Securing Containerized Workloads on Amazon ECS

Ravindu Nirmal Fernando

AWS User Group Colombo | August Meetup | 2023

About Me

Head of DevSecOps @ Emojot Inc.

2x AWS Community Builder

AWS Solution Architect - Professional

Certified Kubernetes Administrato

MSc in Computer Science (Specialized in Cloud Computing)

01

Security & Compliance in AWS

02

Shared Responsibility Model

03

AWS Container Service Landscape

04

Amazon ECS Launch Types & Security Model Overview

05

Amazon ECS Components Overview

06

Enhancing Security Posture: Key Features of Amazon ECS

07

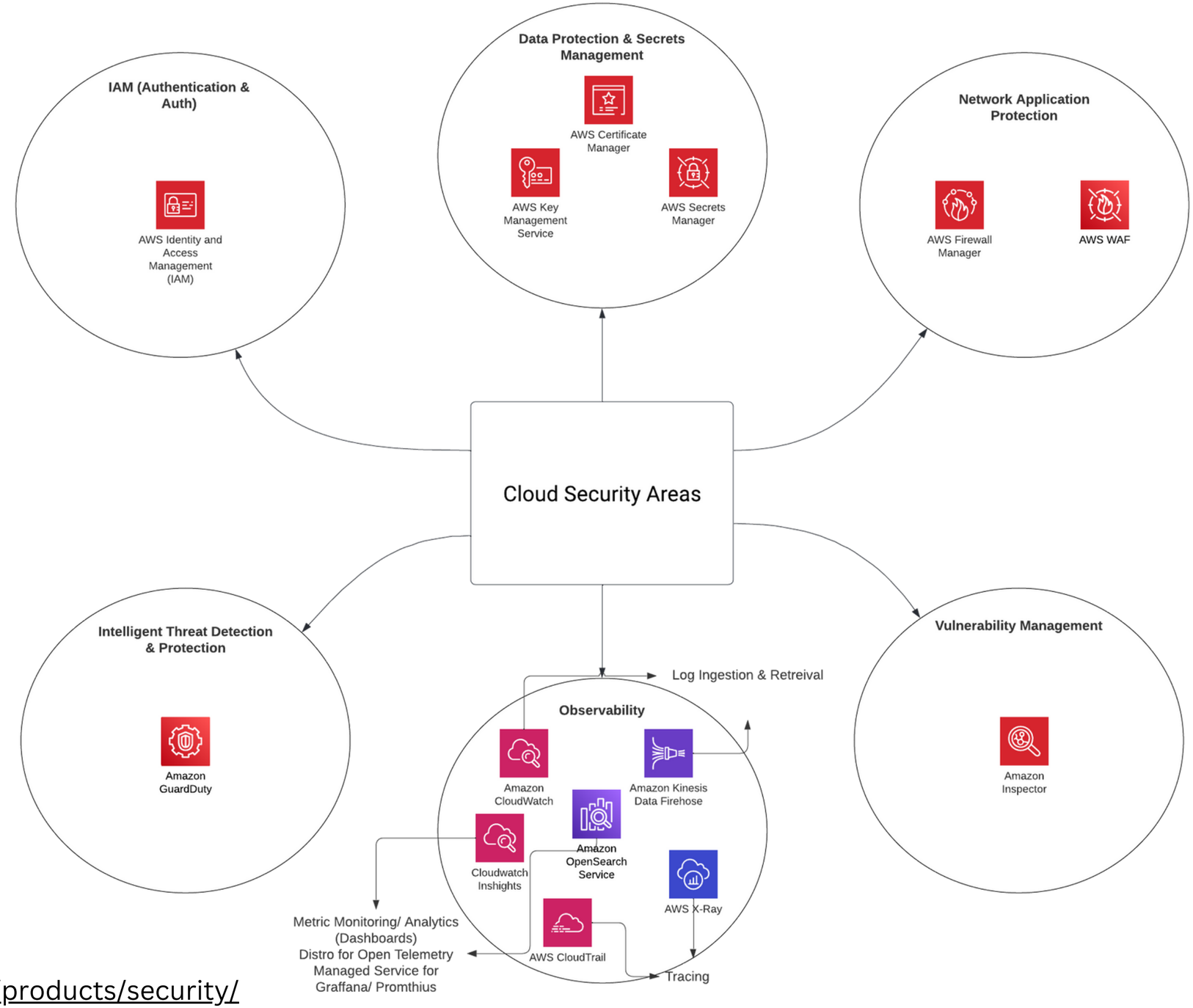
Demo

Agenda

Let's go!!!

1

Security and Compliance on AWS



https://aws.amazon.com/products/security/

Amazon ECS Compliance

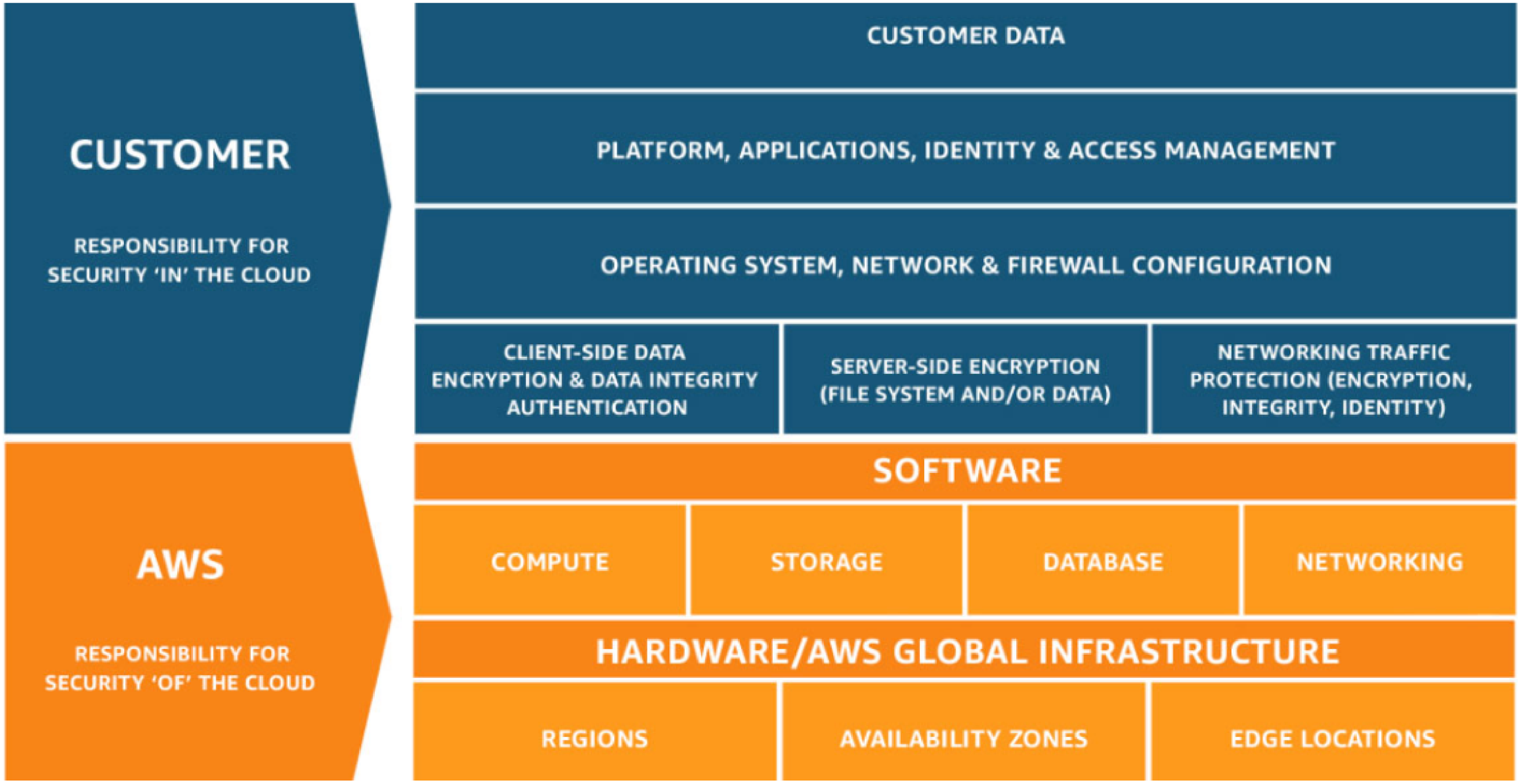
ECS is certified on:

PCI, HIPAA, SOC, ISO/CSA Star, ISMAP, MTCS, C5, HITUST CSF, FINMA, FEDRAMP, DoD CC, IRAP, K-ISMS, ENS-High, OSPAR, GSMA, CCCS etc..

https://aws.amazon.com/compliance/services-in-scope/

2

Shared Responsibility Model


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

SRM also extends to IT controls…

Inherited Controls – Controls which a customer fully inherits from AWS.

- Physical and Environmental controls

Shared Controls – Controls which apply to both the infrastructure layer and customer layers, but in completely separate contexts or perspectives. In a shared control, AWS provides the requirements for the infrastructure and the customer must provide their own control implementation within their use of AWS services. Examples include:

- Patch Management – AWS is responsible for patching and fixing flaws within the infrastructure, but customers are responsible for patching their guest OS and applications.
- Configuration Management – AWS maintains the configuration of its infrastructure devices, but a customer is responsible for configuring their own guest operating systems, databases, and applications.
- Awareness & Training - AWS trains AWS employees, but a customer must train their own employees.

