AWS

Introduction to AWS for Non-Engineers

Module 3 Core Services

Technology domain

There are four domains in the AWS certified Cloud Practitioner Exam. They are: Cloud Concepts, Security, Technology and Billing and Pricing. The four courses in the Introduction to AWS for Non-Engineers series, follow these four domains. This course that you're watching now covers the largest domain in the certification, the Technology Domain.
There are four parts of the Domain which are:
1. Define methods of deploying and operating in the AWS cloud,
2. Define the AWS global infrastructure,
3. Identify the core AWS services and
4. Identify resources for technology support.

With almost 200 services available on 24 categories as of winter 2020 and ever expanding, the technology section of the domain might seem extremely daunting. Heck, they even have a satellite as a service. However there's good news, most of the services will not be on the AWS Certified Cloud Practitioner Exam. There's a certain subset of services that are often considered core services in AWS. And while dozens of services are announced every year, these core services don't change much. Generally, you do not have to worry about having to acquaint yourself with the newest product launch announcement from AWS re:Invent conference in Las Vegas every December. Study the services we talk about in this course and at least be familiar with the names of the other services, and you should be good to go. You can find all the products AWS offers at aws.amazon.com.products.

Aside from understanding the fundamentals of the core AWS services, we will also review concepts that make up the AWS global infrastructure, like Availability Zones, Regions and Edge locations. We will also briefly go over different ways you can deploy and operate in the AWS Cloud, like utilizing the AWS Management Console, through AWS Command Line Interface or CLI, and Infrastructure as Code. All of these words may seem confusing at first but don't worry, we'll get them squared away. We'll provide study breaks and exam study tips so you can begin preparing for the AWS (mumbles) Cloud Practitioner Exam's Technology domain.

EC2

Your manager comes by and taps you on the shoulder. "Hey, we need to get a new server for an app the dev team is creating. Can you get it set up for them?" "Sure!" You say, enthusiastically. After all, you love researching and buying new computers. How hard could this be? You soon realize that there are a lot of moving parts in purchasing a server. You need to know how much storage it needs, how much memory the dev team requires, the operating system, the bells and whistles. Even once you manage to find a server that fits all the requirements posed by the dev team, you then have to get it through all the bureaucratic ladders up and down the command chain in your team, and the engineering team. And once you thought you were finally ready to get it purchased, you find out you have to push it through finance, too. And the finance department is not happy with the huge, unexpected cost of buying an expensive server. You spend a little while convincing the finance department to approve the cost, and you order it. By this point it's already been a month since your manager initially asked you to purchase this and the dev team is getting impatient. Now, you wait for it to be delivered in a week. But wait, you aren't done just because it's delivered. Now you have to set it up and make it ready for dev team to use. All in all, it would take you months to get that server up and running that the dev team wanted. And by the time you deliver it to them, they have one thing to say. "Oh, yeah, we've been going further into the development process, and turns out we need double the amount of storage that we initially requested." Back to the first step.

There's an easier way to do this instead of spending months of your time drowning in bureaucratic headaches. You can log into AWS and spin up a virtual server, called an instance, in seconds with the exact specifications that your dev team requested. And if their needs change, you can easily adjust the existing server or just spin up a new one. This service is called Amazon Elastic Compute Cloud, or EC2, and it's one of the most widely used services in the AWS. Amazon EC2 allows you to launch applications and servers when you need them without upfront financial commitments. It's integrated with many other AWS services, is reliable and secure, and allows you or your company to quickly and inexpensively spin up instances of virtual servers for all of your different needs.

Elastic Beanstalk

Do you find yourself doing more administration of infrastructure than coding, even though you are a programmer? Are you spending more time managing your platform, provisioning, and load balancing than developing? Do you wish you could worry about your code instead of whether or not your application stack can handle your app? Or wish you could provide this environment for your team's developers? AWS Elastic Beanstalk could be your solution, Elastic Beanstalk is an easy to use AWS service to help you deploy and scale web applications by simply uploading your code.

Elastic Beanstalk handles the deployment process, including the capacity provisioning, load balancing, auto scaling and application health monitoring. You can upload services developed using JAVA, .NET, PHP, Node.js, Python, Ruby, Go and Docker and you retain full control over the underlying resources at all times, it's free to use and you only pay for the AWS resources needed to store and run the web applications you've deployed. You never have to worry about outgrowing the resources Elastic Beanstalk provisioned for you because it automatically scales your applications up and down based on its specific needs. You also have complete freedom to select the AWS resources such as the EC2 Instance type that you want to use for your application. If you decide that you want to take over the manual management of the infrastructure you can do so at any time. Elastic Beanstalk provisions and operates the infrastructure for you so that you can focus on coding. If you're constantly frustrated by the amount of time you spend managing configuring servers, databases, firewalls and networks, perhaps its time you give AWS Elastic Beanstalk a try.

Elastic Load Balancing

Have you even experienced a situation where suddenly everyone around you at work seems to want something from you? And now you probably got completely overwhelmed and maybe found yourself unable to do any of the tasks requested much less all of them. When a server is overloaded with requests, it reacts similarly. It becomes unable to send out the responses to the paralyzing amount of traffic. It becomes completely overwhelmed and you might notice that the website is loading very slowly or goes down completely.

Let's think about fashion company A. Fashion company A decided to have a sale to celebrate their first anniversary. All shirts are 50% off. Their marketing department did an amazing job and everyone is talking about it on social media. As soon as the sale starts, boom, their site goes down. Uh oh, what happened? Chances are the infrastructure team wasn't prepared for the crushing amount of traffic that suddenly inundated the company website server which quickly became overloaded. Now the whole company is running around trying to get the website back up and communicate with angry customers. How could fashion company A have prepared better for the sale so that their website didn't go down from too many excited customers accessing it at the same time? They could have utilized elastic load balancing to automatically distribute incoming traffic across multiple servers. This means that the infrastructure department could have set up multiple replicated servers which means multiple servers with the same content on them and place the elastic load balancer in front of them to distribute the traffic to multiple servers. That way each server is only taking care of a fraction of the overall traffic. You can think of elastic load balancers like air traffic controllers telling incoming airplanes to go to different runways. They make sure that any one runway doesn't get overwhelmed with airplanes to cause delays by equally utilizing all available runways for landing. Elastic load balancers help your applications achieve fault tolerance by ensuring scalability, performance, and security. They can also monitor the health of your servers. If a server goes down, the load balancers can send the traffic to the remaining healthy servers. Elastic load balancers are highly available, secure, flexible, and monitorable allowing you to glean robust information about your traffic as well as providing you the confidence that your applications are up at all times.

