AWS(Amazon Web Services):

*AWS or Amazon Web Services, is a cloud computing platform that offers on-demand computing services such as virtual servers and storage that can be used to build and run applications and websites. AWS is known for its security, reliability, and flexibility, which makes it a popular choice for organizations that need to store and process sensitive data.*

*Amazon Web Services is the world’s most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers- including faster-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.*

*AWS is a comprehensive cloud computing platform that includes infrastructure as a service (IaaS) and platform as a service (PaaS) offerings.*

*AWS provides a wide range of services with a****pay-as-per-use pricing model****over the Internet such as Storage, Computing power, Databases, Machine Learning services, and much more. AWS facilitates for both businesses and individual users with effectively hosting the applications, storing the data securely, and making use of a wide variety of tools and services improving management flexibility for IT resources.*

***\*AWS Services:***

*~ Compute Services:*

*`AWS Elastic Compute Cloud (EC2).*

*`AWS Lambda.*

*~ Storage Services:*

*`AWS Simple Storage Service(AWS S3).*

*`AWS Elastic Block Store(EBS).*

*`AWS Relational Database System(RDS).*

*~Networking Services:*

*`AWS Virtual Private Cloud(VPS).*

*~ Security and Identity:*

*` AWS Identity and Access Management(IAM).*

*~ Deployment and Automation:*

*`AWS Elastic Beanstalk(EBS).*

*~ Machine Learning Services:*

*`AWS SageMaker*

**Application of AWS**

Amazon Web Services (AWS) is being increasingly adopted by many large enterprises such as Netflix, McDonald’s, Airbnb, NASA, and Samsung to expand their businesses. AWS offers a variety of applications, some of which include:

* Storage and Backup
* Social Networking
* Mobile Apps
* Websites
* Gaming
* ec2(Elastic compute cloud):

EC2 is a web service which is provided by the AWS cloud which is secure, resizable, and scalable. Instead of managing the infrastructure AWS will do that so you can just launch and terminate the EC2 instance whenever you want. You can deploy your applications in EC2 servers without any worrying about the underlying infrastructure. You configure the EC2-Instance in a very secure manner by using the VPC, Subnet and Security group. You can scale up and down the EC2 instance depending on the incoming traffic. The other advantage of AWS EC2 is that you need to pay only for how much you use it is like the pay-as-you-go model.

**Use Cases of Amazon EC2 (Elastic Compute Cloud)**

1. **Deploying Application:** In the AWS EC2 instance, you can deploy your application like **.jar,.war,** or **.ear** application without maintaining the underlying infrastructure.
2. **Scaling Application:**Once you deployed your web application in the EC2 instance know you can scale your application based upon the demand you are having by scaling the AWS EC2-Instance.
3. **Deploying The ML Models:**You can train and deploy your ML models in the EC2-instance because it offers up to 400 Gbps), and storage services purpose-built to optimize the price performance for ML projects.
4. **Hybrid Cloud Environment:**You can deploy your web application in EC2-Instance and you can connect to the database which is deployed in the on-premises servers.
5. **Cost-Effective:**Amazon EC2-instance is cost-effective so you can deploy your gaming application in the Amazon EC2-Instances

**\*\*EC2 Instances:** EC2 allows you to rent virtual servers (instances) on which you can run your own applications.These instances can be configured with varying amounts of CPU, memory, storage, and networking capacity, depending on your needs. EC2 instances are highly customizable and can be launched and terminated as needed, providing flexibility and scalability for your applications.

Different Amazon EC2 instance types are designed for certain activities. Consider the unique requirements of your workloads and applications when choosing an instance type. This might include needs for computing, memory, or storage.

**The AWS EC2 Instance Types are as follows:**

1. General Purpose Instances: These instances provide a balance of compute, memory, and networking resources. They are suitable for a wide range of workloads including web servers, development environments, gaming servers, personal projects and small to medium databases.

