**Additional compute services**

Video Transcript :

EC2 instances are virtual machines that you can provision with minimal friction to get up and running on AWS. EC2 is great for all sorts of different use cases like running basic web servers to running high performance computing clusters and everything in between.

That being said, though EC2 is incredibly flexible, reliable, and scalable, depending on your use case, you might be looking at alternatives for your compute capacity. EC2 requires that you set up and manage your fleet of instances over time. When you're using EC2, you are responsible for patching your instances when new software packages come out, setting up the scaling of those instances as well as ensuring that you've architected your solutions to be hosted in a highly available manner. This is still not as much management as you would have if you hosted these on-premises. But management processes will still need to be in place.

You might be wondering what other services does AWS offer for compute that are more convenient from a management perspective. This is where the term serverless comes in. AWS offers multiple serverless compute options. Serverless means that you cannot actually see or access the underlying infrastructure or instances that are hosting your application. Instead, all the management of the underlying environment from a provisioning, scaling, high availability, and maintenance perspective are taken care of for you. All you need to do is focus on your application and the rest is taken care of.

AWS Lambda is one serverless compute option. Lambda's a service that allows you to upload your code into what's called a Lambda function. Configure a trigger and from there, the service waits for the trigger. When the trigger is detected, the code is automatically run in a managed environment, an environment you do not need to worry too much about because it is automatically scalable, highly available and all of the maintenance in the environment itself is done by AWS. If you have one or 1,000 incoming triggers, Lambda will scale your function to meet demand. Lambda is designed to run code under 15 minutes so this isn't for long running processes like deep learning. It's more suited for quick processing like a web backend, handling requests or a backend expense report processing service where each invocation takes less than 15 minutes to complete.

If you weren't quite ready for serverless yet or you need access to the underlying environment, but still want efficiency and portability, you should look at AWS container services like Amazon Elastic Container Service, otherwise known as ECS. Or Amazon Elastic Kubernetes Service, otherwise known as EKS.

Both of these services are container orchestration tools, but before I get too far here, a container in this case is a Docker container. Docker is a widely used platform that uses operating system level virtualization to deliver software in containers. Now a container is a package for your code where you package up your application, its dependencies as well as any configurations that it needs to run. These containers run on top of EC2 instances and run in isolation from each other similar to how virtual machines work. But in this case, the host is an EC2 instance. When you use Docker containers on AWS, you need processes to start, stop, restart, and monitor containers running across not just one EC2 instance, but a number of them together which is called a cluster.

The process of doing these tasks is called container orchestration and it turns out it's really hard to do on your own. Orchestration tools were created to help you manage your containers. ECS is designed to help you run your containerized applications at scale without the hassle of managing your own container orchestration software. EKS does a similar thing, but uses different tooling and with different features.

Both Amazon ECS and Amazon EKS can run on top of EC2. But if you don't want to even think about using EC2s to host your containers because you either don't need access to the underlying OS or you don't want to manage those EC2 instances, you can use a compute platform called AWS Fargate. Fargate is a serverless compute platform for ECS or EKS. That's a bit high level and it might be confusing, so let's clear that up.

If you are trying to host traditional applications and want full access to the underlying operating system like Linux or Windows, you are going to want to use EC2. If you are looking to host short running functions, service-oriented or event driven applications and you don't want to manage the underlying environment at all, look into the serverless AWS Lambda. If you are looking to run Docker container-based workloads on AWS, you first need to choose your orchestration tool. Do you want to use Amazon ECS or Amazon EKS? After you choose your tool, you then need to chose your platform. Do you want to run your containers on EC2 instances that you manage or in a serverless environment like AWS Fargate that is managed for you?

Those are just some of your compute options with AWS and it's not even a complete list. Check out the notes for more information on AWS compute services as well as others that we didn't get to talk about in this video.

**Serverless computing**

Earlier in this module, you learned about Amazon EC2, a service that lets you run virtual servers in the cloud. If you have applications that you want to run in Amazon EC2, you must do the following:

* 1

1

Provision instances (virtual servers).

