**Module 1**

I'm Blaine Sundrud, AWS Training and Certification. I've been teaching technology for more years than I'm willing to admit. After spending time teaching the newspaper industry, I moved to AWS, where I've taught classes globally on many different disciplines, such as security, cloud architecture, DevOps, big data, AI and ML, and theater history. My momma was a teacher. My daddy was a teacher. My grandpa was a bartender. I was born for this.

Hi, I'm Morgan Willis, a Senior Cloud Technologist at AWS. I started in the IT world about 10 years ago. And along the way, I decided that I was missing something. I missed the help and teaching aspect of IT that I had in my first job in IT support. So, I went into teaching software development in different areas around the U.S. And then I eventually landed here at AWS, where, as a Cloud Technologist, I get to support others in their cloud journey every day.

And I'm Rudy Chetty. I come from sunny Cape Town, South Africa, home of biltong, boerewors, and bunny chow. I'm a Solutions Architect, and have been with AWS for over three years. Teaching is my passion. And I can't wait for you to dive into the course, and learn. Thank a lot. and good luck.

This course is gonna cover all the essential information that you need to understand, to be comfortable discussing AWS, to know why it's beneficial to your business.

AWS offers a massive range of services for every business, starting with basic elements, like compute, storage, and network security tools, through complex solutions like blockchain, machine learning, or artificial intelligence, and robot development platforms, all the way through very specialized tool sets, like video production management systems, and orbital satellites you can rent by the minute.

All that, however, is way more than we have time to cover in a foundational class like this one. So let's simplify the conversation by starting with the fundamental cloud compute model.

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.

We hope you stick around for the entire course, as we dive deeper into these concepts, and help launch you on your journey to being a Cloud Practitioner.

Topic 1: Introduction

**Learning objectives**

In this module, you will learn how to:

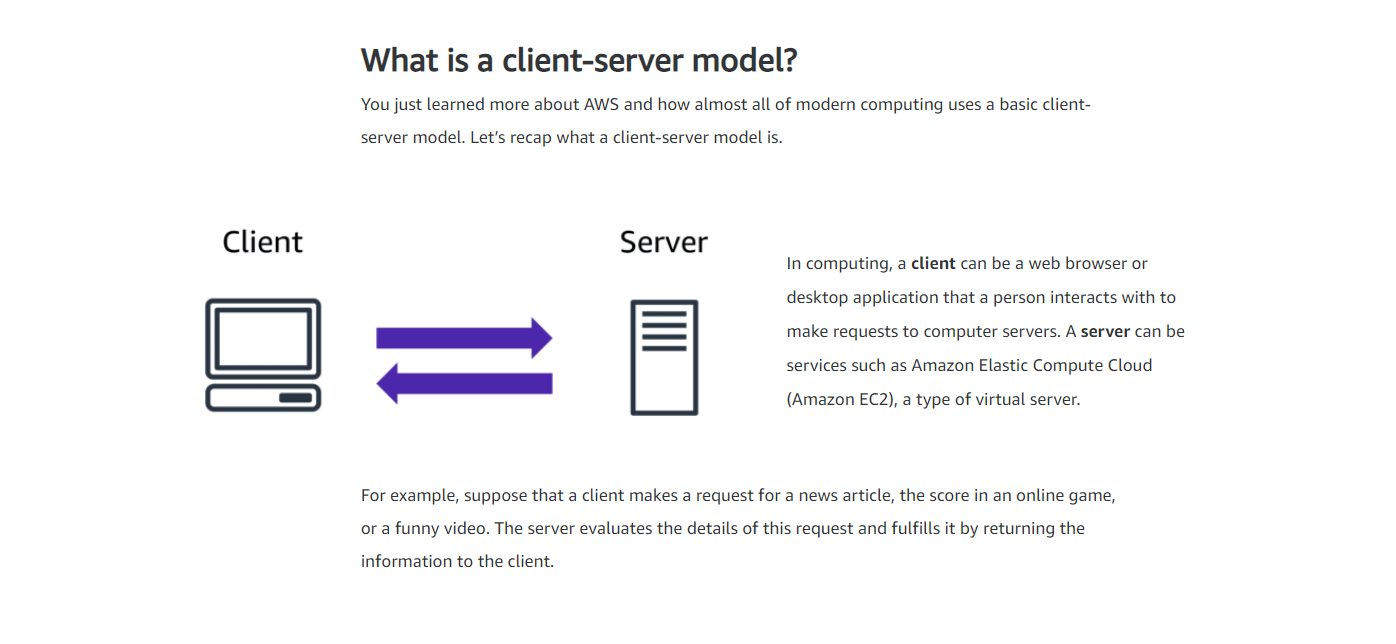
* Summarize the benefits of AWS.
* Describe differences between on-demand delivery and cloud deployments.
* Summarize the pay-as-you-go pricing model.

