CS 4037
Introduction to Cloud Computing
Lecture 26.2

Danyal Farhat

FAST School of Computing

NUCES Lahore

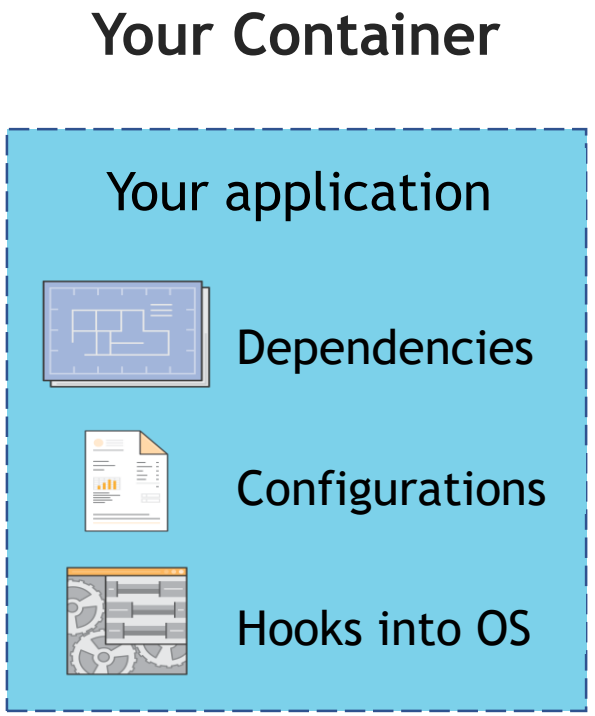
AWS Compute – Part 2

Lecture's Agenda

- Compute Services Overview
- Amazon EC2
- EC2 Cost Optimization
- **Container Services**
- Introduction to AWS Lambda
- Introduction to AWS Elastic Beanstalk

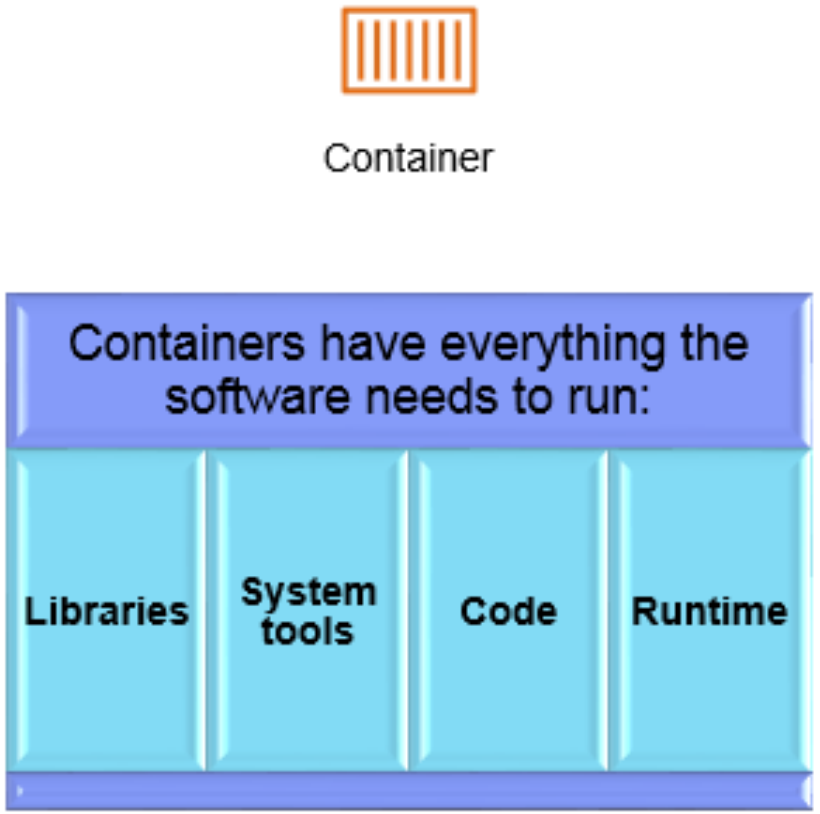
Containers

- **Containers are a method of operating system virtualization**
- **Benefits:**
  - **Repeatable**
  - **Self-contained environments**
  - **Software runs the same in different environments**
    - ✓ **Developer's laptop, test, production**
  - **Faster to launch and stop or terminate than virtual machines**



Docker

- **“Docker is a software platform that enables to build, test, and deploy applications quickly”**
- **Docker packages software (such as applications) into containers**
- **Docker is installed on each server that will host containers**
  - **Ensure OS Virtualization**

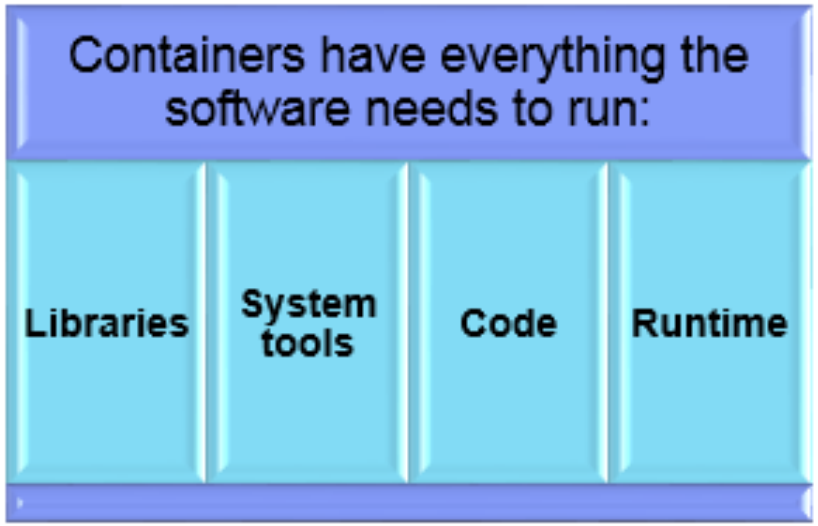


Docker (Cont.)

