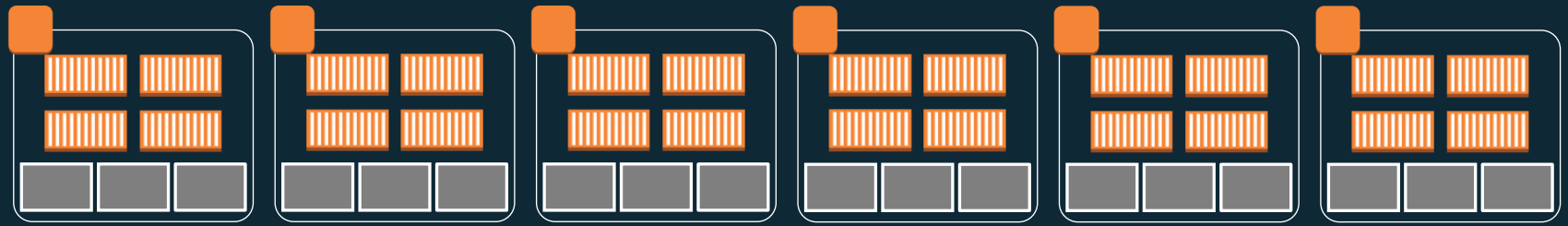
But cluster management is only half the equation…



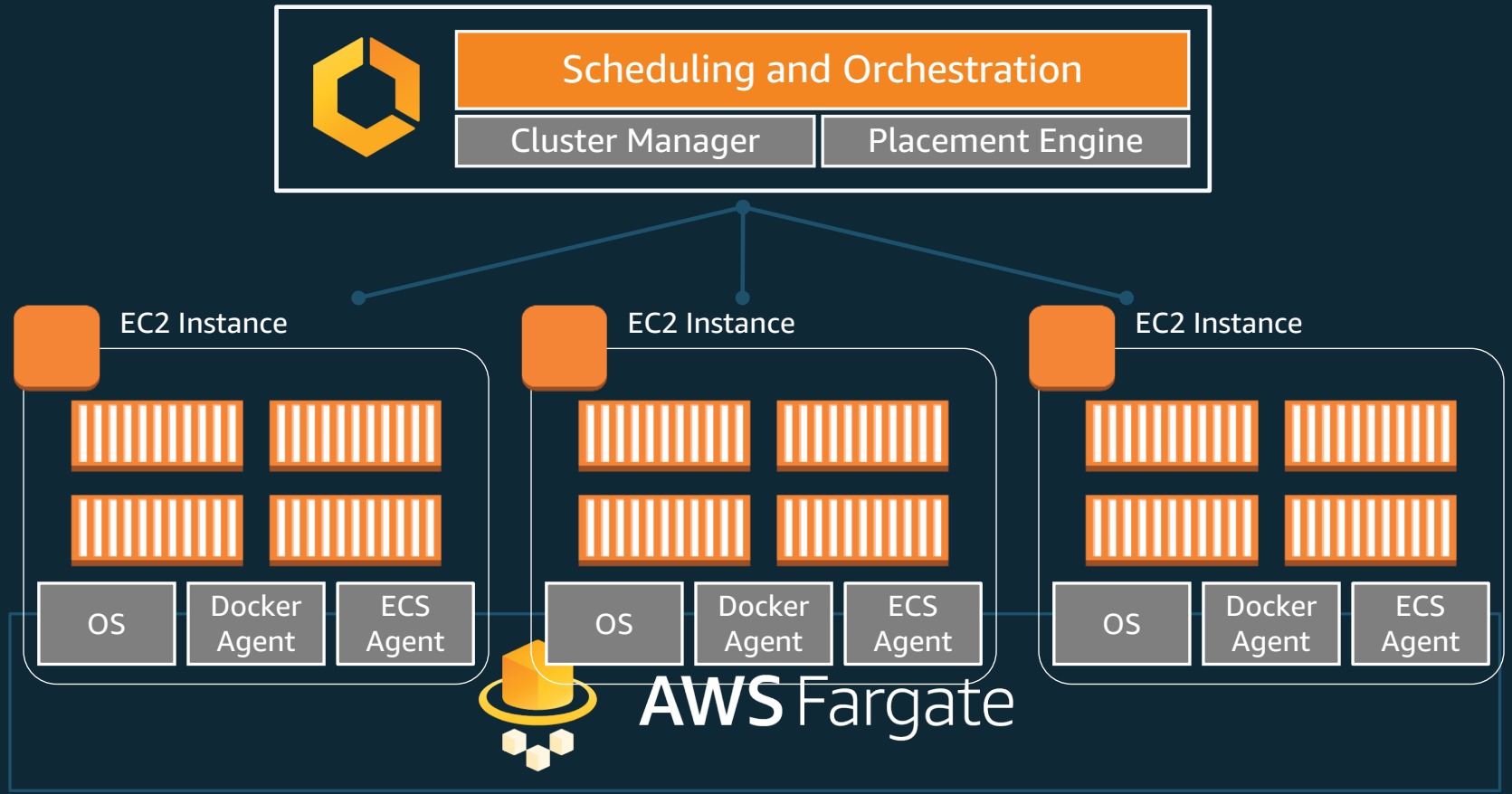
Managing instance fleets is hard work too!