Service Offerings:

1. Compute
2. Storage
3. Network security
4. Blockchain
5. Machine learning
6. Artificial intelligence
7. Robot development
8. Video production
9. Orbital satellites

Amazon Elastic Compute Cloud AEC2 (Virtual Server)

Key Values: Pay for what you need



Topic 2: Cloud computing

Before we get deeper into the pieces and parts of AWS, let's zoom out and get a good working definition of cloud. Cloud computing is the on-demand delivery of IT resources over the internet with pay-as-you-go pricing. Let's break this down. On-demand delivery indicates that AWS has the resources you need, when you need them. You don't need to tell us in advance that you're going to need them. Suddenly you find yourself needing 300 virtual servers. Well, just a few clicks and launch them. Or you need 2000 terabytes of storage. You don't have to tell us in advance, just start using the storage you need, when you need it. Don't need them anymore, just as quickly, you can return them and stop paying immediately. That kind of flexibility is just not possible when you're managing your own data centers.

The idea of IT resources is actually a big part of the AWS philosophy. We often get asked why AWS has so many products and the answer is really simple: Because businesses need them. If there are IT elements that are common across a number of businesses, then this is not a differentiator.

Take a MySQL database as an example. If your business runs a MySQL database, does your ability to install the MySQL engine make you a better company than your competitors? Well, probably not that. Do you keep backups in a way that makes you superior to other players in your vertical? Again, doubtful. 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.

At AWS, we call that the undifferentiated heavy lifting of IT. Tasks that are common, often repetitive and ultimately time-consuming; these are the tasks AWS wants to help you with. So you can focus on what makes you unique. Over the internet, seems simple enough, but it implies that you can access those resources using a secure webpage console or programmatically.

No additional contracts or sales calls are needed. With pay-as-you-go pricing, we re-emphasize what we pointed out here in the coffee shop. You don't staff a shop with employees 24 hours a day at the same levels you do during peak hours. In fact, some hours, you might not even staff them at all. So why pay for developer environments, for example, on weekends, if your developers aren't working on the weekends?

Its on-demand delivery of IT resources over the internet with pay-as-you-go pricing

Undifferentiated heaving lifting of IT.. tasks that are common, repetive and ultimately time-consuming

**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.

Select each tab to learn about each category.

ONE: CLOUD-BASED DEPLOYMENT

* 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.

TWO: 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.

THREE: 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.

**Benefits of cloud computing**

* **Trade upfront expense for variable expense**

Upfront expense refers to data centers, physical servers, and other resources that you would need to invest in before using them. Variable expense means you only pay for computing resources you consume instead of investing heavily in data centers and servers before you know how you’re going to use them.

By taking a cloud computing approach that offers the benefit of variable expense, companies can implement innovative solutions while saving on costs.

* **Stop spending money to run and maintain data centers**

Computing in data centers often requires you to spend more money and time managing infrastructure and servers.

A benefit of cloud computing is the ability to focus less on these tasks and more on your applications and customers.

* **Stop guessing capacity**

With cloud computing, you don’t have to predict how much infrastructure capacity you will need before deploying an application.

For example, you can launch Amazon EC2 instances when needed, and pay only for the compute time you use. Instead of paying for unused resources or having to deal with limited capacity, you can access only the capacity that you need. You can also scale in or scale out in response to demand.

* **Benefit from massive economies of scale**

By using cloud computing, you can achieve a lower variable cost than you can get on your own.

Because usage from hundreds of thousands of customers can aggregate in the cloud, providers, such as AWS, can achieve higher economies of scale. The economy of scale translates into lower pay-as-you-go prices.

* **Increase speed and agility**

The flexibility of cloud computing makes it easier for you to develop and deploy applications.