Lambda

You've created a mobile application that allows people to create profiles of dogs and friend other dogs all over the internet. The first thing a new user does is upload their dog's photo, because what's a dog social media without dog photos, right? You want the dog photo's thumbnail to become available immediately after upload, so that the user can pick and choose which photos to feature on their profile. In order to make this happen, you need to provision and manage servers to run your code and make your app function. As more dog lovers sign up, you find yourself having to devote more and more time to maintaining and scaling your server infrastructure, instead of writing code to make your app better and adding new features. How can you focus on writing code for your dog social media instead of spending so much time on the infrastructure upkeep? You can check out AWS Lambda.

AWS Lambda runs code, called a Lambda function, in response to an event. An event could be anything from a user uploading an image into an S3 bucket to a user tapping a button on your mobile app to buy an item. All you have to do is upload your code, and AWS Lambda automatically runs your code and scales your application for you. You only pay for the time your code spends running, and each event trigger, which helps you keep your costs low. There are no servers to provision or manage, and nothing to scale, because Lambda takes care of all of that for you so you can focus on writing code. You can have a Lambda function up and running with just a few clicks, saving you hours and hours of back-end provisioning work. More dogs, less admin work.

Lightsail

Love the idea of having your website, database, or application running on AWS, but just don't have the energy or technical know-how to get it done? Need a quick development or test environment spun up? AWS has a service that'll do just the trick, Amazon Lightsail. Amazon Lightsail is perfect for simpler workloads, quick deployments, and getting started on AWS. It's a snap to get going, while still being designed to scale with you as you grow. You can use it to deploy simple web applications, create websites, run your business's software, or spin up developer or test environments, while maintaining cost-effective monthly fees. There are many preconfigured and ready-to-use operating systems, web apps, and development stacks.

Some of the more popular resources available are WordPress, Windows OS, Ubuntu, and Node.js. They are one-click-to-launch services, so getting started is a breeze. You can quickly deploy projects ranging from creating your first WordPress blog to running a database with just a few clicks. If you're considering using AWS to quickly spin up projects or resources, but don't have the time or engineering know-how to deploy full-on services, consider trying out Amazon Lightsail.

Deploying and operating in AWS

In this video, we're going to take a break from all the services, and discuss an important topic when it comes to creating and maintaining your IT resources on the AWS Cloud, deploying on to AWS Cloud. AWS users can manage and deploy resources since the AWS cloud in three ways, by utilizing AWS Management Console, Command-Line Interface, or CLI, and Software Development Kits, or SDKs. They all reference the AWS API's to help you deploy, and manage your AWS Cloud infrastructure.

The AWS Management Console is a graphical interface that supports the majority of AWS services. You can think of this as a web portal that you log into, as you would your social media account to see everything that's offered on that website. Through it you can look at your billing statements, launch new services, and find out how your apps are doing, all while using a interface that feels like you're browsing through another website. If you don't have much familiarity with utilizing command lines, or SDK's, the AWS Management Console is extremely user-friendly and easy to navigate. The AWS Command-Line Interface, or CLI, allows you to access services via the command line. The command line is a way to access and change resources with text-based command entry. You can tell an easy two service to shut down, or add new file to an S3 bucket, all by typing in some command into the AWS CLI. It is programming language agnostic, and allows you to create scripts to run on AWS.

The AWS Software Development Kits, or SDKs, let's you incorporate connectivity and functionality of a wide range of AWS cloud services into your code, helping you to deploy AWS services and resources using a variety of popular programming languages like NODEJS, C++, Java, Ruby, and PHP. SDK allows you to use AWS Cloud resources in existing applications. You can use just one, two, or all three ways of deployment to AWS Cloud. AWS CLI's and SDK's allow you to create tools that are specific to your organization, and help create an environment that utilize infrastructure as code. With infrastructure as code, you can write code that describes the configurations for specific AWS Cloud services, and they can be deployed for you by AWS. It helps to speed up the deployment process, and removes the risk of human error when spinning up new resources. Some AWS Cloud services that utilize infrastructure as code are Elastic Beanstalk, AWS Lambda, and AWS CloudFormation.

AWS Global Infrastructure

Amazon web services is global. Many companies and users around the world rely on AWS cloud to help their businesses succeed and grow. AWS has data centers around the world called Availability Zones. Each availability zone is independent of each other in network and power source. There are currently almost six dozen availability zones, or Azs, around the world. A region is made up of two or more availability zones, and there are currently two dozen AWS regions around the world. Some regions have more AWS cloud services than others. When a brand new service is introduced, it's generally first introduced in a few specific regions as opposed to the whole world. Some of the regions that receive new services earlier are US East North Virginia, US West North California, some Asia Pacific regions like Singapore, Sydney, and Tokyo, and some areas in the European Union like Frankfurt and Ireland. This fact may influence which region you decide to use to host your infrastructure.

Generally, you would choose a region closest to your physical location to host your AWS cloud infrastructure, because you can reduce network latency for your end users. For example if your company is based out of Washington D.C., you might pick the region US-East-1, which is based out of North Virginia in the United States. Some regions cost more than others. And service level agreements, or SLAs, also vary by region. You may also have compliance requirements to meet, which may require you to host your resources in specific regions or multiple regions. Hosting your resources in multiple availability zones, or even regions, help create what is known as high availability. The ability to provide uninterrupted performance even during natural disasters is called resiliency. Having multiple copies of your data in different data centers is called having redundancy. By architecting you AWS cloud infrastructure across multiple regions, you can prepare for events like power failures, natural disasters, and other potential operational mishaps to make sure that your web applications and resources don't experience extended down times when disaster strikes. Every data center, availability zone, and region is interconnected with highly available, low latency, private global network. This means that data transfers between data centers, availability zones, and regions are super fast, allowing customers to take advantage of the resources without lag.