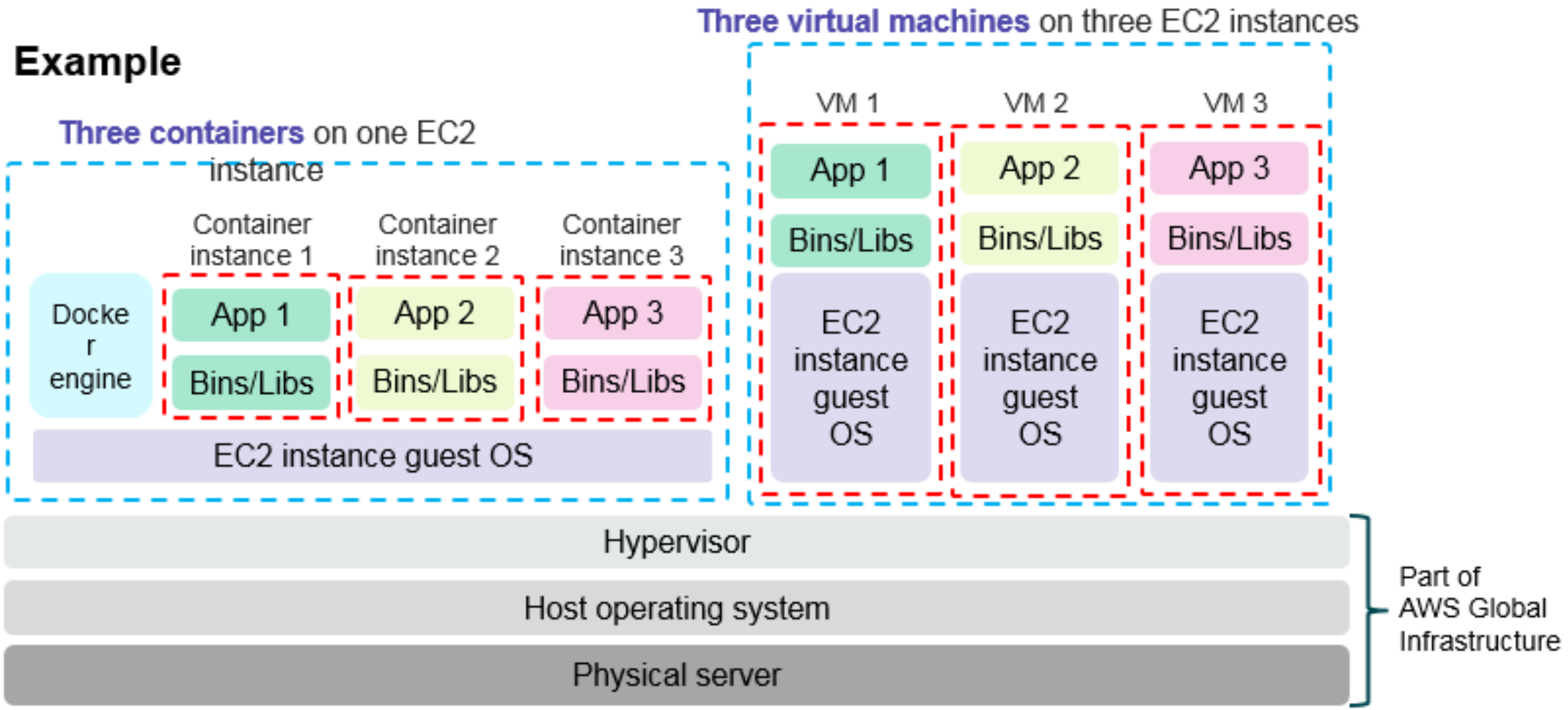
- **Containers are created from a template called Docker image**

- **A container has everything a software application needs to run**

Container



Containers vs. Virtual Machines



Amazon Elastic Container Service (Amazon ECS)

- **A highly scalable, managed container management service**
- **Key benefits are:**
  - **Orchestrates the running of Docker containers**
  - **Maintains and scales the fleet of nodes (instances) that run the containers**
  - **Removes the complexity of standing up the infrastructure**
- **Integrated with features that are familiar to Amazon EC2 service users**
  - **Elastic Load Balancing**
  - **Amazon EC2 security groups**
  - **Amazon EBS volumes**
  - **IAM roles**

Amazon Elastic Container Service

Amazon ECS – Task Definition

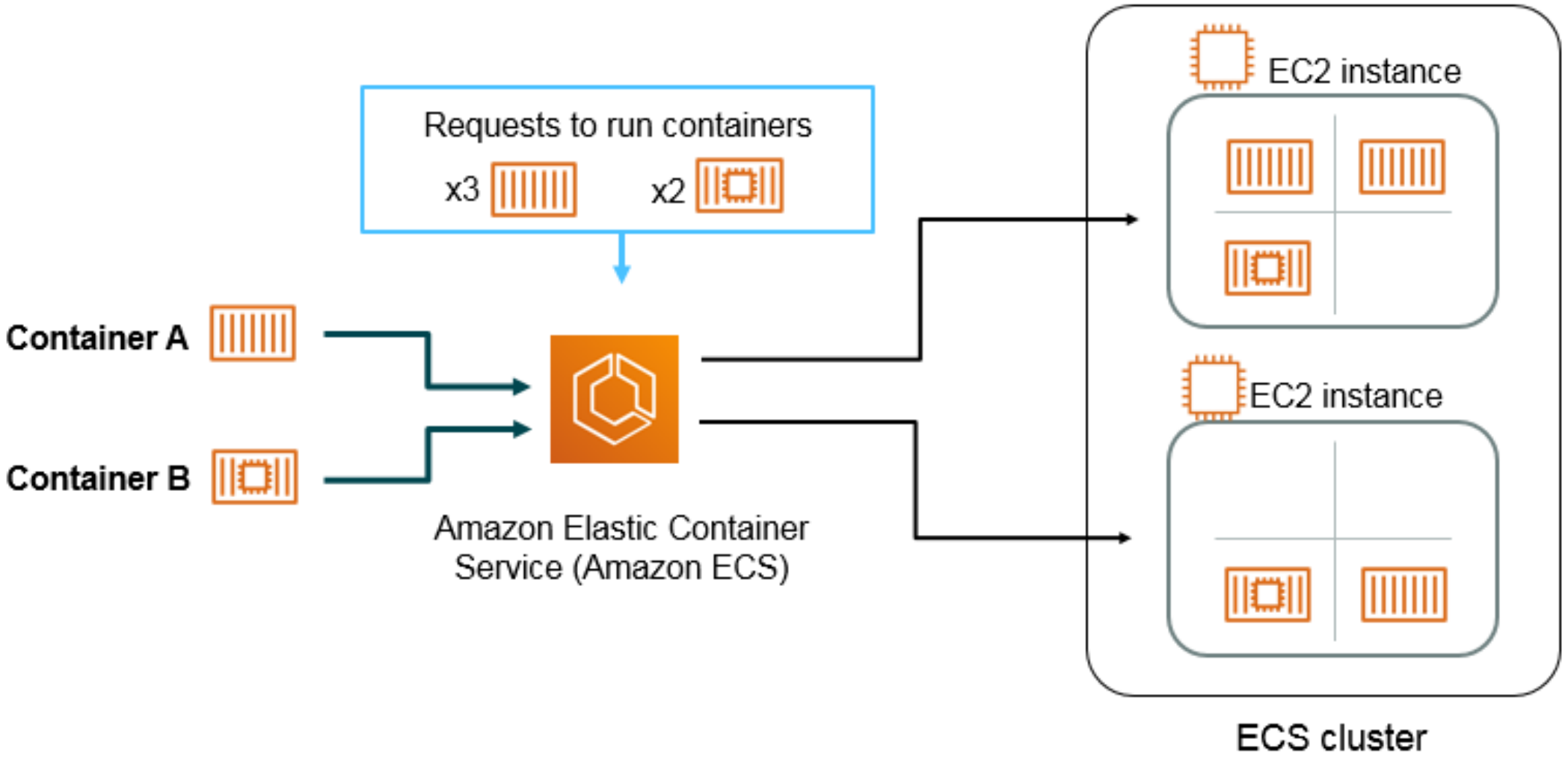
- **To run an app on ECS, customer creates a task definition**
  - **Blueprint for customer's application**
  - **Text file that describes one or more (up to maximum ten) containers**

