AWS re:Invent

CON414-R

Security best practices for AWS Fargate

Len Henry
Senior Solutions Architect
Amazon Web Services

AWS re:Invent

aws

Agenda

AWS Fargate terminology

Build and deploy a container on AWS Fargate using Amazon Elastic Container Service (ECS)

Explore threat vectors to a container on AWS Fargate

“Secure by default” AWS Fargate constructs

Additional best practices for securing containers on AWS Fargate

Objectives

Building container images and storing them securely in an Amazon Elastic Container Registry (ECR) repository

Creating resources required to run a container securely in AWS Fargate

Securing access to containers running in AWS Fargate using security groups

Using AWS Secrets Manager to store secrets

Using ASM and AWS Identity and Access Management (IAM) roles to securely bootstrap containers with secrets

AWS Fargate terminology

Cluster

- Logical grouping of tasks or services
- Fargate clusters do not require EC2 instances to be managed by customers
- Can be used to create custom scoped-down IAM policies
- Can also be used as a scaling and isolation construct

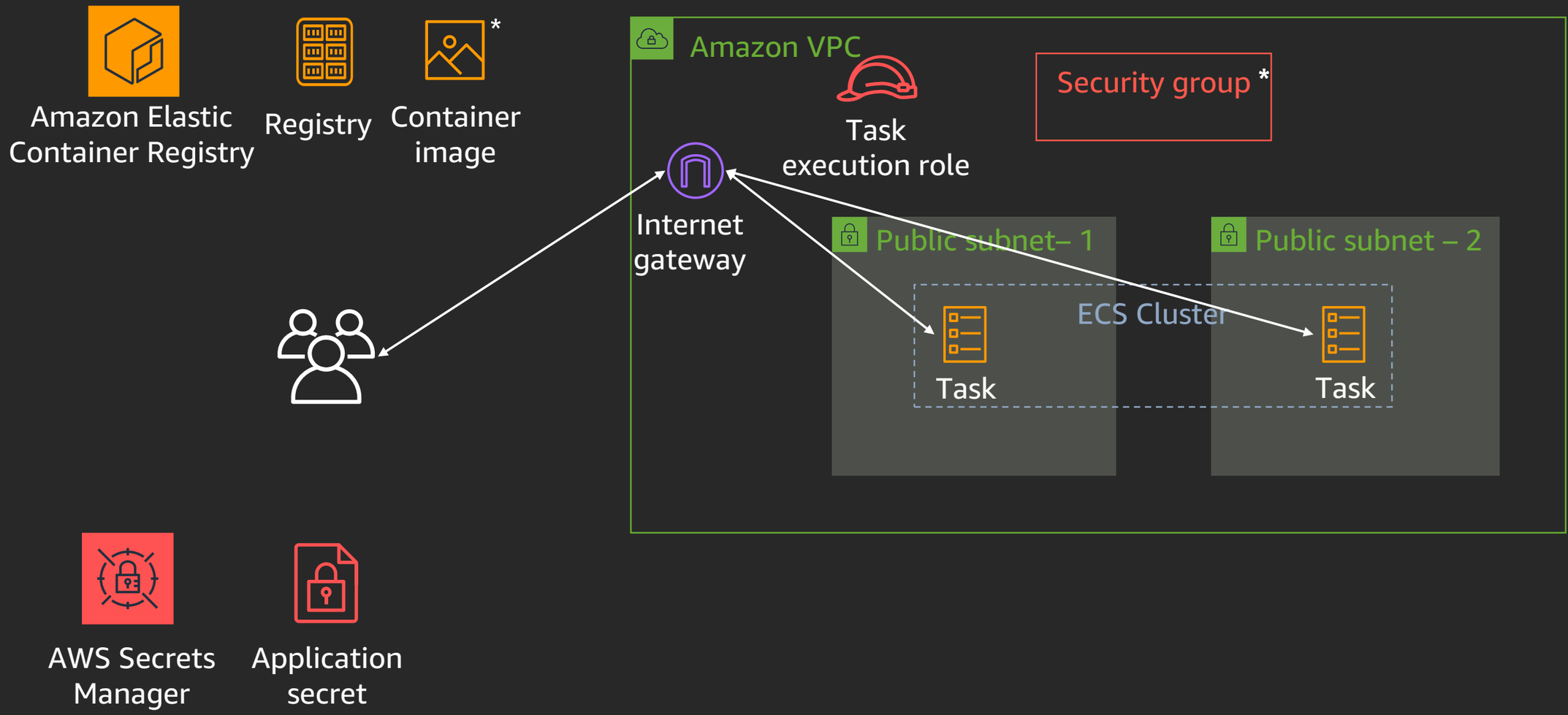
Task

- Basic unit of work in ECS
- Composed of multiple containers that are co-located on a single VM
- Allocated unique set of compute, storage & network resources in Fargate

Service

- Allows you to run and maintain a specified number of copies of a task simultaneously
- ECS Scheduler launches another copy of your task if a task should fail or stop

Sample web app architecture



Code on GitHub:
https://github.com/aaithal/reinvent2019-con414

Demo – Setting up the infrastructure

Setting up the infrastructure

What's been set up?

1. An Amazon Virtual Private Cloud (VPC) with public subnets and a security group
2. An ECS cluster
3. An AWS Identity and Access Management (IAM) role used by ECS to bootstrap the container (aka 'Task Execution Role')
4. An ECR repository
5. An "httpd-login" container image

What's been secured?

1. Your build & release artifacts (container image using ECR)
2. Your runtime environment
   - Network: Security groups and gateways in Amazon VPC
   - Application/container permissions using IAM roles

Demo - Running a Fargate task

Running a Fargate task

What's been set up?

1. An ECS task definition
2. An ECS task running on AWS Fargate

What's been secured?

AWS Fargate is secure by default

1. Task isolation: isolated compute, storage & network
2. Immutable environment (task cannot be modified once it is RUNNING)
3. Container image accessed in a secure manner
4. Application logs accessed in a secure manner

What needs to be secured yet?

1. Security group allows access to your task for any IP address on the internet
2. Application secret is built into the container image itself

Demo – Securing access to the Fargate task

Securing access to the Fargate task

What's been secured?

Network access to Fargate task has been restricted

What needs to be secured yet?

Application secret is built into the container image itself

Demo – Hardening the security posture of the Fargate task using AWS Secrets Manager

Hardening the security posture using Secrets Manager

What's been set up?

1. A Secrets Manager secret
2. A restricted security group
3. A container image that does not have secret built into the image

What's been secured?

1. Container image
2. Application secret

Demo – Running a Fargate task with Secrets Manager

Running a Fargate task with Secrets Manager

What's been set up?

1. An ECS task definition that integrates with Secrets Manager
2. An ECS task running on AWS Fargate, where secrets are bootstrapped during runtime

What's been secured?

1. Application secret exposure in Fargate task
2. Application network

Code on GitHub:
https://github.com/aaithal/reinvent2019-con414

Thank you!

Anirudh Aithal

@aaithal

AWS re:Invent

aws

Please complete the session survey in the mobile app.