

# TECHNOLOGY



## Designing Applications and Architectures in AWS

(Aligned with AWS Solution Architect Associate Certification)

## Container Service





# A Day in the Life of a Cloud Architect

You are a cloud architect in an organization and have been asked to design a strategy for deploying the company's e-commerce web application in the cloud.

Your primary objectives include enabling the deployment of microservices in containers, ensuring continuous monitoring, facilitating automatic scaling, and incorporating other vital features to boost performance and reliability.

To achieve these, you will learn a few concepts in this lesson that will help you find a solution for the given scenario.



# Learning Objectives

By the end of this lesson, you will be able to:

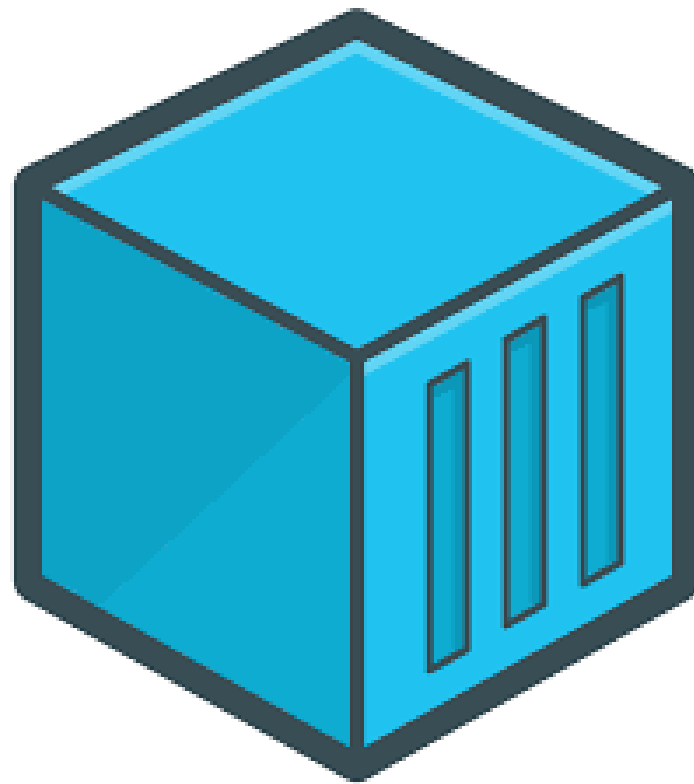
- Execute a Docker container on an EC2 instance to improve resource utilization
- Push and pull container images from Amazon ECR to implement container-based applications
- Create and manage container clusters using Amazon ECS for various AWS service integration
- Deploy a container cluster using Amazon Fargate for serverless container management



## Elastic Container Service

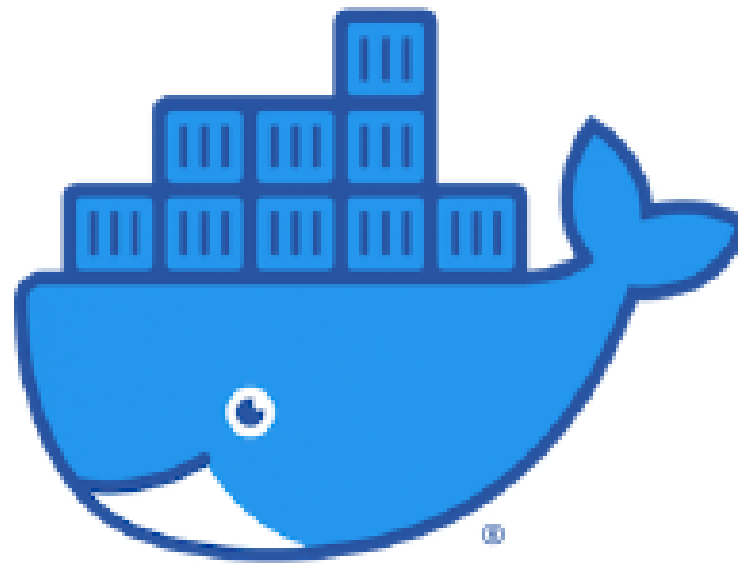
# What Is a Container?

A container is a standardized software development unit encompassing all elements necessary for a software application. These elements include code, runtime, system tools, system libraries, and more.



# What Is Docker?

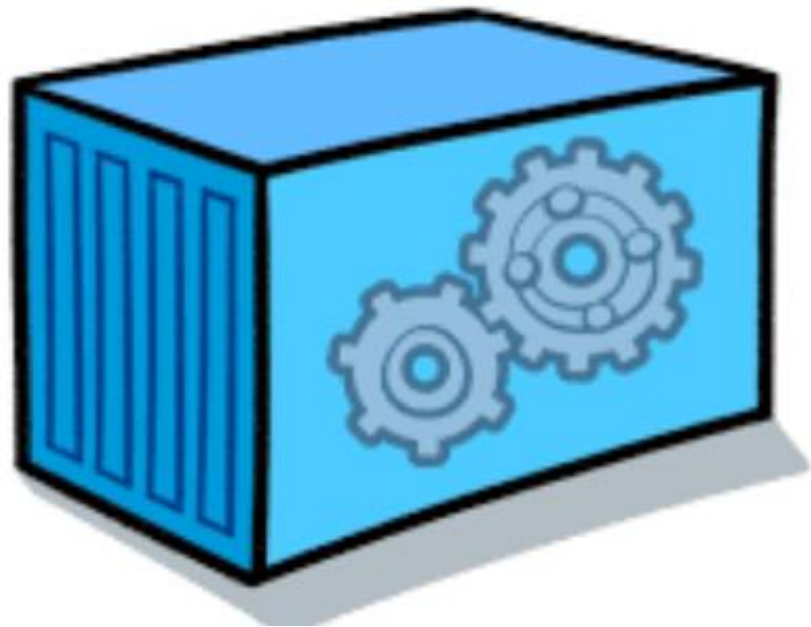
Docker is a tool designed to automate the deployment of applications as lightweight containers, enabling efficient operation across diverse environments.



A Docker container is a streamlined software package that encompasses all necessary dependencies, such as code, frameworks, and libraries, to execute an application effectively.



# Advantages of a Docker Container



01

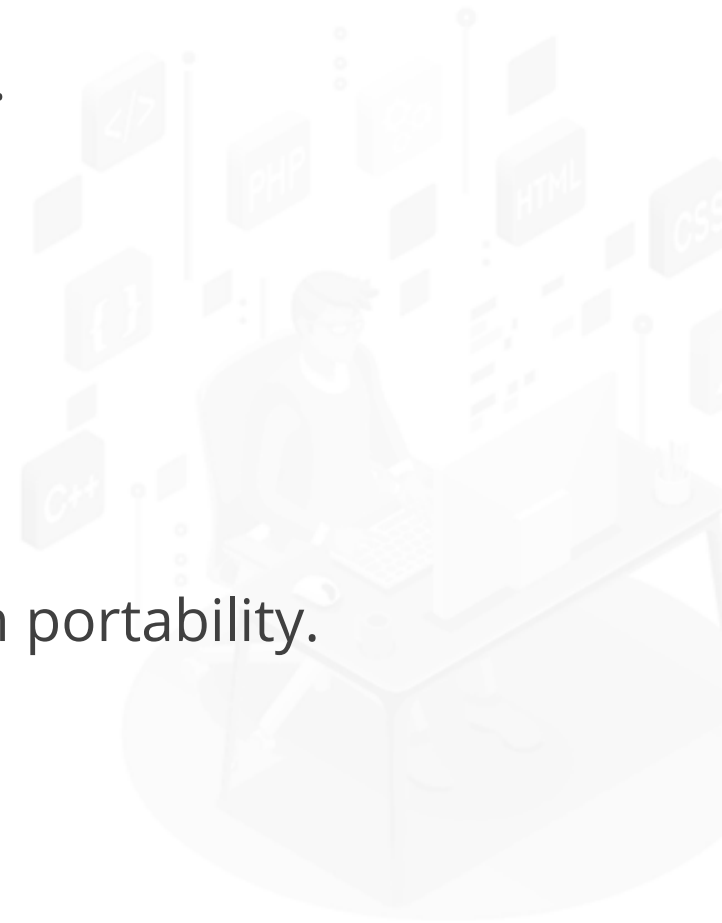
They are highly scalable and efficient.