* 2

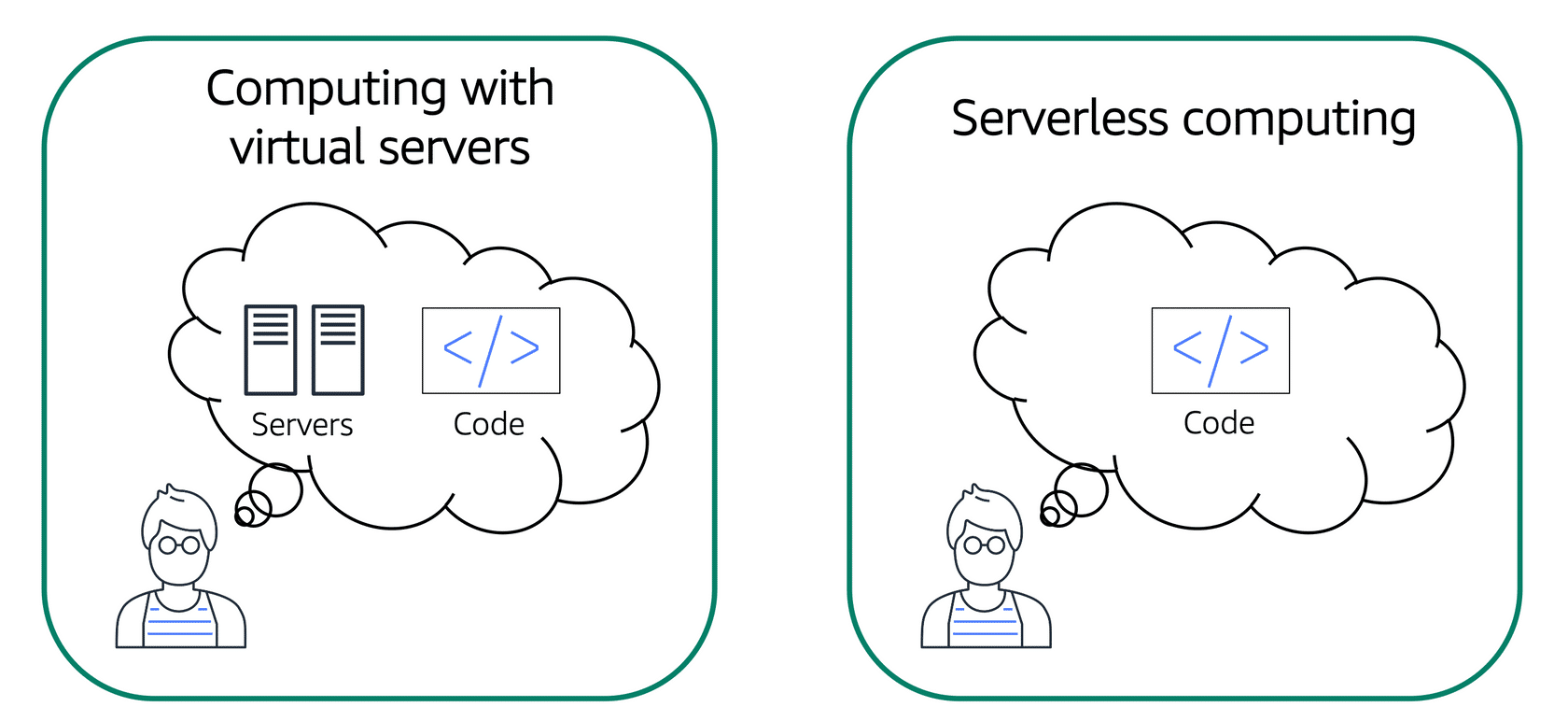
2

Upload your code.

* 3

3

Continue to manage the instances while your application is running.



The term “serverless” means that your code runs on servers, but you do not need to provision or manage these servers. With serverless computing, you can focus more on innovating new products and features instead of maintaining servers.

Another benefit of serverless computing is the flexibility to scale serverless applications automatically. Serverless computing can adjust the applications' capacity by modifying the units of consumptions, such as throughput and memory.

An AWS service for serverless computing is **AWS Lambda**.