**EC2 General-Purpose Instance Types**

**T2. micro:** The most well-known instance in AWS is t2.micro, which gives 1 CPU and 1 GB of memory with low to moderate network performance.

**M6a Instance:** In m6a there are different sizes like m6a.large, m6a.2xlarge, m6a.4xlarge, and so on. m6a.large offers 2 CPUs, 8GiB memory, and network performance up to 12.5 Gigabit.

**M5 instance: M5 instance:**The newest generation of general-purpose instances, known as M5, are powered by Intel’s Xeon Platinum 8175 processors.

1. Compute Optimized Instances:These instances are designed for workloads that require high-performance compute capabilities, such as batch processing, media transcoding, and high-performance web servers

**Compute-Optimized Some Instance Types**

**c5d.24large:**The c5d instance, which has 96 CPUs, 192 GiB of RAM, 3600 GB of SSD storage, and 12 Gigabit of network performance.

1. Memory-Optimized Instances:These instances are optimized for memory-intensive workloads, such as in-memory databases, real-time big data analytics, and high-performance computing (HPC) applications.

**Memory-Optimized Some Instance Types**

**The R and X categories belong to the memory-optimized family.**

**R7g.medium:** It is run on AWS Gravitation processors with ARM architecture. with 1 CPU, 8 (GiB) of memory, an EBS storage type, and a maximum of 12.5% network bandwidth.

**x1:**X1 is mainly suited for enterprise edition databases with memory applications and comes with 64 vCPU, 976 GiB of memory, 1 x 1,920 GB of SSD Storage, 7,000 Mbps of dedicated EBS bandwidth, and 10 Gbps of network performance.

1. Storage Optimized Instances: These instances are designed for workloads that require high storage capacity and high disk throughput, such as data warehousing, distributed file systems, and large-scale data processing.

**Storage Optimized Instance Types**

**Im4gn:**  Im4gn is powered by AWS Graviton processors, it offers the best pricing performance for workloads in Amazon EC2 that demand a lot of storage. It offers some other instance types of **ls4gn, l4i, D, and H.**

1. Accelerated Computing Instances:These instances are equipped with specialized hardware accelerators, such as GPUs (Graphics Processing Units) or FPGAs (Field-Programmable Gate Arrays), to speed up compute-intensive workloads such as machine learning, scientific simulations, and video encoding.

**Accelerated Computing Instance Types**

Accelerated computing consists of mainly P1, Inf2, G5, G5g, G4dn, G4ad, G3, F1 and VT1.

\*\*AMI(Amazon Machine Image ):

An **Amazon Machine Image** is a special type of virtual appliance that is used to instantiate (create) a virtual machine within EC2. it's the core unit for deploying services via EC2. Whenever you want to launch an instance, you need to specify AMI. To launch instances, you can also use different AMIs. If you want to launch multiple instances from a single AMI,  then you need multiple instances of the same configuration. An AMI acts as a template for the root volume of an instanc( essential components like operating systems, application servers, and applications).

## **Why do We Need AMI?**

Let us suppose that we want to launch 5 servers with the same configuration. One way of doing that would be to launch a new EC2 instance every time and install the required packages every time. While the other way of doing it would be to configure your EC2 instance once and then create an image of that instance. Using that image you can deploy four more EC2 servers.

*AMI is used to create EC2-Instances and EC2-Instance are used to create the virtual machines.*

**\*SNAPSHORT:** Snapshots can be used to create a backup of critical workloads, such as a large database or a file system that spans across multiple EBS Volume.