02

They boot up quickly.

03

Docker containers demonstrate high portability.





# Running a Docker Container on an AWS EC2 Instance



**Duration: 15 min**

## **Problem Statement:**

You have been asked to run the hello world container on an EC2 instance.

ASSISTED PRACTICE

# Assisted Practice: Guidelines

---

## Steps to be followed are:

1. Set up an EC2 instance
2. Install the Docker on Ubuntu
3. Run the Hello World Container



# Creating a Container Registry Using AWS ECR



**Duration: 15 min**

## **Problem Statement:**

You have been asked to create a container image repository using AWS ECR and push images into it.

ASSISTED PRACTICE

# Assisted Practice: Guidelines

---

## Steps to be followed are:

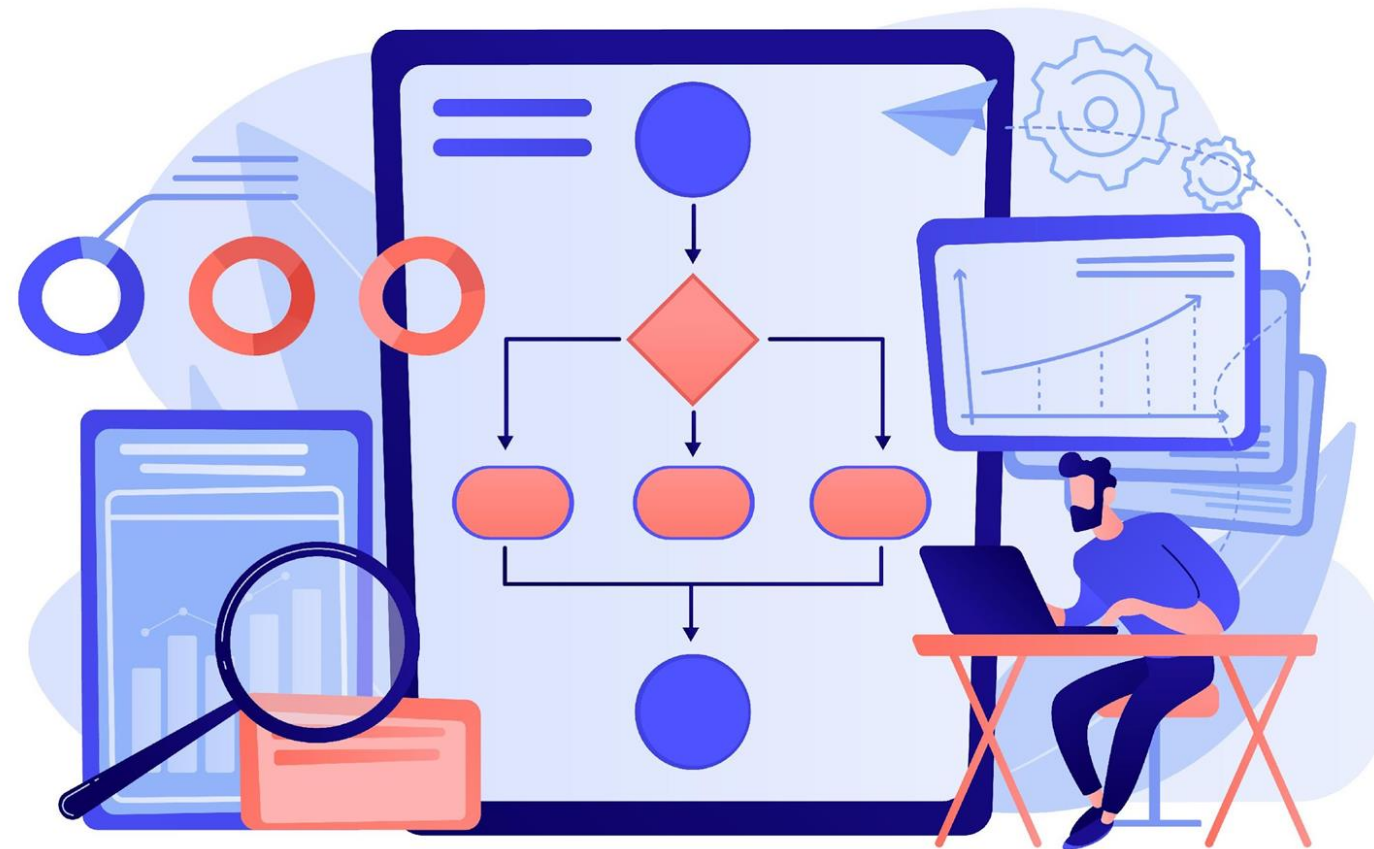
1. Create an ECR repository
2. Launch an EC2 instance
3. Install Docker on the EC2 instance
4. Create and push the Docker image to the repository





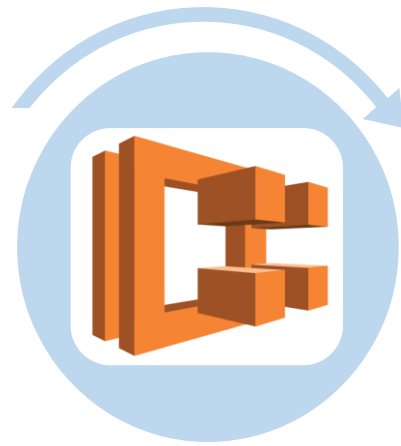
# What Is Container Orchestration?

Container orchestration automates scheduling, development, networking, scaling, health monitoring, and management of containers.



# Container Orchestration Tools in AWS

The AWS services used for container orchestration are as follows:



Amazon ECS



Amazon EKS



# Amazon Elastic Container Registry (ECR)

Amazon Elastic Container Registry (ECR) is a fully managed Docker container registry that simplifies developers' tasks of storing, managing, and deploying Docker container images.



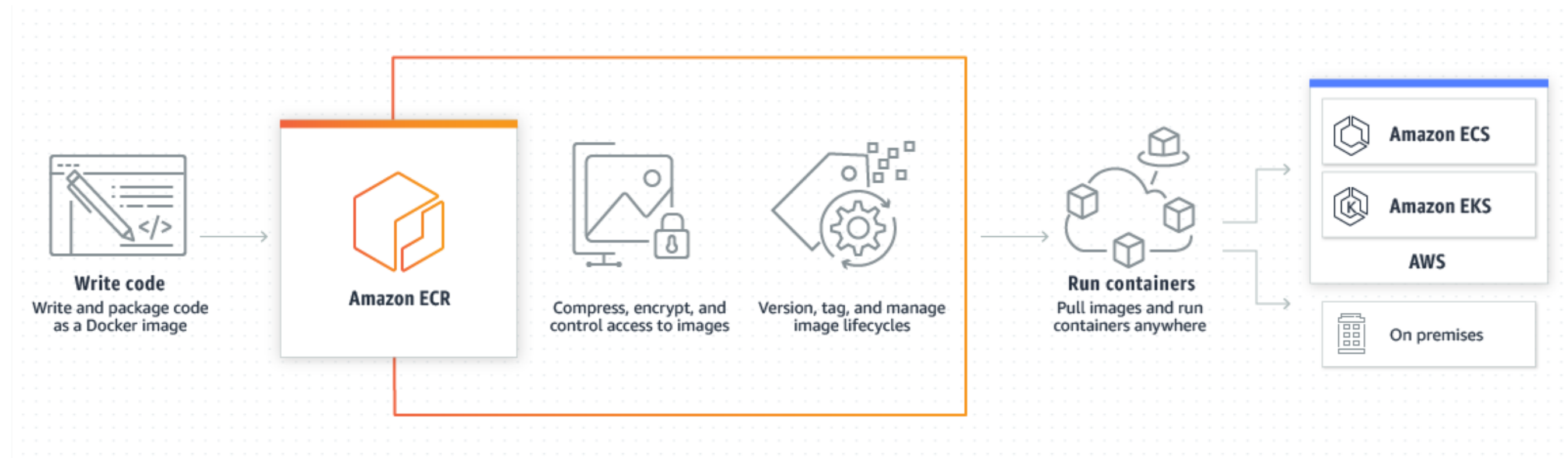
## AWS ECR

It is integrated with Amazon Elastic Container Service (ECS) to streamline the container deployment process.



