Cloud computing is the on-demand delivery of compute power, database, storage, applications, and other IT resources through a cloud services platform through the internet with pay-as-you-go pricing. A cloud services platform provides rapid access to flexible and low-cost IT resources that you can use to build and maintain software and databases, and create applications to delight customers.

Benefits.

Cloud Computing enables you to access as many resources as you need, almost instantly, and only pay for what you use. On-demand, pay-as-you-go access to services is fundamental to the cloud computing model. You don’t need to make large upfront investments in hardware and spend a lot of time on the heavy lifting of managing that hardware.

The cloud provides developers with greater flexibility, scalability, and faster time to innovation.

* Pay only when you use computing resources, and only for how much you use.
* Benefit from massive economies of scale - This translates into lower pay-as-you-go prices.
* Stop guessing capacity - When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, you can access as much or as little capacity as you need, and scale up and down as required with only a few minutes notice.
* Increase speed and agility - IT resources are only a click away, which means that you reduce the time to make resources available to your developers from weeks to minutes. This dramatically Cloud Computing increases agility for the organization, because the cost and time it takes to experiment and develop is significantly lower.
* Realize cost savings – Thus giving Companies more time to focus on their customers and innovation. can focus on projects that differentiate their business instead of maintaining data centers. With cloud computing, you can focus on your customers, rather than on the heavy lifting of racking, stacking, and powering physical infrastructure.
* Helps you Go global in minutes, How? – By lowering latency,(latent period means an interval between stimulation and response),and this means a better experience for your customers at a minimal cost.

What is AWS Cloud?

The AWS Cloud encompasses a broad set of global cloud-based products that includes compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing.

the AWS Cloud has what you need to develop, deploy, and operate your applications, all while lowering costs, becoming more agile, and innovating faster.

One benefit of the AWS global infrastructure network is that you can provision resources in the Region or Regions that best serve your specific use case. When you are done with the resources, you can simply delete them. With this built-in flexibility and scalability, you can build an application to serve your first customer, and then scale to serve your next 100 million.

On Premises and Cloud Computing.

In an on-premises solution, If accompany wants to add an new feature or environment to an existing software, this requires them to buy and install hardware, connect the necessary cabling, provision power, install operating systems, and more. These tasks can be time consuming and expensive. Meanwhile, the team needs to delay the release of the new features while they wait for the QA environment. In contrast, if you run your application in the cloud, you can replicate an entire production environment, as often as needed, in a matter of minutes or even seconds. Instead of physically installing hardware and connecting cabling, the solution is managed over the internet.

Using cloud computing saves time during setup and removes the undifferentiated heavy lifting.

**IaaS, PaaS, and SaaS**

Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)

**Cloud computing, has several different service models** to help meet specific needs of different users, and these are IaaS, PaaS, and SaaS

**Infrastructure as a Service (IaaS)** contains the basic building blocks for cloud IT, and typically provides access to networking features, computers (virtual or on dedicated hardware), and data storage space.

**Platform as a Service (PaaS),** removes the need for you to manage the underlying infrastructure (usually hardware and operating systems) and allows you to focus on the deployment and management of your applications. This increases your efficiency.

and **Software as a Service (SaaS)** provides you with a completed product that is run and managed by the service provider. These are commonly refeered as end-user applications, example your web-based email

Module 2: AMAZON ELASTIC COMPUTE CLOUD(EC2)

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

1-Provision instances (virtual servers).

2-Upload your code.3

3-Continue to manage the instances while your application is running.

AWS provides secure, resizable compute capacity in the cloud as Amazon EC2 instances.

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.

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 compute 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. No hassles, no stress. 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. (What is an Amazon EC2 instance? Basically, An Amazon EC2 instance is a virtual server in Amazon’s Elastic Compute Cloud (EC2) used for running applications on the Amazon Web Services(AWS) infrastructure.

EC2 runs on top of physical host machines managed by AWS using virtualization technology. When you spin up an EC2 instance (Spin is an old IT term coming from the era in which data was stored on rotating devices, it’s a slang for “start a machine”, In the context of AWS, it just means start a virtual machine), 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. (a hypervisor, also known as a virtual machine monitor or VMM is a software that creates and runs virtual machines, VMs. A hypervisor allows one host computer to support multiple quest VMs by virtually sharing its resources, such as memory and processing). 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. They are secure and separate from each other.