Study break: Reviewing compute services

In this chapter, we went over five of the major compute services in AWS. Amazon Elastic Compute Cloud, or EC2, Amazon Elastic Beanstalk, Elastic Load Balancing, AWS Lambda and Amazon Lightsail. Let's quickly review all of them to make sure we've got the fundamental concepts down before moving on.

Amazon Elastic Compute Cloud, more commonly referred to as Amazon EC2, is a virtual server hosted on AWS cloud. You can instantly launch applications and servers wherever you want with an extremely versatile range of capabilities. It is one of the most widely used services in AWS and you can spin up an instance with no upfront financial commitments.

AWS Elastic Beanstalk helps you deploy and scale web applications by simply uploading your code. It handles the deployment process like capacity provisioning, load balancing, auto-scaling, and application health monitoring so you and your team can focus on coding. You can upload code in many of the popular programming languages like PHP, Node.js, and python. And you can retain full control over the underlying resources at all times. Better yet, it's free. You only pay for the AWS resources utilized as a result of deploying the code.

Elastic Load Balancing helps your application achieve fault tolerance by ensuring scalability, performance, and security. They can monitor the health of your servers, and if one goes down, it can reroute incoming web traffic to healthy servers. Elastic Load Balancers are highly available, secure, flexible, and monitorable which means you can feel confident that your applications are up at all times and even get insightful information about web traffic going to your applications.

AWS Lambda runs code called a lambda function in response to an event. An event could be someone uploading an image into your application, or someone visiting a specific web page. When an event happens, your code runs. Instead of having to spin up and maintain servers to take care of set events, lambda allows you to deploy the lambda function and only charges you for the time your code spends running in each event trigger. No servers to provision, manage, or scale. Lambda functions can be up and running with just a few clicks, saving you tons of time.

You can think about *Amazon Lightsail* as a EC2 on bumper bowling mode. AWS provides many preconfigured and ready to use operating systems, web applications, and development stacks to help you get your applications or websites up and running with minimal configurations. When you want to focus on launching and creating instead of configuring and managing your virtual servers, Amazon Lightsail might be the better option to you over Amazon EC2. It's a snap to spin up a virtual server, but the servers are still designed to scale with you with cost effective monthly fees. Some popular resources available to you are Word Press, Windows OS, Ubuntu, and Node.js. With a one click to launch service like Amazon Lightsail, there's no real excuse to not get your project up and running on the cloud.

Of the services we went over, AWS Elastic Beanstalk and AWS Lambda are what we consider infrastructure as code because they allow us to deploy resources to the cloud using code.

There are three ways to deploy to the AWS cloud.

1. The AWS Management Console,
2. AWS Command-Line Interface, or CLI, and
3. AWS Software Development Kits, or SDKs.

Finally, AWS has a global infrastructure.
They have independent data centers called Availability Zones.
Two or more Availability Zones make up a Region. Your AWS cloud infrastructure is generally hosted in a Region closest to your organization's physical location. By creating redundancy by replicating your resources in multiple Availability Zones or Regions, you can create an infrastructure that is resilient to natural disasters and highly available.

S3

I have my files, folders and videos backed up to the cloud everywhere, Dropbox, Box, Google Drive, you name it, I probably have some files stored there. Most of the time it's because I've run out of storage in one service, so I move to another, but this makes it so difficult to keep tabs on all of my files, and I could lose important documents if I can't get into my account or remember which account I stored them in. Many of us have had hazard file storage situations when it comes to backing up to the cloud. The storage service might go down, we might be wasting money on paying for storage space through a subscription service that we may not end up using. Also, when we upload the files to the cloud, it's difficult to hotlink to it, meaning you can't use an image or upload it onto a service like Dropbox or Google Drive and embed it into your webpage.

AWS has a solution to all of your static file storage called Amazon Simple Storage Service or Amazon S3. Amazon S3 is an object storage service, which means that you're storing each file as an entity called an object. It offers industry-leading data availability, security, performance and scalability. Scalability refers to the way you can scale your usage up or down with extreme flexibility and be charged only for what you use. It's designed for 99.999999999% or 11 nine's of durability, which means that there is almost no chance of the data become corrupted. You can upload files of all sizes to serve a wide variety of needs such as websites, mobile apps, backup and archiving, enterprise applications, IoT devices, and big data analytics, as three boasts easy-to-use management features to fine-tune access controls for your organization's specific compliance requirements.

There are many storage classes available with S3, which support different data access levels at corresponding pricing. You can even setup S3 lifecycle policies, which would automatically transfer files from one storage class to a cheaper one after a certain number of days. These options range from using S3 Standard class to store your frequently accessed files to using S3 Glacier Deep Archive to store backup data that's rarely accessed for very cheap rates. Whatever your needs for object-based files storage may be, ranging from using photographs for your website or spending as little money as possible to backup your organization's files, Amazon S3 would have an option that works for your budget and needs.

Elastic Block Store

- You've span up a virtual server on AWS, using amazon EC2 to run a database. But you notice that you are running out of space on your virtual machine. What should you do so that you can continue making your databases larger without impacting the virtual machine's performance? AWS has a solution for you called Amazon Elastic Block Store which allows you to add extra block stores to your EC2 instance and you don't even have to reboot your server.

Amazon Elastic Block Store or Amazon EBS behaves like raw , unformatted blocks devices, which can be mounted or attached to your EC2 instance to expand your server storage. You can add multiple volumes to the same instance, and you can use these volumes as file systems, or hard drives. You can dynamically change the configurations of a volume attached to an instance, which means you can change the settings and sizes with just a few clicks on the management consule. These volumes are automatically replicated within their availability zone, making them available and durable.

Many organizations use EBS to host their huge databases. There are different EBS stores types available to fit your needs and budgets as well as the option to encrypt the volumes for compliance. EBS provides persistence block storage volumes which means that they don't disappear when EC2s are rebooted. They also exist independently of the virtual servers they are mounted on to, and can therefore be moved around on to other EC2 instances. You can think of EBS volumes as external hard drives for your virtual servers, taking advantage of scalable, durable, and reliable storage options using Amazon EBS will make scaling your IT operations a breeze.

