

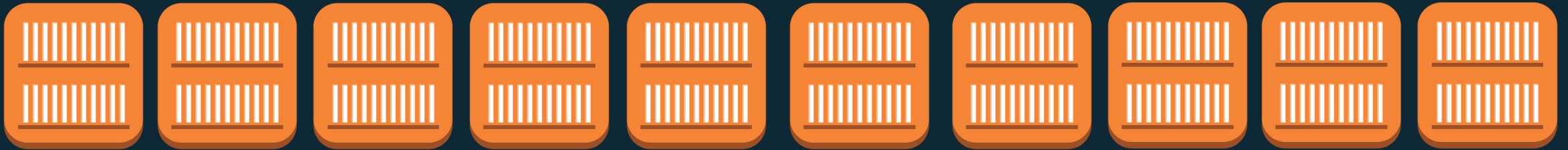
# Containerized Applications with AWS Fargate

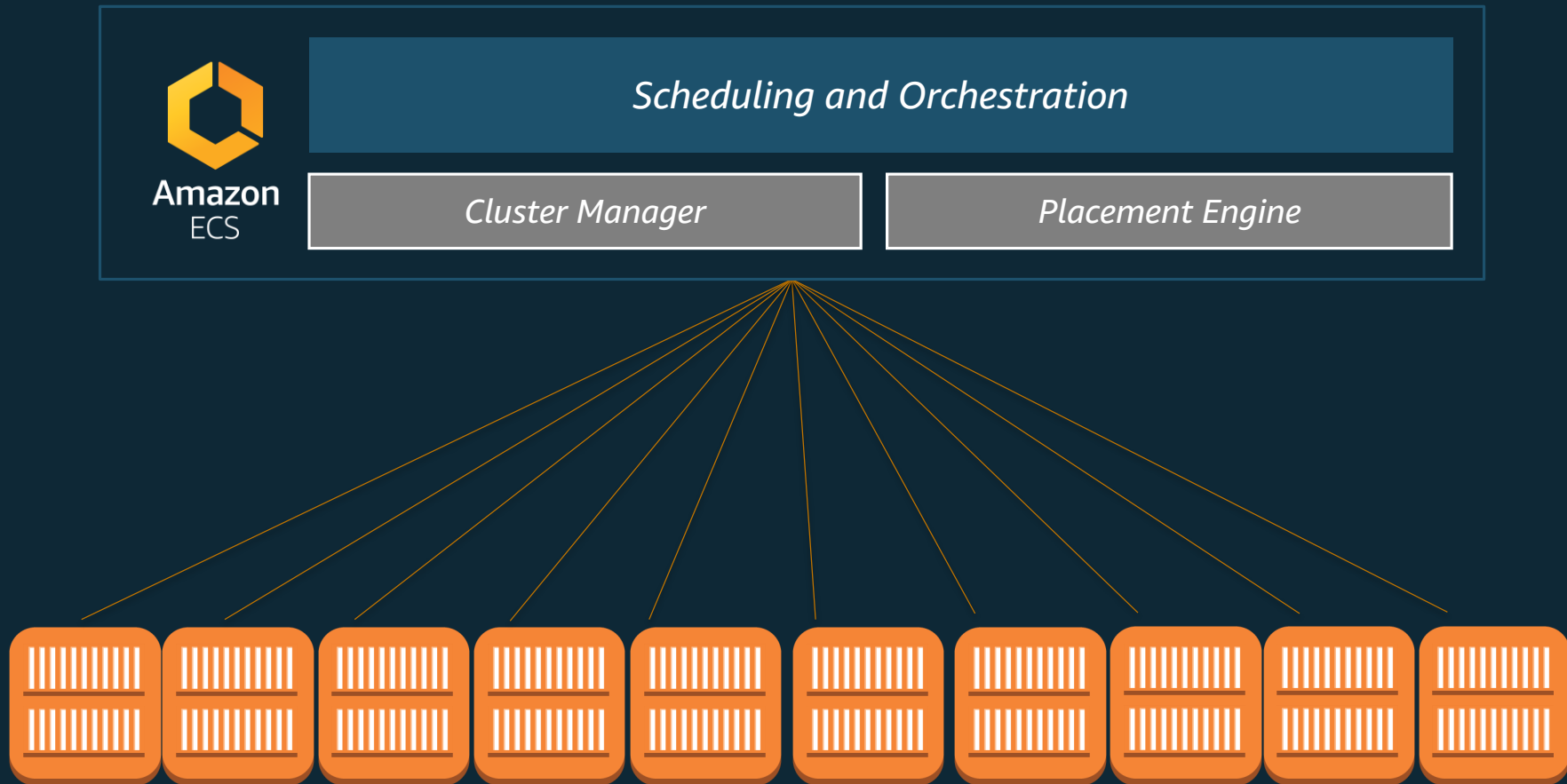
Erin McGill

Partner Solutions Architect

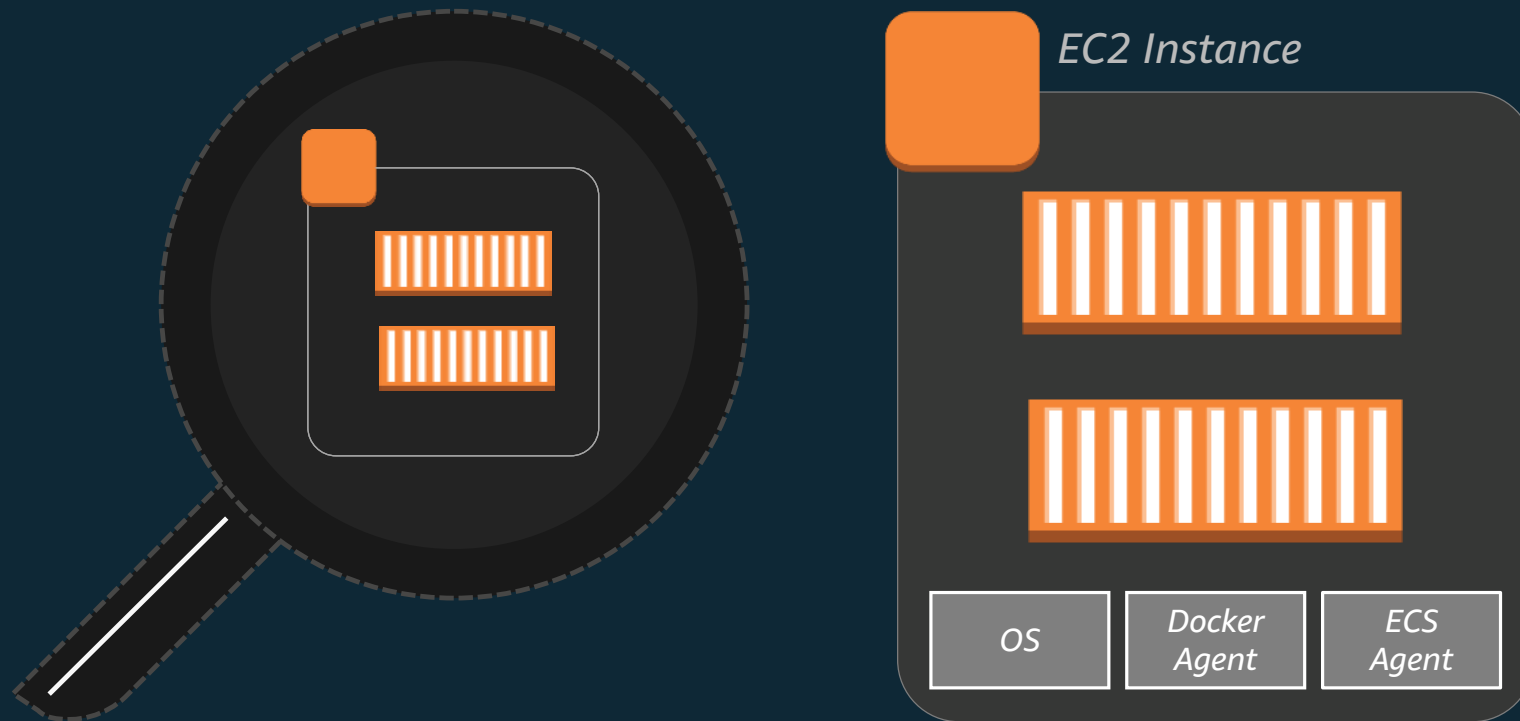
Amazon Web Services

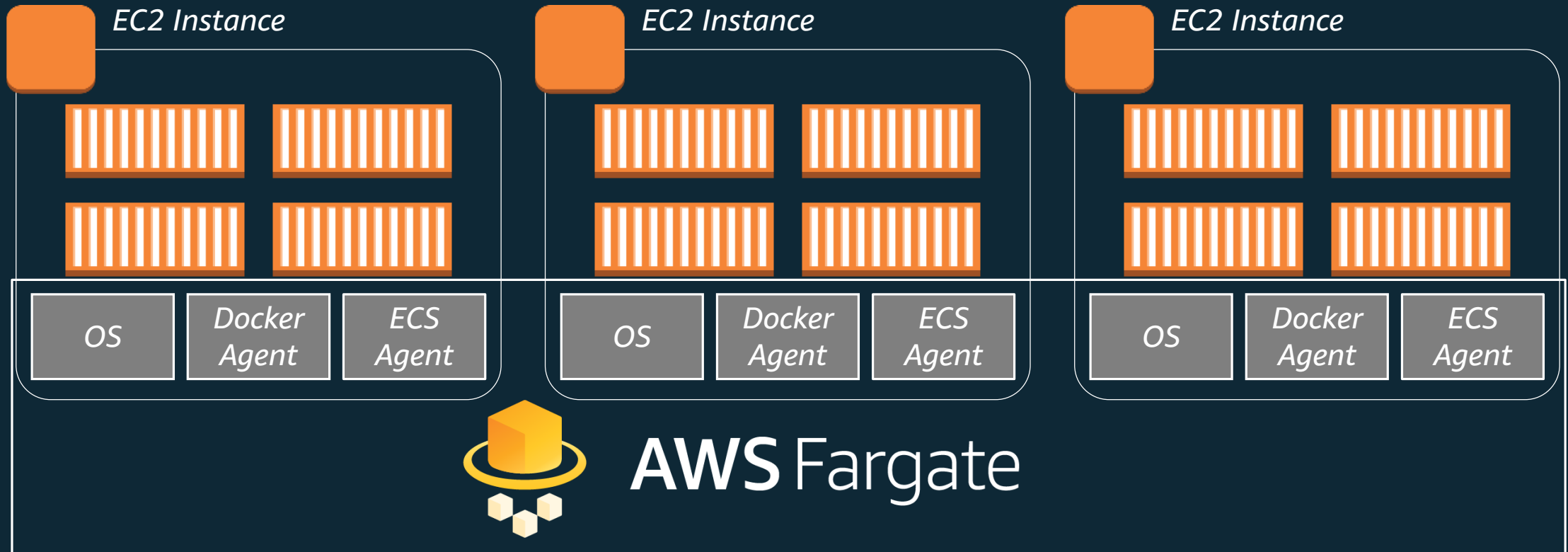
*Containers made it easy to build and scale  
cloud-native applications*





# You still end up managing more than just containers





# AWS FARGATE



**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.*

# 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**

# Fargate is now available in EU (London) region!

Region Name	Region
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (Oregon)	us-west-2
EU (Ireland)	eu-west-1
<b>EU (London)</b>	<b>eu-west-2</b>
EU (Frankfurt)	eu-central-1
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2



**Pricing:** <https://aws.amazon.com/fargate/pricing/>



# WORKING WITH FARGATE

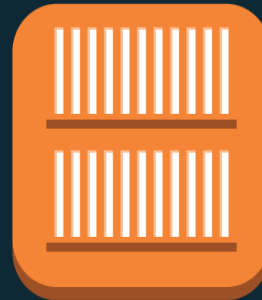
# CONSTRUCTS WHEN USING FARGATE WITH ECS