Customer Specific – Controls which are solely the responsibility of the customer based on the application they are deploying within AWS services. Examples include:

- Service and Communications Protection or Zone Security which may require a customer to route or zone data within specific security environments.

https://aws.amazon.com/compliance/shared-responsibility-model/

3

AWS Container Service Landscape

Application Networking

Service discovery and service mesh

AWS Cloud Map

Amazon ECS Connect

Management

Deplyoment, scheduling, scaling and cofiguration of containerized applications

Amazon ECS

Amazon EKS

Hosting

Where the containers run

Amazon EC2

AWS Fargate

Image Registry

Container image repositroy

Amazon ECR

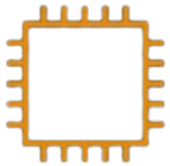
4

Amazon ECS Launch Types & Security Model Overview

ECS on EC2 - Security Model

Resource Isolation is at EC2 compute instance level

Dynamic Patch Mechanism, so instance should not start/stop every time. Have some manual steps involved



Hardware virtualization

isolated compute

Network isolation*

dedicated ENIs

Storage isolation

dedicated storage

Credentials isolation*

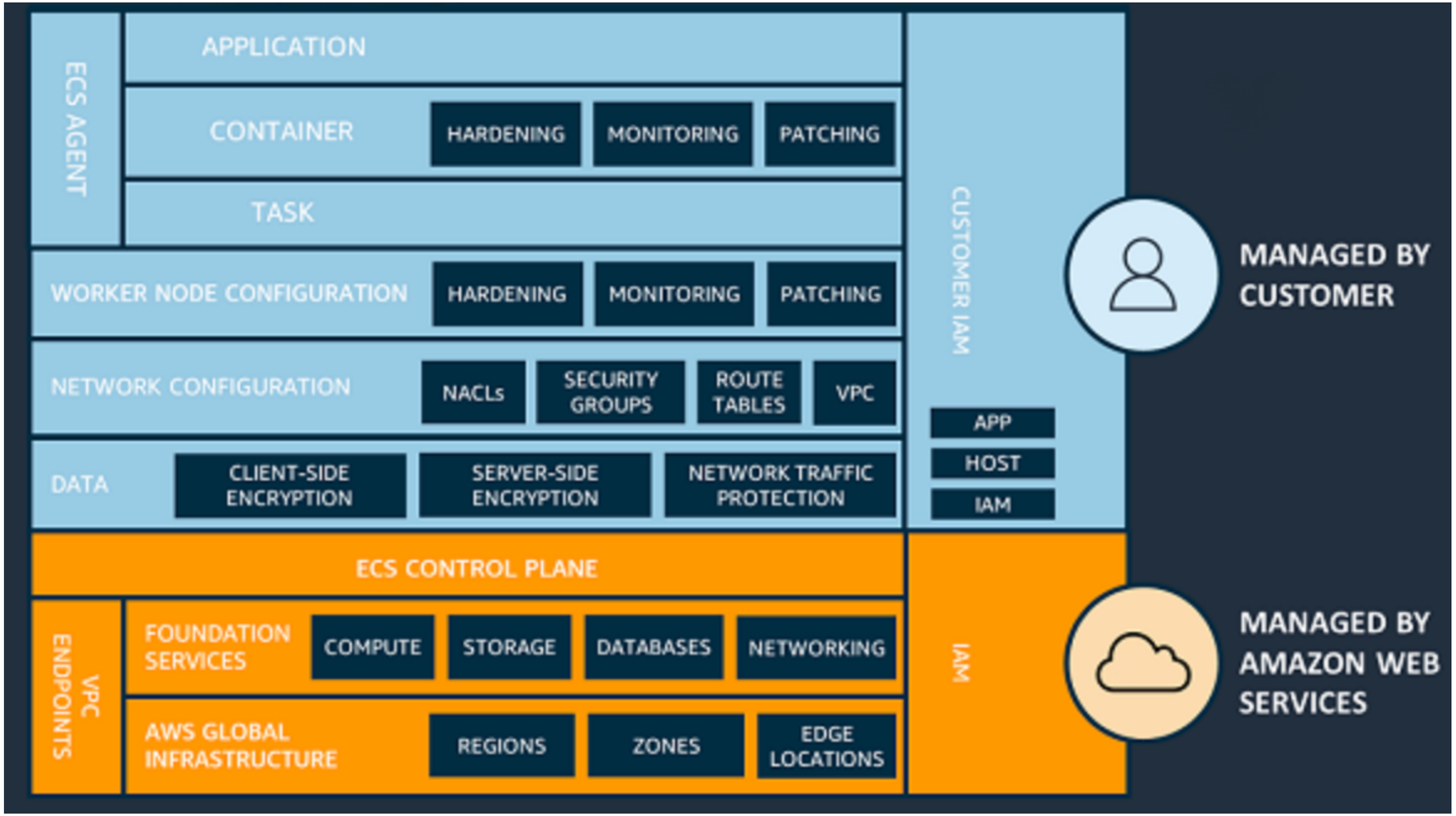
Isolated long-term credentials

Hands-off patching

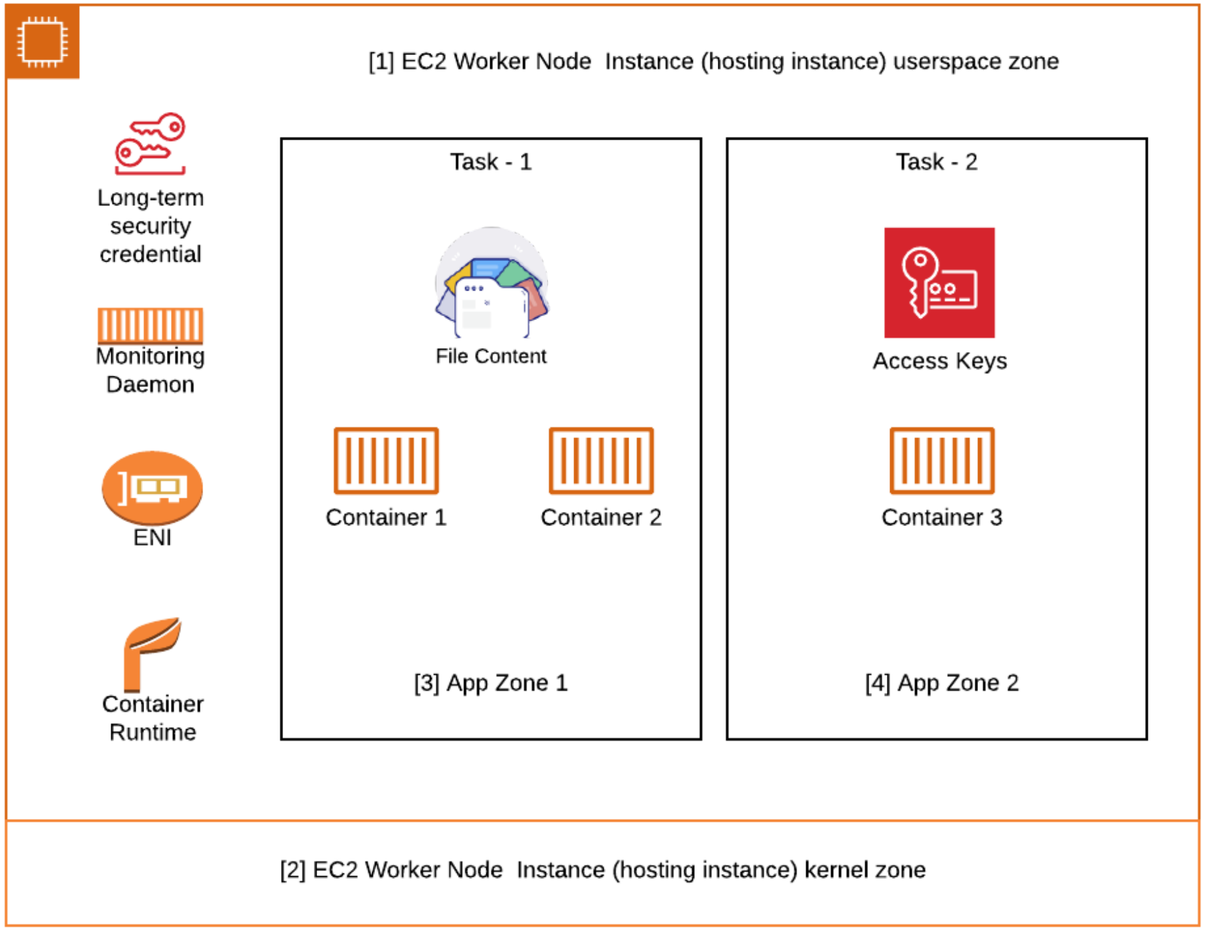
Infrastructure always patched

*ECS can further isolate ENI, Credentials into the Task levels

ECS on EC2 - Shared Responsibility Model



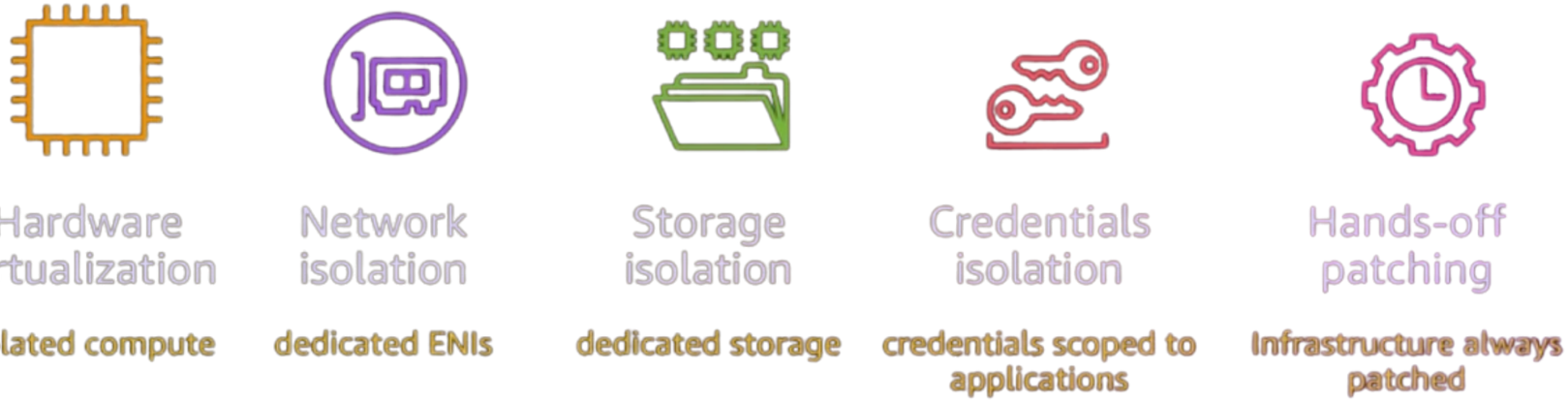
ECS on EC2 - Trust Boundaries



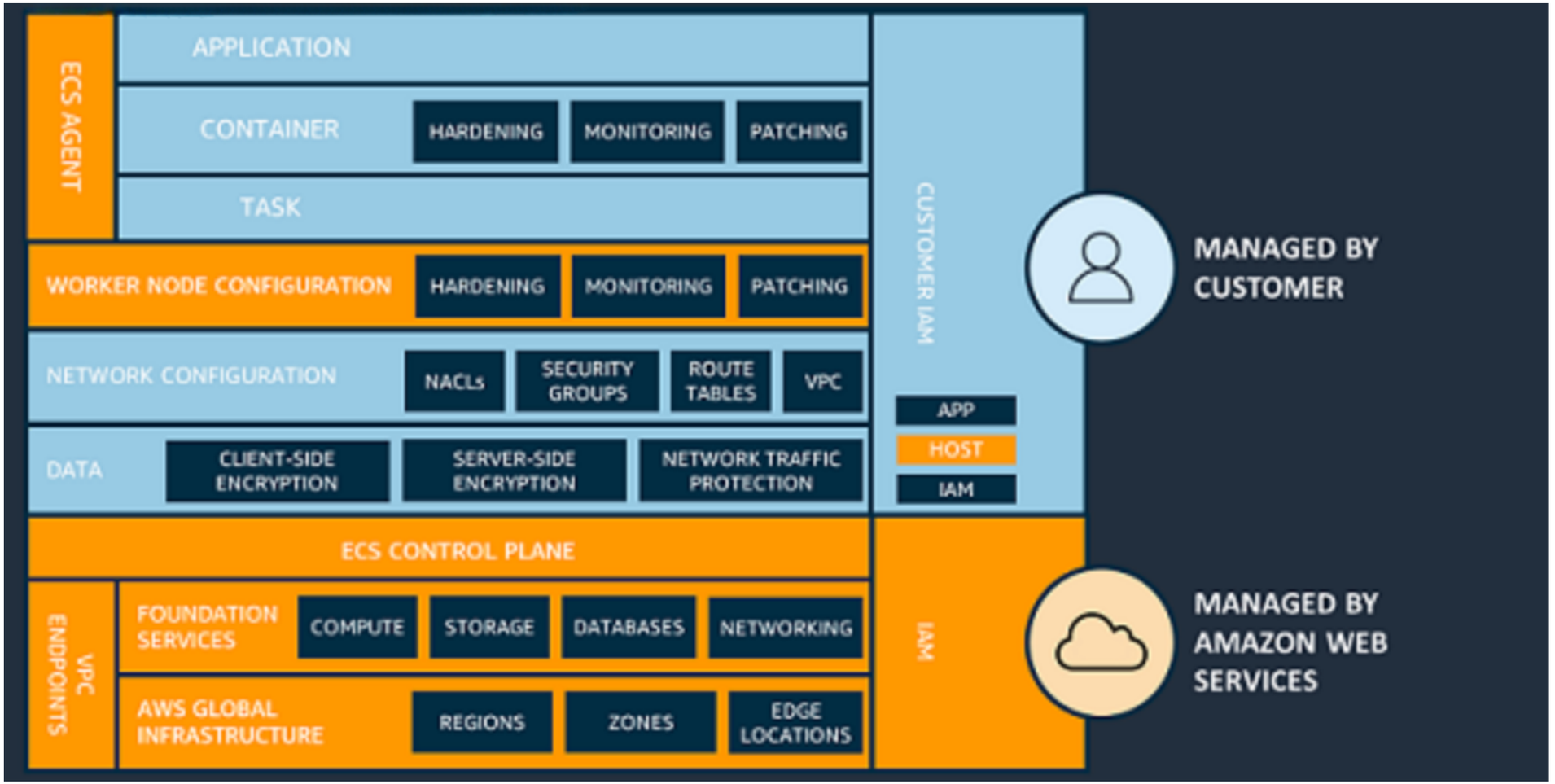
ECS on Fargate - Security Model

Resource Isolation is at Task level

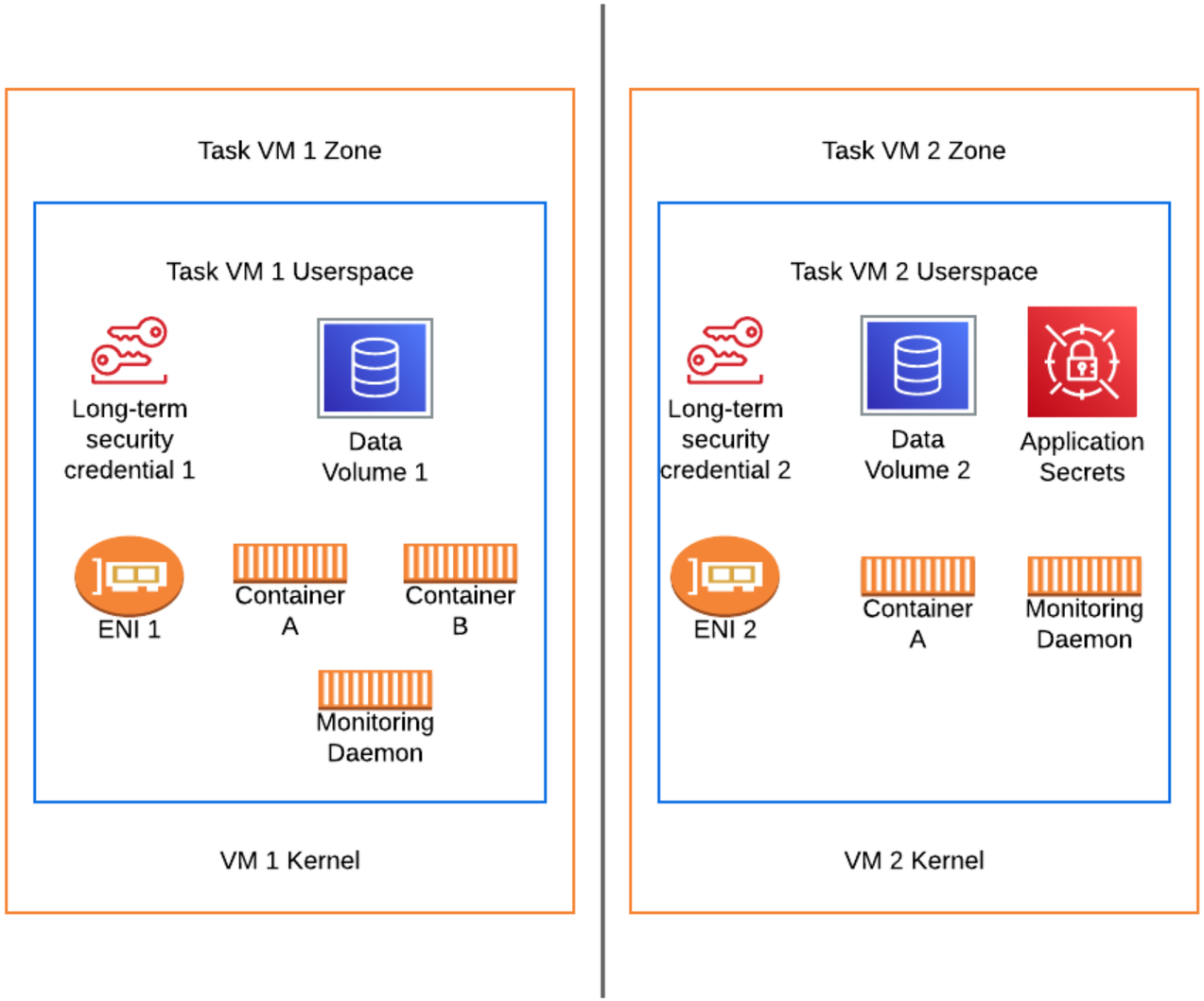
Completely Hands-Off



ECS on Fargate - Shared Responsibility Model