This flexibility provides you with more time to experiment and innovate. When computing in data centers, it may take weeks to obtain new resources that you need. By comparison, cloud computing enables you to access new resources within minutes.

* **Go global in minutes**

The global footprint of the AWS Cloud enables you to deploy applications to customers around the world quickly, while providing them with low latency. This means that even if you are located in a different part of the world than your customers, customers are able to access your applications with minimal delays.

Later in this course, you will explore the AWS global infrastructure in greater detail. You will examine some of the services that you can use to deliver content to customers around the world.

Resources:

<https://docs.aws.amazon.com/general/latest/gr/glos-chap.html>

<https://d0.awsstatic.com/whitepapers/aws-overview.pdf>

<https://aws.amazon.com/getting-started/fundamentals-overview/>

<https://aws.amazon.com/what-is-cloud-computing/>

<https://aws.amazon.com/types-of-cloud-computing/>

<https://aws.amazon.com/what-is-aws/>

**Module 1**

In this video, we are going to talk at a high level about a service called Amazon Elastic Compute Cloud or EC2. If you remember from our coffee shop, the employees are a metaphor for the client/server model where a client sends a request to the server; the server does some work, and then sends a response. That example is for the coffee shop. But the same idea applies to other businesses. Your business, whether it be in healthcare, manufacturing, insurance, or delivering video content to millions of users all around the world, are also using this model to deliver products, resources, or data to your end users. And you're going to need servers to power your business and your applications. You need raw compute capacity to host your applications and provide the compute power that your business needs. When you're working with AWS, those servers are virtual. And the service you use to gain access to virtual servers is called EC2.

Using EC2 for compute is highly flexible, cost effective, and quick when you compare it to running your own servers on premises in a data center that you own. The time and money it takes to get up and running with on-premises resources is fairly high. When you own your own fleet of physical servers, you first have to do a bunch of research to see what type of servers you want to buy and how many you'll need. Then you purchase that hardware up front. You'll wait for multiple weeks or months for a vendor to deliver those servers to you. You then take them to a data center that you own or rent to install them, rack and stack them, and wire them all up. Then you make sure that they are secure and powered up and then they're ready to be used. Only then can you begin to host your applications on top of these servers. The worst part is, once you buy these servers you are stuck with them whether you use them or not.

With EC2, it's much easier to get started. AWS took care of the hard part for you already. AWS already built and secured the data centers. AWS has already bought the servers, racked and stacked them, and they are already online ready to be used. AWS is constantly operating a massive amount of compute capacity. And you can use whatever portion of that capacity when you need it. All you have to do is request the EC2 instances you want and they will launch and boot up, ready to be used within a few minutes. Once you're done, you can easily stop or terminate the EC2 instances. You're not locked in or stuck with servers that you don't need or want. Your usage of EC2 instances can vary greatly over time. And you only pay for what you use. Because with EC2, you only pay for running instances, not stopped or terminated instances.

EC2 runs on top of physical host machines managed by AWS using virtualization technology. When you spin up an EC2 instance, you aren't necessarily taking an entire host to yourself. Instead, you are sharing the host with multiple other instances, otherwise known as virtual machines. And a hypervisor running on the host machine is responsible for sharing the underlying physical resources between the virtual machines. This idea of sharing underlying hardware is called multitenancy. The hypervisor is responsible for coordinating this multitenancy and it is managed by AWS. The hypervisor is responsible for isolating the virtual machines from each other as they share resources from the host. This means EC2 instances are secure. Even though they may be sharing resources, one EC2 instance is not aware of any other EC2 instances also on that host. They are secure and separate from each other.

Luckily, this is not something you, yourself, need to set up. But it's important to know the idea of multitenancy and have a high level understanding of how this works. EC2 gives you a great deal of flexibility and control. Not only can you spin up new servers or take them offline at will, but you also have the flexibility and control over the configuration of those instances.

When you provision an EC2 instance, you can choose the operating system based on either Windows or Linux. You can provision thousands of EC2 instances on demand. With a blend of operating systems and configurations to power your business' different applications.