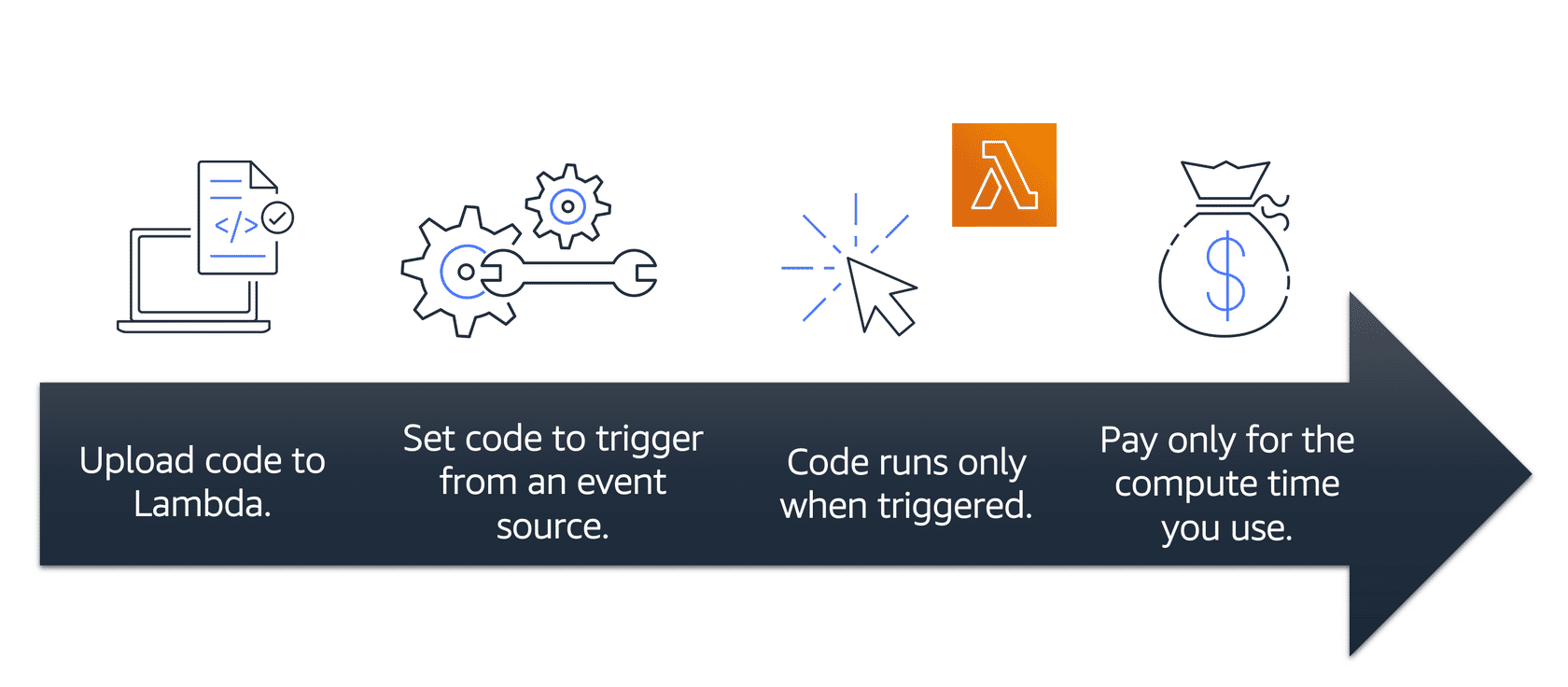
**AWS Lambda**

[**AWS Lambda**](https://aws.amazon.com/lambda) is a service that lets you run code without needing to provision or manage servers.

While using AWS Lambda, you pay only for the compute time that you consume. Charges apply only when your code is running. You can also run code for virtually any type of application or backend service, all with zero administration.

For example, a simple Lambda function might involve automatically resizing uploaded images to the AWS Cloud. In this case, the function triggers when uploading a new image.

**How AWS Lambda works**



* 1

1

You upload your code to Lambda.

* 2

2

You set your code to trigger from an event source, such as AWS services, mobile applications, or HTTP endpoints.

* 3

3

Lambda runs your code only when triggered.

* 4

4

You pay only for the compute time that you use. In the previous example of resizing images, you would pay only for the compute time that you use when uploading new images. Uploading the images triggers Lambda to run code for the image resizing function.