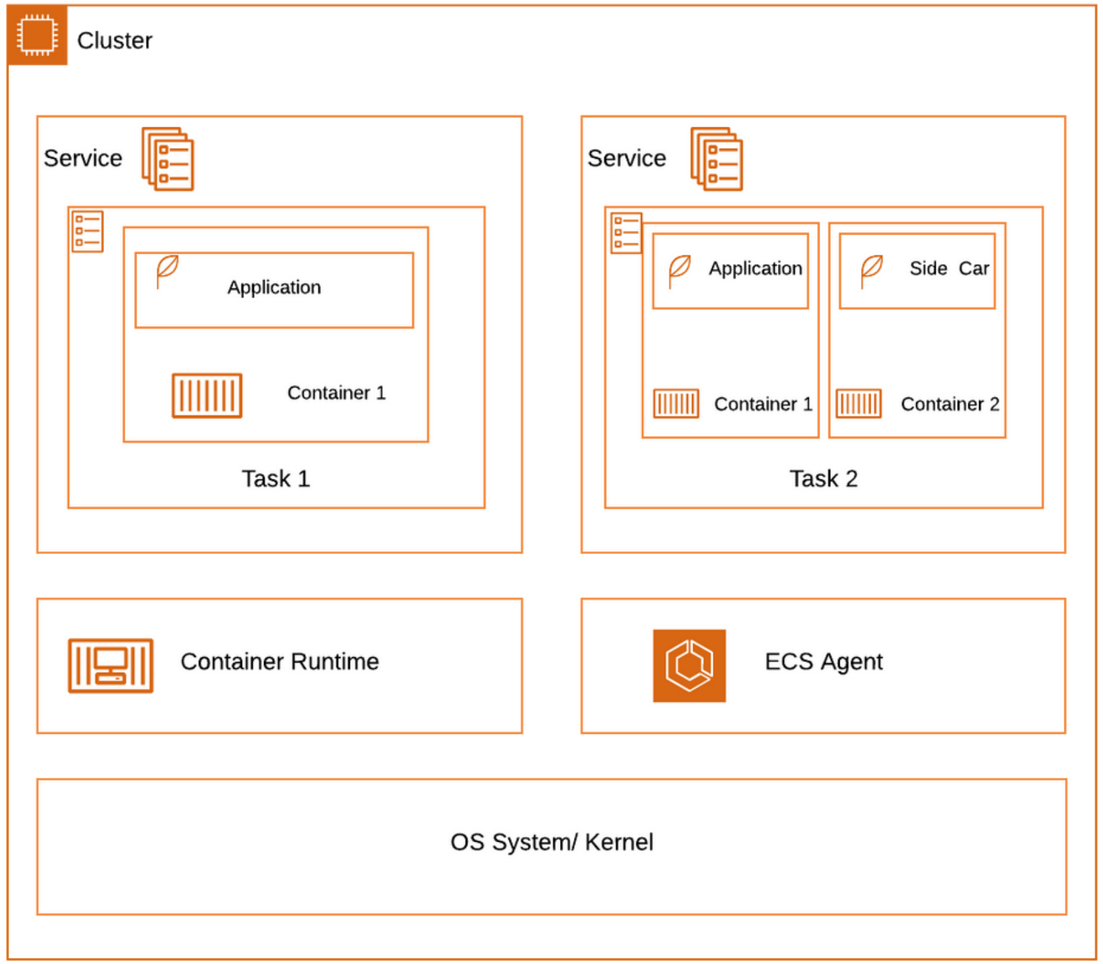
ECS on Fargate - Trust Boundaries

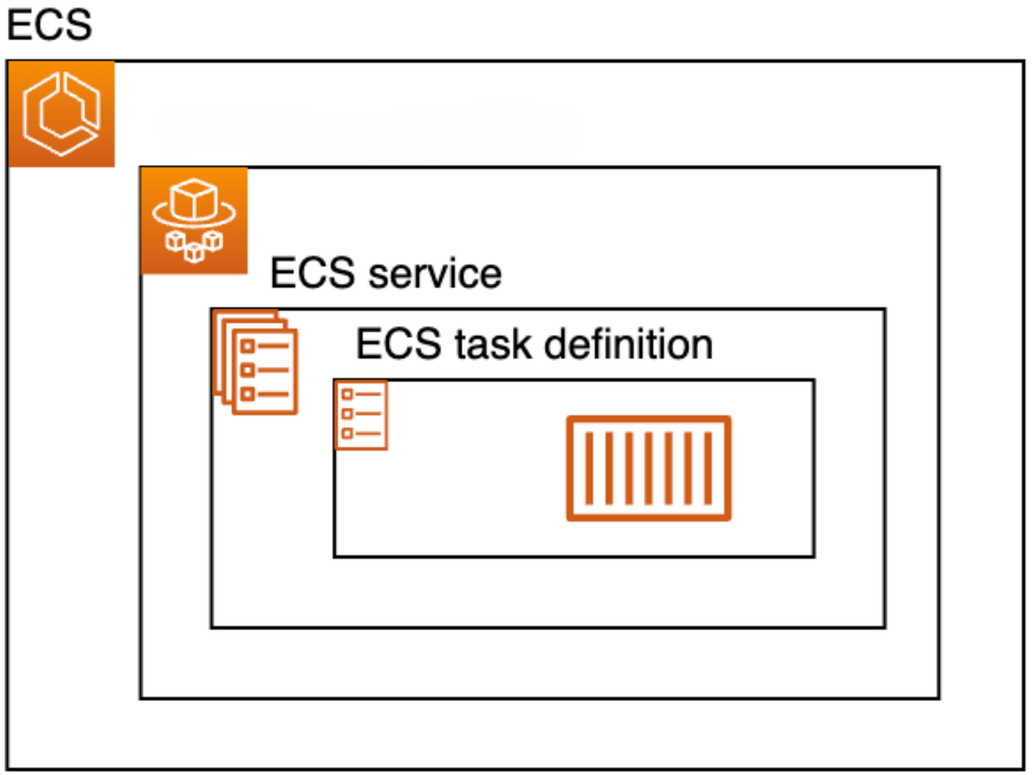


5

Amazon ECS Components Overview

Amazon ECS Architecture





Application Types: Service (ex:- web application) vs Standalone Task (ex:- batch job which runs and exits)

Amazon ECS Service Types

Applicable only when application type is a Service type.

- Daemon **(Supported only on ECS on EC2)** - The scheduler places exactly one task on each active container instance that meets all of the task placement constraints specified in your cluster.

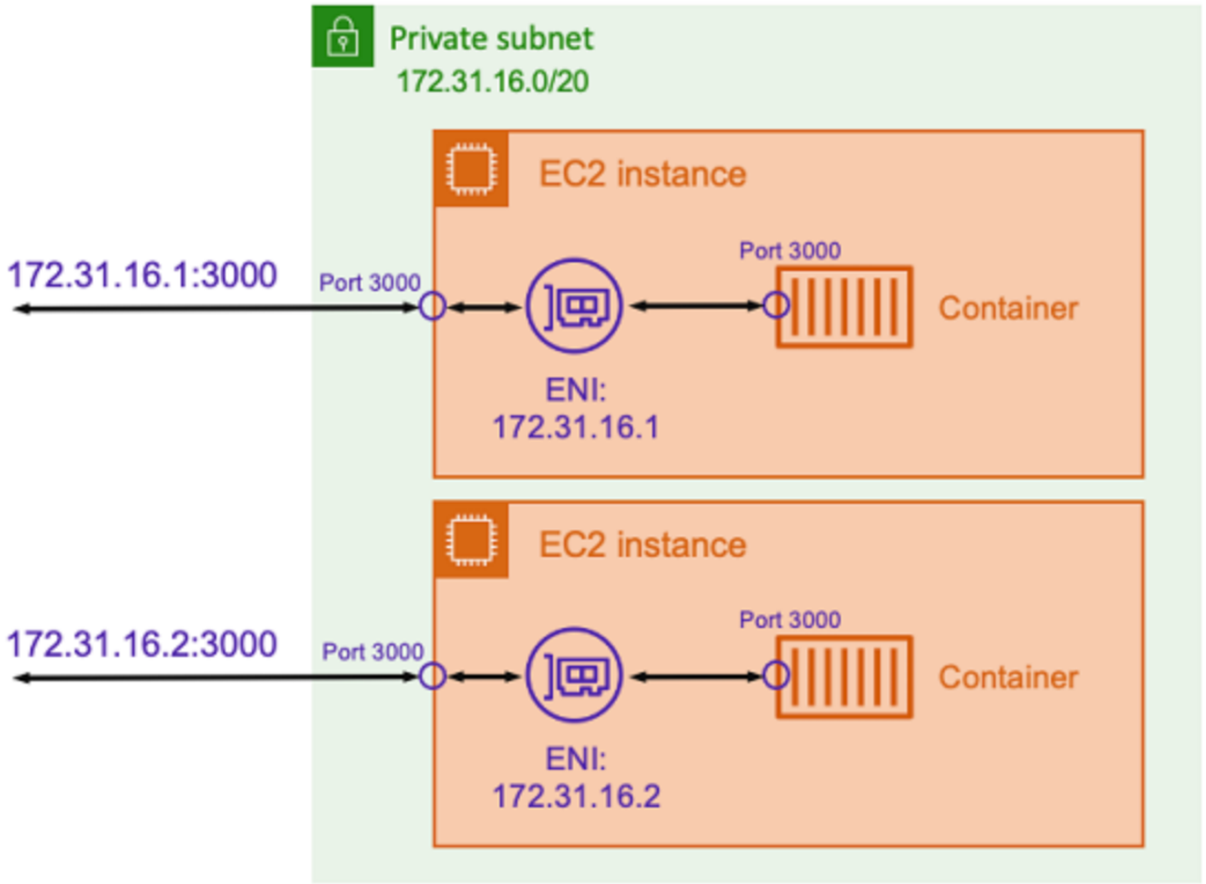
- Replica - The scheduler places and maintains the number of tasks that you specify across your cluster. By default the scheduler spreads tasks across Availability Zones. Can even add customized placements strategies for ECS services.