Beyond the OS, you also configure what software you want running on the instance. Whether it's your own internal business applications, simple web apps, or complex web apps, databases or third party software like enterprise software packages, you have complete control over what happens on that instance. EC2 instances are also resizable. You might start with a small instance, realize the application you are running is starting to max out that server, and then you can give that instance more memory and more CPU. Which is what we call vertically scaling an instance.

In essence, you can make instances bigger or smaller whenever you need to. You also control the networking aspect of EC2. So what type of requests make it to your server and if they are publicly or privately accessible is something you decide.

We will touch more on this later in the course in detail. Virtual machines are not a new thing. But the ease of provisioning EC2 instances allows for programmers and businesses to innovate more quickly. AWS has just made it much, much easier and more cost effective for you to acquire servers through this Compute as a Service model. There's a lot more to learn about EC2. We talked about virtualization and the types of software you can run on an EC2 instance. But there is more you can configure with EC2 as well.

**Learning objectives**

In this module, you will learn how to:

* Describe the benefits of Amazon EC2 at a basic level.
* Identify the different Amazon EC2 instance types.
* Differentiate between the various billing options for Amazon EC2.
* Summarize the benefits of Amazon EC2 Auto Scaling.
* Summarize the benefits of Elastic Load Balancing.
* Give an example of the uses for Elastic Load Balancing.
* Summarize the differences between Amazon Simple Notification Service (Amazon SNS) and Amazon Simple Queue Service (Amazon SQS).
* Summarize additional AWS compute options.

**Amazon Elastic Compute Cloud (Amazon EC2)**

**It’s a server to gain access to other virtual servers**

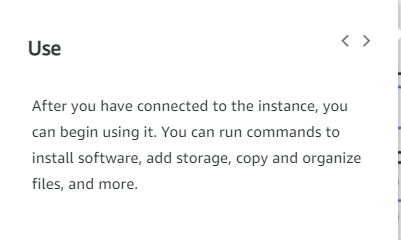
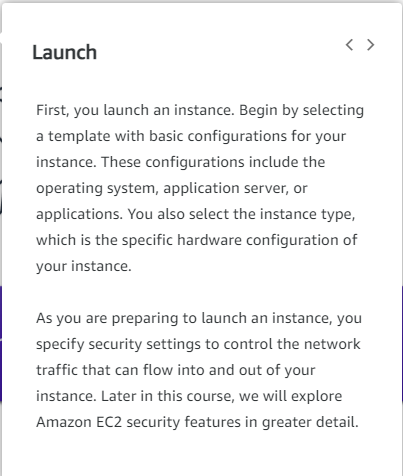
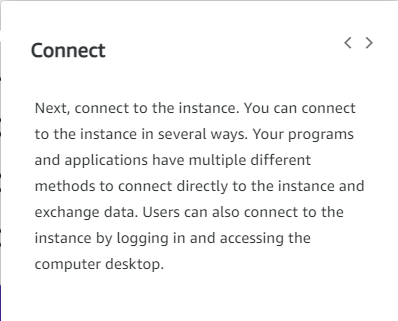
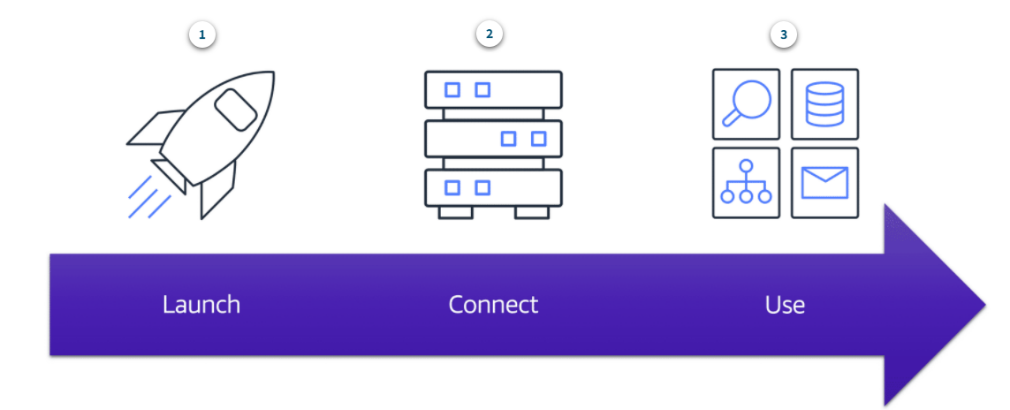
[Amazon Elastic Compute Cloud (Amazon EC2)](https://aws.amazon.com/ec2/) provides secure, resizable compute capacity in the cloud as Amazon EC2 instances.