- **Task definition specify parameters for app**
  - **Which containers to use?**
  - **Which ports should be opened?**
  - **What data volume should be used with the containers in the task?**

Amazon ECS – Task Scheduling

- **Task is the instantiation of a task definition within a cluster**
  - **Customers specify the number of tasks that will run on cluster**
- **Amazon ECS task scheduler is responsible for placing tasks within the cluster**
- **A task will run anywhere from one to ten containers, depending on the task definition that customer defined**
- **ECS cluster consists of a group of EC2 instances each of which is running an ECS container agent**

Amazon ECS Orchestrates Containers



Amazon ECS Cluster Options

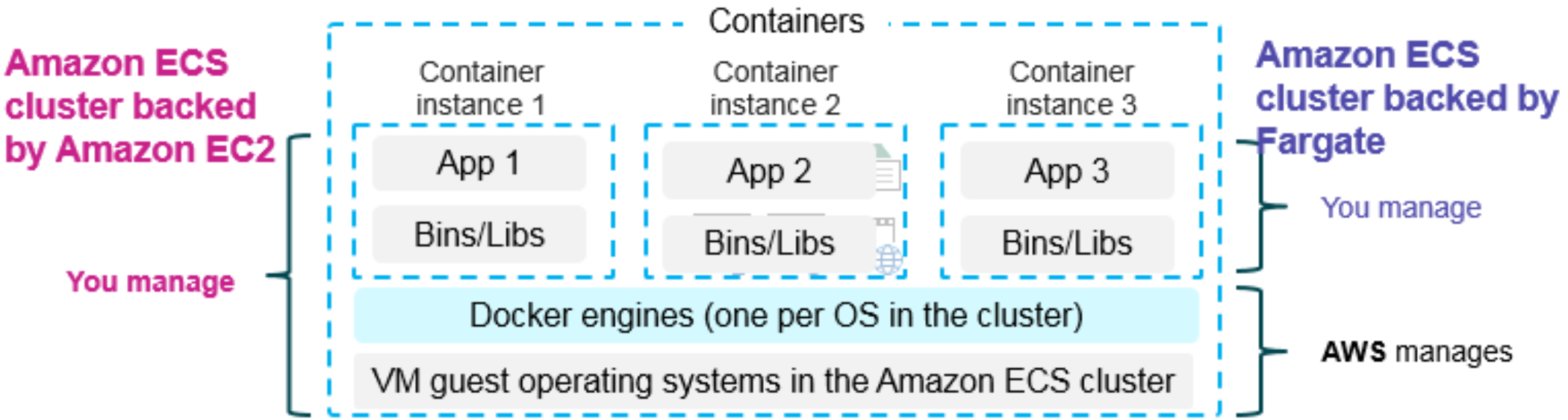
Do you want to manage the Amazon ECS cluster that runs the containers?

- **If yes, create an Amazon ECS cluster backed by Amazon EC2 (provides more granular control over infrastructure)**

- **If no, create an Amazon ECS cluster backed by AWS Fargate (easier to maintain, focus on your applications)**

Amazon ECS Cluster Options (Cont.)

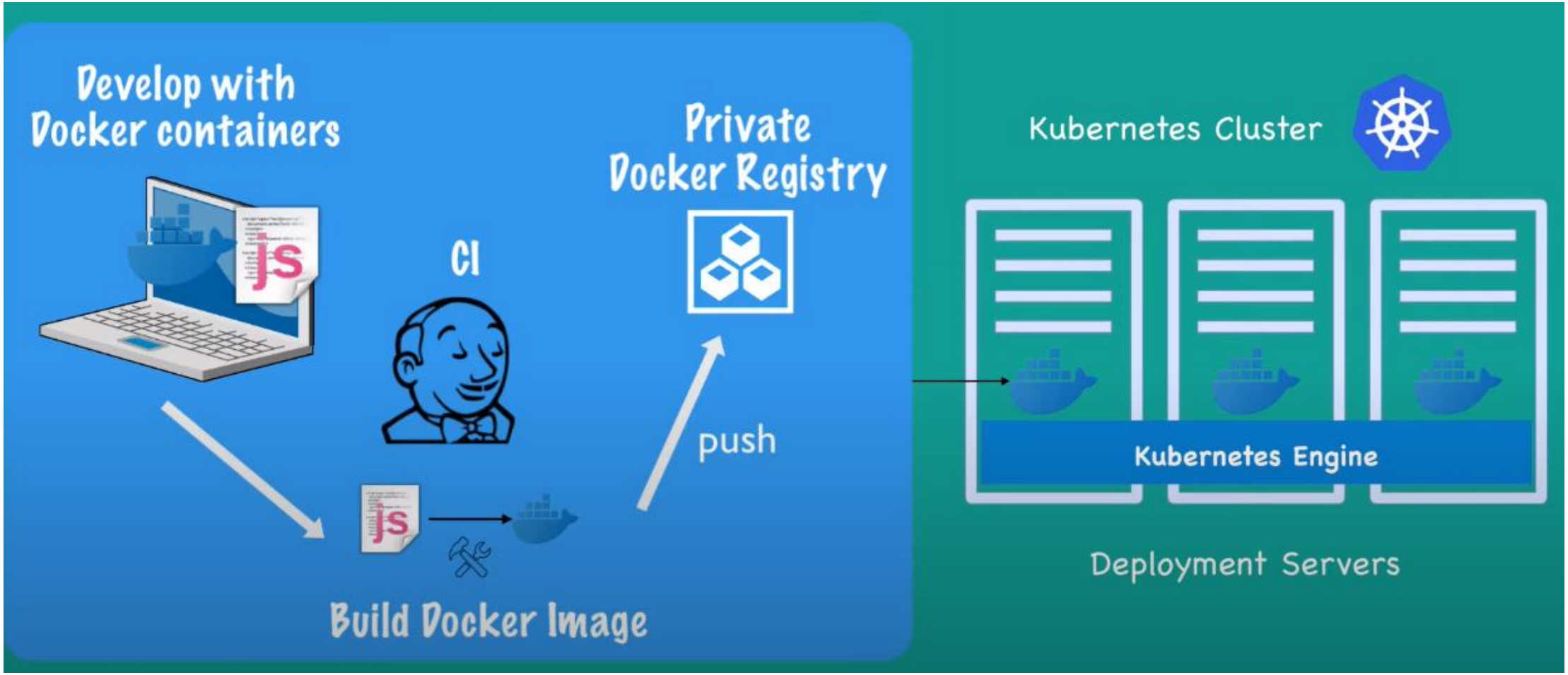
- **When user creates an ECS cluster, user has three options:**
  - **A Networking Only cluster (powered by AWS Fargate)**
  - **An EC2 Linux + Networking cluster**
  - **An EC2 Windows + Networking cluster**