The idea of EC2 gives you control and flexibility over the configuration of the instances(virtual servers).

When you provision an EC2 instance, you can choose the operating system based on either Windows or Linux. What is provisioning an instance? Provisioning an instance is the process of setting up an IT infrastructure. It can also refer to the steps required to manage access to data and resources, and make them them available to users and sytems. Provisioning is not the same thing as configuration, but they are both steps in the deployment process. Onces something has been provisioned, the next step is configuration. You can provision (set up) thousands of EC2 instances on demand with a blend of operating systems and configurations to power your business' different applications. 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.

AMAZON EC2 PRICING/BILLING OPTIONS.

Varities of Amazon EC2, pricing options;

-On Demand Instances: This is preferably for short term, irregular workloads that cannot be interrupted, the instances run until you stop them. You only pay for the computing time that you use, or that your instance (Virtual server) runs for, either per second or per hour depending on the instance type and OS you choose to run.

No upfront costs, no long-term commitments or minimum contracts apply. This type of pricing is usually for when you get started and want to spin(start) up servers to test out workloads and play around. You can also use them to get a baseline for your average usage. Example of use cases include developing and testing application, and running apps that have unpredictable usage patterns.

-Amazon EC2 Savings Plans: This plan is ideal for workloads that involve a consistent amount of compute usage over a 1-year or 3-year term. With this plan, you can reduce your compute cost by up to 72% (for example, $10 for an hour), over On-Demand costs on your AWS compute usage, thus lowering prices on your EC2 usage, regardless of instance family, size, OS, tenancy, or AWS region.

-Reserved instances; These are a billing discount applied to the use of On-Demand Instances in your account. And are suited for steady-state workloads or ones with predictable usage and offer you up to a 75% discount compare with On-Demand pricing. You qualify for a discount once you commit to a one or three-year term. But, you save more on the 3-year option. 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, and can pay for them with three payment options: namely

-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.

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: They are ideal for workloads that has flexible start and end time, or that can withstand interruptions. With Spot Instancess, you can reduce your compute cost up to 90% over On-demand costs. These Instances use unused Amazon EC2 computing capacity, where they allow you to request spare or unused Amazon EC2 computing capacity for up to 90% off of the On-Demand prices. Unlike Amazon EC2, Spot Instances do not require contracts or a commitment to a consistent amount of compute usage.

There are disadvantages to choosing Spot Instances;

-There are possibilities or unexpected interruptions. AWS can reclaim the instance at any time they need it and this might likely interrupt your workload or instance.

-You might make a spot instance request and Amazon EC2 capacity is unavailable, this might delay the launch of your job.

- Assuming you have launched a Spot Instance, and capacity is no longer available or demand for Spot Instances increases, your instance may be interrupted. This might not pose issues to your job.

Dedicated Hosts: These are physical servers that have Amazon EC2 Instance Capacity fully dedicated to your use. To maintain license compliance requirements, you can use your existing per-socket, per-core, or per-VM software licenses. An Advantage is that nobody else will share tenancy of your host with you.

-Disadvantage; Dedicated Hosts are the most expensive.

**What is the difference between Amazon EC2 Savings Plans and Spot Instances?**

* Amazon EC2 Savings Plans: This plan is idea for workloads that involve a consistent amount of compute usage over a 1-year or 3-year term. With this plans, you can reduce your compute cost by up to 72% (for example, $10 for an hour), over On-Demand costs. on your AWS compute usage, thus lowering prices on your EC2 usage, regardless of instance family, size, OS, tenancy, or AWS region.
* SPOT INSTANCES: They are ideal for workloads that has flexible start and end time, or that can withstand interruptions. With Spot Instancess, you can reduce your compute cost up to 90% over On-demand costs. Unlike Amazon EC2, Spot Instances do not require contracts or a commitment to a consistent amount of compute usage.

SCALING AMAZON EC2

**Scalability** involves beginning with only the resources you need and designing your architecture to automatically respond to changing demand by scaling out or in. As a result, you pay for only the resources you use.