Imagine you are responsible for the architecture of your company's resources and need to support new websites. With traditional on-premises resources, you have to do the following:

* Spend money upfront to purchase hardware.
* Wait for the servers to be delivered to you.
* Install the servers in your physical data center.
* Make all the necessary configurations.

By comparison, with an Amazon EC2 instance you can use a virtual server to run applications in the AWS Cloud.

* You can provision and launch an Amazon EC2 instance within minutes.
* You can stop using it when you have finished running a workload.
* You pay only for the compute time you use when an instance is running, not when it is stopped or terminated.
* You can save costs by paying only for server capacity that you need or want.



**Amazon EC2 instance types**

Now that we've learned about EC2 instances and the crucial role they play in AWS, let's talk about the different types of EC2 instances that are available. Thinking back to our coffee shop analogy, you'll remember that EC2 instances are like our employees and that they serve client requests. If we want to have a cafe that can serve a lot of customers, then we're probably going to need multiple employees, right? And they all can't just be cashiers. We also need someone to make the drinks, someone to handle the food, and maybe someone to do that cool latte art that our customers love so much. Like any business, there are a variety of tasks that need to be done, and they often require different skillsets.

If we want our business to operate as efficiently as possible, it's important to make sure that an employee's skillset suits their role. In the same way that our coffee shop has different kinds of employees, AWS has different types of EC2 instances that you can spin up and deploy into your AWS environment.

Each instance type is grouped under an instance family and are optimized for certain types of tasks. Instance types offer varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. The different instance families in EC2 are general purpose, compute optimized, memory optimized, accelerated computing, and storage optimized.

General purpose instances provide a good balance of compute, memory, and networking resources, and can be used for a variety of diverse workloads like web service or code repositories.

Compute optimized instances are ideal for compute-intensive tasks like gaming servers, high performance computing or HPC, and even scientific modeling.

Similarly, memory optimized instances are good for memory-intensive tasks. Accelerated computing are good for floating point number calculations, graphics processing, or data pattern matching, as they use hardware accelerators.

And finally, storage optimized are good for, can you guess it? Workloads that require high performance for locally stored data.

Now, if we map this back to our coffee shop, our cashier becomes a memory optimized EC2 instance, baristas become compute optimized instances, and our latte art employee is an accelerated computing instance type. And there you have it, EC2 instance types.

**General purpose instances**

**General purpose instances** provide a balance of compute, memory, and networking resources. You can use them for a variety of workloads, such as:

* application servers
* gaming servers
* backend servers for enterprise applications
* small and medium databases

Suppose that you have an application in which the resource needs for compute, memory, and networking are roughly equivalent. You might consider running it on a general purpose instance because the application does not require optimization in any single resource area.

**Compute optimized instances**

**Compute optimized instances** are ideal for compute-bound applications that benefit from high-performance processors. Like general purpose instances, you can use compute optimized instances for workloads such as web, application, and gaming servers.

However, the difference is compute optimized applications are ideal for high-performance web servers, compute-intensive applications servers, and dedicated gaming servers. You can also use compute optimized instances for batch processing workloads that require processing many transactions in a single group.

**Memory optimized instances**

**Memory optimized instances** are designed to deliver fast performance for workloads that process large datasets in memory. In computing, memory is a temporary storage area. It holds all the data and instructions that a central processing unit (CPU) needs to be able to complete actions. Before a computer program or application is able to run, it is loaded from storage into memory. This preloading process gives the CPU direct access to the computer program.

Suppose that you have a workload that requires large amounts of data to be preloaded before running an application. This scenario might be a high-performance database or a workload that involves performing real-time processing of a large amount of unstructured data. In these types of use cases, consider using a memory optimized instance. Memory optimized instances enable you to run workloads with high memory needs and receive great performance.

**Accelerated computing instances**

**Accelerated computing instances** use hardware accelerators, or coprocessors, to perform some functions more efficiently than is possible in software running on CPUs. Examples of these functions include floating-point number calculations, graphics processing, and data pattern matching.