# Amazon ECR Workflow

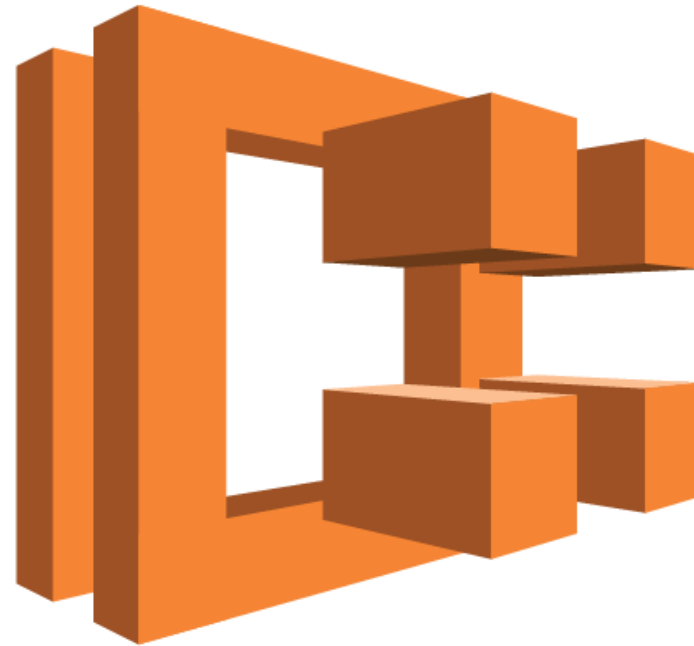
This image illustrates the role of Amazon ECR in the container deployment process:





# Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) is a highly scalable and rapid container management service that deploys and manages containers on a cluster.



ECS is integrated with the broader AWS platform, offering a secure and user-friendly solution for executing container workloads in the cloud.

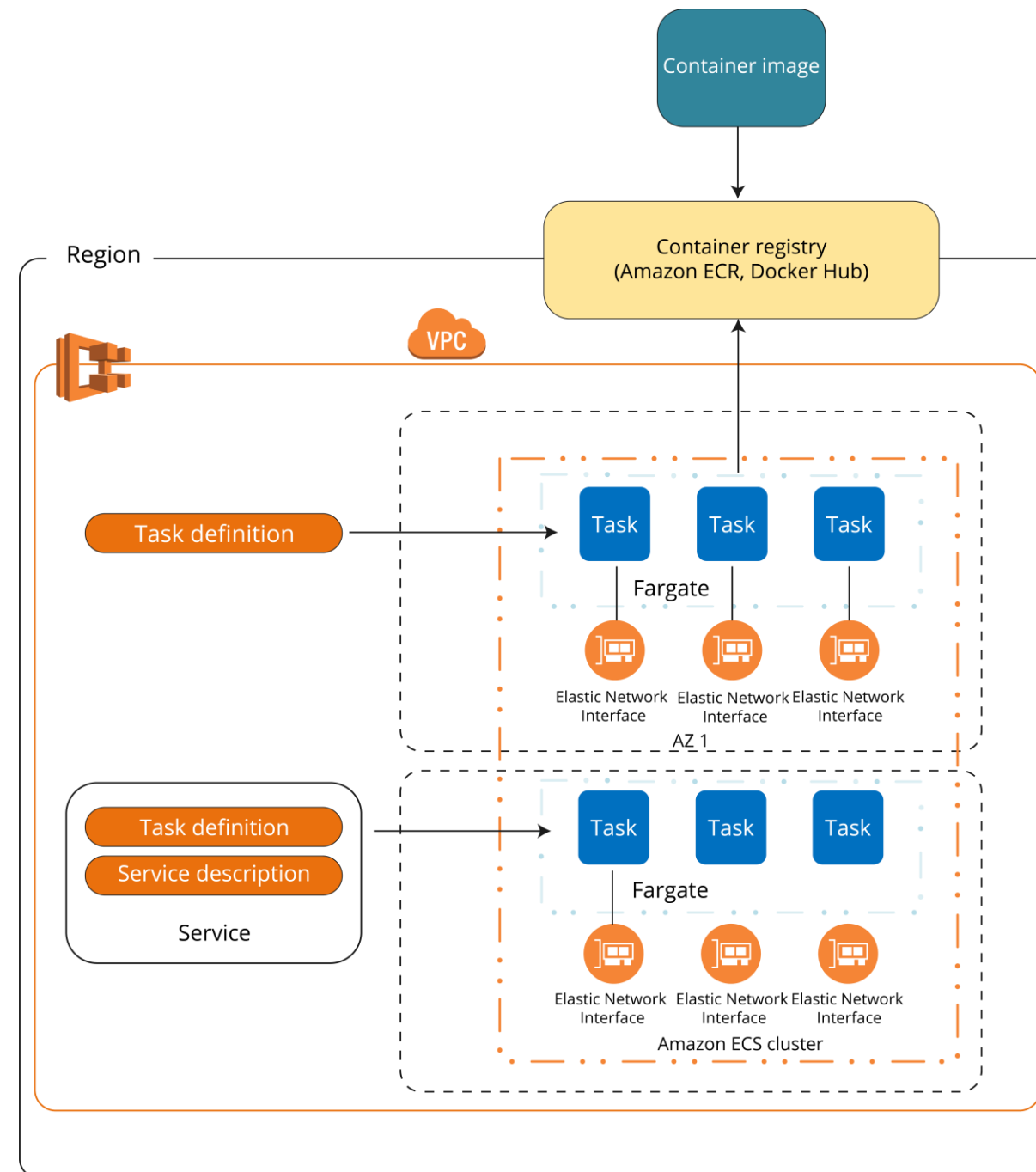
# ECS: Features



- Amazon ECS fully supports Docker, enabling users to run and manage Docker containers effectively.
- It provides flexibility for users to integrate and extend their service by making API calls.
- It automatically restores failed containers, ensuring consistent application availability and reliability.
- It facilitates cluster scheduling based on diverse requirements, such as configuring clusters according to resource needs like CPU or RAM.

# ECS Architecture

The image below illustrates how Amazon ECS employs AWS Fargate to execute containers:



# ECS: Components



01

## **Cluster:**

It is a logical grouping of tasks or services within ECS.

02

## **Container:**

It is a standardized software unit containing all the components necessary for an application to function.