The AWS service that provides this functionality for Amazon EC2 instances is **Amazon EC2 Auto Scaling, by emabling the scaling process to happen automatically**.

**Amazon EC2 Auto Scaling:** enables you to automatically add or remove Amazon EC2 instances in response to changing application demand. By automatically scaling your instances in and out as needed and terminate them when no longer needed, thus enabling you able to maintain a greater sense of application availability.

Within Amazon EC2 Auto Scaling, you can use two approaches:

1: Dynamic scaling- responds to changing demand.

2: predictive scaling. - automatically schedules the right number of Amazon EC2 instances based on predicted demand.

REMINDER: To scale faster, you can use dynamic scaling and predictive scaling together.

AUTO SCALING GROUP CONFIGURATION TYPES:

-Minimum Capacity

-Desired Capacity’

-Maximum Capacity.

When configuring the size of your Auto Scaling group, you might set the minimum number of Amazon EC2 instances at one (called minimum capacity). This means that at all times, there must be at least one Amazon EC2 instance running.

When you create an Auto Scaling group, you can set the minimum number of Amazon EC2 instances. The **minimum capacity** is the number of Amazon EC2 instances that launch immediately after you have created the Auto Scaling group. In this example, the Auto Scaling group has a minimum capacity of one Amazon EC2 instance. Next, you can set the **desired capacity** at two Amazon EC2 instances even though your application needs a minimum of a single Amazon EC2 instance to run.

REMINDER: If you do not specify the desired number of Amazon EC2 instances in an Auto Scaling group, the desired capacity defaults to your minimum capacity.

In Maximum Capacity Configuration, you might configure the Auto Scaling group to scale out in response to increased demand, but only to a maximum of four Amazon EC2 instances.

Because Amazon EC2 Auto Scaling uses Amazon EC2 instances, you pay for only the instances you use, when you use them. You now have a cost-effective architecture that provides the best customer experience while reducing expenses.

**Directing traffic with Elastic Load Balancing**

A load balancer is an application that takes in requests and routes them to the instances to be processed. A load balancer acts as a single point of contact for all incoming web traffic to your Auto Scaling group. This means that as you add or remove Amazon EC2 instances in response to the amount of incoming traffic, these requests route to the load balancer first. Then, the requests spread across multiple resources that will handle them.

**Elastic Load Balancing, ELB,** is the AWS service that is engineered to address the undifferentiated heavy lifting of load balancing by automatically distributing incoming application traffic across multiple resources, such as Amazon EC2 instances. Elastic Load Balancing is a regional construct, the service is automatically highly available with no additional effort on your part. It can be used for internal and external traffic

ELB is automatically scalable. As your traffic grows, ELB is designed to handle the additional throughput with no change to the hourly cost. When your EC2 fleet auto-scales out, as each instance comes online, the auto-scaling service just lets the Elastic Load Balancing service know that it's ready to handle the traffic, and off it goes. Once the fleet scales in, ELB first stops all new traffic, and waits for the existing requests to complete, to drain out. Once they do that, then the auto-scaling engine can terminate the instances without disruption to existing customers.

For example, if you have multiple Amazon EC2 instances, Elastic Load Balancing distributes the workload across the multiple instances so that no single instance has to carry the bulk of it.

{so, in a nut shell, If you are using Amazon EC2 Auto Scaling configuration, when there is regular web traffic(during a low-demand period), the Elastic load balancer utilizes the minimum capacity of EC2 instances in the your auto scaling group, then when there is an increase in incoming web traffic the elastic load balancer automatically distributes these excess traffic across the remaining EC2 Instances in your Auto scaling group?

Elastic Load Balancing and Amazon EC2 Auto Scaling are separate services, they work together to help ensure that applications running in Amazon EC2 can provide high performance, cost-efficient, highly available, and automatically scalable system that makes your applications more stable and more fault tolerant.

**Monolithic applications and microservices**

Applications are made of multiple components. The components communicate with each other to transmit data, fulfill requests, and keep the application running. These components might include databases, servers, the user interface, business logic, and so on.

-Monolithic Applications – an application architecture with tightly coupled components. In this approach to application architecture, if a single component fails, other components fail, and possibly the entire application fails.