Amazon ECS Network Modes

- Host - The host network mode is the most basic network mode that's supported in Amazon ECS.

- Bridge - The bridge network mode allows you to use a virtual network bridge to create a layer between the host and the networking of the container. This is the default network mode for Amazon ECS EC2 type.

- AWSVPC - With the awsvpc network mode, Amazon ECS creates and manages an Elastic Network Interface (ENI) for each task and each task receives its own private IP address within the VPC. ECS on Fargate should use this mode. AWS recommend this network mode for ECS on EC2 as well.

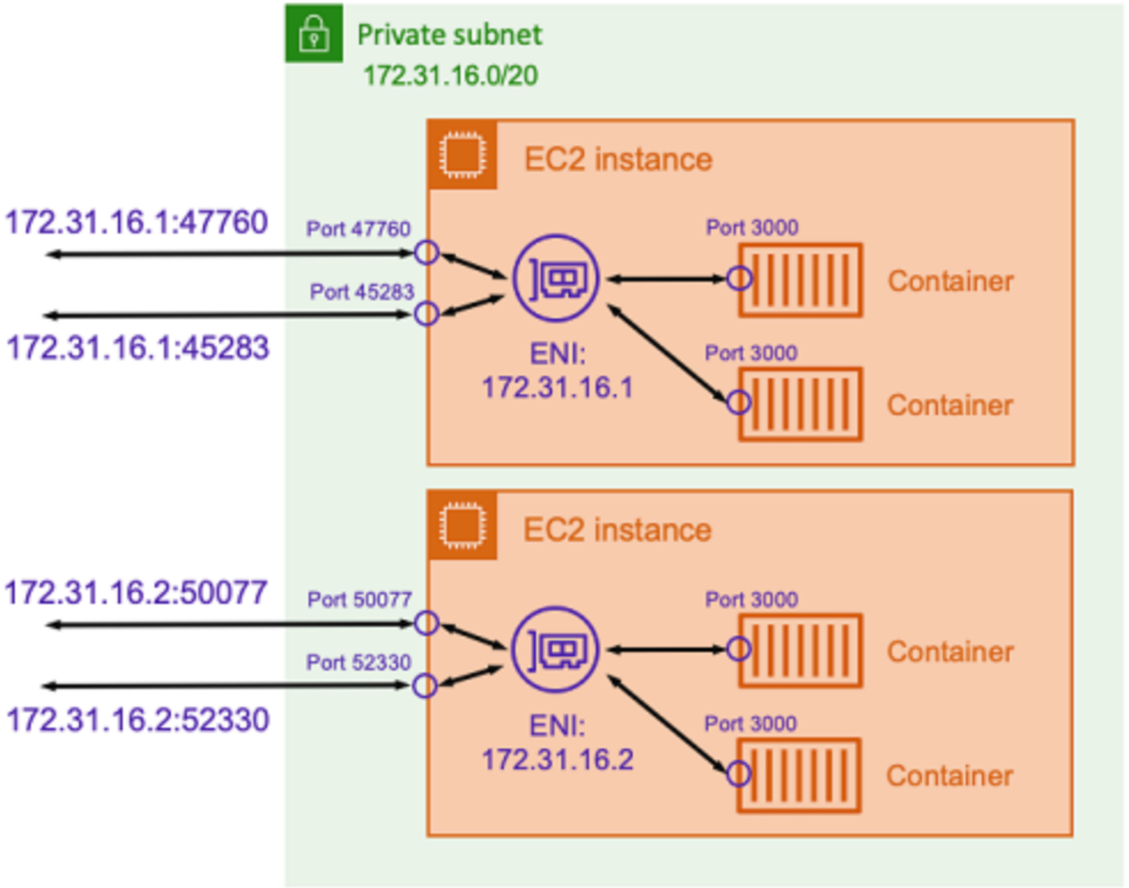


Host Network Mode

Using host mode, the networking of the container is tied directly to the underlying host that's running the container.

This mode is not recommended as only 1 container can be running on a EC2 instance with this mode.

AWS ECS on Fargate doesn't support this mode.

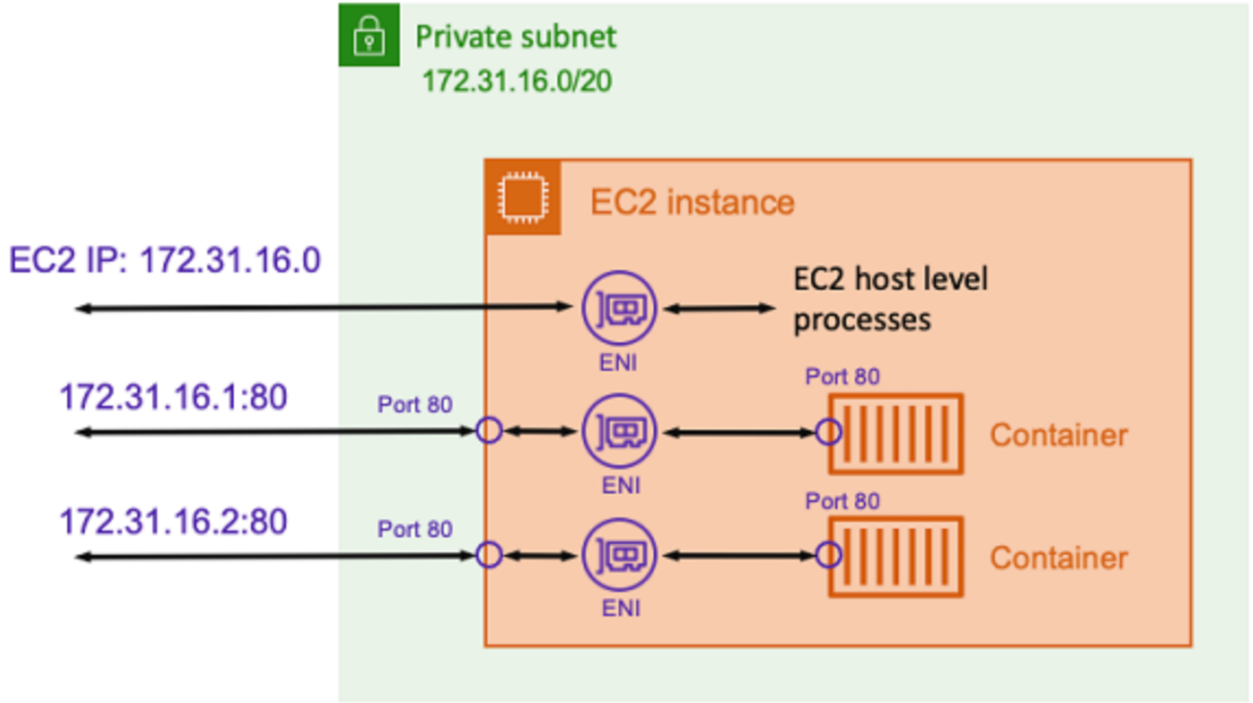


Bridge Network Mode

A virtual network bridge is used create a layer between the host and the networking of the container. This way, you can create port mappings that remap a host port to a container port. The mappings can be either static or dynamic.

AWS ECS on Fargate doesn't support this mode.

AWSVPC Network Mode



Amazon ECS creates and manages an Elastic Network Interface (ENI) for each task and each task receives its own private IP address within the VPC.