Snowball

Okay, you've made your case and management has approved the migration of the company's backups from onsite servers to AWS Cloud. Here we go! 10 years worth of data. Time to upload. Wait, how long will uploading 10 years worth of data even take through the Internet connection at your company? Sometimes the amount of data you want to upload to AWS Cloud is small enough that you can do it through your high-speed Internet connection. Other times, you realize that the amount it takes for you to upload gigabytes or terabytes of data would potentially take years on the Internet circuit you have. But, of course, AWS has already come up with a solution to make your life easier. AWS Snowball.

AWS Snowball is one of the very few hardware solutions from AWS, and it is a data migration tool. This means that AWS will physically ship you a Snowball to move your data onto, and then mail it back so you can migrate huge amounts of data. The amount of data you can move into AWS Cloud ranges from 50 terabytes with regular Snowball up to 100 petabytes with a Snowmobile, a 45-foot-long ruggedized shipping container pulled by a semi-trailer truck. In case you had any doubts, this is the Snowmobile, pulled by semi-trailer truck to haul your petabytes of data back to AWS Data Centers. The usage fee of a Snowball device is free for 10 days of onsite usage, with small extra onsite usage fees for every day you keep it beyond that. There is a service fee per job which ranges from $200 for a 50 terabyte snowball to $320 for an 80 terabyte snowball used in Asia-Pacific. The data transferred into AWS is stored in S3 and there is no transfer fee. However, you will be paying for S3 storage space as a storage fee, to host your transferred data.

The Snowmobile has larger costs associate with use as you will be chartering a huge truck with extremely large storage capacities. You can follow up with an AWS sales representative for an estimate for your particular need. Requesting and using AWS Snowball is simple. You make a request to have the hardware mailed to you using the AWS Management Console. When it arrives you attach the device to your local network, run the Snowball client on your machine, and select folders and files you want to encrypt and transfer onto the AWS Cloud. Once the transfers are completed you can mail it back to AWS and once received your files will be uploaded onto S3. 100 terabytes of data will take more than 100 days to transfer over a dedicated 100 megabits per second connection. For large amounts of data, the transfer up to the cloud could take months, even with a high speed Internet connection. With AWS Snowball the same transfer can be accomplished in less than a week using two Snowball devices, with a few days tacked on for shipping. If you're considering moving a lot of data onto AWS Cloud, take a peek at AWS Snowball to help you out.

Storage Gateway

So, this idea of using cloud computing to keep your costs down sounds like a great idea but your organization uses the data a lot so you also don't want to sacrifice latency or the length of time it takes for your resources to be accessible. Going up to the cloud to download a 500 megabyte file every time you want to make an edit sounds like a horrible idea when your whole company is sharing a single circuit. What should you do to maintain very low latency but still take advantage of the costs and time-saving benefits of cloud computing? AWS Storage Gateway may be the best of both worlds solution you're looking for. It connects your on-premises storage with AWS Cloud Storage, providing a hybrid storage solution for your IT infrastructure.

The service seamlessly integrates on-premises enterprise applications and corporate workflows with AWS's Cloud Storage Services through the use of a virtual machine installed onto an on-premises data centers host server. Basically, it creates a gate that connects your on-site users and devices to the resources stored in AWS Cloud with minimal latency.

AWS offers three types of storage solutions to fit your needs.

1. File based,
2. volume based, and
3. tape based.

Files backed up using the *File Gateway* are stored as objects in S3. There is a one-to-one representation of each file backed up to the cloud in S3 and the gateway asynchronously updates the objects in S3 as local files are updated. Local cache is maintained to provide low-latency access to recently access resources.

The *Volume Gateway* on the other hand, uploads files in blocks as opposed to single files. You can think of the Volume Gateway as backing resources up as a virtual hard disk instead of individual files. These blocks can be asynchronously backed up as point-in-time snapshots and stored as Elastic Block Store or EBS snapshots. There are two types of Volume Gateways available:

1. stored volume, and
2. cached volume.

The major difference is where the complete copy of your data is stored. Stored volume keeps the complete copy on-premises while sending snapshots or incremental backups to AWS. Cached volume keeps only the most recently accessed data on-premises and keeps the complete copy on the cloud.

The last type of storage gateway is the *Tape Gateway* which utilizes virtual tapes. You can use your existing tape based backup infrastructure to backup data onto virtual tapes on S3. You can think about Tape Gateway as taking backups on physical tapes except instead of physical tapes they are digital tape cartilages stored on S3. Data is stored locally then asynchronously uploaded to S3. The data can then be archived using Amazon Glacier which is like sending your physical tape backups to an off-site tape holding facility like Iron Mountain. You pay for storage and data retrieval. The quicker you can access the backed up data the more expensive the solution becomes. For example, data stored via Tape Gateway is much cheaper saved to S3 Glacier Deep Archive than S3 Glacier because the data retrieval takes a longer period of time.

Depending on where you want to store the complete copy of your data and how you would like to backup your data there are multiple options available for utilizing AWS Cloud as a backup and storage resource for your frequently accessed data through AWS Storage Gateway.

Study break: Reviewing storage services

In this chapter, we run over four of the major storage services in AWS, Amazon Simple Storage Service or S3, Amazon Elastic Block Store or EBS, AWS Snowball, and AWS Storage Gateway.
Let's quickly review all of them to make sure we've got the fundamental concepts down before moving on.

Amazon Simple Storage, more commonly known as Amazon S3, is an object storage service, which you can conceptualize like storing each file as an individual object, like you would in your My Documents folder. It's designed for scalability, data availability, security, and performance, and is used in industries of all sizes. There are many storage classes available to fit every organizations budget and needs. You can even set up S3 lifecycle policies which will automatically transfer files from one storage class to a cheaper one after a certain number of days. You can use S3 for a variety of needs whether it's for hosting images your users upload to your web app as your as inexpensive backup solutions.

In contrast, *Amazon Elastic Block Store or EBS* is a block storage service while S3 stores files individually as objects, Amazon EBS stores them as blocks. They're not a 100% accurate representation, you can think of it like making a zip file of your Christmas holiday photos and sharing that zip file with a friend. They can't just download a portion of the zip file, they have to download the whole thing to unzip it and see your individual photos. Amazon EBS behaves like raw, unformatted block devices, which can be attached to your EC2 instances to expand your server storage. Think about an external hard drive that helps you up your laptop storage capacity. Amazon EBS is scalable, durable, and reliable storage option to make sure you always have enough storage available for all of your applications and servers.

