**Module 1 : Introduction**

**Learning objectives**

In this module, you will learn how to:

* Summarize the benefits of AWS.
* Describe differences between on-demand delivery and cloud deployments.
* Summarize the pay-as-you-go pricing model.

**Video Transcript**

Almost all modern computing centers around a basic client-server model. Now I know it can be more complicated than that, so let's take a look at our coffee shop.

This coffee shop is going to give us some real world metaphors to help you understand why AWS can change the way your IT operates.

Let's make Morgan the server, the barista. And I am the client, the customer. I make a request. In this case, it is for coffee. Now in the computing world, the request could be anything. It could be rain pattern analysis in South Africa, or the latest x-rays of your knee, or videos of kittens. Whatever is the business, basically a customer makes a request, and with permissions, the server responds to that request. All I want is a caffeinated beverage.  
  
Morgan represents the server part of the client-server model. In AWS, she would be called an Amazon Elastic Compute Cloud, or EC2, an EC2 instance, a virtual server. So from an architectural point of view, the transaction we did is really simple to explain. I, the user, made a request to Morgan, the server. Morgan validated that the request was legitimate, in this case, did I give her money? Then she returned a response, which in this case, is a berry blaster with extra caramel shots.

Now in the real world, applications can get more complicated than just a single transaction with a single server. In a business solution that is more mature, it can get beautifully complex.

To avoid this complexity, we're going to start simple. We will build this discussion out so that it is easy for anyone to understand how these concepts build on each other. So, by the end, those complex concepts, they'll be easy to understand. Let's start with a key concept to AWS, and that is, you only pay for what you use.

This principle makes sense when you run a coffee shop. Employees are only paid when they're in the store working. If Rudy and Morgan are off the clock, well then they don't get paid. The store owner simply decides how many baristas are needed and then just pays for the hours they work. For example, the coffee shop is about to release a new drink, the Pumpkin Monster Spice. In anticipation of this launch, you could always staff your shop with a dozen baristas all day long, just in case you suddenly get an unexpected rush at some point in the day. Only, let's be honest. For most of your day, you don't have near enough customers to justify paying for all those employees.

And yet, the is exactly what happens in an on-premises data center. You can't just snap your fingers and triple your capacity. At AWS, you don't pre-pay for anything. And you don't have to worry about capacity constraints.

When you need instances, or baristas, you just click a button, and you have them. And when you don't need them, another click, and they go away, and you stop paying for them. The same way you don't pay for employees for hours that they're not working.

So, pay for what you need, becomes the first key value of many for running your business on AWS. And that is really why we're here, to help you understand how AWS is built to help you run your business better.

**What is a client-server model?**

You just learned more about AWS and how almost all of modern computing uses a basic client-server model. Let’s recap what a client-server model is.



In computing, a**client** can be a web browser or desktop application that a person interacts with to make requests to computer servers. A **server** can be services such as Amazon Elastic Compute Cloud (Amazon EC2), a type of virtual server.

For example, suppose that a client makes a request for a news article, the score in an online game, or a funny video. The server evaluates the details of this request and fulfills it by returning the information to the client.

Cloud Computing:

Video Transcript

The data inside your database, now that's critically different. The way you build your tables and manage the structures, absolutely separates you from the competition. But the engine is just the engine.

**Deployment models for cloud computing**

When selecting a cloud strategy, a company must consider factors such as required cloud application components, preferred resource management tools, and any legacy IT infrastructure requirements.

The three cloud computing deployment models are cloud-based, on-premises, and hybrid.

**cloud-based deployment**

Select each tab to learn about each category.

* Run all parts of the application in the cloud.
* Migrate existing applications to the cloud.
* Design and build new applications in the cloud.

In a **cloud-based deployment** model, you can migrate existing applications to the cloud, or you can design and build new applications in the cloud. You can build those applications on low-level infrastructure that requires your IT staff to manage them. Alternatively, you can build them using higher-level services that reduce the management, architecting, and scaling requirements of the core infrastructure.

For example, a company might create an application consisting of virtual servers, databases, and networking components that are fully based in the cloud.

**On-premises deployment**

* Deploy resources by using virtualization and resource management tools.
* Increase resource utilization by using application management and virtualization technologies.

**On-premises deployment**is also known as a *private cloud* deployment. In this model, resources are deployed on premises by using virtualization and resource management tools.

For example, you might have applications that run on technology that is fully kept in your on-premises data center. Though this model is much like legacy IT infrastructure, its incorporation of application management and virtualization technologies helps to increase resource utilization.

**hybrid deployment**

* Connect cloud-based resources to on-premises infrastructure.
* Integrate cloud-based resources with legacy IT applications.

In a **hybrid deployment**, cloud-based resources are connected to on-premises infrastructure. You might want to use this approach in a number of situations. For example, you have legacy applications that are better maintained on premises, or government regulations require your business to keep certain records on premises.

For example, suppose that a company wants to use cloud services that can automate batch data processing and analytics. However, the company has several legacy applications that are more suitable on premises and will not be migrated to the cloud. With a hybrid deployment, the company would be able to keep the legacy applications on premises while benefiting from the data and analytics services that run in the cloud.