Patching and Upgrading OS, agents, etc.

Scaling the instance fleet for optimal utilization



Customers wanted to run containers without having to manage EC2 instances



aws

AWS FARGATE

Your Docker Containers

NO INSTANCES TO MANAGE

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.

aws

AWS Container Services Landscape

IMAGE REGISTRY
Container Image Repository

MANAGEMENT
Deployment, Scheduling, Scaling & Management

HOSTING
Where the containers run

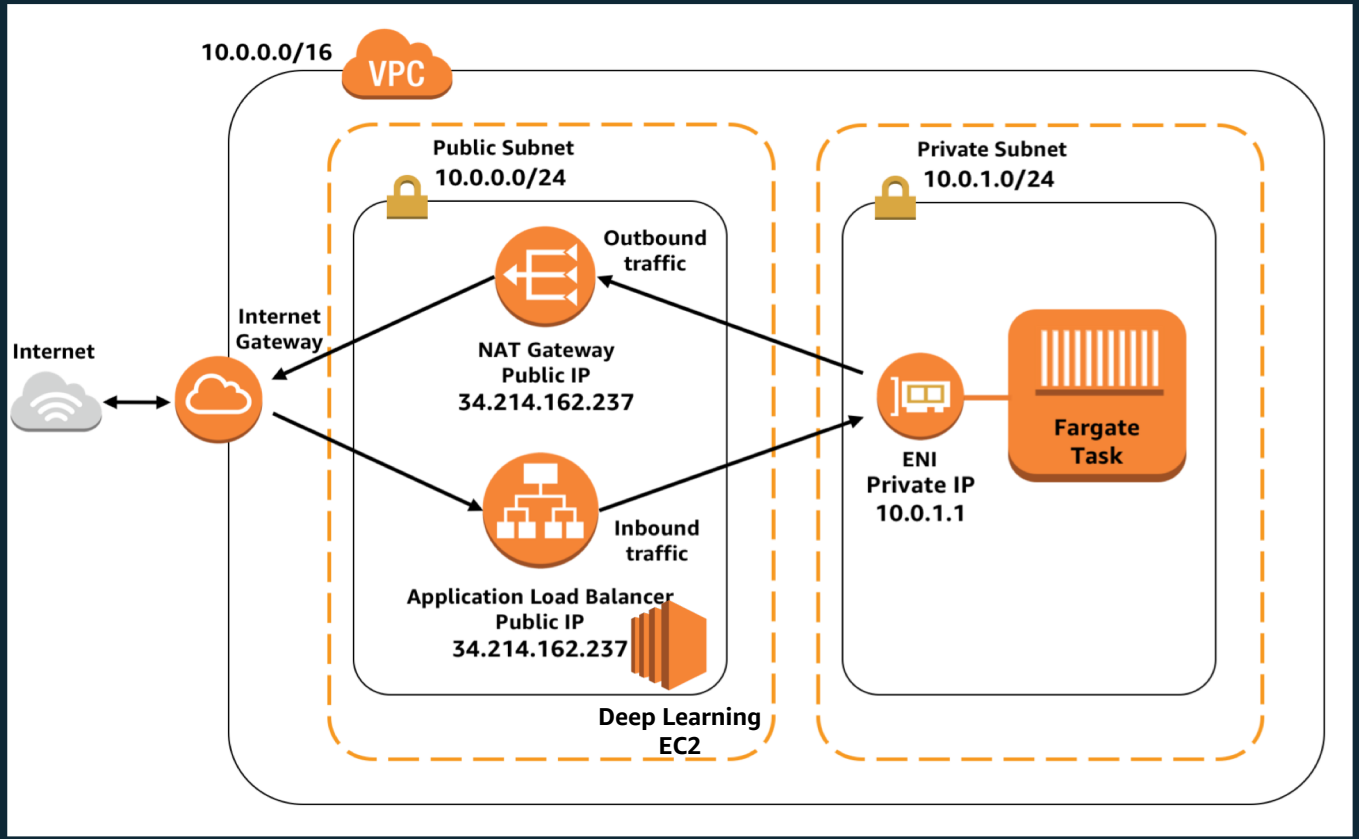
AWS Fargate

Amazon EC2

aws

WORKING WITH FARGATE

WE WILL BUILD



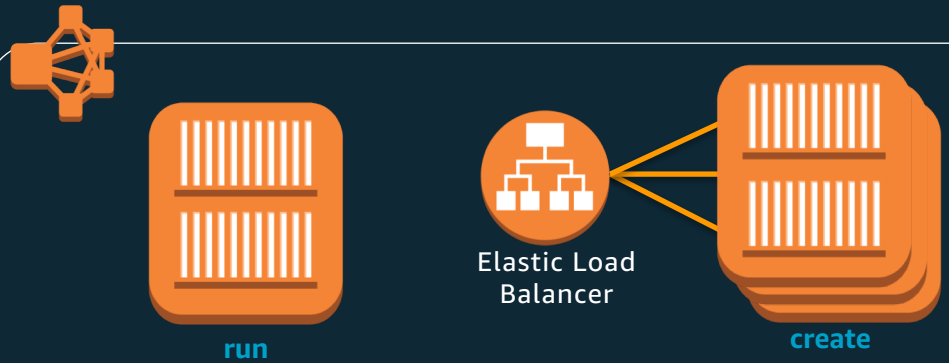
FARGATE CONSTRUCTS

CONSTRUCTS

register

Task Definition

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



run

Task

- A running instantiation of a task definition
- Use FARGATE launch type

create

Service

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

create

Cluster

- Infrastructure Isolation boundary
- IAM Permissions boundary

aws

TASK DEFINITION

Task Definition Snippet

```
{
  "family": “mxnet-model-server-
fargate-app",
  "containerDefinitions": [
    {
     "name":“mxnet-model-server-
fargate-app",
     "image":"xxx.dkr.ecr.us-east-
1.amazonaws.com/fe"
    }
  ]
}
```

Immutable, versioned document

Identified by family:version

Contains a list of up to 10 container definitions

All containers are co-located on the same host

Each container definition has:

- A name
- Image URL (ECR or Public Images)
- And more…stay tuned!

aws

REGISTRY SUPPORT

Amazon Elastic Container Registry (ECR)

Public Repositories

aws

COMPUTE

CPU & MEMORY SPECIFICATION

Units

- CPU : cpu-units. 1 vCPU = 1024 cpu-units
- Memory : MB

Task Level Resources:

- Total Cpu/Memory across all containers
- Required fields
- Billing axis

Container Level Resources:

- Defines sharing of task resources among containers
- Optional fields

Task Definition Snippet

```
{
  "family": " mxnet-model-se
fargate-app",
  "cpu": "1 vCpu",
  "memory": "2 gb",
  "containerDefinitions": [
    {
     "name":“mxnet-model-server-
fargate-app ",
     "image":"xxx.dkr.ecr.us-east-
1.amazonaws.com/fe“,
     "cpu": 256,
     "memoryReservation": 512
    }
  ]
}
```