In computing, a hardware accelerator is a component that can expedite data processing. Accelerated computing instances are ideal for workloads such as graphics applications, game streaming, and application streaming.

**Storage optimized instances**

**Storage optimized instances** are designed for workloads that require high, sequential read and write access to large datasets on local storage. Examples of workloads suitable for storage optimized instances include distributed file systems, data warehousing applications, and high-frequency online transaction processing (OLTP) systems.

In computing, the term input/output operations per second (IOPS) is a metric that measures the performance of a storage device. It indicates how many different input or output operations a device can perform in one second. Storage optimized instances are designed to deliver tens of thousands of low-latency, random IOPS to applications.

You can think of input operations as data put into a system, such as records entered into a database. An output operation is data generated by a server. An example of output might be the analytics performed on the records in a database. If you have an application that has a high IOPS requirement, a storage optimized instance can provide better performance over other instance types not optimized for this kind of use case.

**Amazon EC2 instance types**

We talked about EC2 instance types, but you're all probably wondering how much is this gonna cost me? Well, don't fret. For EC2, we have multiple billing options available.

The first one and the one that most people are familiar with is called On-Demand. What that means is that you only pay for the duration that your instance runs for. This can be per hour or per second, depending on the instance type and operating system you choose to run. Plus, no long-term commitments or upfront payments are needed. This type of pricing is usually for when you get started and want to spin up servers to test out workloads and play around. You don't need any prior contracts or communication with AWS to use On-Demand pricing. You can also use them to get a baseline for your average usage, which leads us to our next pricing option, Savings Plan.

Savings Plan offers low prices on EC2 usage in exchange for a commitment to a consistent amount of usage measured in dollars per hour for a one or three-year term. This flexible pricing model can therefore provide savings of up to 72% on your AWS compute usage. This can lower prices on your EC2 usage, regardless of instance family, size, OS, tenancy, or AWS region. This also applies to AWS Fargate and AWS Lambda usage, which are serverless compute options that we will cover later in this course.

Another option is Reserved Instances. These are suited for steady-state workloads or ones with predictable usage and offer you up to a 75% discount versus On-Demand pricing. You qualify for a discount once you commit to a one or three-year term and can pay for them with three payment options: all upfront, where you pay for them in full when you commit; partial upfront, where you pay for a portion when you commit; and no upfront, where you don't pay anything at the beginning.

The next option is Spot Instances, and they allow you to request spare Amazon EC2 computing capacity for up to 90% off of the On-Demand price. The catch here is that AWS can reclaim the instance at any time they need it, giving you a two-minute warning to finish up work and save state. You can always resume later if needed. So when choosing Spot Instances, make sure your workloads can tolerate being interrupted. A good example of those are batch workloads.

And finally, we have Dedicated Hosts, which are physical hosts dedicated for your use for EC2. These are usually for meeting certain compliance requirements and nobody else will share tenancy of that host.

**On-Demand**

–

**On-Demand Instances** are ideal for short-term, irregular workloads that cannot be interrupted. No upfront costs or minimum contracts apply. The instances run continuously until you stop them, and you pay for only the compute time you use.  
  
Sample use cases for On-Demand Instances include developing and testing applications and running applications that have unpredictable usage patterns. On-Demand Instances are not recommended for workloads that last a year or longer because these workloads can experience greater cost savings using Reserved Instances.

**Amazon EC2 Savings Plans**

–

AWS offers Savings Plans for several compute services, including Amazon EC2. **Amazon EC2 Savings Plans** enable you to reduce your compute costs by committing to a consistent amount of compute usage for a 1-year or 3-year term. This term commitment results in savings of up to 72% over On-Demand costs.

Any usage up to the commitment is charged at the discounted Savings Plan rate (for example, $10 an hour). Any usage beyond the commitment is charged at regular On-Demand rates.

Later in this course, you will review AWS Cost Explorer, a tool that enables you to visualize, understand, and manage your AWS costs and usage over time. If you are considering your options for Savings Plans, AWS Cost Explorer can analyze your Amazon EC2 usage over the past 7, 30, or 60 days. AWS Cost Explorer also provides customized recommendations for Savings Plans. These recommendations estimate how much you could save on your monthly Amazon EC2 costs, based on previous Amazon EC2 usage and the hourly commitment amount in a 1-year or 3-year Savings Plan.