# CONSTRUCTS WHEN USING FARGATE WITH ECS



## 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*



*Elastic Load Balancer*



## create **Service**

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

## create **Cluster**

- *Infrastructure Isolation boundary*
- *IAM Permissions boundary*

# REGISTRY SUPPORT

*Amazon Elastic Container Registry (ECR)*

---



*Public Repositories supported*

---



*3<sup>rd</sup> Party Private Repositories*



# TASK CPU MEMORY CONFIGURATIONS

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

*50 different CPU/Memory configurations to choose from*

# PRICING

*Pay for what you provision*

---

*Billed for Task level CPU and Memory*

---

*Per-second billing. 1 minute minimum*

---

# NETWORKING

# VPC INTEGRATION

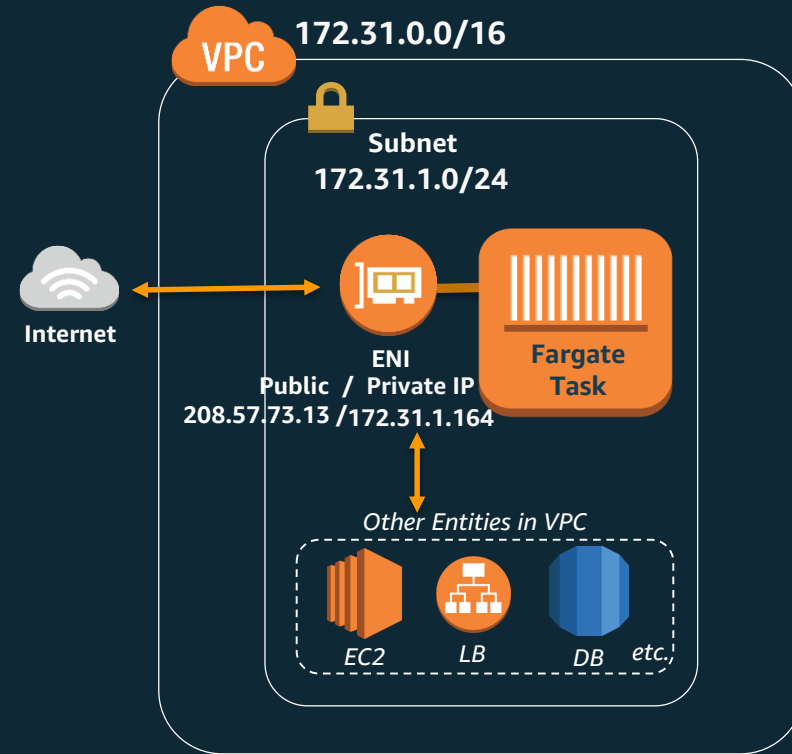
*Launch your Fargate Tasks into subnets*