AWS Snowball is one of the very few hardware services AWS offers. It's a data migration tool that can also function as a storage service. When you begin using AWS Snowball, AWS will actually ship you a physical Snowball to move your data onto. Once you finish moving the data onto Snowball, you mail it back and AWS will migrate the data onto Amazon S3 for you. The amount of data you can transfer to AWS cloud at one time using this service ranges from 50 gigabytes with a regular Snowball to up to 100 petabytes with a Snowmobile, a 45 foot long shipping container pulled by a semi-trailer truck. Why bother with a physical device? Because transferring such large amounts of data over the internet will take a lot of time. By moving the data to a physical device and then shipping it to AWS to upload to S3 on their end, you can save a lot of time, bandwidth and even money.

AWS Storage Gateway is a hybrid storage solution for your IT infrastructure providing both low latency for file access and benefit of cost and time saving with cloud computing. It is a gate that connects your onsite users and devices to resources stored in the AWS cloud with minimal latency, and offers three types of storage solutions to fit your needs, file gateway, tape gateway, and volume gateway, all addressing different kinds of needs. In the most fundamental sense, the difference in the three gateways is where you want to keep the complete copy of your data, onsite or on the cloud. There was a lot to unpack with the storage services. They all store things for you in the cloud, but they all have different uses in different ways of getting your data there. If any of them seem like they need a little bit more reviewing, feel free to go back and review the videos.

DynamoDB

Your team has created a mobile application and plans on releasing it next week. And a prominent TEC Publication has picked up the story. The publicity has done wonders and the application has 100 times the number of anticipated downloads, that's fabulous. But what's not fabulous is that your back and infrastructure is not prepared for the amount of data that's going to be sent back and forth, and your database solution can't scale up quickly enough to meet the new demands, while maintaining the efficiency your audience is looking for. Thankfully, AWS has a low maintenance solution for your database that scales up and down with your needs.

Do you need a secure scalable, fast and flexible NoSQL database?
Amazon DynamoDB is a one stop solution for enterprise ready database needs that can help you build applications with virtually unlimited throughput and storage.

This means that no matter how small or large your project is, DynamoDB will adapt with your needs so that your application speed and stability remains high. DynamoDB is serverless which means that you don't have to provision, patch or manage any servers. AWS automatically scales your tables up and down to adjust for capacity and to maintain performance as well as maintaining stability with redundancies and fault tolerance. You can focus on getting rid of that pesky bug from your new Superstar mobile application, instead of provisioning and maintaining the databases. You have the option of paying only for what you use, or specifying a workload amount and paying for the provision resource depending on your organization's preference.

DynamoDB is being utilized in many well known companies like Nikee, Snapchat Lyft and Netflix. your organization's application may benefit from utilizing AWS's scalable NoSQL database service.

RDS

You just learnt about Amazon solution to a non-relational database: Amazon DynamoDB. But what if your application requires a relational database, and an efficient yet affordable one at that. As you'd imagine, AWS has a solution for that as well, and it's called Amazon Relational Database Service, or Amazon RDS. Don't you love it when the name of the service tells you exactly what it's meant for?

With Amazon RDS, you can set up, operate, and scale a relational database with just a few clicks. It's cost-efficient, and you pay only for the resources you actually consume. You also have the option of on-demand pricing or even lower hourly rates with Reserved Instance pricing. Because you don't need to be responsible for provisioning, monitoring or maintaining the database, you can focus on building your application secure in the knowledge that AWS make sure your databases are highly available, durable, scalable, fast and secure.

Best of all, while AWS does provide its own database engine called Amazon Aurora as an option, you can utilize very familiar database engines like PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server. So you can dive right into creating a database without the learning curve. If you already have existing databases, you can use AWS Database Migration Service to migrate or replicate them into Amazon RDS as well. In the next video, let's check out AWS' own relational database engine, Amazon Aurora, which is fully managed by Amazon RDS.

Aurora

Amazon Aurora is one of the six relational database engines that you can use in Amazon Relational Database Service, or RDS. It's fully managed by Amazon RDS, which relieves engineers of having to engage in time consuming administrative work, like provisioning, database setup, and maintenance. Instead, it allows the engineers to focus on coding.

You can monitor performance using various AWS monitoring and alerting services so you can quickly detect performance issues. It's MySQL and PostgreSQL compatible, but up to five times faster than standard MySQL and three times faster than standard PostgreSQL databases. You can get the same security, availability, and reliability of commercial databases for just 1/10 of the cost. You can have databases scaling all the way up to 64 terabytes per instance, hosted on distributed, fault-tolerant, self-healing storage systems with low latency. You can migrate existing MySQL or PostgreSQL databases onto AWS using AWS Database Migration Service. Inexpensive, highly scalable, available, durable, and secure, Amazon Aurora is a great relational database service to look into.

Redshift

Got lots of data? Petabytes of data? Looking for a data warehouse to efficiently and affordably analyze it all? Well, of course AWS has a service to help you manage all that data, and it's called Amazon Redshift.

Data warehouses are large stores of data accumulated from a wide range of sources within a company, which is then used to guide management decisions after analysis. Amazon Redshift is a cloud-based, fully managed, petabyte-scale data warehouse service. Always striving to improve, Redshift in 2019 is 10 times faster than it was just two years ago. You only pay for what you use, and it starts as low as 25 cents per hour, all the way up to petabytes for under $1,000 per terabyte per year. Redshift is fully integrated with your data lakes, which are repositories of data stored in their raw formats. You can deploy a new data warehouse in minutes, and it's easily scalable with the click of a button. Security is built in too, so you can secure your whole database with the click of a mouse. Data encryption is also simple, and Redshift's encryption is compliant with many common requirements like HIPAA. If you're looking for a data warehouse solution that's cheap, secure, and easy to deploy and manage, Amazon Redshift might be a good place to start.

Study break: Reviewing database services

In this chapter, we went over four of the major database services in AWS. Amazon DynamoDB, Amazon Relational Database Service, or RDS. Amazon Aurora, and Amazon Redshift. Let's quickly review all of them to make sure we've got the fundamental concepts down before moving on.