To help maintain application availability when a single component fails, you can design your application through a **microservices** approach.

-Microservices Approach- Here, application components are loosely coupled. The loose coupling prevents the entire application from failing. In this case, if a single component fails, the other components continue to work.

Two services that facilitate application integration when designing applications on AWS, are:

Amazon Simple Notification Service (Amazon SNS) and Amazon Simple Queue Service (Amazon SQS).

**Messaging and queuing**

Applications send messages to each other to communicate. For example, if we have Application A and it is sending messages directly to Application B, if Application B has a failure and cannot accept those messages, Application A will begin to see errors as well. This is a tightly coupled architecture. A more reliable architecture is loosely coupled. This is an architecture where if one component fails, it is isolated and therefore won't cause cascading failures throughout the whole system.

In this case, we introduced a message queue. Messages are sent into the queue by Application A and they are processed by Application B. If Application B fails, Application A doesn't experience any disruption. Messages being sent can still be sent to the queue and will remain there until they are eventually processed. This idea of placing messages into a buffer is called messaging and queuing.

AWS services that can assist in this regard are;

1; Amazon Simple Queue Service or SQS : This is a message queuing service that enables messages between decoupled application complements. You can send, store, and receive messages between software components, without losing messages or requiring other services to be available. In Amazon SQS, an application sends messages into a queue. A user or service retrieves a message from the queue, processes it, and then deletes it from the queue.

2: Amazon Simple Notification Service or SNS :  is a publish/subscribe service. In Amazon SNS, subscribers can be web servers, email addresses, AWS Lambda functions, or several other options.  Using Amazon SNS topics, a publisher publishes messages to subscribers.

**Additional compute services**

Amazon EC2, a service that lets you run your applications through virtual servers in the cloud. If you have applications that you want to run in Amazon EC2, you must do the following:

1-Provision(setting up a server for use) instances (virtual servers), (In a simple word, This may refer to installing the operating system and other system software, as well as making adjustments in the software control panels, or it may refer to assigning an already configured server to a particular customer).

2-Upload your code.3

3-Continue to manage the instances while your application is running.

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

Benefits of Serverless Computing:

1: the flexibility to scale serverless applications automatically.

2: Serverless computing can adjust the applications' capacity by modifying the units of consumptions, such as throughput and memory.

**AWS LAMBDA**; This is an AWS service for serverless computing that lets you run code without needing to provision or manage servers.

While using AWS Lambda, you pay only for the compute time that you consume. Charges apply only when your code is running.

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

**How AWS Lambda works**

**1;** You upload your code to Lambda.

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

**3;** Lambda runs your code only when triggered.

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

**Containers**

**Containers** provide you with a standard way to package your application's code, configuration and dependencies(this is an essential functionality library or piece of code that’s essential for a different part of the code to work) into a single object. Containers share an operating system installed on the server and runs as resource-isolated process, ensuring quick, reliable, and consistent deployments, regardless of the environment (Operating system).

AWS container services make it easier to manage your underlying infrastructure, whether on-premises or in the cloud, so you can focus on innovation and your business needs. AWS for security, reliability, and scalability.

Why Use Containers? Containers are a powerful way for developers to package and deploy their applications. They are lightweight and provide a consistent, portable software environment for applications to easily…. run and scale anywhere. Building and deploying microservices, running batch jobs, for machine learning applications, and moving existing applications into the cloud are just some of the popular use cases for containers.

**-****Amazon Elastic Container Service (Amazon ECS);** is a highly scalable, high-performance container management system that enables you to run and scale containerized applications on AWS.

Amazon ECS supports Docker containers

-Docker- is a software platform that enables you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.. With Amazon ECS, you can use API calls to launch and stop Docker-enabled applications.

- **Amazon Elastic Kubernetes Service (Amazon EKS);** is a fully managed service that you can use to run Kubernetes on AWS.

1-Kubernetes: is open-source software that enables you to deploy and manage containerized applications at scale.

**Excerpt:** Kubernetes is an open-source container management and orchestration system. On AWS, you can choose to run and manage Kubernetes infrastructure yourself with Amazon EC2, or use Amazon EKS for a managed, automatically provisioned Kubernetes control plane.