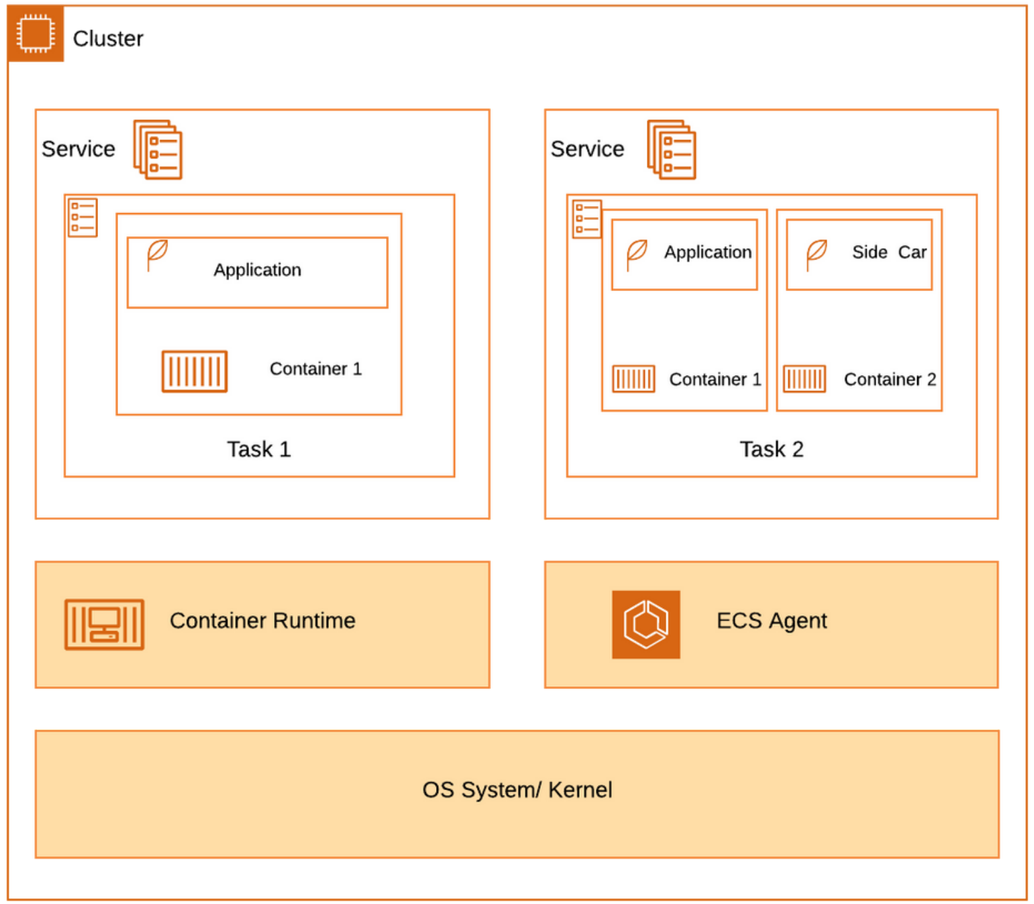
This ENI is separate from the underlying hosts ENI. If an Amazon EC2 instance is running multiple tasks, then each task's ENI is separate as well.

The awsvpc network mode is supported for Amazon ECS tasks hosted on both Amazon EC2 and Fargate. For Fargate awsvpc is required.

6

Enhancing Security Posture: Key Features of Amazon ECS

Compute/ Infrastructure Security



- Compute Segregation
- Infrastructure Patching (ECS on EC2/ ECS on Fargate) [Next Slide]
- Security Groups, NACLs and IAM roles for container instances within ECS on EC2 mode
- AWS PrivateLink

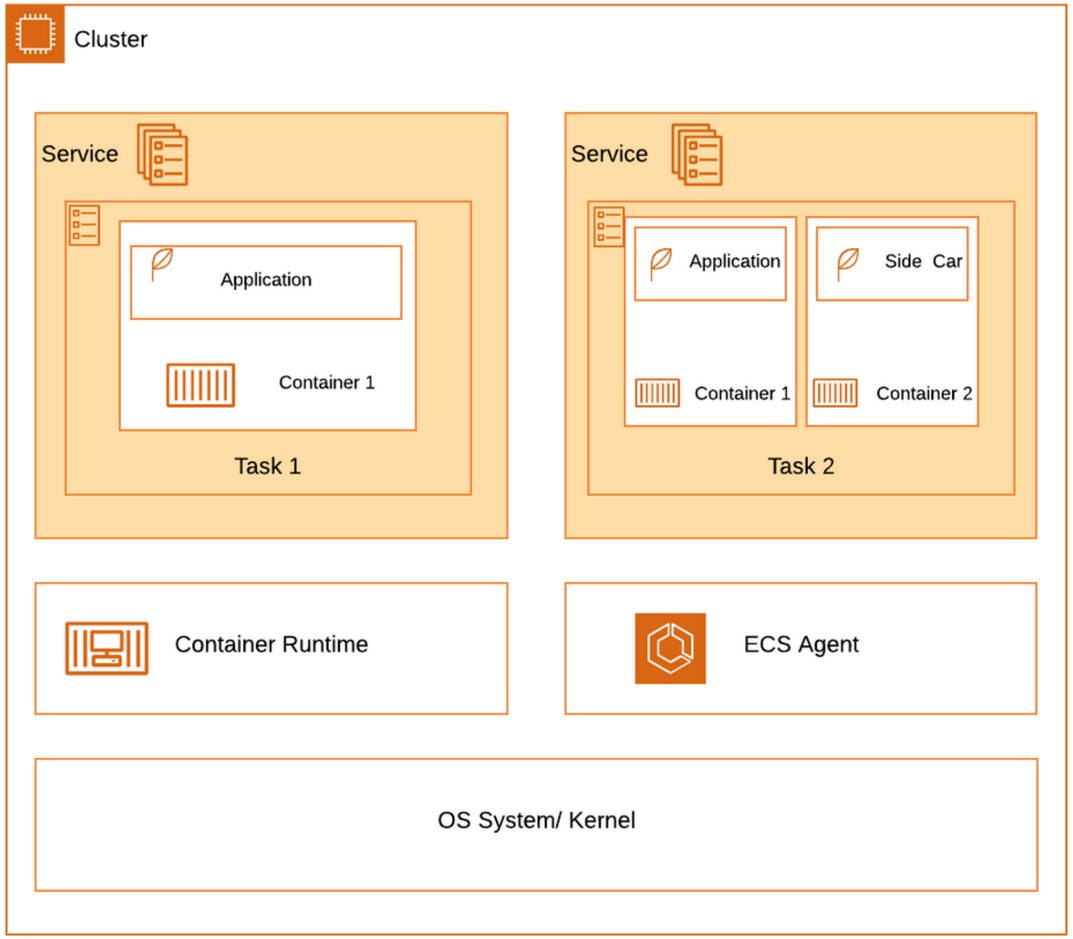
Infrastructure Patching ECS on EC2

- ECS releases AMIs with updated kernels, packages and ECS agents with security updates.
- Customer is responsible for using the latest AMIs by updating the existing ones.
- You can do it using,
    - SSM Parameters
        - aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended
    - In-place updates
        - yum update using SSM agent
    - ECS Auto Scaling/ Capacity Providers

Infrastructure Patching ECS on Fargate

- Hands off patching
    - AWS Fargate Platform Versions are used to refer to a specific runtime environment for Fargate task infrastructure. It is a combination of the kernel and container runtime versions.
    - Comes with OS, Kernel and Container Runtime updates in the form of automatic PV upgrades.
    - ECS Fargate sends out Task Termination notifications if you are running on a vulnerable PV version and notify about the retirement date. If you do not take action, ECS Service scheduler stops the ECS service running on old PV version and schedule it to be running on a new PV version. If its a standalone Task, it simply stopped.

Task Security

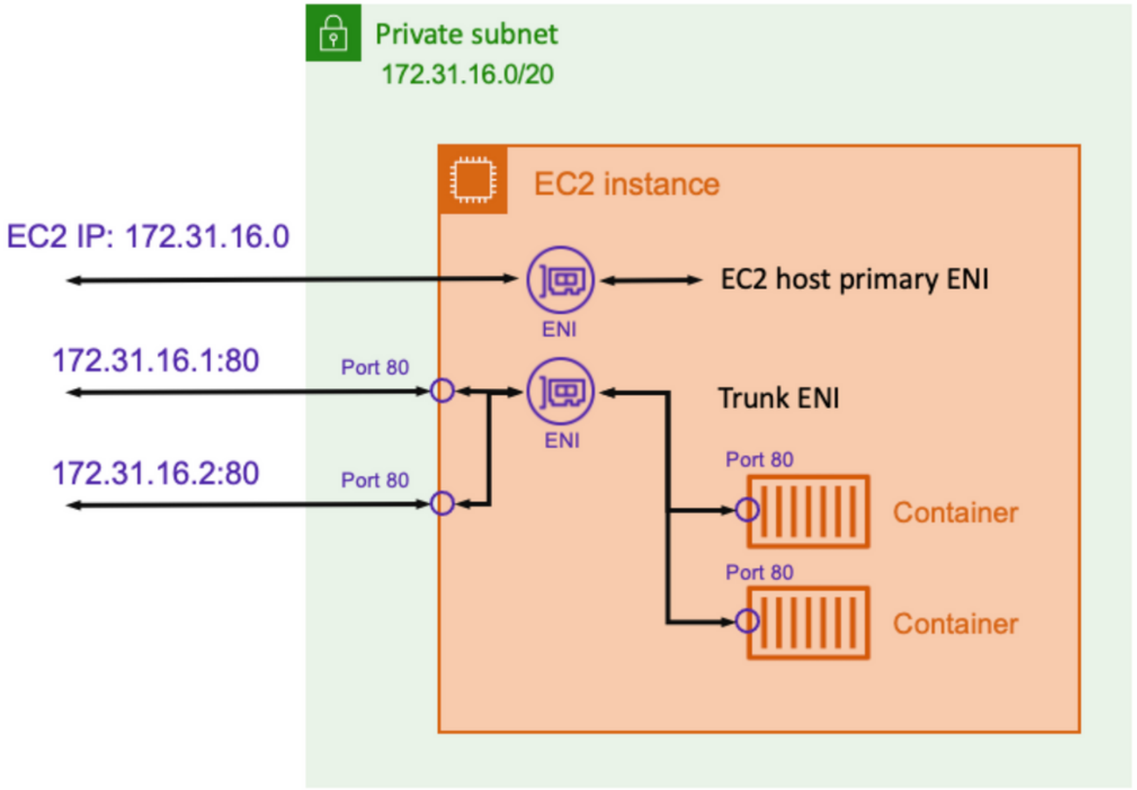


- IAM Role for Tasks
  - Credential isolation at the Task level (applications) to access AWS resources via AWS API.
- Task Execution Role
  - AWS API permissions required by the underneath ECS agents.
- Dedicated ENIs for Tasks
- Storage Isolation
  - Ephemeral
  - AWS EFS volumes