EBS Snapshots are point-in-time images or copies of your EBS Volume. These are stored on S3, which can be accessed through Elastic Compute Cloud APIs or AWS Console

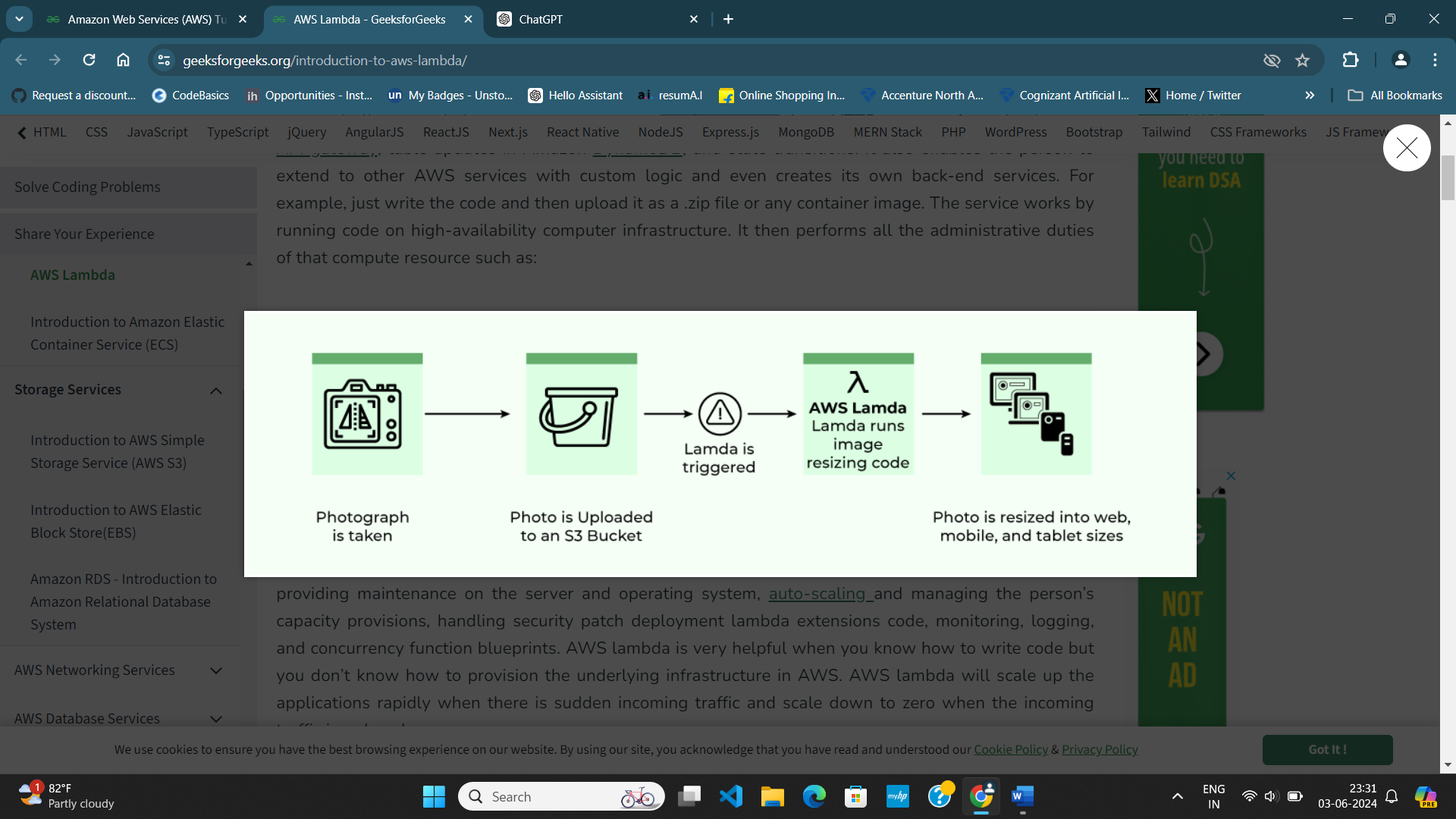
*AWS Snapshots are the copy of EBS volumes and stores in the AWS S3 bucket for longer use.*

* Amazon Lambda:

AWS lambda are server-less compute functions are fully managed by the AWS where developers can run there code without worrying about servers.AWS lambda functions will allow you to run the code with out provisioning or managing servers.

Once you upload the source code file into AWS lambda in the form of ZIP file then AWS lambda will automatically run the code without you provision the servers and also it will automatically scaling your functions up or down based on demand.