Kubernetes

- **Open source software for container orchestration**
  - Deploy and manage containerized applications at scale
  - The same toolset can be used on premises and in the cloud
- **Complements Docker**
  - Docker enables you to run multiple containers on a single OS host
  - Kubernetes orchestrates multiple Docker hosts (nodes)
- **Automates**
  - Container provisioning
  - Networking
  - Load distribution
  - Scaling

Docker and Kubernetes in Software Development Process



Kubernetes

- Kubernetes manages a cluster of compute instances (called nodes)
- It runs containers on the cluster based on where compute resources are available and the resource requirements of each container
- Containers are run in logical groupings called pods
- Customer can run and scale one or many containers together as a pod
  - Each pod is given an IP address and a single DNS name, which Kubernetes uses to connect the services with each other and external traffic

Amazon Elastic Kubernetes Service (Amazon EKS)

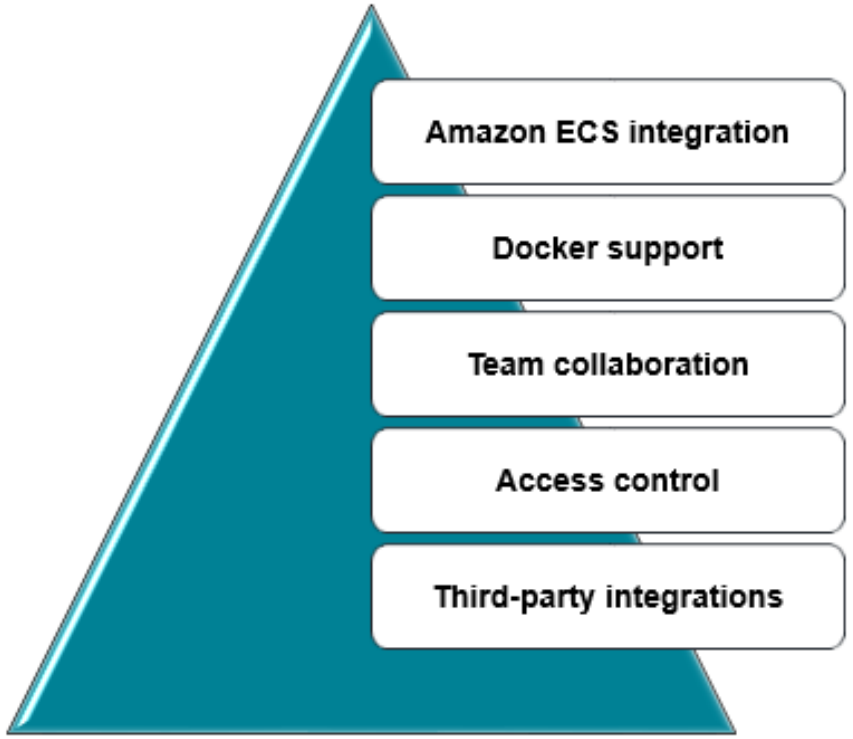
- **Managed service to run Kubernetes**
- **Certified Kubernetes conformant (supports easy migration)**
- **Supports Linux and Windows containers**
- **Compatible with Kubernetes community tools and supports popular Kubernetes add-ons**

Amazon Elastic Kubernetes Service

- **Amazon EKS used to:**
  - **Manage clusters of Amazon EC2 compute instances**
  - **Run containers that are orchestrated by Kubernetes on those instances**

Amazon Elastic Container Registry (Amazon ECR)

- **“Amazon ECR is a fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images.”**