Task Level Resources

Container Level Resources

aws

TASK 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

50 different CPU/Memory configurations to choose from

aws

PRICING

Pay for what you provision

Billed for Task level CPU and Memory

Per-second billing. 1 minute minimum

aws

NETWORKING

VPC INTEGRATION

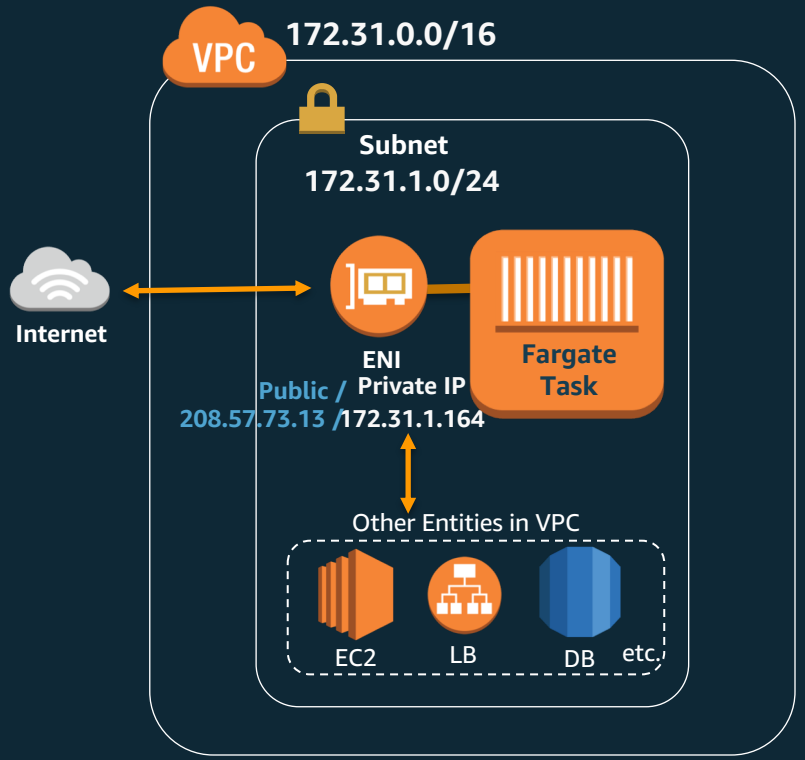
Launch your Fargate Tasks into subnets

Under the hood :

- We create an Elastic Network Interface (ENI)
- The ENI is allocated a private IP from your subnet
- The ENI is attached to your task
- Your task now has a private IP from your subnet!

You can assign public IPs to your tasks

Configure security groups to control inbound & outbound traffic



VPC CONFIGURATION

Task Definition

```
{
  "family": " mxnet-model-server-
fargate-app",
  "cpu": "1 vCpu",
  "memory": "2 gb",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
     "name":“mxnet-model-server-
fargate-app",
     "image":"xxx.dkr.ecr.us-east-
1.amazonaws.com/fe",
     "cpu": 256,
     "memoryReservation": 512
    }
  ]
}
```

Enables ENI creation & attachment to Task

Run Task

```
$ aws ecs run-task ...
    -- task-definition scorekeep:1
    -- network-configuration
         “awsvpcConfiguration = {
            subnets=[subnet1-id, subnet2-id],
            securityGroups=[sg-id]
         }”
```


aws

INTERNET ACCESS

The Task ENI is used for all inbound & outbound network traffic to and from your task

It is also used for:

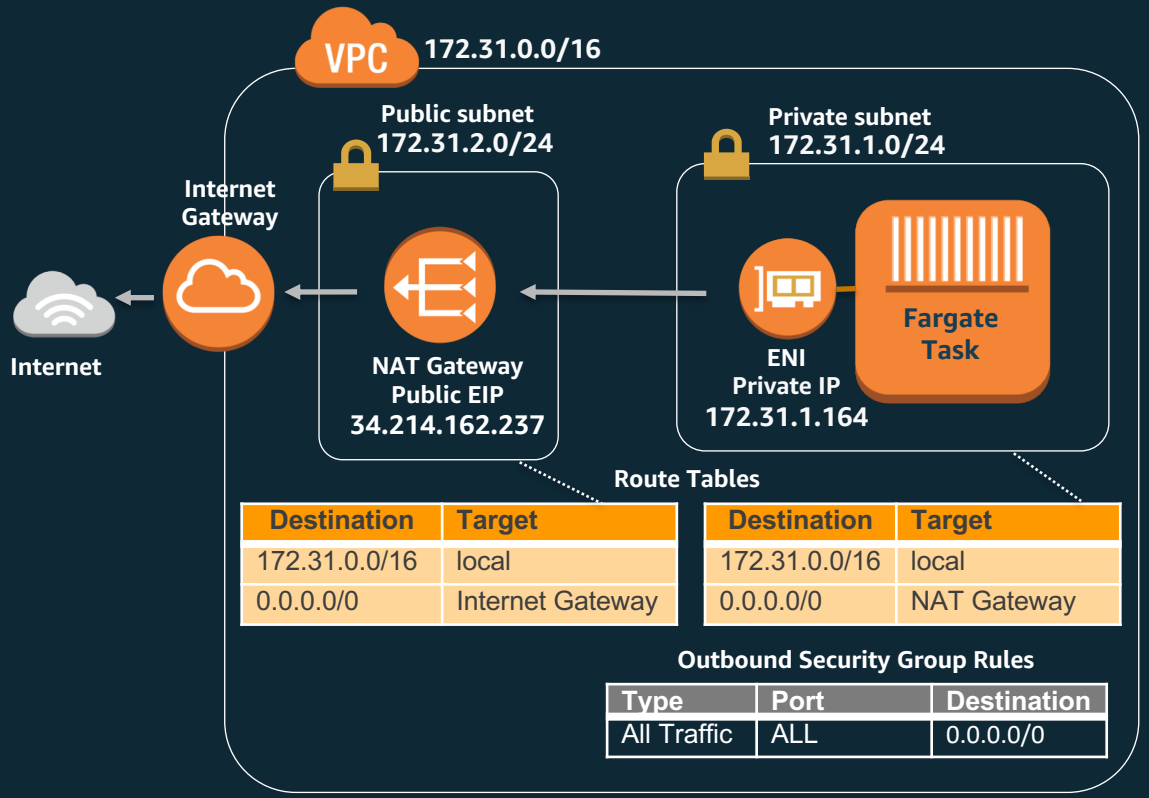
- Image Pull (from ECR or a public repository)
- Pushing logs to Cloudwatch