# ECS: Components



03

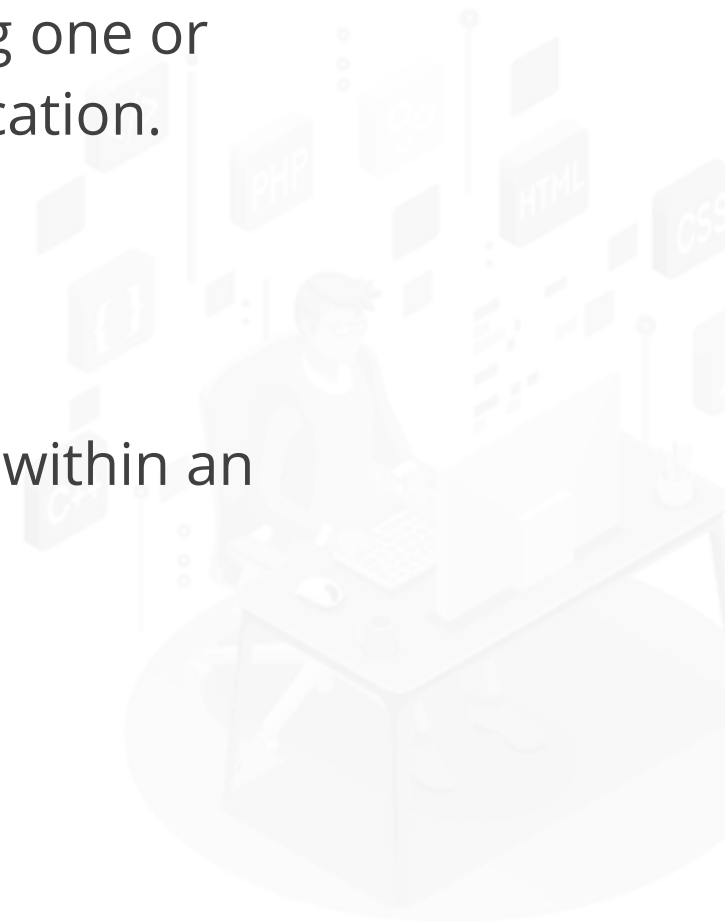
## Task Definition:

It is a JSON-formatted text file describing one or more containers that compose an application.

04

## Task:

It is the instantiation of a task definition within an ECS cluster.



# ECS: Components



05

## **Service:**

The purpose of a service is to concurrently execute and uphold the desired count of tasks within an Amazon ECS cluster.

06

## **Container Agent:**

It operates on each container instance within an Amazon ECS cluster and transmits details about ongoing tasks and the container's resource utilization to Amazon ECS.

# Elastic Container Service (ECS)

Amazon ECS can be launched in the following modes:



Fargate launch

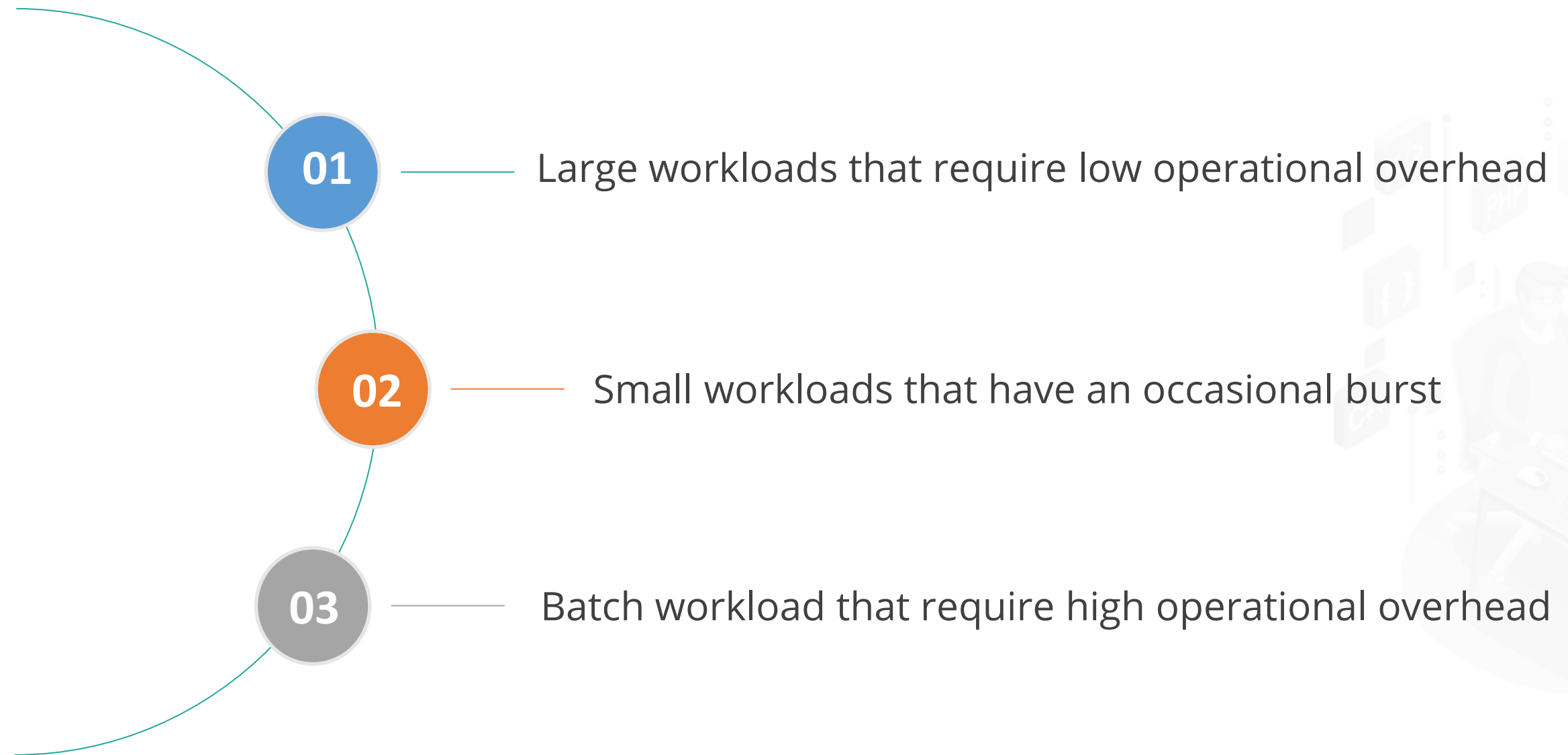


EC2 launch



# When to Use Fargate Launch Type?

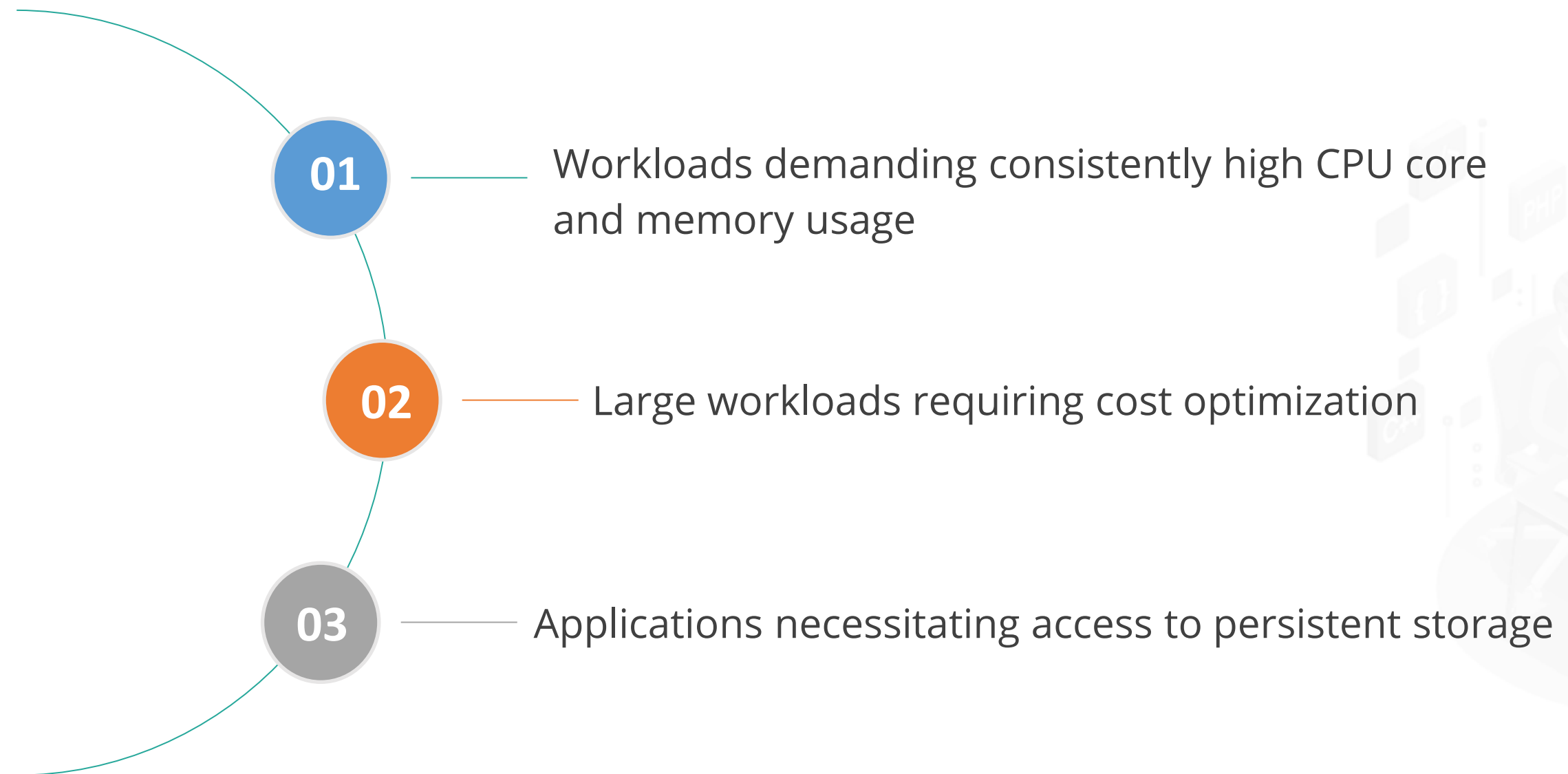
The Fargate launch type is well-suited for the following workloads:





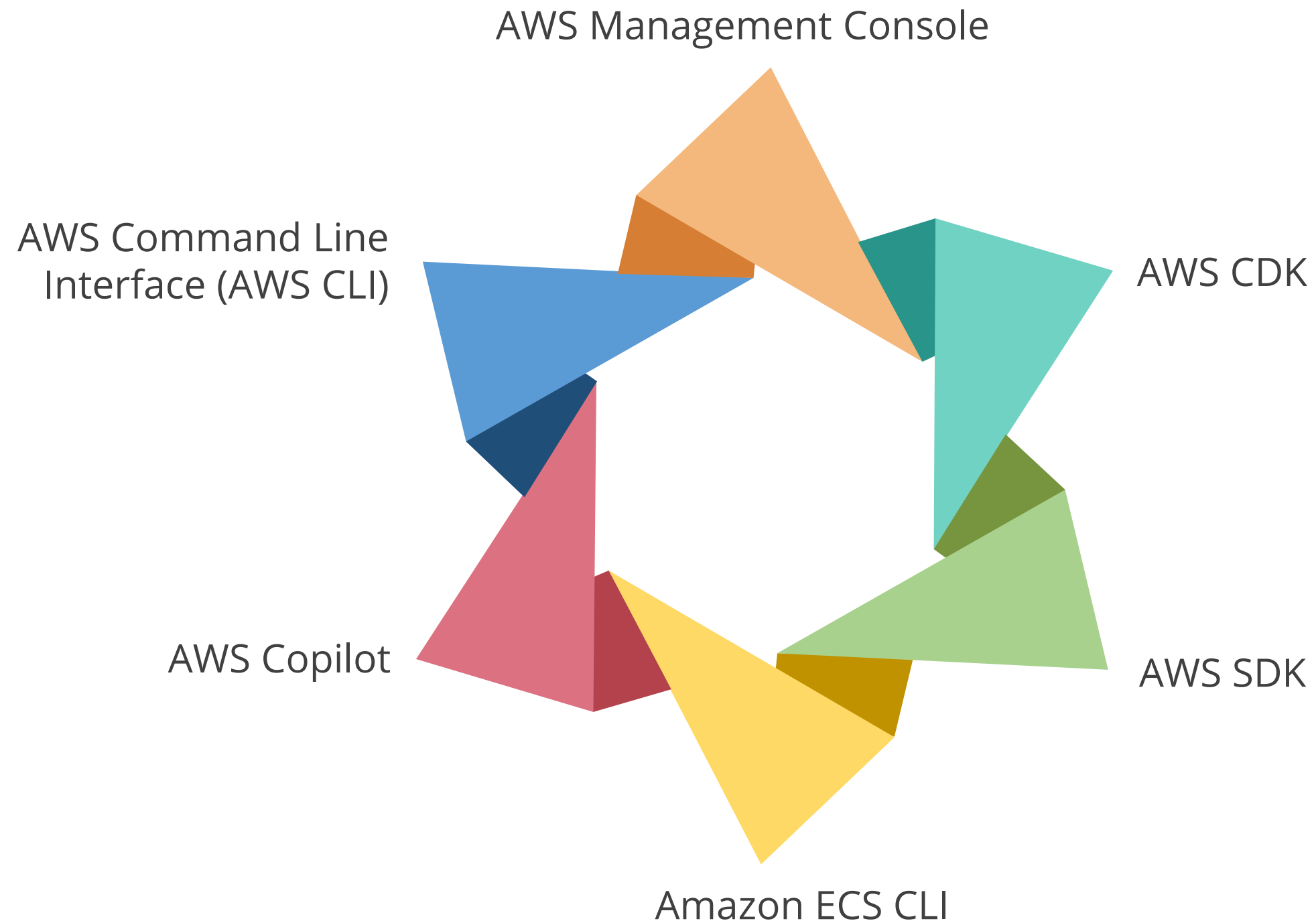
# When to Use EC2 Launch Type?

The EC2 launch type is ideal for the following workloads:



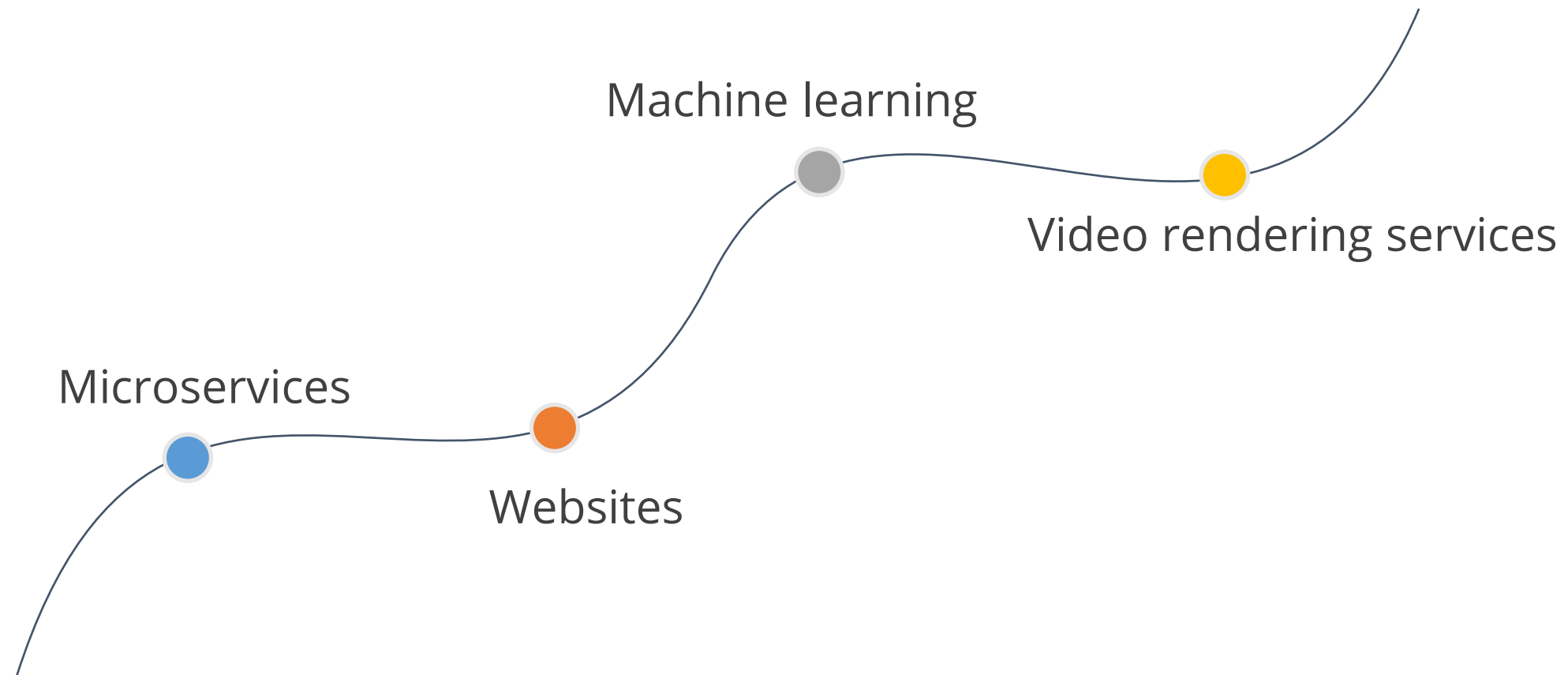
# How to Access ECS?

ECS can be accessed from the following interfaces:



# ECS: Use Cases

Amazon ECS is suited for a variety of use cases, including:



# Companies Using Amazon ECS

The following companies use Amazon ECS:



# Creating an EC2 Cluster with Auto Scaling



**Duration: 15 min**

## **Problem Statement:**

You have been asked to create an ECS cluster with auto scaling and Container Insights, and how to run a task definition for an Nginx container on the cluster.

ASSISTED PRACTICE

# Assisted Practice: Guidelines

---

## Steps to be followed are:

1. Create an ECS cluster
2. Create a task definition
3. Run the task definition on the cluster

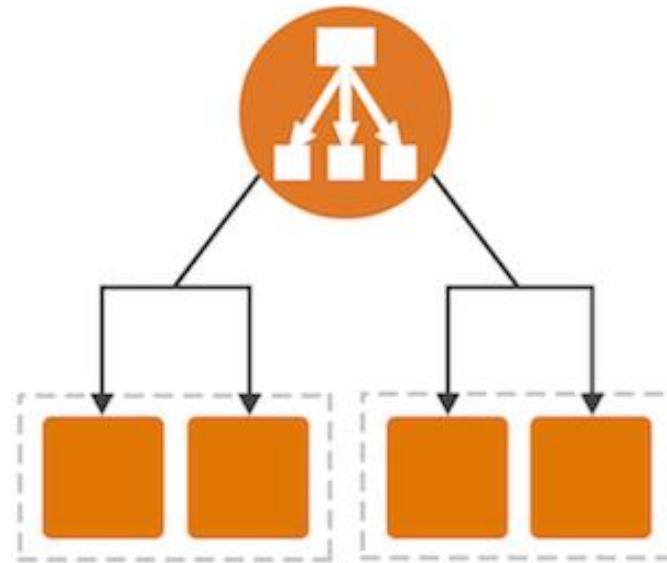




## Dynamic Port Mapping

# What Is Dynamic Port Mapping?

It is a feature of an Application Load Balancer that enables users to execute multiple tasks from a single service on the same container instance.

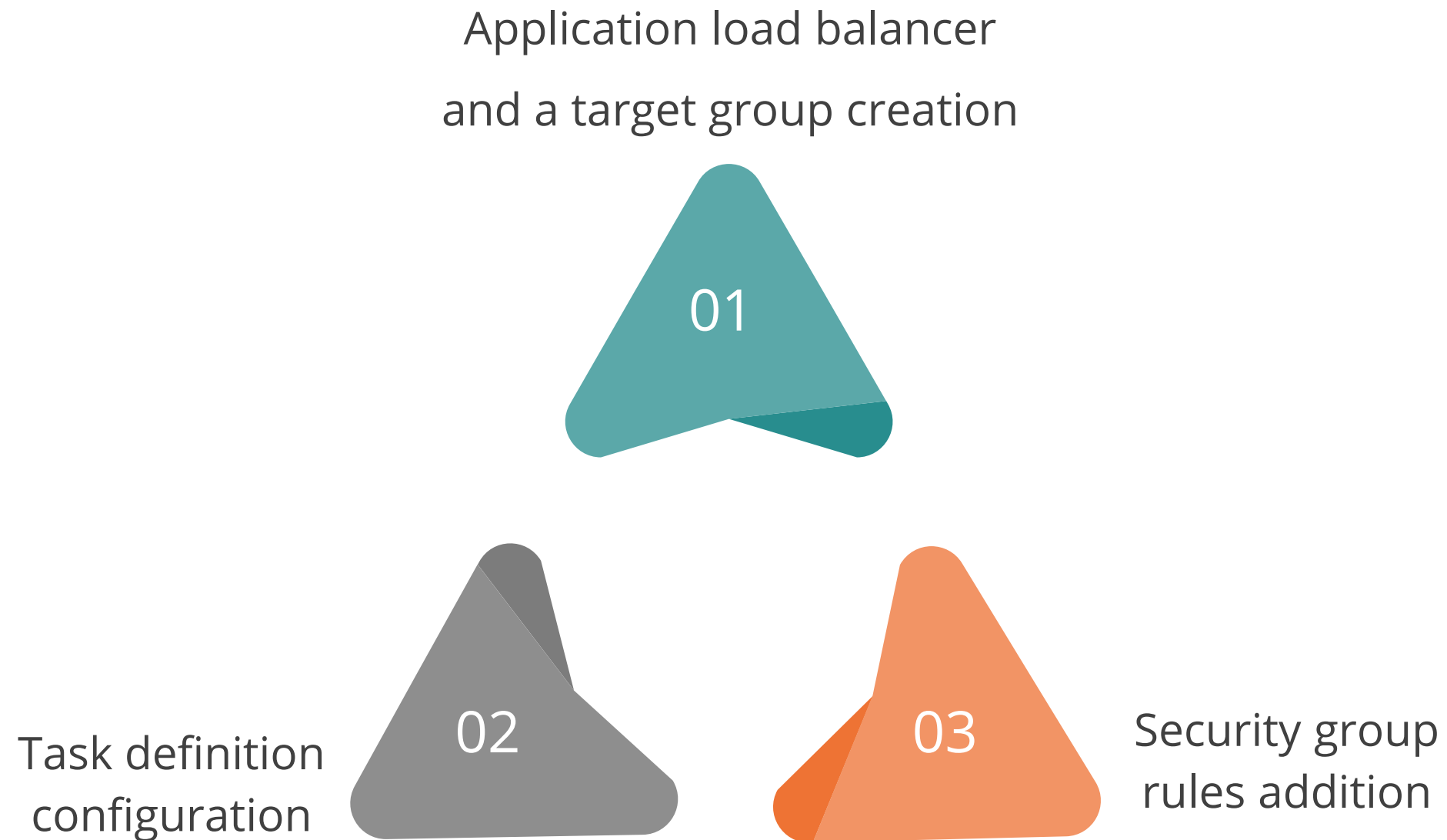


Dynamic port mapping can be achieved using network load balancers.