In AWS, you can also build and run **containerized** applications.

**Containers**

**Containers** provide you with a standard way to package your application's code and dependencies into a single object. You can also use containers for processes and workflows in which there are essential requirements for security, reliability, and scalability.

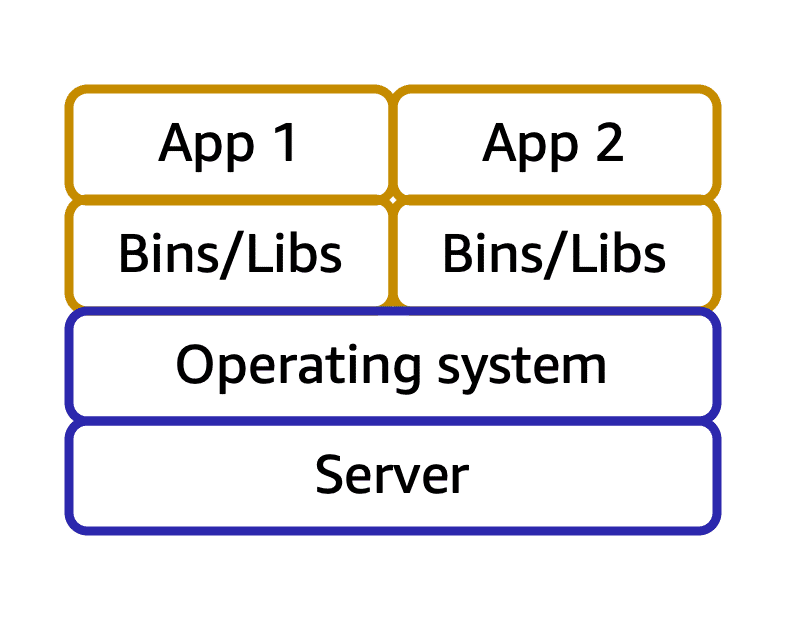
## Example: Containers

To review an example that can help you to understand how containers work, select **Start**.