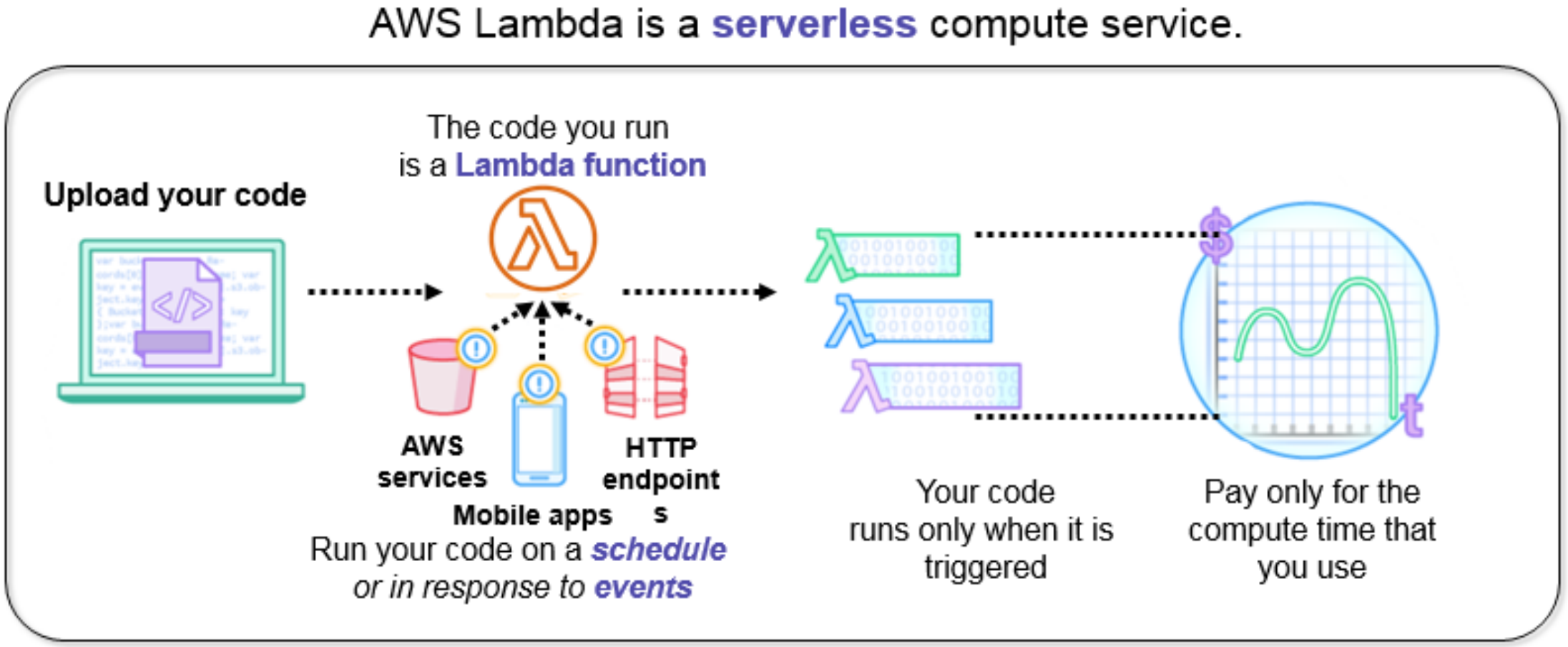
Container Services – Key Points

- **Containers** can hold everything that an app needs to run
- **Docker** is a software platform that packages software into containers
  - A single application can span multiple containers
- **Amazon ECS** orchestrates the running of Docker containers
- **Kubernetes** is open source software for container orchestration
- **Amazon EKS** enables to run Kubernetes on AWS
- **Amazon ECR** enables to store, manage, and deploy Docker based containers

Lecture's Agenda

- Compute Services Overview
- Amazon EC2
- EC2 Cost Optimization
- Container Services
- **Introduction to AWS Lambda**
- Introduction to AWS Elastic Beanstalk

AWS Lambda: Run Code Without Servers



Benefits of Lambda

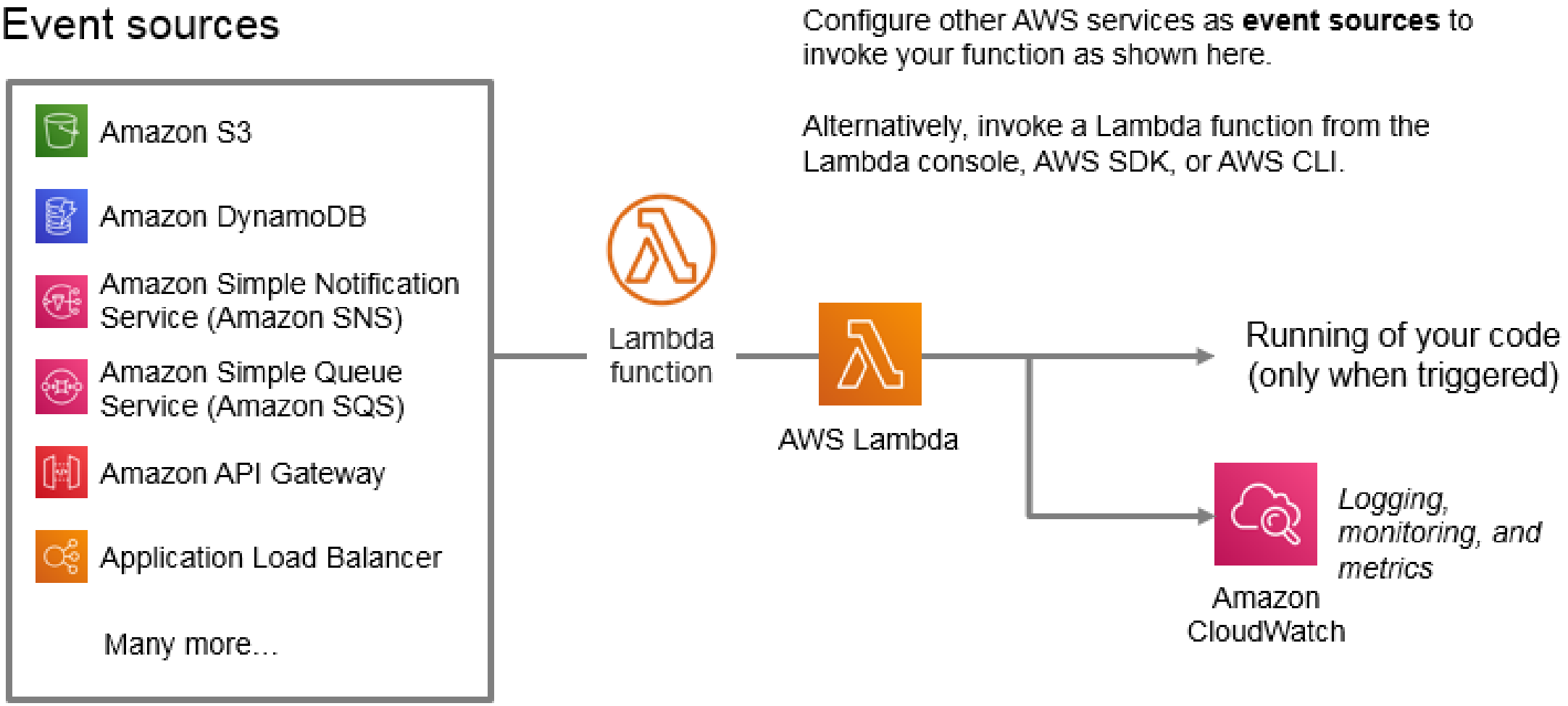
- **Supports multiple programming languages**
  - **Java, Go, PowerShell, Node.js, C#, Python, and Ruby**
- **Your code can use any library, either native or third-party**
- **Completely automated administration**
- **Built-in fault tolerance**
- **Pay-per-use pricing**
- **Supports orchestration of multiple functions**
  - **AWS Step Functions**