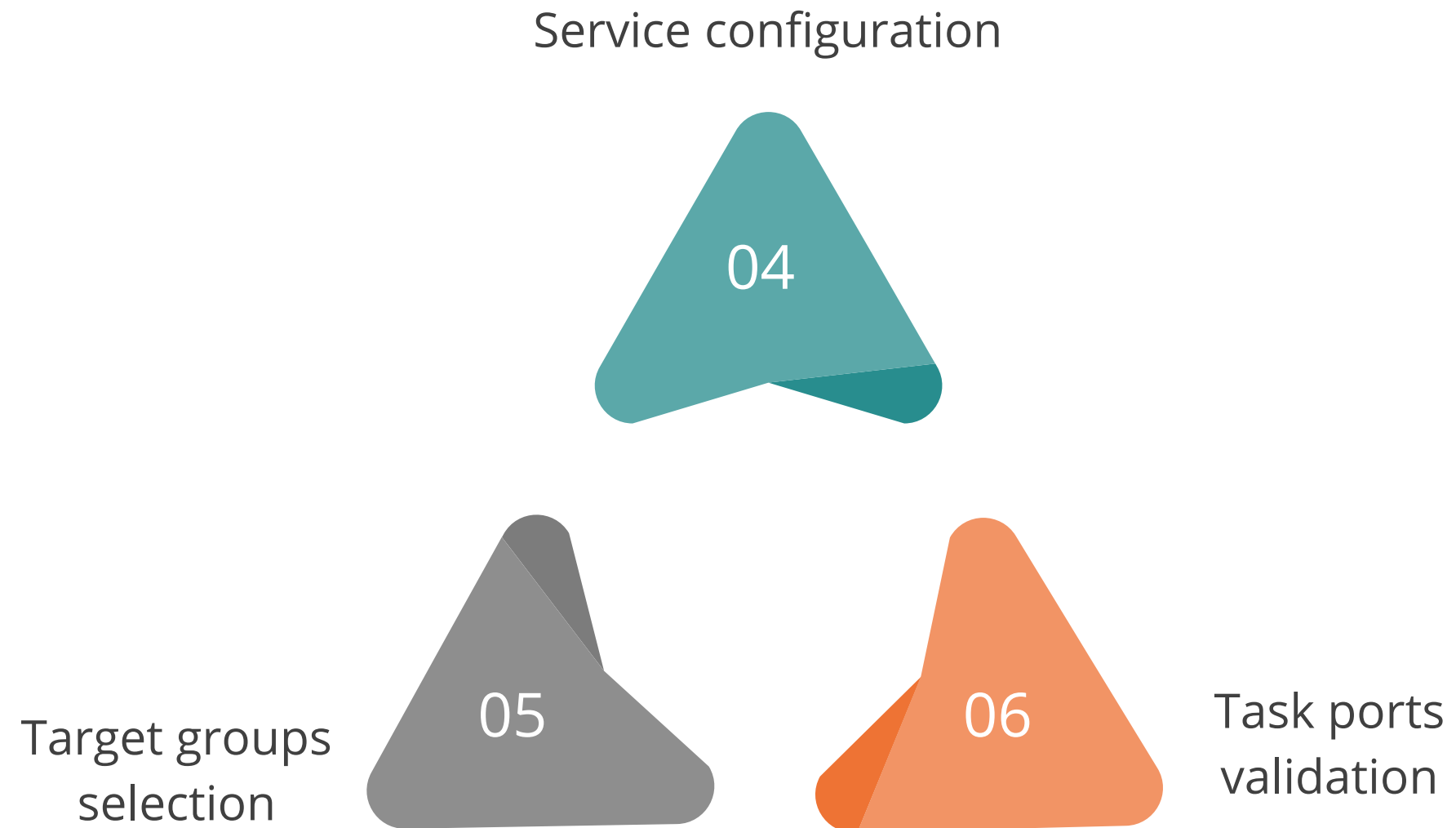
# Setting Up Dynamic Port Mapping

The steps to set up dynamic port mapping are as follows:



# Setting Up Dynamic Port Mapping

The steps to set up dynamic port mapping are as follows:



# Dynamic Port Integration with Application Load Balancer



Duration:15 min

## Problem Statement:

You have been asked to enable dynamic port mapping in a container.

ASSISTED PRACTICE

# Assisted Practice: Guidelines

---

## Steps to be followed:

1. Create a custom VPC and enable DNS hostname
2. Create Internet Gateway
3. Create three subnets
4. Create a Route table and attach it to three subnets
5. Create a Target group
6. Create an application load balancer
7. Create a Cluster
8. Create a Task definition
9. Run the task on the cluster

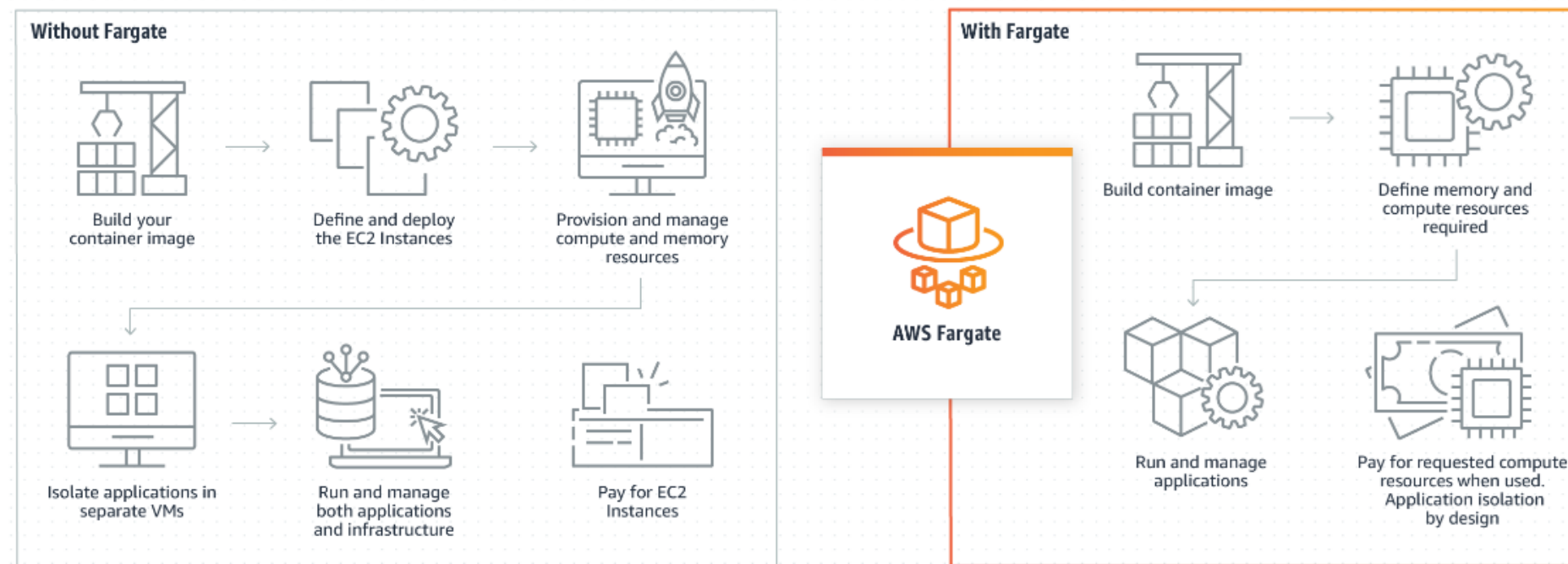




## Introduction to AWS Fargate

# AWS Fargate

AWS Fargate is an AWS service that acts as the compute engine for both Amazon ECS and Amazon EKS.