*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*





# STORAGE

# DISK STORAGE

*EBS backed Ephemeral storage provided in the form of:*

---

*Writable Layer Storage*

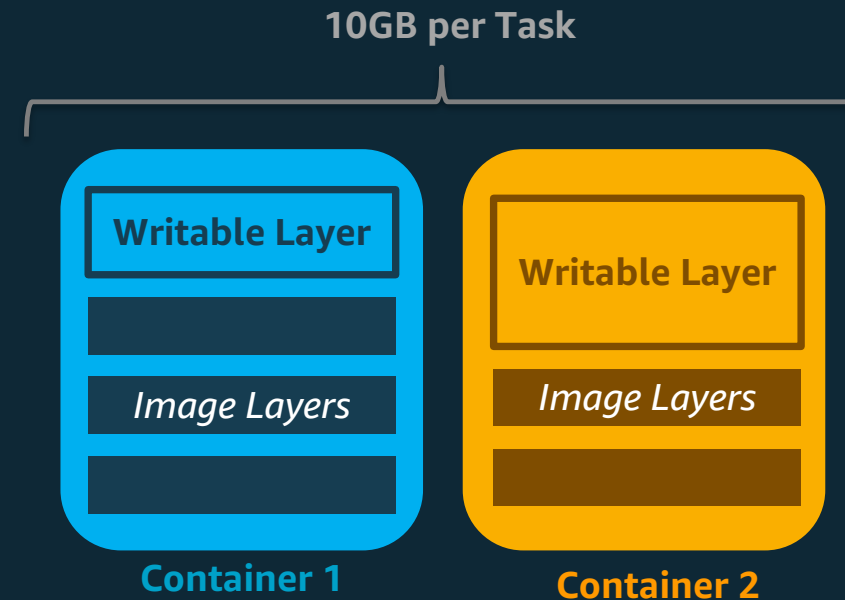
---

*Volume Storage*

---

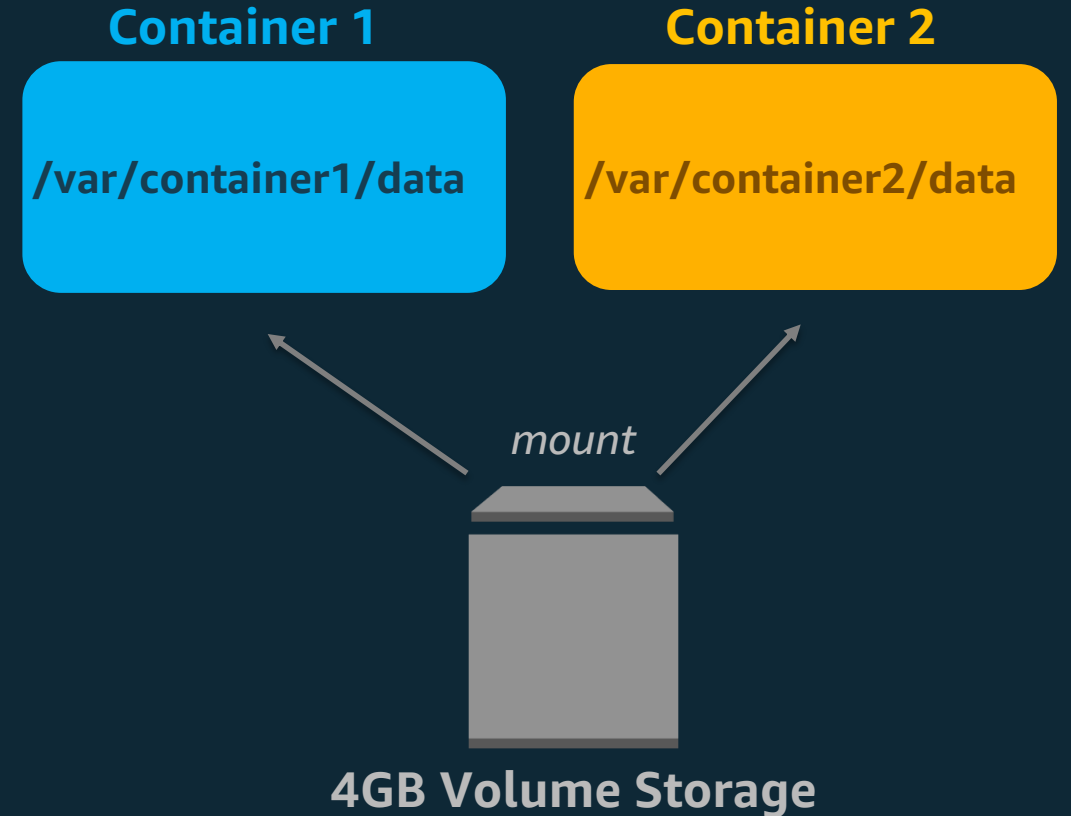
# LAYER STORAGE

- *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.*



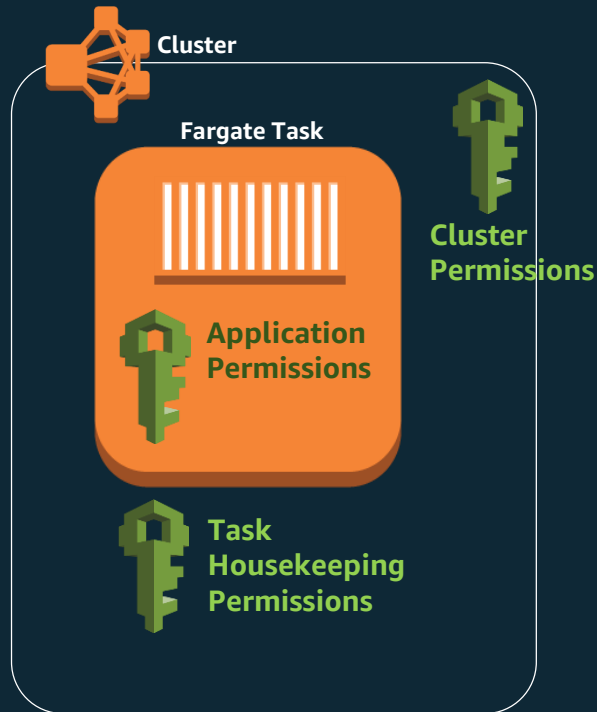
# VOLUME STORAGE

- *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*



# IAM PERMISSIONS

# PERMISSION TIERS



## *Cluster Permissions:*

*Control who can launch/describe tasks in your cluster*

## *Application Permissions:*

*Allows your application containers to access AWS resources securely*

## *Housekeeping Permissions:*

*Allows us to perform housekeeping activities around your task:*

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

# VISIBILITY & MONITORING

# CLOUDWATCH LOGS

*View logs in the ECS or Cloudwatch Console*

Clusters > default > Task: 33f9f994-9650-4ba4-ac39-644145563394

Task : 33f9f994-9650-4ba4-ac39-644145563394

Run more like this Stop

Details Logs

Select a container to view logs: scopekeep-api

scopekeep-frontend

scopekeep-api

Last updated on November 22, 2017 5:07:58 PM (2m ago)

⌕ All 30s 5m 1h 6h 1d 1w < 1-65 >

Timestamp (UTC+00:00)	Message