**2-****AWS Fargate**; is a serverless compute engine for containers. AWS Fargate is a serverless, pay-as-you-go compute engine that lets you focus on building applications without managing servers. It works with both Amazon ECS and Amazon EKS.

AWS Fargate manages your server infrastructure for you. You can focus more on innovating and developing your applications, and you pay only for the resources that are required to run your containers.

**MODULE 3.**

AWS Global Infrastructure consists of Availability Zones and Regions.

An Availability Zone is a single data center or a group of data centers within a Region.

**A Region consists of two or more Availability Zones**

**It is best practice to run applications across at least two Availability Zones in a Region.**

**There are 4 factors to consider when determining and selecting the right region for your** services, data, and applications.

1. Compliance with data governance and legal requirements
2. Proximity to your customers- closer regions will help you to get content to your customers faster.
3. Available services within a Region
4. Pricing

Amazon CloudFront

Amazon CloudFront is a content delivery service. It uses a network of edge locations to cache content and deliver content to customers all over the world.

Edge locations

An edge location is a site that Amazon CloudFront uses to store cached copies of your content closer to your customers for faster delivery.

**Ways to interact with AWS services**

**There are 3 ways/methods to launch or configure Cloud services using AWS services, namely**

**1-AWS Management Console; -** **The AWS Management Console is a web-based interface for accessing and managing AWS services. It includes wizards and automated workflows that can simplify the process of completing tasks by utilizing names, keywords and acronym.**

**(What is a console? Traditionally, a console refers to a computer terminal where a user may input commands and view output). AWS Management Console is a web-based console that you can access through any web browser).**

**2-AWS Command Line Interface (AWS CLI);-** AWS CLI enables you to control multiple AWS services directly from the command line within one tool. Here, you are using a command prompt or terminal program to run a command.

**3-Software Development Kits (SDK);**- SDKs enable you to use AWS services with your existing applications or create entirely new applications that will run on AWS. SDKs make it easier for you to use AWS services through an API designed for your programming language or platform.

(Simply put, SDKs is just some software that’s provided from AWS to help developers to create applications that work with AWS services.) This is where we’re interacting programmatically, directly with the API. In this case, a developer would write code in an integrated development environment known as IDE, eg Microsoft Visual Studio code. The code is leveraging (to use (sth) to a maximum advantage) what’s called a software development kit to work with services.

**AWS Elastic Beanstalk**

Elastic Beanstalk is a service for deploying and scaling resources (web applications and services) in AWS Cloud, without having to learn about the infrastructure that runs those applications. With **AWS Elastic Beanstalk**, you provide code and configuration settings, and Elastic Beanstalk handles the details of

- Adjusting capacity

* **Load balancing**
* **Automatic scaling, and**
* **Application health monitoring.**

**AWS CloudFormation** is a service that helps you model and set up your AWS resources. With **AWS CloudFormation**, you can treat your infrastructure as code. This means that you can build an environment by writing lines of code instead of using the AWS Management Console to individually provision resources.

AWS Outposts is a family of fully managed solutions delivering AWS infrastructure and services to virtually any on-premises or edge location for a truly consistent hybrid experience

MODULE 4.

**NETWORKING**

**CONNECTIVITY TO AWS.**

Assume you are connected to AWS, in order isolate your workload, you need your own corner or niche or privacy. In AWS service, this is called Virtual Private Cloud.

**[Amazon Virtual Private Cloud (Amazon VPC)](https://aws.amazon.com/vpc/" \t "_blank)**.

Is a networking service that you can use to provision (set up) an isolated section of the AWS Cloud, establish boundaries around your AWS resources, organize your resources into **subnets,** and launch these resources in a virtual network that you have defined.

What is a subnet? A **subnet** is a section of a VPC that can contain resources such as Amazon EC2 instances, and Database.

To allow public traffic from the internet to access your VPC, you attach an **internet gateway** to the VPC. An internet gateway is a connection between a VPC and the internet.

If VPC includes only private resources, access to these private resources can only be possible through a **virtual private gateway**, a component that allows protected internet traffic to enter into the VPC, If is coming from an approved network, by establishing a network of protection called **Virtual Private Network** which will protect(encrypts) your internet traffic from all other requests around it.