- IAM Role for Tasks
  - Gives scoped permissions for Tasks to call AWS APIs.
  - Gives Authorization
  - Give Auditability with CloudTrail logs using Task ARNs or IDs

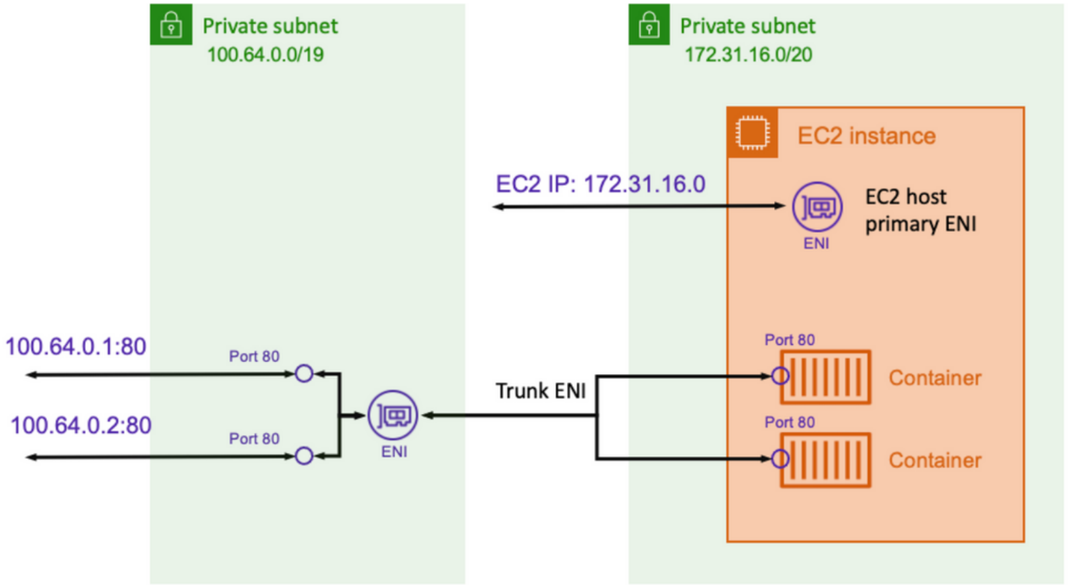
- Dedicated ENIs for Tasks
  - Networking isolation for ECS Tasks which means security groups can be attached directly to the ECS task itself.
  - Allows to pull container images, secrets and push logs to CloudWatch directly by the Task ENI itself
  - Can monitor traffic using VPC-Flow Logs for audit and monitor inbound/ outbound traffic from ENI level.
  - **[Only on EC2 option]** Can be enabled by using awsvpc network mode for the Task definition. Note the ENI trunking limitation for **ECS on EC2 [Covered in next slide]**

- Storage Isolation
  - Ephemeral - Fargate by default provides 20 GB storage (can be expanded upto 200GB) and encrypted with AES-256 algo using a key managed by AWS. You can even use KMS.
  - AWS EFS volumes - Provides simple, scalable, and persistent file storage for use with your Amazon ECS tasks. Supports both Fargate and EC2 options.
  - FSx for Windows File server, Docker volumes, Bind mounts only supports EC2 option.



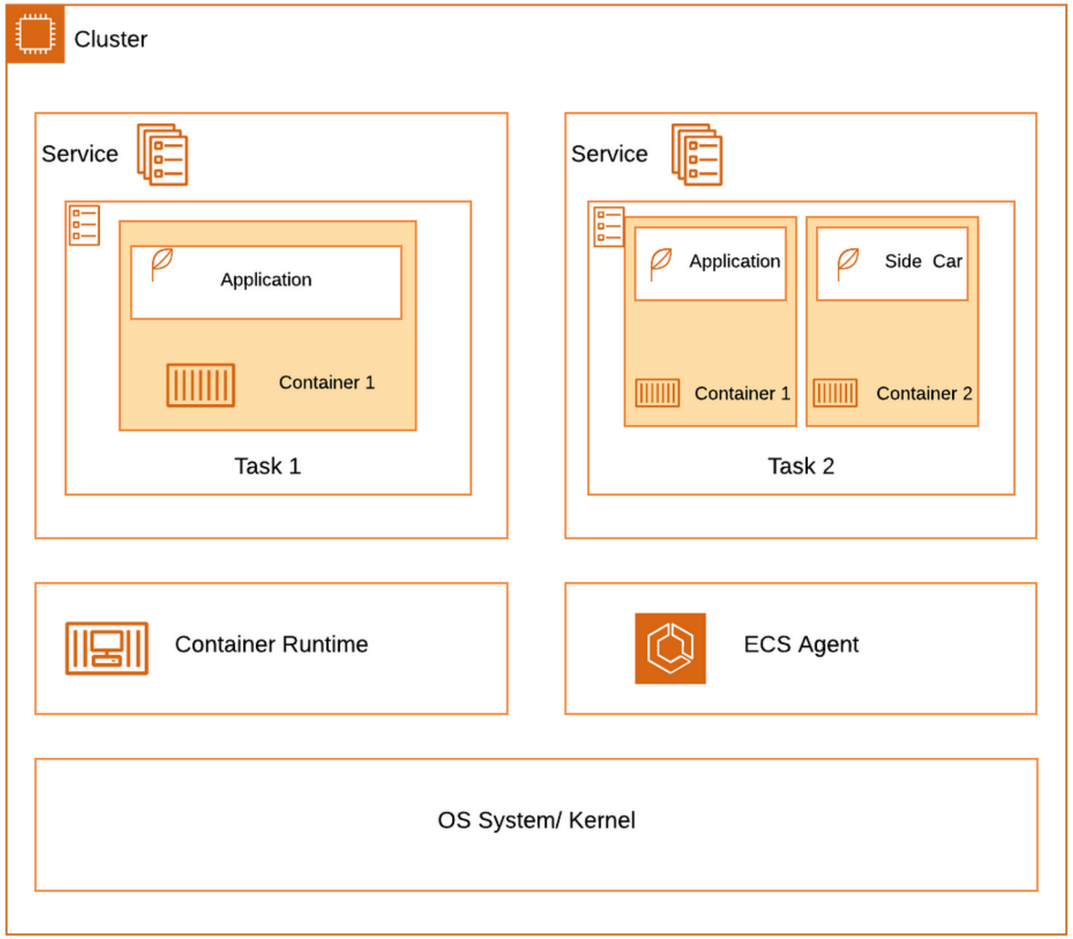
ENI Trunking (Only Applicable to ECS on EC2 on awsvpc network mode)

- EC2 instances have a limit on the number of ENIs that can be attached to them.
- This brings a limit to the number of Tasks that can be running on EC2
- Amazon ECS provides the ENI trunking feature which increases the number of available ENIs to achieve more task density.
- For example: c5.large server can only attached 2 ENIs by default vs with ENI trunking we can expand that up to 10 ENIs



- **With ENI trunking,**
    - Can avoid IP address exhaustion Amazon VPC CNI can be configured to use ENIs in a different IP address space than the host. By doing this, you can give your Amazon EC2 host and your tasks different IP address ranges that don't overlap.