Amazon DynamoDB is a fast, flexible, fully managed and secure non relational, or NoSQL. Database, that can handle more than 10 trillion requests per day, and support peaks of more than 20 million requests per second. It's serverless, so you don't have to provision, patch or manage any servers, and it automatically scales up or down, to adjust for capacity. Instead of worrying about managing your database, you can just worry about scaling your application.

Amazon relational database service, or Amazon RDS is a fully managed relational database, because it's fully managed, like Amazon DynamoDb. You don't have to provision, or manage any servers. Instead of spending your time doing administrative tasks, you can devote your time to working on your products.

You have six database engines to choose from:

1. Amazon Aurora,
2. PostgreSQL,
3. MySQL,
4. MariaDB,
5. Oracle database, and
6. SQL Server.

We just mentioned *Amazon Aurora*, as one of the database engines you can use with Amazon RDS. Amazon Aurora is fully managed by Amazon RDS, and it's MySQL and PostgreSQL compatible. You can get the same security, availability and reliability of commercial databases, but for faster and cheaper. *Amazon Redshift* is a cloud based, fully managed, petabyte-scale data warehouse service, that's faster and cheaper than other data warehouse providers. Data warehouses store extremely large amount of data, collected from a wide range of sources to analyze. It is quick to set up, and easy to scale, and its encryption, is compliant with many industry regulations.

Databases, databases. You may not be very familiar with databases, which could make trying to decipher these options a little more difficult. *Amazon DynamoDB,* is a **non** relational or NoSQL database. Whereas Amazon RDS, and Amazon Aurora, are relational databases. Amazon Redshift, is a data warehouse for lots and lots of data. All are scalable, secure, and less expensive than industry alternatives. If you want to review any of the databases, feel free to take a few minutes, going over to videos.

VPC

Think about your wi-fi setup at home. That's a private network. You likely have a cable that runs into your house from the street, which connects your home's private network to your ISP, like Comcast or Verizon. That cable is connected to your modem, which is your connection to the internet, also thought of as a gateway. The modem is connected to a router, or a switch, via another cable which routes traffic between devices in the network, and also the internet. You connect your devices, like your laptop and tablet, to the router, using the wireless network. Your home wi-fi setup is a private network where you can create your own ecosystem for connecting devices and resources. And a private network in the cloud is what AWS calls Amazon Virtual Private Cloud, more commonly referred to as Amazon VPC.

Amazon VPC creates a logically isolated section in the cloud where you can provision your AWS resources. Think of it like your corner of the cloud, where you define what comes in, what goes out, and what lives inside. Amazon VPC is very flexible and secure, allowing you to control almost every aspect of your virtual network. It's completely scalable, allowing you to instantly scale your resources up or down. It also boasts advanced security features, like security groups and network access control lists to help you filter inbound and outbound traffic at the instance level and sub-net level.

When you sign up for an AWS cloud account, you automatically get a VPC provision to you, along with automatically configured sub-nets, IP ranges, route tables, and security groups to help you get started. Going back to the analogy of your home network, the virtual private cloud is your home network. The modem is the internet gateway. The router is the route table. And your network's firewall is the network access control list. Your laptops and tablets are resources like your EC2 instances that are launched inside your VPC, or private network. When you create your first AWS cloud account, you will be creating a logically isolated corner in this vast realm of the cloud, where you are free to create and scale resources for your organization.

CloudFront

Faster, faster, faster. That seems to be the trend with everything these days. Back in the days, we were happy just to have objects we ordered online reliably delivered to us. But then Amazon's two-day shipping happened and suddenly waiting a week for your shoes to arrive became almost intolerable. In some cities, Amazon even has one-hour shipping for that emergency toilet paper you just can't seem to leave the house to buy.

In the digital space, we went from having to buffer a 10-minute YouTube video to being able to watch a whole 4K movie with no apparent lag on Netflix. Gone are the days when you had to wait for images to load slice by slice or a whole webpage to load a few rows of text at a time. We want things fast, digital or physical, and we want them now.

On the Internet, content delivery networks or CDNs are working behind the scenes to deliver your content faster and faster. Amazon's global CDN service that securely delivers data, applications, and APIs is called Amazon CloudFront.

CloudFront seamlessly integrates with many AWS services to provide optimal performance and security, including AW Shield for DDOS mitigation and Amazon EC2 as origins for your applications. A CDN is a system of distributed servers around the world that delivers website and application content to end users based on a few factors. These factors are location of the user, origin of the website or application, and the location of the content delivery server. The main purpose of CDNs is to make loading websites and applications for end users faster. And Amazon CloudFront does this by using edge locations to cache files and resources for quicker retrieval.

Imagine your favorite fruit. Most of us go to the grocery store to pick up an apple or a watermelon. Not very many of us live close enough to a farm to directly buy a fruit from a farmer. Instead, there are distribution networks set up between us and the farmer, which simplified bring the fruit to a grocery store nearby. All we have to do is pick up the fruit from the local grocery store where a truck traveled for days to bring the fruits from the farm. We have the convenience of driving 15 minutes as opposed to hours or maybe even days to buy some fruit. The farm is the origin, which on AWS could be S3 bucket, EC2 instance, or Elastic load balancer amongst a few other services. The truck then takes the apples to a grocery store where they are left to be sold to consumers or, in web terms, cached.

In AWS, files and data are cached at edge locations. Once the data is downloaded to an edge location it stays there for a certain period of time at which point users near the data center can retrieve the webpages or application resources from the edge location close to their location rather than having to go all the way to the origin which could even be on a different continent. This allows for data to be retrieved faster with the best possible performance because users are not going all the way back to the origin server to download the resources but rather accessing a location close to themselves. CloudFront is scalable, allowing you to start small and scale up as traffic to your application or webpage increases. It automatically manages traffic load without any intervention from you and utilizes application acceleration and optimization. There is no minimum commitment or a fixed-term contract and you only pay for content delivered using the service. Amazon CloudFront acts as a supermarket in a busy city making cached data quickly accessible to users around the world using edge locations.

Route 53

If you've ever set up a website with yourpersonaldomain.com, you probably used a domain name registrar to purchase and set up your domain. You might have used something like GoDaddy or Namecheap, to name a few popular commercial domain name registrars. AWS, of course, has its own service where you can purchase and set up domain names, but it can do so much more. It's called Amazon Route 53, and it's a highly scalable cloud Domain Name System, or DNS.