AWS Lambda

AWS Lambda & AWS Step Functions

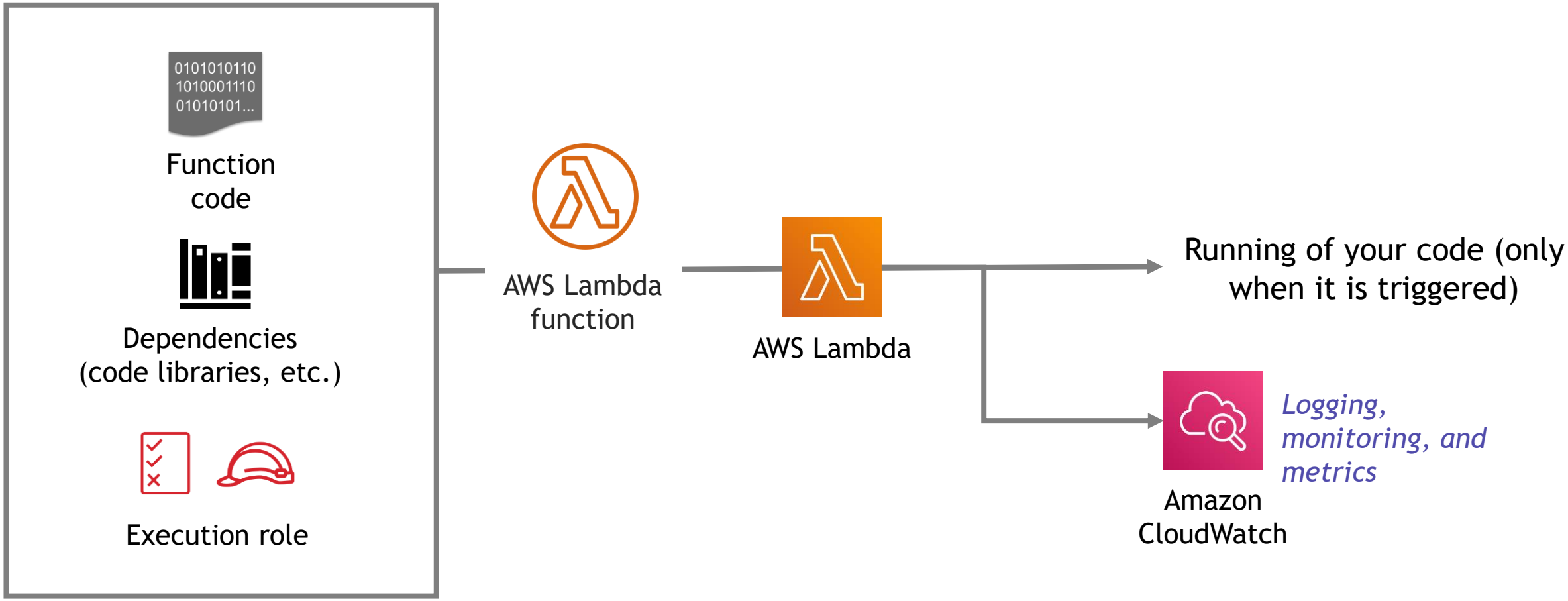
- You can orchestrate multiple Lambda functions for complex or long-running tasks by building workflows with AWS Step Functions
- Use Step Functions to define workflows
- Workflows trigger a collection of Lambda functions by using sequential, parallel, branching, and error-handling steps
- With Step Functions and Lambda, you can build stateful, long-running processes for applications and back-ends