**Reserved Instances**

–

**Reserved Instances** are a billing discount applied to the use of On-Demand Instances in your account. You can purchase Standard Reserved and Convertible Reserved Instances for a 1-year or 3-year term, and Scheduled Reserved Instances for a 1-year term. You realize greater cost savings with the 3-year option.

At the end of a Reserved Instance term, you can continue using the Amazon EC2 instance without interruption. However, you are charged On-Demand rates until you do one of the following:

* Terminate the instance.
* Purchase a new Reserved Instance that matches the instance attributes (instance type, Region, tenancy, and platform).

**Spot Instances**

–

**Spot Instances** are ideal for workloads with flexible start and end times, or that can withstand interruptions. Spot Instances use unused Amazon EC2 computing capacity and offer you cost savings at up to 90% off of On-Demand prices.  
  
Suppose that you have a background processing job that can start and stop as needed (such as the data processing job for a customer survey). You want to start and stop the processing job without affecting the overall operations of your business. If you make a Spot request and Amazon EC2 capacity is available, your Spot Instance launches. However, if you make a Spot request and Amazon EC2 capacity is unavailable, the request is not successful until capacity becomes available. The unavailable capacity might delay the launch of your background processing job.  
  
After you have launched a Spot Instance, if capacity is no longer available or demand for Spot Instances increases, your instance may be interrupted. This might not pose any issues for your background processing job. However, in the earlier example of developing and testing applications, you would most likely want to avoid unexpected interruptions. Therefore, choose a different EC2 instance type that is ideal for those tasks.

**Dedicated Hosts**

–

**Dedicated Hosts**are physical servers with Amazon EC2 instance capacity that is fully dedicated to your use.

You can use your existing per-socket, per-core, or per-VM software licenses to help maintain license compliance. You can purchase On-Demand Dedicated Hosts and Dedicated Hosts Reservations. Of all the Amazon EC2 options that were covered, Dedicated Hosts are the most expensive.

**Scaling Amazon EC2**

So we have a good idea now on the basics of EC2 and how it can help with any compute needs, like making coffee. Well, like metaphorically making coffee. The coffee represents the, whatever your instance is producing. The next thing we want to talk about is another major benefit of AWS, scalability and elasticity, or how capacity can grow and shrink, based on business needs.  
  
Here is the on-prem data center dilemma. If your business is like 99% of all businesses out in the world, your customer workloads vary over time: perhaps over a simple 24 hour period, or you might have seasons where you're busy, and weeks that are not in demand. If you're building out a data center, the question is, what is the right amount of hardware to purchase? If you buy for the average amount, the average usage, you won't be wasting money on average. But when the peak loads come in, you won't have the hardware to service the customers, especially during the critical moments to expect to be making all your results.

Now, if you buy for the top max load, you might have happy customers, but for most of the year, you'll have idle resources. Which means your average utilization is very low. And I've seen data centers with average utilization under 10%, just out of fear of missing peak demand. So how do you solve the problem on-premises? Well the truth is, you can't. And here's where AWS changes the conversation entirely.

What if you could provision your workload to exactly the demand, every hour, every day? Well, now you have happy customers, because they can always get the services they want. And you have a happy financial officer because they get the ROI your company needs.

And here's how it works. Morgan is behind the counter. She's taking orders and we have a decoupled system. Morgan isn't doing all of the work here, so we need somebody making the drinks. It looks like Rudy's up. Now, the first problem I wanna solve, is the idea that we plan for a disaster. There's a great quote by Werner Vogels that says, "Everything fails all the time, so plan for failure and nothing fails." Or in other words, we ask ourselves what would happen if we lost our order taking instance? Well, we'd be out of business until we'd get another person to work the line, sorry, another instance up and running.

So here is where AWS makes it really simple. Using the same programmatic method we used to create the original Morgan, we can create a second Morgan. So if one fails, we have another one already on the front line and taking orders. The customers never lose service. Well, don't forget about the backend. Let's make our processing instances redundant as well. So that gets our regular operating capacity. We now have a highly available system, with no single point of failure. And as long as the number of customers in line stays the same, we're good. But you know that's gonna change, right? So let's take a look at what's going to happen when we have a rush of customers, an increase in demand.