It allows you to reliably and cost-effectively route your end users to your internet applications. It can connect user requests to infrastructure running on AWS, like an EC2 instance or an S3 bucket. It can also route users to infrastructure outside of AWS, acting as a DNS service for domains purchased at other domain name registrars.

Route 53 is designed to be integrated with other AWS services, like mapping your domain names to your EC2 instance or S3 bucket. It's simple to set up, fast, secure, and cost effective. You are charged only for what you use, without any upfront fees or minimum usage commitments. It's also designed to automatically scale to handle large query volumes. Route 53's basic functions are domain registration, domain name system, or DNS, service, health checking of web application accessibility, and auto naming for service discovery. Utilizing the more robust features of Route 53 allows you to create websites and applications with high availability by automatically rerouting traffic catered to demand and integrating with services that send alerts when down times occur.

One of the ways you could use Route 53 is to purchase yourpersonaldomain.com, then to route users who visit yourpersonaldomain.com to a static website, hosted on your S3 bucket. When your visitors type in yourpersonaldomain.com into a browser, they will be able to load your static website, because Route 53 does the legwork of routing the user to the resources you identified. You can think of Route 53 like a telephone operator. Back in the days, you would call the telephone operator so you could speak to your grandmother. When the telephone operator got your request, he or she would use a switchboard to connect you to your grandmother. Route 53 works in similar ways, except with routing internet traffic.

Study break: Network and content delivery

In this chapter, we went over three of the major network and content delivery services in AWS.

1. Amazon Virtual Private Cloud or VPC,
2. Amazon CloudFront, and
3. Amazon Route 53.

Let's quickly review all of them to make sure we've got the fundamental concepts down, before moving on.

Amazon Virtual Private Cloud or VPC is an isolated corner of AWS Cloud made just for you. You can provision your AWS resources into a virtual network that you define with complete control over your virtual networking environment. From IP address range, to configurational route tables, and network gateways. It's free, and it's automatically created for you when you create your AWS account. Inside your very own Amazon VPC, you can create and scale your AWS Cloud resources to your heart's content.

Amazon CloudFront is a content delivery network or CDN. The main purpose of CDNs is to make websites and applications load faster. Amazon CloudFront achieves this by using Edge Locations all around the world to cache files and resources for quicker retrieval. By caching, say a video, at an Edge Location in Sydney Australia, someone who lives in Australia can stream the video much quicker than if there was no content delivery networks. Because they would have to download the video all the way from the content origin, which could be anywhere in the world. Amazon CloudFront sees where you're based and routes your traffic to the closest cache location. So you can enjoy the content without having to wait. It's scalable, and you only pay for content delivered using the service.

Amazon Route 53 sounds like a highway, and in a sense, it is kind of like a highway. If highways help take you from here to there, it is a highly scalable domain name system or DNS. It allows you to route your users to your internal applications. This could be in form of your users accessing infrastructure running on AWS, like an EC2 instance. It's basic functions are: domain registration, Domain Name System or DNS, health checking of web applications' accessibility, and auto-naming for service discovery. It helps route your users to the appropriate resources you want them to access. AWS's networking content delivery services mainly occupy themselves with helping you create secure networks for your resources to live within, and to route traffic to proper services. If any of these services we just talked about needs a bit of review, feel free to go back and check out the videos again.

CloudFormation

You've built an awesome system in AWS using many different services and many different settings. Together, they work great, and you'd like to replicate the setup for a new project. Unfortunately, setting it up take days and you don't exactly remember every step you took. Worse yet, some resources have to be provisioned before others for the system to work. We love recipes when we cook because they tell us what materials to buy, and when to do what for that perfect meal. In tech, we also love recipes. Better yet, we love recipes that cook themselves and present us with the finished dish.

That's what Amazon CloudFormation does for your IT infrastructure hosted on the cloud. You create templates, like recipes for your resources to be set up a certain way. And you can run it over and over to provision and deploy fully configured infrastructure. Best yet, unlike all those cookbooks filled with your favorite recipes in the bookstores, using CloudFormation is free. You just pay for the resources you use when you run the service, like the EC2 instances or S3 bucket storage.

With CloudFormation you can provision anything ranging from a simple EC2 instance, to a multiregion multitier application quickly and efficiently, using a simple text file written in JSON or YAML. You can update or manage the templates, referred to a stacks at any point using the AWS Management Console, command line or Software Development Kit, commonly known as SDK. Basically, you can change up the recipe whenever you see fit, even making different versions for different uses.

Version control is always available so you can revert back to previous settings whenever you want. AWS CloudFormation brings to life what is known as Infrastructure as Code where you can deploy IT infrastructure based on a text file filled with code, that specifies resources and configurations you need for each service you want to deploy. With CloudFormation, you can bring orderly and predictable back into resource deployment, no longer leaving things up to human error or chance.

CloudTrail

Your AWS IT infrastructure, like any IT infrastructure, needs to be monitored and audited, to make sure that the resources remain compliant with any government, industry, or company policies. In addition to compliance, the AWS CloudTrail service, helps to track user activity and API usage. Which allows for operational and risk auditing of your AWS infrastructure.

With CloudTrail, you can log and monitor account activities, provide event history of account activities, simplify compliance audits, discover and troubleshoot security and operational issues. Provide visibility into user and resource activities, and track and automatically respond to security threats within your AWS infrastructure. For example, you can utilize CloudTrail to automatically respond to security vulnerabilities. You can create a workflow to add a specific policy to an S3 bucket, when CloudTrail finds API call that made the bucket public. You track many account activities, including actions taken through the AWS Management Console, AWS SDKs, and command-line tools.

You can review logs using CloudTrail event history. Have the reports delivered to S3 buckets or send reports to CloudWatch logs and events for more granular monitoring of AWS resources. You can view, filter, and download account activities for the most recent 90 days for free. You can also, set up a trail that delivers a copy of management events in every region free of charge. However, the data is sent to S3 so you will be charged for storage usage. Data events, which are operations performed on, or within the resources itself also have very small charges. AWS CloudTrail is an invaluable resource in simplifying events security analysis and troubleshooting for your AWS cloud IT infrastructure.