These endpoints need to be reachable via your task ENI

Two common modes of setup:

- Private with no inbound internet traffic, but allows outbound internet access
- Public task with both inbound and outbound internet access

PRIVATE TASK SETUP



Attach Internet Gateway to VPC

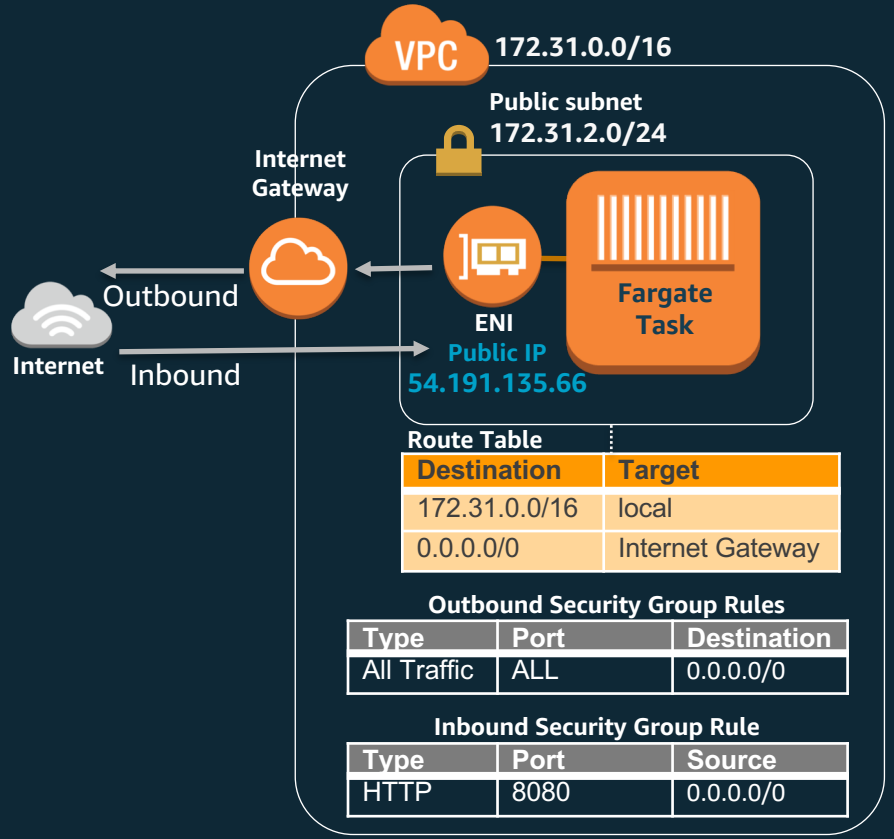
Setup a Public Subnet with
- Route to Internet Gateway
- NAT Gateway