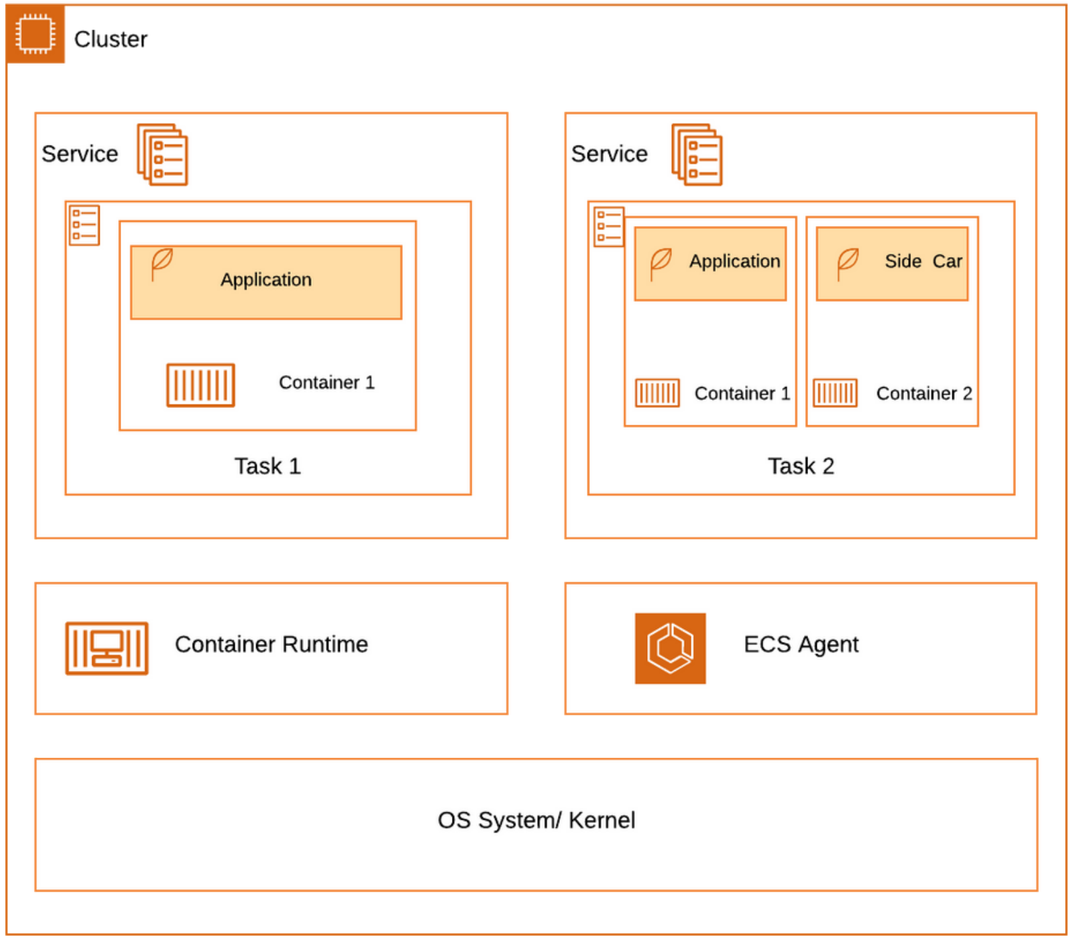
Container Security



- Container Secrets
  - Can inject sensitive information to containers rather than adding those as plain texts.
- Linux Kernel Capabilities
- Ulimits
- User Options
- Image Scanning with AWS Inspector
- Image Pruning

- Container Secrets
    - You can store secrets within AWS Secrets Manager or AWS Systems Manager - Parameter Store and inject those into container as environment variables or as part of log configuration.
    - Access to those resources can be controlled using IAM role attached to Task Execution Role defined in Task definition.
- Linux Capabilities on ECS
    - Privileged Mode - Provides container elevated privileges on the host machine similar to ROOT user. This is not a recommended approach for most use cases.
    - Add and drop capabilities - Allows you to add fine-grained permissions to your containers. Can be defined within Task container definition via linuxParameters option.
    - Best approach is to drop all the capabilities and and add the capabilities that the container require
        - ECS on Fargate - Only allows you to add “SYS_PTRACE” capability. Tracing system calls made by containers on the Kernel
        - ECS on EC2 - You can add/ drop all available Linux capabilities

Application Security



- TLS Security with AWS App Mesh
- Tag Based ECS Resource Controls
- Log Routing
- Metrics and Reporting
- Audit and Tracing
- Execute command for debugging

```
{
  Statement : [
    {
      "sid": "clusterCreation"
      "Effect": "Allow",
      "Action": [
         "ecs:createCluster"
      ],
      "Resource": "*",
      "Condition": {
         "ForAllValues:StringEquals": {
             "aws:TagKeys" : [
               "dev-ecs-security-workshop"
             ]
         }
       }
    },
    {
      "sid": "clusterDeletion"
      "Effect": "Allow",
      "Action": [
         "ecs:deleteCluster"
      ],
      "Resource": "*",
      "Condition": {
         "Null": {
           "ecs:ResourceTag/dev-ecs-security-workshop": false
         }
       }
    }
  ]
}
```

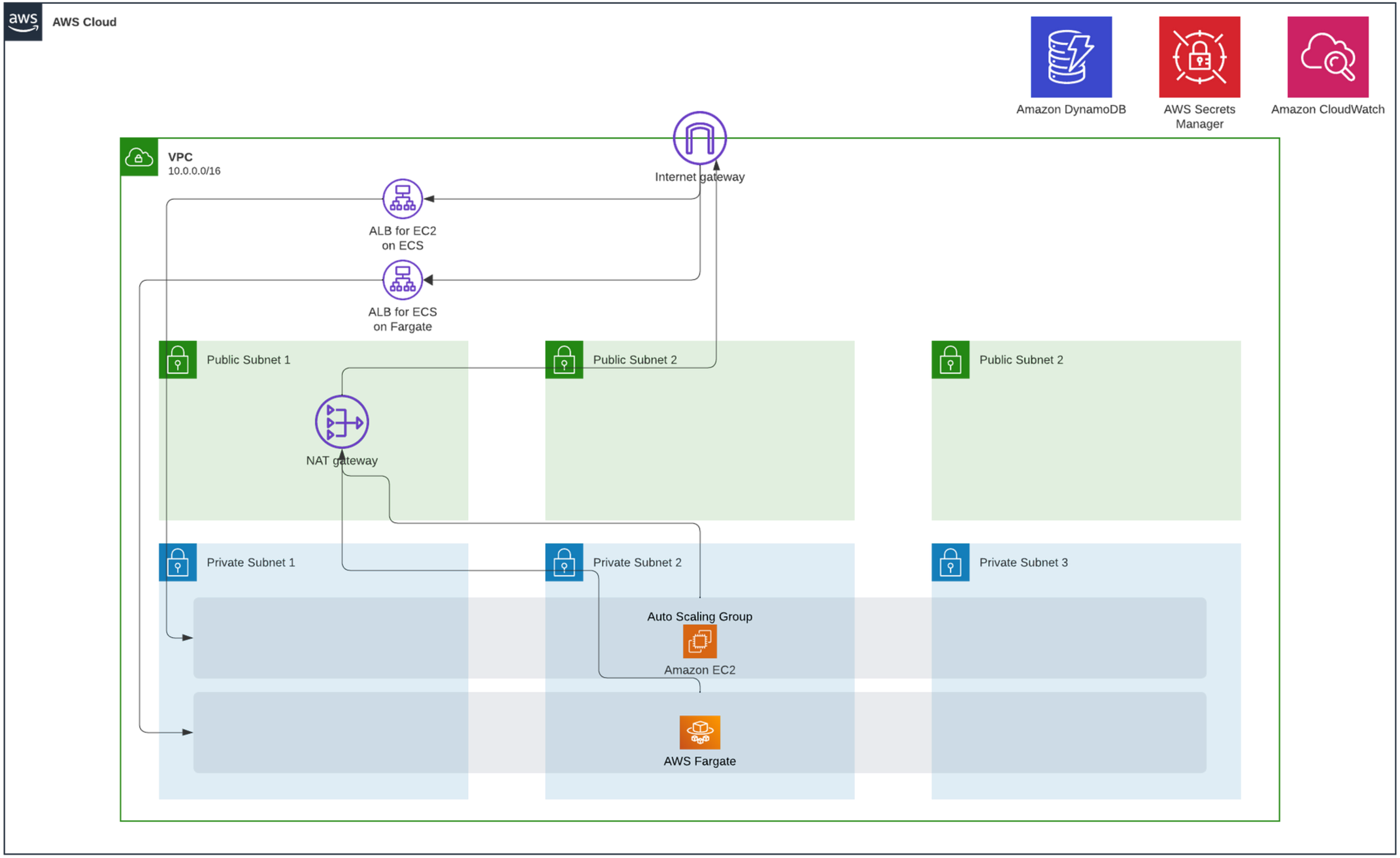
Tag Based ECS Resource Controls

Tag Based ECS Resource Controls (Not an option limited to ECS)

- User who is assuming a IAM Role which is attached with this IAM policy,
  - Can only create ECS clusters with “dev-ecs-security-workshop” tag in it
  - Can only delete ECS clusters which exists the dev-ecs-security-workshop

7

Demo


AWS Cloud
Amazon DynamoDB
AWS Secrets Manager
Amazon CloudWatch
VPC
10.0.0.0/16
Internet gateway
ALB for EC2 on ECS
ALB for ECS on Fargate
Public Subnet 1
Public Subnet 2
Public Subnet 2
NAT gateway
Private Subnet 1
Private Subnet 2
Private Subnet 3
Auto Scaling Group
Amazon EC2
AWS Fargate

Scan this QR code to view the Github Repo

https://github.com/rav94/aws-ecs-security-workshop

Resources

- Protecting production with Amazon ECS security features - https://www.youtube.com/watch?v=5-kgXY74Fpg
- Container Security in AWS Container Services - https://www.youtube.com/watch?v=ibW5YkoUSpQ&t=649s
- A deep dive into container security on AWS - https://www.youtube.com/watch?v=V3NCG_TWlEQ&t=459s
- Task Networking - https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task-networking.html
- Task Definitions - https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ecs-taskdefinition-ulimit.html
- Fargate Platform Version - https://docs.aws.amazon.com/AmazonECS/latest/developerguide/platform_versions.html
- Fargate Security - https://docs.aws.amazon.com/AmazonECS/latest/bestpracticesguide/security-fargate.html
- Using Data Volumes - https://docs.aws.amazon.com/AmazonECS/latest/developerguide/using_data_volumes.html
- Container Security Guide - https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_atomic_host/7/html/container_security_guide/index
- Best Practices for running your application with Amazon ECS - https://docs.aws.amazon.com/AmazonECS/latest/bestpracticesguide/application.html