START

**Step 1**

## One host with multiple containers



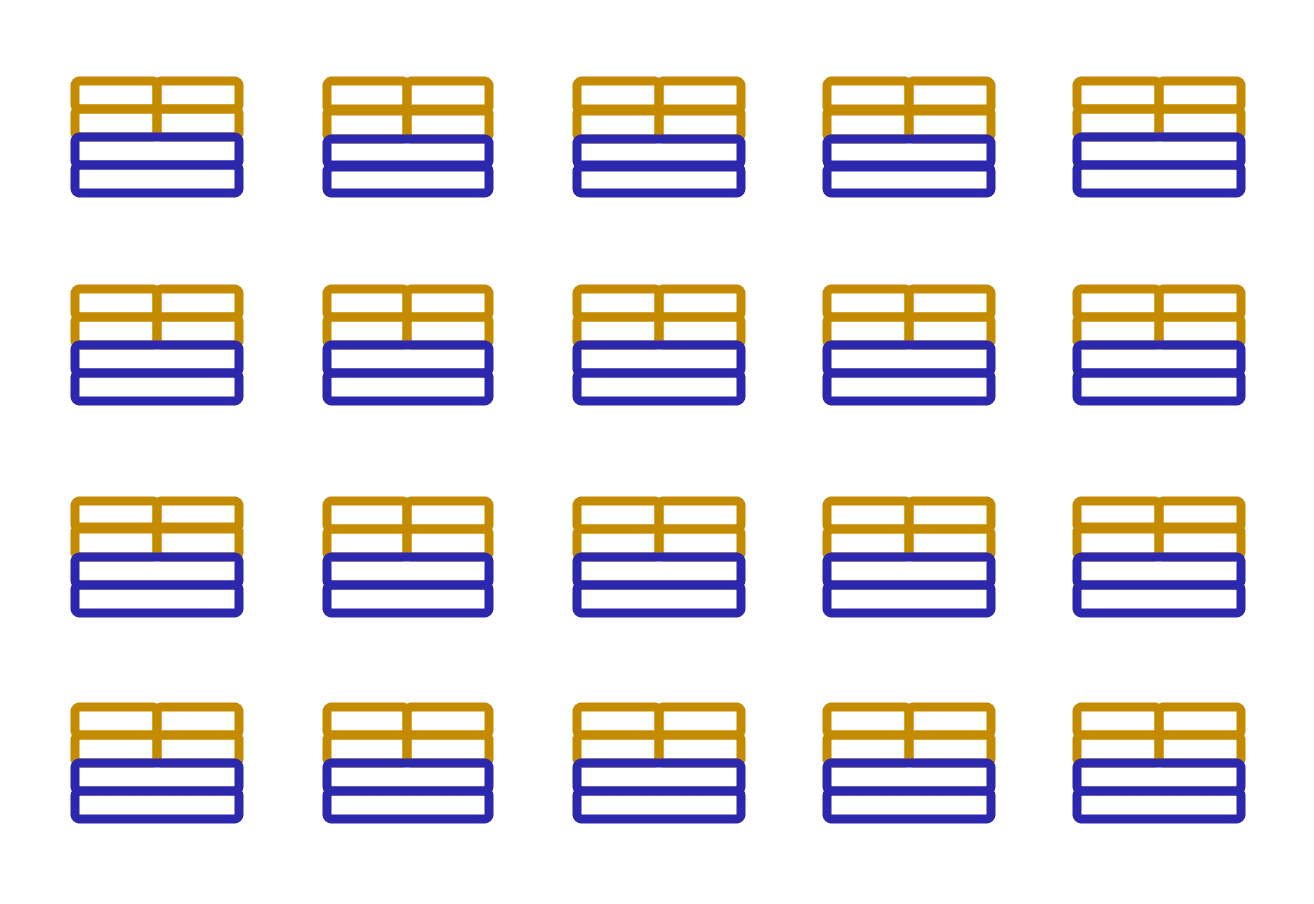
Suppose that a company’s application developer has an environment on their computer that is different from the environment on the computers used by the IT operations staff. The developer wants to ensure that the application’s environment remains consistent regardless of deployment, so they use a containerized approach. This helps to reduce time spent debugging applications and diagnosing differences in computing environments.

1

2

**Step 2**

## Tens of hosts with hundreds of containers



When running containerized applications, it’s important to consider scalability. Suppose that instead of a single host with multiple containers, you have to manage tens of hosts with hundreds of containers. Alternatively, you have to manage possibly hundreds of hosts with thousands of containers. At a large scale, imagine how much time it might take for you to monitor memory usage, security, logging, and so on.

1

2

## Summary

Container orchestration services help you to deploy, manage, and scale your containerized applications. Next, you will learn about two services that provide container orchestration: Amazon Elastic Container Service and Amazon Elastic Kubernetes Service.

1

2

**START AGAIN**

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

[**Amazon Elastic Container Service (Amazon ECS)**](https://aws.amazon.com/ecs/) is a highly scalable, high-performance container management system that enables you to run and scale containerized applications on AWS.

Amazon ECS supports Docker containers. [Docker](https://www.docker.com/) is a software platform that enables you to build, test, and deploy applications quickly. AWS supports the use of open-source Docker Community Edition and subscription-based Docker Enterprise Edition. With Amazon ECS, you can use API calls to launch and stop Docker-enabled applications.

**Amazon Elastic Kubernetes Service (Amazon EKS)**

[**Amazon Elastic Kubernetes Service (Amazon EKS)**](https://aws.amazon.com/eks/)is a fully managed service that you can use to run Kubernetes on AWS.

[Kubernetes](https://kubernetes.io/) is open-source software that enables you to deploy and manage containerized applications at scale. A large community of volunteers maintains Kubernetes, and AWS actively works together with the Kubernetes community. As new features and functionalities release for Kubernetes applications, you can easily apply these updates to your applications managed by Amazon EKS.

**AWS Fargate**

[**AWS Fargate**](https://aws.amazon.com/fargate/)is a serverless compute engine for containers. It works with both Amazon ECS and Amazon EKS.

When using AWS Fargate, you do not need to provision or manage servers. AWS Fargate manages your server infrastructure for you. You can focus more on innovating and developing your applications, and you pay only for the resources that are required to run your containers.