2017-11-22 11:26:41	2017-11-22 19:26:41.558 INFO 6 --- [ost-startStop-1] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'SimpleCORSFilter' to: [/]
2017-11-22 11:26:41	2017-11-22 19:26:41.553 INFO 6 --- [ost-startStop-1] o.s.b.w.servlet.ServletRegistrationBean : Mapping servlet: 'dispatcherServlet' to [/]
2017-11-22 11:26:40	2017-11-22 19:26:40.258 INFO 6 --- [ost-startStop-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2017-11-22 11:26:40	2017-11-22 19:26:40.347 INFO 6 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 17798 ms
2017-11-22 11:26:39	2017-11-22 19:26:39.353 INFO 6 --- [ main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.4
2017-11-22 11:26:39	2017-11-22 19:26:39.351 INFO 6 --- [ main] o.apache.catalina.core.StandardService : Starting service Tomcat
2017-11-22 11:26:39	2017-11-22 19:26:39.152 INFO 6 --- [ main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat initialized with port(s): 5000 (http)
2017-11-22 11:26:22	2017-11-22 19:26:22.647 INFO 6 --- [ main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@28...
2017-11-22 11:26:21	2017-11-22 19:26:21.357 INFO 6 --- [ main] scorekeep.Application : Starting Application with PID 6 (/scorekeep-api-1.0.0.jar started by root in /)
2017-11-22 11:26:21	2017-11-22 19:26:21.449 INFO 6 --- [ main] scorekeep.Application : No active profile set, falling back to default profiles: default
2017-11-22 11:26:19	:: Spring Boot :: (v1.4.0.RELEASE)