Setup Private Subnet with
- Fargate Task
- Route to NAT Gateway

Security Group to allow outbound traffic

aws

PUBLIC TASK SETUP



Route Table

Destination	Target
172.31.0.0/16	local
0.0.0.0/0	Internet Gateway

Outbound Security Group Rules

Type	Port	Destination
All Traffic	ALL	0.0.0.0/0

Inbound Security Group Rule

Type	Port	Source
HTTP	8080	0.0.0.0/0

Run Task

```
$ aws ecs run-task ...
    -- network-configuration
          "awsvpcConfiguration = {
               subnets=[public-subnet],
               securityGroups=[sg-id],
               assignPublicIp=ENABLED
          }"
```

Launch the task into a Public subnet

Give it a public IP address

Security Group to allow the expected inbound traffic

aws

ELB CONFIGURATION

Task Definition

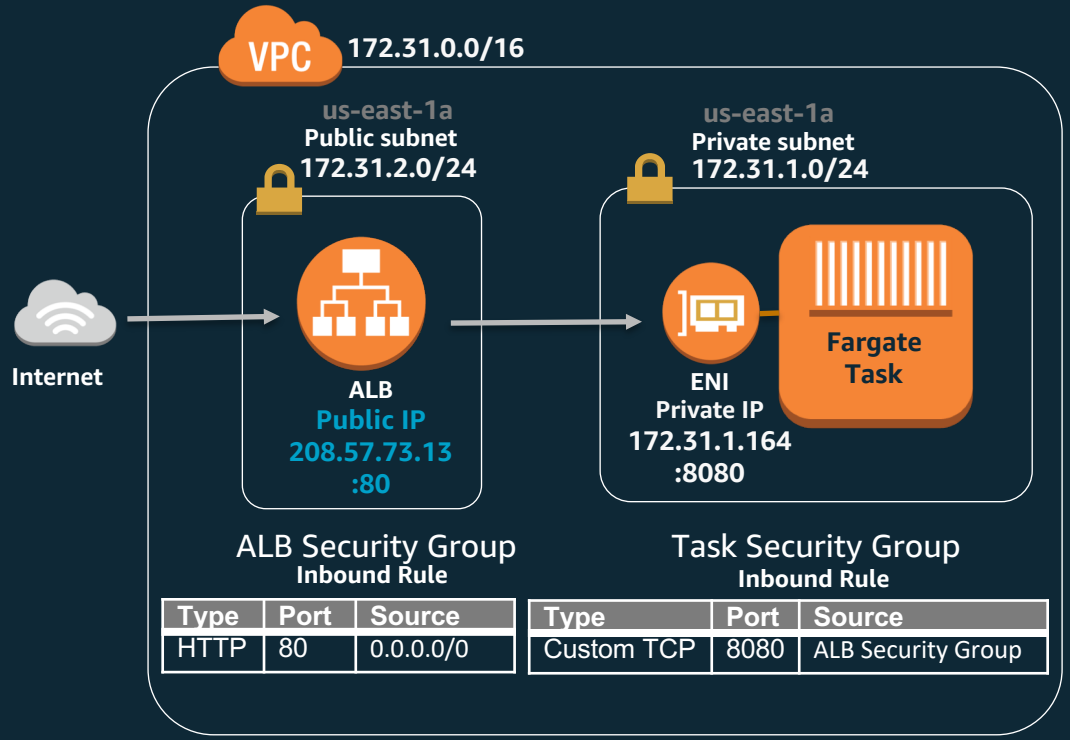
```
{
  "family": "scorekeep",
  "cpu": "1 vCpu",
  "memory": "2 gb",
  "networkMode": “awsvpc“,
  "containerDefinitions": [
    {
     "name":“mxnet-model-server-
fargate-app",
     "image":"xxx.dkr.ecr.us-east-
1.amazonaws.com/fe",
     "cpu": 256,
     "memoryReservation": 512,
     "portMappings": [
          { "containerPort": 8080 }
      ]
    } ]
}
```

Create Service

```
$ aws ecs create-service ...
    -- task-definition scorekeep:1
    -- network-configuration
         “awsvpcConfiguration = {
             subnets=[subnet-id],
             securityGroups=[sg-id]
         }”
    -- load-balancers
       “[
          {
            "targetGroupArn": “<insert arn>",
            "containerName":  “mxnet-model-server-
fargate-app",
            "containerPort":  8080
          }
        ]”
```


