# JON BONSO AND KENNETH SAMONTE





# Tutorials Dojo Study Guide and Cheat Sheets - AWS Certified DevOps Engineer Professional by Jon Bonso and Kenneth Samonte

### **Amazon Elastic Container Registry (ECR)**

 A managed AWS Docker registry service. Stores your Docker images that you can use to deploy on your EC2, ECS, or Fargate deployments.

#### **Features**

- ECR supports Docker Registry HTTP API V2 allowing you to use Docker CLI commands or your preferred Docker tools in maintaining your existing development workflow.
- You can transfer your container images to and from Amazon ECR via HTTPS.

#### Components

#### Registry

- A registry is provided to each AWS account; you can create image repositories in your registry and store images in them.
- The URL for your default registry is https://aws\_account\_id.dkr.ecr.region.amazonaws.com.

#### Authorization token

 Your Docker client needs to authenticate to ECR registries as an AWS user before it can push and pull images. The AWS CLI get-login command provides you with authentication credentials to pass to Docker.

#### Repository

- o An image repository contains your Docker images.
- **ECR lifecycle policies** enable you to specify the lifecycle management of images in a repository.

#### Repository policy

You can control access to your repositories and the images within them with repository policies.

#### Image

- You can push and pull Docker images to your repositories. You can use these images locally on your development system, or you can use them in ECS task definitions.
- You can replicate images in your private repositories across AWS regions.



# Tutorials Dojo Study Guide and Cheat Sheets - AWS Certified DevOps Engineer Professional by Jon Bonso and Kenneth Samonte

### **Amazon Elastic Container Service (ECS)**

- A container management service to run, stop, and manage Docker containers on a cluster.
- ECS can be used to create a consistent deployment and build experience, manage, and scale batch and Extract-Transform-Load (ETL) workloads, and build sophisticated application architectures on a microservices model.

#### **Features**

- You can create ECS clusters within a new or existing VPC.
- After a cluster is up and running, you can define task definitions and services that specify which Docker container images to run across your clusters.

#### **Components**

- Containers and Images
  - Your application components must be architected to run in containers containing everything that your software application needs to run: code, runtime, system tools, system libraries, etc.
  - o Containers are created from a read-only template called an **image**.
- Task Components
  - **Task definitions** specify various parameters for your application. It is a text file, in JSON format, that describes one or more containers, up to a maximum of ten, that form your application.
  - o Task definitions are split into separate parts:
    - Task family the name of the task, and each family can have multiple revisions.
    - IAM task role specifies the permissions that containers in the task should have.
    - Network mode determines how the networking is configured for your containers.
    - Container definitions specify which image to use, how much CPU and memory the container are allocated, and many more options.
- Tasks and Scheduling
  - A task is the instantiation of a task definition within a cluster. After you have created a task
    definition for your application, you can specify the number of tasks that will run on your cluster.
    - Each task that uses the Fargate launch type has its own isolation boundary and does not share the underlying kernel, CPU resources, memory resources, or elastic network interface with another task.
  - You can upload a new version of your application task definition, and the ECS scheduler automatically starts new containers using the updated image and stop containers running the previous version.
- Clusters
  - When you run tasks using ECS, you place them in a **cluster**, which is a logical grouping of resources.
  - Clusters can contain tasks using both the Fargate and EC2 launch types.



# Tutorials Dojo Study Guide and Cheat Sheets - AWS Certified DevOps Engineer Professional by Jon Bonso and Kenneth Samonte

- When using the Fargate launch type with tasks within your cluster, ECS manages your cluster resources.
- Enabling managed Amazon ECS cluster auto scaling allows ECS to manage the scale-in and scale-out actions of the Auto Scaling group.

#### Services

- ECS allows you to run and maintain a specified number of instances of a task definition simultaneously in a cluster.
- In addition to maintaining the desired count of tasks in your service, you can optionally run your service behind a load balancer.
- There are two deployment strategies in ECS:

## Rolling Update

■ This involves the service scheduler replacing the current running version of the container with the latest version.

#### **Blue/Green Deployment with AWS CodeDeploy**

- This deployment type allows you to verify a new deployment of a service before sending production traffic to it.
- The service must be configured to use either an Application Load Balancer or Network Load Balancer.
- Container Agent (AWS ECS Agent)
  - o The **container agent** runs on each infrastructure resource within an ECS cluster.
  - It sends information about the resource's current running tasks and resource utilization to ECS, and starts and stops tasks whenever it receives a request from ECS.
  - Container agent is only supported on Amazon EC2 instances.

### **AWS Fargate**

- You can use Fargate with ECS to run containers without having to manage servers or clusters of EC2 instances.
- You no longer have to provision, configure, or scale clusters of virtual machines to run containers.
- Fargate only supports container images hosted on Elastic Container Registry (ECR) or Docker Hub.

## **Task Definitions for Fargate Launch Type**

- Fargate task definitions require that the network mode is set to *awsvpc*. The *awsvpc* network mode provides each task with its own elastic network interface.
- Fargate task definitions only support the *awslogs* log driver for the log configuration. This configures your Fargate tasks to send log information to Amazon CloudWatch Logs.
- Task storage is **ephemeral**. After a Fargate task stops, the storage is deleted.

#### **Monitoring**