# Container Health Checks

- Define *custom* health check commands in the ECS Task Definition

▼ Advanced container configuration

**HEALTHCHECK**

Command  ⓘ

Interval\*  second(s) ⓘ

Timeout\*  second(s) ⓘ

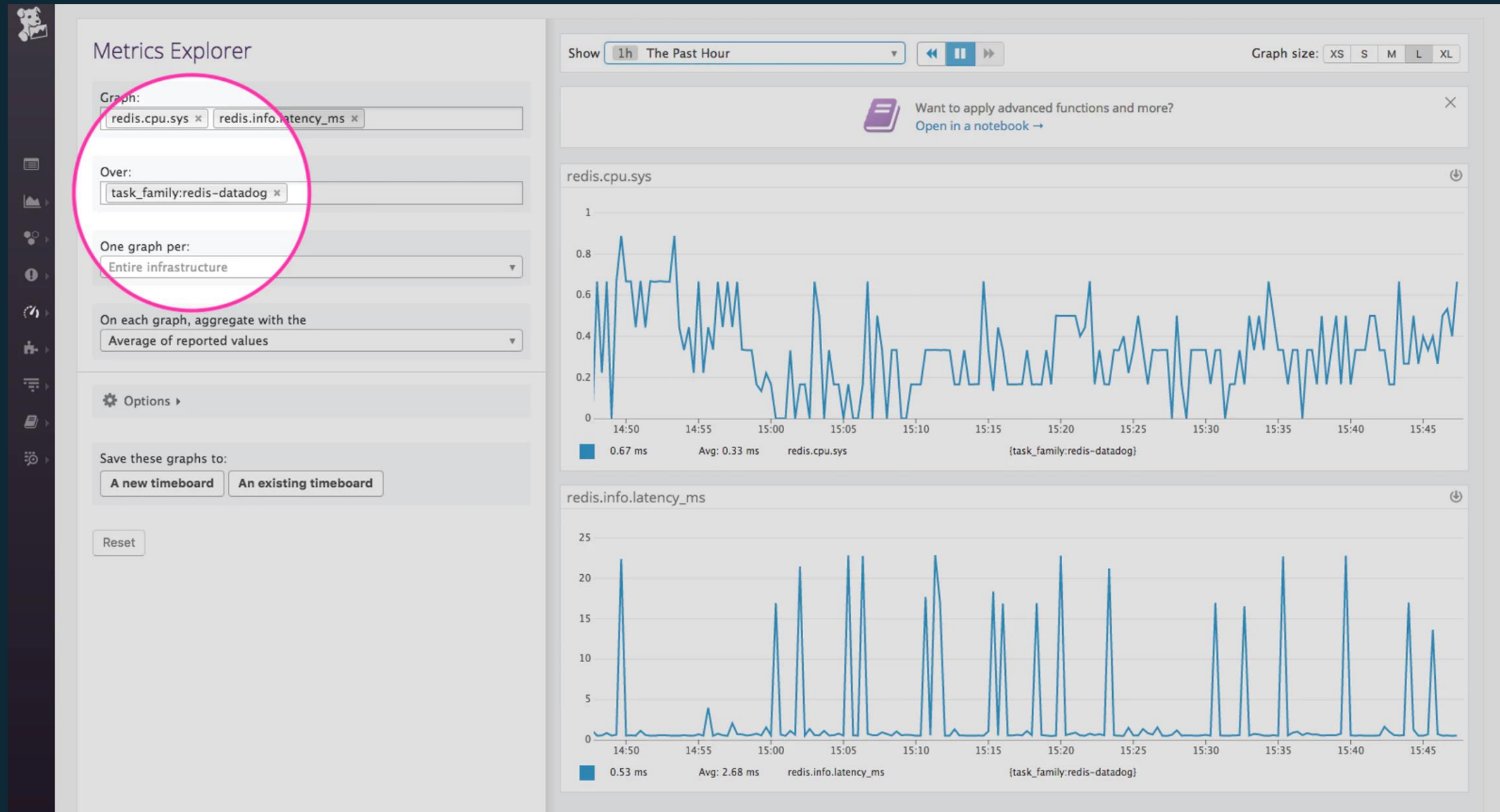
Start period\*  second(s) ⓘ

Retries\*  ⓘ

# Task Metadata

- Query environmental data and statistics for running tasks from within the Task!  
Enables monitoring tools like Datadog, etc
- Endpoints available:
  - Task Level (for all containers)
    - 169.254.170.2/v2/metadata – Metadata JSON for Task
    - 169.254.170.2/v2/stats - Docker stats JSON for all containers in the Task
  - Container Level
    - 169.254.170.2/v2/metadata/<container-id>
    - 169.254.170.2/v2/stats/<container-id>

# Monitoring with Datadog Autodiscovery and Fargate



# Thank you!

Learn more at

[aws.amazon.com/Fargate](https://aws.amazon.com/Fargate)

[www.datadoghq.com/blog/monitor-aws-fargate/](https://www.datadoghq.com/blog/monitor-aws-fargate/)