aws

INTERNET FACING ELB VPC SETUP



ALB Security Group
Inbound Rule

Type	Port	Source
HTTP	80	0.0.0.0/0

Task Security Group
Inbound Rule

Type	Port	Source
Custom TCP	8080	ALB Security Group

Task in private subnet with private IP

ALB in public subnet with public IP

Make sure the AZs of the two subnets match

ALB security group to allow inbound traffic from internet

Task security group to allow inbound traffic from the ALB's security group

aws

STORAGE

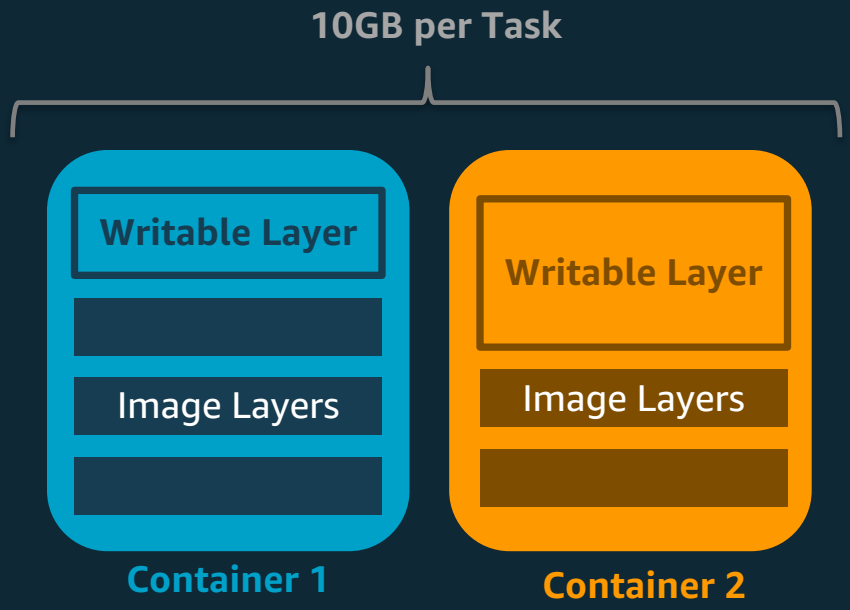
DISK STORAGE

EBS backed Ephemeral storage provided in the form of:

- Writable Layer Storage
- Volume Storage

LAYER STORAGE

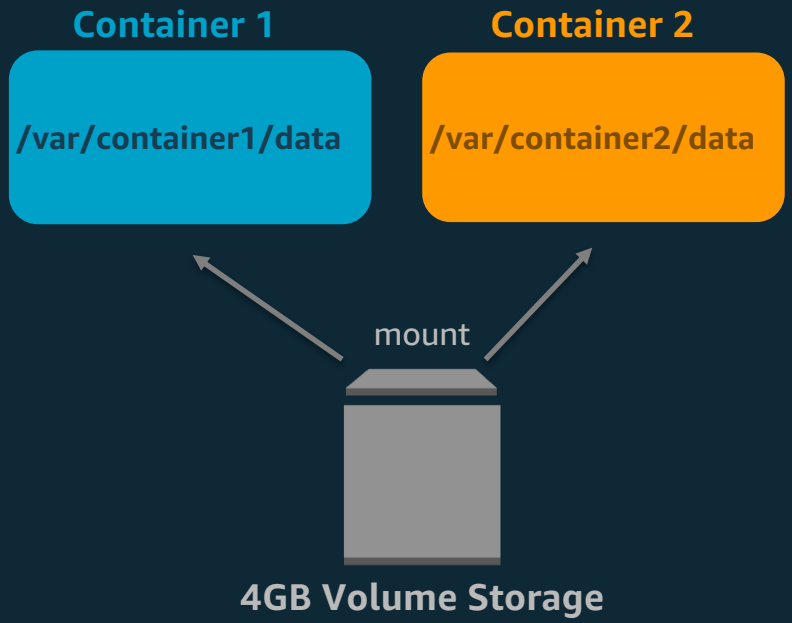
- Docker images are composed of layers
  The topmost layer is the “writable” layer to capture file changes made by the running container
- 10GB Layer storage available per task, across all containers, including image layers
- Writes are not visible across containers
- Ephemeral. Storage is not available after the task stops.



aws

VOLUME STORAGE

- Need writes to be visible across containers?
- Fargate provides 4GB volume space per task
- Configure via volume mounts in task definition
  - Can mount at different containerPaths
  - Do not specify host sourcePath
- Remember this is also ephemeral, i.e. not available after the task stops



aws

Thank You

Learn more aws.amazon.com/fargate

Try it yourself https://github.com/awslabs/eb-java-scorekeep/tree/fargate

aws