**A virtual private gateway** enables you to establish a virtual private network (VPN) connection between your VPC and a private network, such as an on-premises data center or internal corporate network.

**AWS Direct Connect**

[**AWS Direct Connect**](https://aws.amazon.com/directconnect/) is a service/component that enables you to establish a dedicated private connection between your data center

in a VPC and AWS.  The private connection helps you to reduce network costs and increase the amount of bandwidth that can travel through your network.

How can you use AWS networking services to isolate resources and determine exactly how network traffic flows?

We are going to talk about the role of subnets within a VPC.

In a VPC, **subnets** are separate area/section within the VPC where you can group together resources based on security or operational needs. Subnets can be public or private.  (A typical illustration is a Dresser with Drawers, The dresser is your VPC, the drawers are your Subnets).

Subnets can be public or private.

**Public subnets** contain resources that need to be accessible by the public.

**Private subnets** contain resources that should be accessible only through your private network..

**Network traffic in a VPC.**

**A Network Access Control List (ACL)** is a VPC component that checks packet permissions by allowing or denying specific inbound or outbound traffic at the subnet level, by acting as a virtual firewall that controls inbound and outbound traffic at the subnet level.

Recall that in AWS Cloud, you have your VPC isolated with your resources which are organized in subnets. Data transmission between your VPC and a customer is made possible through an **internet gateway.**

**Youtube..**EC2 inatance reaches the internet through a gateway called internet gateway by enabling access, think of it as a door to the internet from your VPC , the outward. But just leaving the internet is not enough, bcos the subnet has to know how to reach the internet. And that’s where the route tables come in. ROUTE TABLES determine where network traffic from your subnets are directed. So you’d create a route in your route table to say , hey, route table, go here and go out to the internt. Alright, now that we have a way to the internet, and we can launch resourcesinto our subnets, what about security. Security groups acts as security at an Instance level’

Access to the private subnets are made possible through a **virtual private gateway**. In other to protect your data, the gateway establishes a network of protection called **Virtual Private Network** which will protect(encrypts) your internet traffic. from all other requests around it.

A virtual private gateway enables you to create a VPN connection between your VPC and a private network, such as your company’s data center. Although this connection is private and encrypted, it travels through the public internet, not through a dedicated connection, like AWS direct connection.

Now, when a customer requests data from an application stored in the AWS Cloud, this request is sent as a packet, (A **packet** is a unit of data sent over the internet or a network), which enters into a VPC through an internet gateway.

Before a packet (unit of data) can enter into a private subnet or exit from a subnet, a virtual firewall called A Network **Access Control List (ACL)**, checks for permissions, by controlling inbound and outbound traffic at the subnet level.

**Network access control lists (ACLs) perform stateless packet filtering**. They remember nothing and check packets that cross the subnet border each way: inbound and outbound. The rules in the network ACL are created by default when you set up an AWS account, or you can modify the rules in the list to suit your needs when configuring your VPC.

* Protection of the public subnet which contains an Amazon EC2 instance, is made possible through a virtual firewall called **Security groups** which controls inbound and outbound traffic. These **Security groups also perform stateful packet filtering** for public subnet by remembering previous decisions or request made for incoming packets. **A security group** is a virtual firewall that controls inbound and outbound traffic for an Amazon EC2 instance. **Security groups** are stateful and deny all inbound traffic by default.

Both network ACLs and security groups enable you to configure custom rules for the traffic in your VPC.

**Global Network.**

**Domain Name System (DNS)** resolution.; is the process of translating a domain name to an IP address. DNS is a directory used for matching domain names to IP addresses .When customers enter the web address of any Company that has a website hosted in the AWS Cloud into their browser, they are able to access the website because a DNS resolution involves a customer DNS resolver communicating with a company DNS server by asking the web server for the IP address that corresponds to that Company’s website. The company DNS server responds by providing the IP address for that Company’s website, 192.0.2.0.