AWS Lambda Event Sources

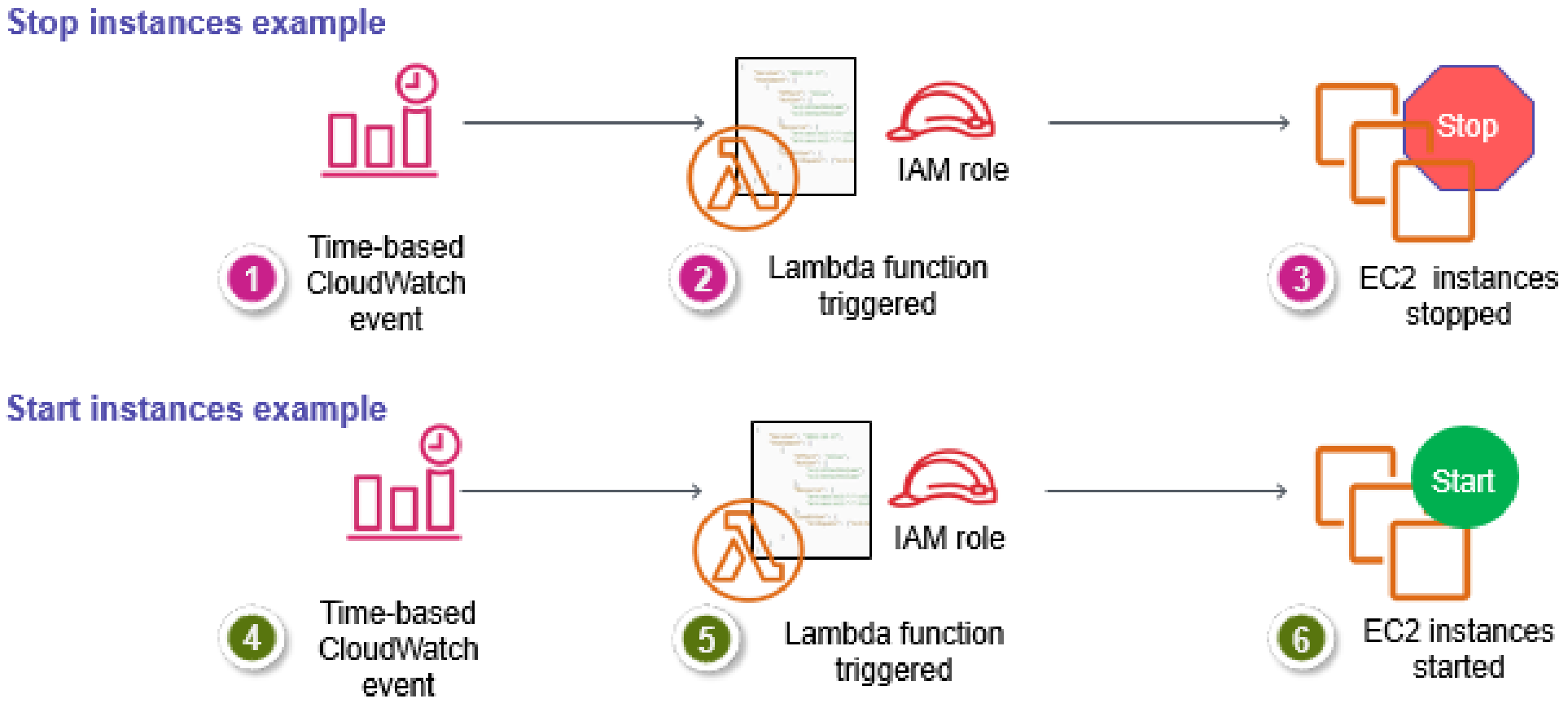


AWS Lambda Function Configuration

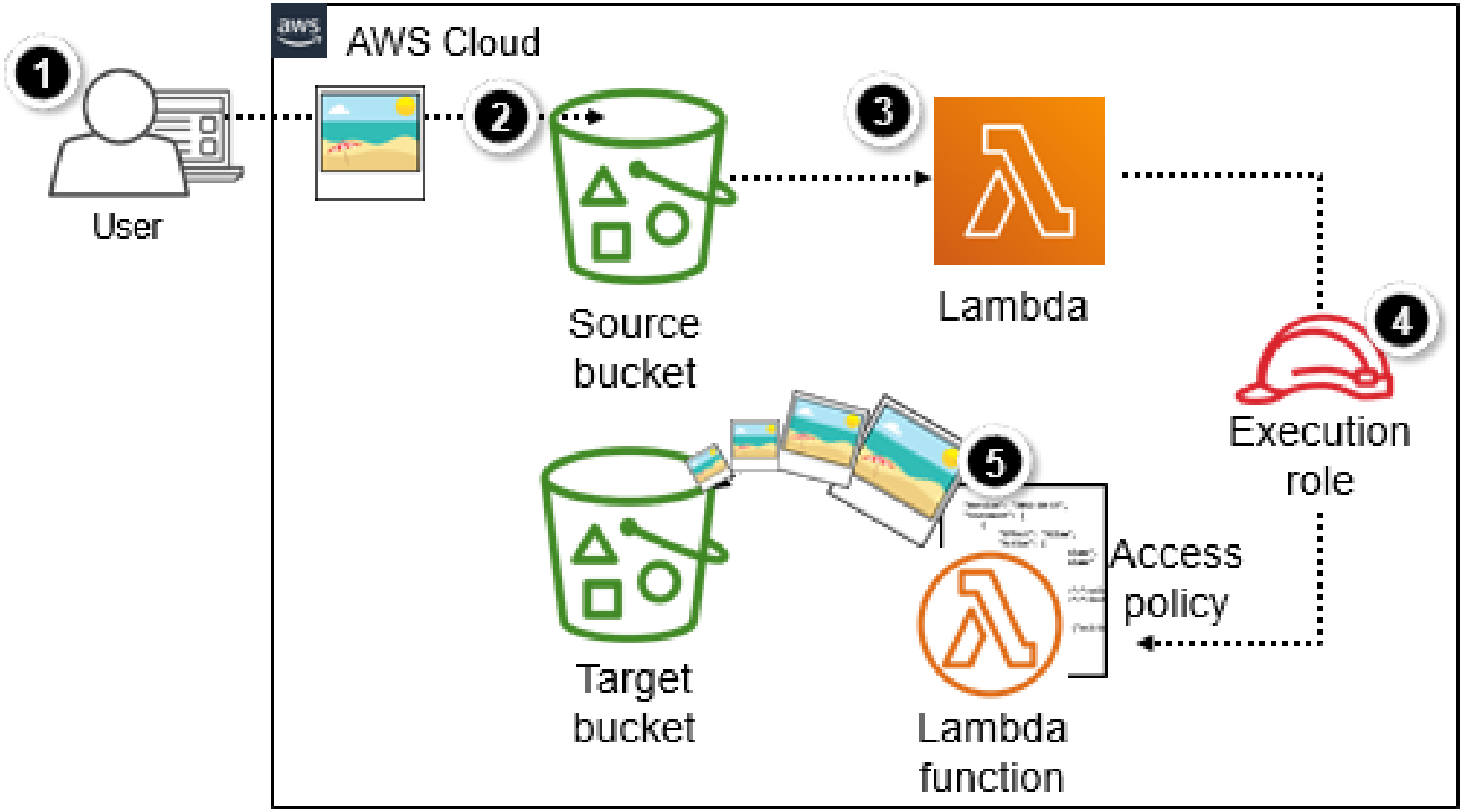
Lambda function configuration



Schedule-Based Lambda Function Example: Start and Stop EC2 Instances



Event-Based Lambda Function Example: Create Thumbnail Images



AWS Lambda Quotas

- **Soft limits per Region**
  - **Concurrent executions = 1,000**
  - **Function and layer storage = 75 GB**

- **Hard limits for individual functions**
  - **Maximum function memory allocation = 10,240 MB (10 GB)**
  - **Function timeout = 15 minutes**
  - **Deployment package size = 250 MB unzipped, including layers**
  - **Container image code package size = 10 GB**

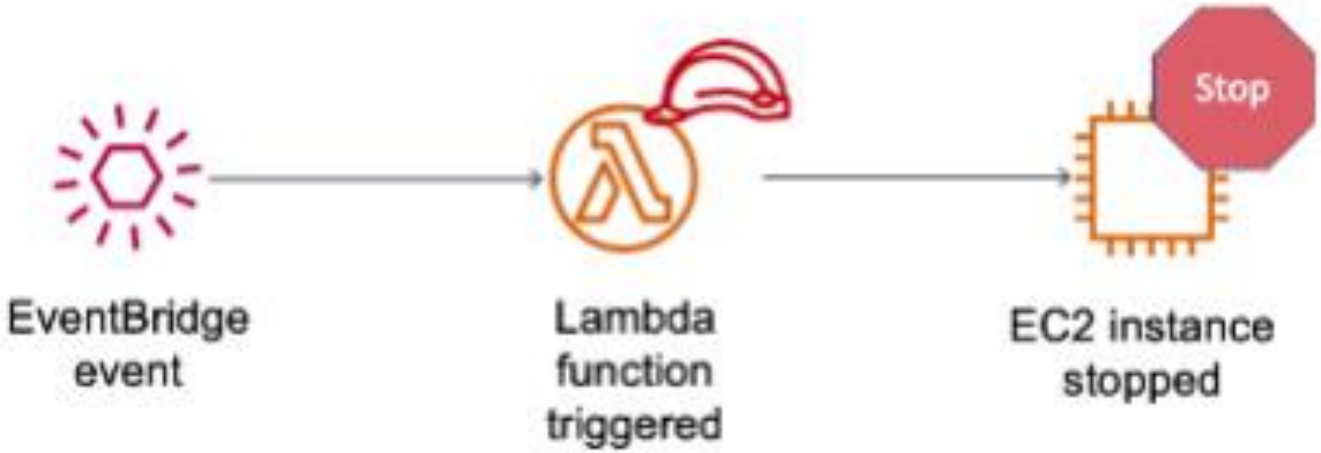
Lab 3.2: AWS Lambda Activity

Lab Scenario:

In this hands-on activity, you will create an AWS Lambda function. You will also create an Amazon EventBridge event to trigger the function every minute. The function uses an AWS IAM role. This IAM role allows the function to stop EC2 instance that is running in the AWS account.

Lab 3.2: AWS Lambda Activity (Cont.)

Final Product:



Lab 3.2: AWS Lambda Activity (Cont.)

Lab Tasks:

- **Task 1 – Create a Lambda function**
- **Task 2 – Configure the trigger**
- **Task 3 – Configure the Lambda function**
- **Task 4 – Verify that the Lambda function worked**

Introduction to AWS Lambda – Key Points

- **Serverless computing enables you to build and run applications and services without provisioning or managing servers**
- **AWS Lambda is a event-driven serverless compute service that provides built-in fault tolerance and automatic scaling**
- **An event source is an AWS service or developer-created application that triggers a Lambda function to run**
- **The maximum memory allocation for a single Lambda function is 10,240 MB**
- **The maximum run time for a Lambda function is 15 minutes**

Lecture's Agenda

- Compute Services Overview
- Amazon EC2
- EC2 Cost Optimization
- Container Services
- Introduction to AWS Lambda
- **Introduction to AWS Elastic Beanstalk**

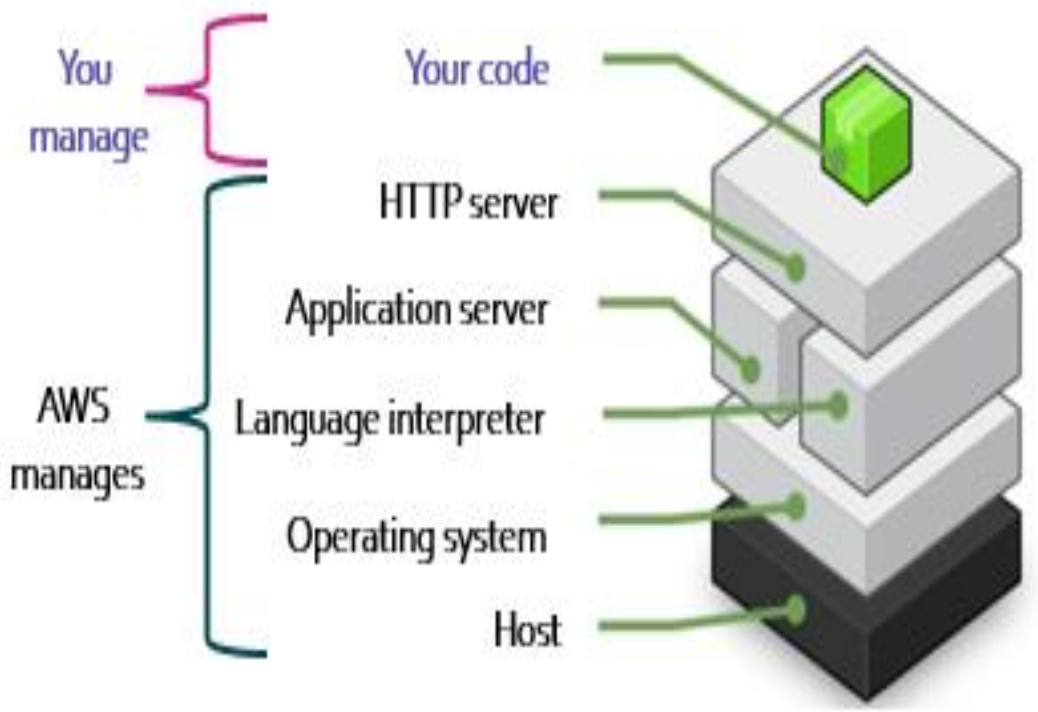
AWS Elastic Beanstalk

- An easy way to get web applications up and running
- A managed service that automatically handles:
  - Infrastructure provisioning and configuration
  - Deployment
  - Load balancing
  - Automatic scaling
  - Health monitoring
  - Analysis and debugging
  - Logging
- No additional charge for Elastic Beanstalk
  - Pay only for the underlying resources that are used

AWS Elastic Beanstalk

AWS Elastic Beanstalk Deployment

- **Supports web applications written for common platforms**
  - **Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker**
- **Customer upload the code**
  - **Elastic Beanstalk automatically handles the deployment**
  - **Deploys on servers such as Apache, NGINX, Passenger, Puma, and Microsoft Internet Information Services (IIS)**

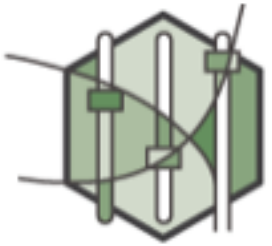


Benefits of Elastic Beanstalk

Fast and simple to start using

Developer productivity

Difficult to outgrow



Complete resource control

AWS Elastic Beanstalk – Key Points

- **AWS Elastic Beanstalk enhances developer productivity**
  - **Simplifies the process of deploying your application**
  - **Reduces management complexity**

- **Elastic Beanstalk supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker**

- **There is no charge for Elastic Beanstalk**
  - **Pay only for the underlying AWS resources that you use**

Additional Resources

- **Amazon ECS Workshop**
  - **https://ecsworkshop.com/**
- **Running Containers on AWS**
  - **https://containersonaws.com/**
- **AWS Lambda Documentation**
  - **https://docs.aws.amazon.com/lambda/**
- **AWS Elastic Beanstalk Documentation**
  - **https://docs.aws.amazon.com/elastic-beanstalk/**
- **Cost Optimization Playbook**
  - **https://d1.awsstatic.com/pricing/AWS_CO_Playbook_Final.pdf**

Questions?