It enables you to run containers without the need to manage the underlying infrastructure.

# AWS Fargate: Advantages



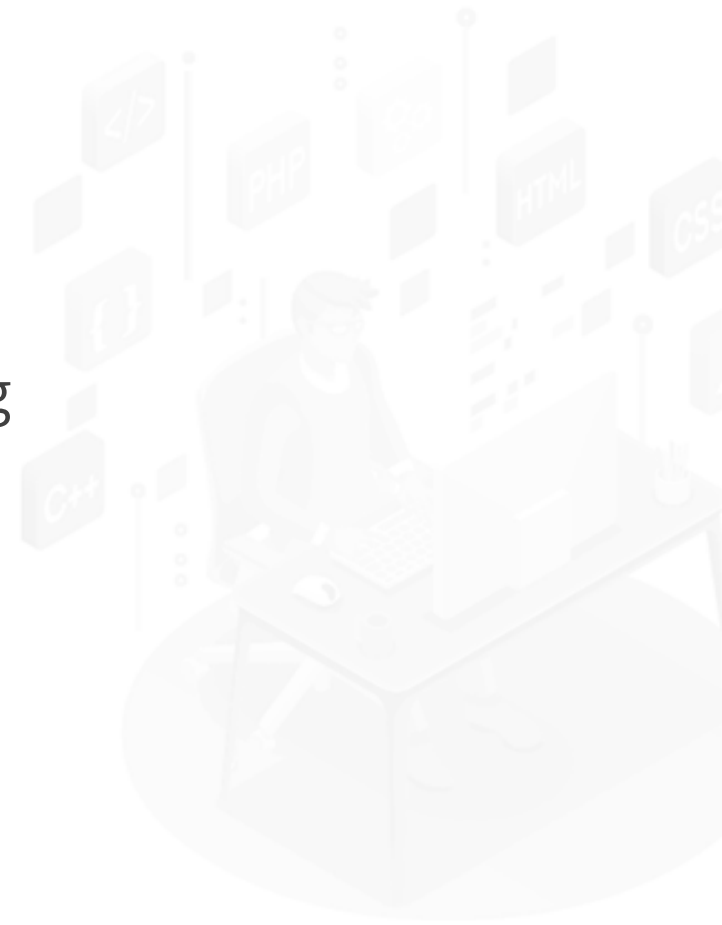
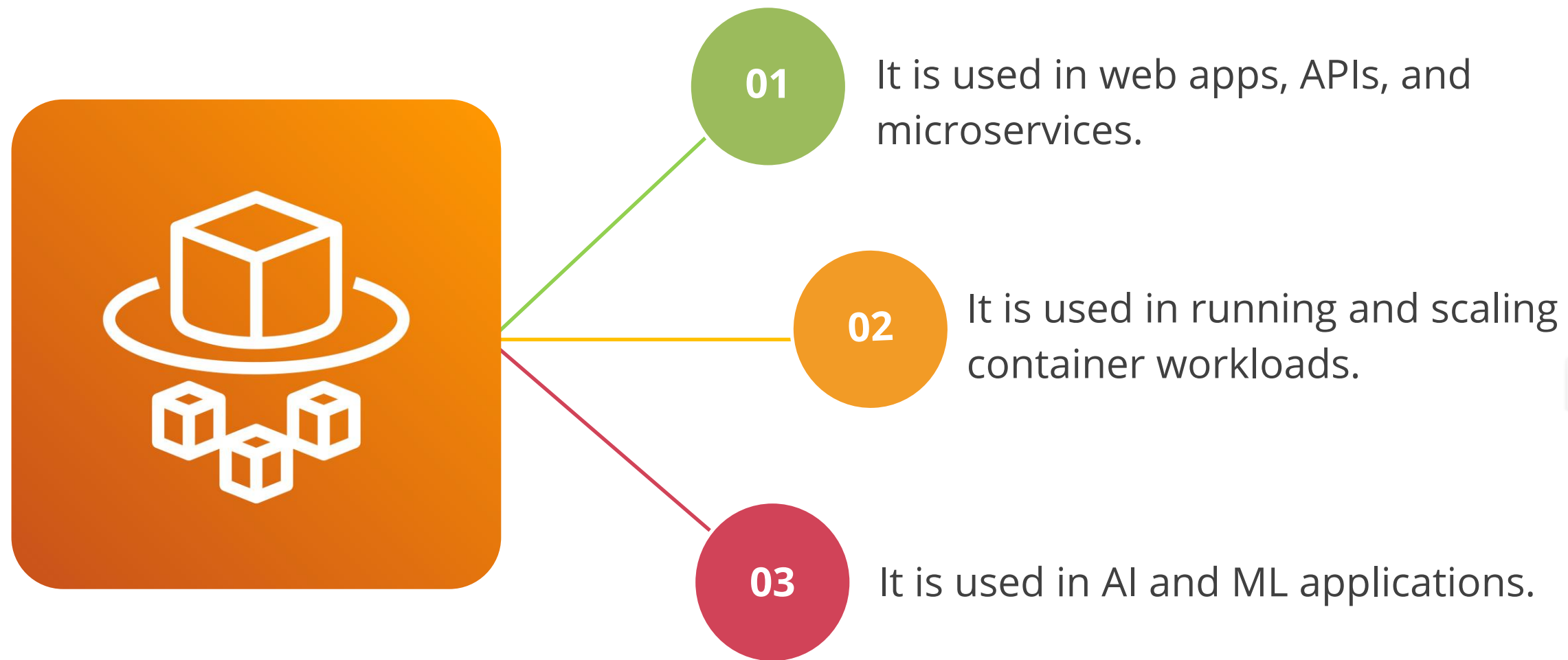
**Easy Management:** Fargate eliminates the need for manual server management.

**Auto-Scaling:** It automatically scales the cluster up or down to match changing demand.

**Health Checks:** It monitors container health and replaces faulty containers automatically.

**Task Definitions:** Users can define tasks and their execution environment precisely.

# AWS Fargate: Use Cases



# Running Task on a Fargate Cluster



**Duration: 15 min**

## **Problem Statement:**

You have been asked to run a task on a Fargate Cluster.

ASSISTED PRACTICE

# Assisted Practice: Guidelines

---

## Steps to be followed:

1. Create a Fargate cluster
2. Create a task definition
3. Run Fargate Cluster





## Key Takeaways

- Containers are software packages that are used to execute an application in any environment.
- Docker is a containerization platform that enables users to package applications into containers.
- The deployment, management, scaling, and networking of containers are all automated through container orchestration.
- Amazon ECS is a container orchestration service that supports Docker containers and allows users to easily run and scale containerized applications on AWS.
- Amazon Fargate is a serverless container platform for running Docker containers.



# Monitoring Insights for ECS Cluster Using CloudWatch

Duration: 30 mins



**Project agenda:** To monitor an Amazon Elastic Container Service (ECS) cluster using CloudWatch Container Insights

**Description:** Your company is going to deploy a container application using ECS clusters. As a cloud architect, you should be able to monitor the cluster's health by measuring various metrics such as CPU utilization, memory utilization, task count, and so on.

**Perform the following:**

1. Monitor the cluster using the CloudWatch container

# TECHNOLOGY

**Thank You**