CloudWatch

But now you need to actively monitor it and collect metrics and react to any events. Unfortunately, you can't be up 24/7 monitoring and neither can your team. Thankfully, Amazon CloudWatch is a monitoring and management system built for developers, system administrators, site-reliability engineers, and IT managers.

Natively integrated with over 70 AWS services, CloudWatch helps you gain system-wide visibility into resource utilization, application performance and operational health. It collects monitoring and operational data as logs, metrics and events to provide insight into your application performances. You can collect and track metrics in real time or have it send off notifications when an event occurs. You can even set up CloudWatch alarms to automatically make changes using pre-defined triggers so you don't have to lift a finger to fix common issues. CloudWatch employs a pay-as-you-go model so you only pay for what you use with no up-front commitment. Keep tabs on your applications hosted on AWS Cloud with Amazon CloudWatch so you and your teammates can get a good night sleep instead of holding round-the-clock monitoring vigils.

Study break: Reviewing management tools

In this chapter, we went over three of the major management tools in AWS:

- AWS CloudFormation,
- AWS CloudTrail, and
- Amazon CloudWatch.

Let's quickly review all of them to make sure we've got the fundamental concepts down before moving on.

AWS CloudFormation allows you to create a recipe for spinning up identical setups for a collection of resources and services for your IT infrastructure. It's free to use and you only pay for the resources you utilize by building a project on CloudFormation. It utilizes infrastructure as code, and you can deploy IT infrastructure based on a text file filled with code that specifies configurations for all of your services and resources. Once that's created, CloudFormation does the actual configurations and deployment for you. You can continue to build out your resources without having to worry about human error and configurations.

AWS CloudTrail can log monitor account activities, provide event history of account activities, simplify compliance audits, discover and troubleshoot security and operational issues, provide visibility into user and resource activities and track and automatically respond to security threats within your AWS infrastructure. And in that show, it's an event tracker and security analysis tool that helps keep your AWS cloud infrastructure compliant and secure.

Amazon CloudWatch helps you gain system wide visibility into resource utilization, application performance and operational health. It collects monitoring and operational data as logs, metrics and events and provides insight into your application performance. You can even set up CloudWatch alarms to automatically make changes using predefined triggers to automatically solve common issues. It's integrated with nearly 70 AWS services helping your team keep comprehensive monitoring data 24 seven.

Now you might be thinking CloudTrail, CloudWatch, what's the difference?

- AWS cloudTrail audits logs.
- Amazon CloudWatch monitors and can react to changes.
- Need access logs because someone did something they shouldn't have, CloudTrail.
- Need to know how much CPU on EC2 instance is using, CloudWatch.

Imagine a detective trailing a trail of footprints for CloudTrail. CloudWatch is watching or monitoring to make sure your resources are functioning as they should be. AWS's management tools help you build and manage your AWS cloud infrastructure. If any of the services we talked about needs a bit of a review, feel free to go check out the videos again.

Study break: Exam tips and resources

Let's quickly go over the services and concepts we'll be going over in this video. We'll be reviewing compute, storage, database, network and content delivery and management tools and services. And we'll also be reviewing concepts like infrastructure as code, deploying on the AWS cloud and availability zones and regions. As mentioned, we covered a lot of services in this course and many of their names are pretty confusing. I'm going to help you create a study cheat sheet so you can refer to it as you study for the exam.

Having a couple words that describe each service will make it easier for you to jog your memory when you're a little unsure about what the specific service does. Here we go.

Let's begin with compute services.

- Amazon EC2 or elastic compute cloud is a virtual server.
- Amazon Elastic Beanstalk helps you automatically grow your applications to meet demands, like Jack's beanstalk growing and growing.
- Elastic Load Balancing balances incoming traffic loads.
- AWS Lambda allows you to run serverless code.
- AWS Lightsail provides preconfigured virtual servers.

Let's move on to storage services.

- Amazon S3 or simple storage service provides object storage. Think of objects like individual files.
- Amazon Elastic Block Store provides block storage. Think of them like external hard-drives you'd attach to your computer. Block storage is, quote unquote, opposite of object storage like a zip file instead of a single image file.
- Amazon Snowball transfers huge amounts of data to AWS with a physical storage device.
- Amazon Storage Gateway provides gateways to connect on-premises resources with the cloud. Next, let's go over the database services.
- Amazon Dynamo DB is a non-relational or no SQL database.
- Amazon RDS or relational database service is a relational database.

- Amazon Aurora is a relational database and can be run on amazon RDS.
- Amazon Redshift is a data warehouse for a lot of data.

Now, to the network and content delivery services.

- Amazon VPC or virtual private cloud is a virtual network. Think of it as your corner of the AWS cloud.
- Amazon CloudFront helps you have speedy websites using edge locations.
- Amazon Route 53 routes domains to services and IP addresses.

Finally, let's go over the AWS management tools.

- Amazon CloudFormation helps you create templates to form cloud services.
- AWS CloudTrail helps you track trails of action, think of audit logs.
- Amazon CloudWatch monitors or watches your AWS cloud instance for you.

Just as a note, sometimes other services come up in the exam. However, if you know what these core services are and what they do, then you'll be able to filter through the ones you're unsure of to come to the correct answer. Thankfully it's a process of elimination because the exam is all multiple choice. So don't get distracted by a service you've never heard about and utilize your process of elimination skills.

Next, let's discuss infrastructure as code. The concept of infrastructure as code is that you can write code that describes the configurations for specific AWS cloud services and they can be deployed for you by AWS. It helps speed up the deployment process and removes the risk of human error when spinning up new resources.
Some AWS cloud services that utilize infrastructure as code are:

- Elastic Beanstalk,
- AWS Lambda, and
- AWS CloudFormation.

Some ways to deploy and manage resources on the AWS cloud are by utilizing AWS Management Console with AWS Command-Line Interface or CLI, and with AWS Software Deployment Kits, or SDKs.

Finally, let's review AWS as global infrastructure. AWS has data centers around the world called availability zones. Each availability zone is independent from each other in network and power source and there are currently almost six dozen availability zones or AZs around the world.

A region is made up of two or more availability zones and there are currently two dozen AWS regions around the world. You should strive to create a highly available, resilient and redundant IT infrastructure by replicating your AWS cloud resources across multiple availability zones and potentially, even regions.