**Amazon Route 53**

Amazon Route 53 is a highly available and scalable [Domain Name System (DNS)](https://aws.amazon.com/route53/what-is-dns/) web service. Route 53 connects user requests to internet applications running on AWS or on-premises. It’s very cost-effective. Amazon Route 53 is a DNS web service. It gives developers and businesses a reliable way to route end users to internet applications that host in AWS.

ROUTE 53 manages the DNS records for domain names. You can transfer DNS records for existing domain names managed by other domain registrars. You can also register new domain names directly in Route 53.

**How Amazon Route 53 and Amazon CloudFront deliver content**

**Process continues from DNS**, --- This information is sent back to the customer. The customer’s request is sent to the nearest edge location through Amazon CloudFront. Amazon CloudFront connects to the Application Load Balancer, which sends the incoming packet to an Amazon EC2 instance.

**MODULE 5**

STORAGE AND DATABASE.

**INSTANCE STORE;** An instance store is disk storage that is physically attached to the host computer for an EC2 instance, that provides temporary block-level storage for an Amazon EC2 instance, and therefore has the same lifespan as the instance..

Amazon EC2 instances are virtual servers. If you start an instance from a stopped state, the instance might start on another host, where the previously used instance store volume does not exist. Therefore, AWS recommends instance stores for use cases that involve temporary data that you do not need in the long term.

**AMAZON ELASTIC BLOCK STORE (AMAZON EBS);** is a service that provides block-level storage volumes that you can use with Amazon EC2 instances. If you stop or terminate an Amazon EC2 instance, all the data on the attached EBS volume remains available.

To create an EBS volume, you define the configuration (such as volume size and type) and provision it, then it can attach to an Amazon EC2 instance.

Because EBS volumes are for data that needs to persist, it’s important to back up the data. You can take incremental backups of EBS volumes by creating Amazon EBS snapshots.

**Amazon EBS snapshots**

An [**EBS snapshot**](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html) is an incremental backup. This means that after your most recent snapshot are saved, for subsequent backups, only new blocks of data that have changed are saved. Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

Instance store; When stopping or terminating an EC2 instance, data is deleted, and its best for temporary data that is not kept long-term.

Amazon EBS volume; Best for data that require retention. When stopping or terminating an EC2 instance, data remains available.

Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

full backups, in which all the data in a storage volume copies each time a backup occurs. The full backup includes data that has not changed since the most recent backup.

**Amazon Simple Storage Service (Amazon S3)**

**Object storage** consists of data, metadata, and a key.

The data might be an image, video, text document, or any other type of file.

Metadata contains information about what the data is, how it is used, the object size, and so on.

An object’s key is its unique identifier.

Recall that when you modify a file in block storage, (Instance Stores and EBS,) only the pieces that are changed are updated. When a file in object storage is modified, the entire object is updated.

**Amazon S3 is** an object storage built to retrieve any amount of data from anywhere.

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, unmatched security, and performance.

Customers of all sizes and industries can store and protect any amount of data for virtually any use case. The maximum file size for an object in Amazon S3 is 5 TB.

When you upload a file to Amazon S3, you can set permissions to control visibility and access to it. You can also use the Amazon S3 versioning feature to track changes to your objects over time.

**Amazon S3 storage classes**

Two factors to consider when selecting an Amazon S3 storage class.

* How often you plan to retrieve your data
* How available you need your data to be

Once an S3 Lifecycle policy is set, your data will automatically transfer to a different storage class without any changes to your application.

Amazon S3 also offers capabilities to manage your data throughout its lifecycle.

**The S3 storage classes** include

**S3 Intelligent** - Tiering for automatic cost savings for data with unknown or changing access patterns,

**S3 Standard** for frequently accessed data.

**S3 Standard-Infrequent Access (S3 Standard-IA**) and **S3 One Zone-Infrequent Access (S3 One Zone-IA)** for less frequently accessed data.

**S3 Glacier Instant Retrieval** for archive data that needs immediate access.

**S3 Glacier Flexible Retrieval (formerly S3 Glacier**) for rarely accessed long-term data that does not require immediate access.

**Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive**) for long-term archive and digital preservation with retrieval in hours at the lowest cost storage in the cloud.

**S3 Outposts** storage class stores data on premises, in situations where data residency requirements can’t be met by an existing AWS Region.

Amazon Virtual Private Cloud, or VPCs,