AWS lambda are mostly used for the event-driven application for the data processing Amazon S3 buckets, or responding to HTTP requests.



**Triggered**: Triggered means that the function is executed in response to a specific event or action. AWS Lambda supports various triggers, such as changes in data stored in Amazon S3, updates to records in DynamoDB tables, incoming HTTP requests through Amazon API Gateway.

**Provisioning**: Provisioning refers to the process of allocating resources, such as servers or computing capacity, to handle the execution of an application or service.

* Amazon S3(SIMPLE STORAGE SERVICE):

Amazon S3 is a Simple Storage Service in AWS that stores files of different types like Photos, Audio, and Videos as Objects providing more scalability and security to. It is designed to store and retrieve any amount of data from anywhere on the web. S3 provides developers and IT teams with secure, durable, and highly-scalable object storage infrastructure at a relatively low cost, and simple connection to other AWS Service.

**\*Why S3 Used:** Amazon S3 is used for scalable, durable, and cost-effective storage of any type of data, from small files to large datasets, offering high availability and low latency access from anywhere on the web.

 **Object Storage**: S3 is an object storage service designed to store and retrieve any amount of data from anywhere on the web. It stores data as objects within buckets, and each object consists of data, a key (unique identifier), and metadata.

 **Scalability**: S3 is highly scalable and can store virtually unlimited amounts of data, making it suitable for storing large volumes of static content, backups, logs, and data lakes.

 **Durability and Availability**: S3 is designed for 99.999999999% (11 nines) durability and is replicated across multiple Availability Zones within a region to ensure high availability.

 **Versatility**: S3 supports various storage classes optimized for different access patterns and cost requirements, including Standard, Infrequent Access (IA), One Zone-IA, Intelligent-Tiering, Glacier, and Glacier Deep Archive.

AWS S3 Bucket: An S3 bucket is a fundamental Storage container for storing objects (files) in S3. Each bucket will have its own set of policies and configurations. It is similar to a folder in a file system, but in the case of S3, it's a top-level container. Each bucket has a unique name globally across AWS, and objects in S3 are stored in these buckets. Buckets can be configured with various settings for access control, versioning, logging, and more. There is a limit of 100 buckets per AWS account. But it can be increased if requested by AWS support.

* AWS Objects:

An object in S3 is any piece of data that is stored in a bucket. It can be anything from a text file to a large binary file, and each object is stored as a key-value pair. Objects have a unique key within the bucket, which is used to retrieve them. Along with the data, objects also contain metadata (such as content type, creation date, etc.) and a unique identifier (called an Object ID) generated by S3. Objects can range in size from 0 bytes to 5 terabytes.

It consists of the following:

* Key, Version ID, Value, Metadata, Subresources, Access control information, Tags.
* AWS classes:

AWS S3 offers different storage classes tailored to different use cases and performance needs.

 **Standard**: This is the default storage class for S3. It provides high durability, availability, and performance for frequently accessed data.

 **Standard-IA (Infrequent Access)**: Designed for data that is accessed less frequently but requires rapid access when needed. It offers lower storage costs compared to the Standard class, with retrieval fees.

 **One Zone-IA**: Similar to Standard-IA, but data is stored in a single Availability Zone, reducing costs further at the expense of slightly lower durability.

**How to Access Amazon S3 Bucket?**

You can work and access the Amazon S3 bucket by using any one of the following

1. AWS Management Console

1. AWS CLI Commands
2. Programming Scripts ( Using boto3 library of Python )

* Amazon EBS(Elastic block STORAGE):

Amazon Elastic Block Store (EBS) is a cloud-based block storage service provided by Amazon Web Services (AWS). Block storage stores files in multiple volumes called blocks, and attach them to Amazon EC2 instances, block-level storage means that data is stored in discrete blocks rather than as a single continuous stream. EBS volumes are highly available and reliable storage resources that can be easily scaled up or down based on demand.

**Block-Level Means:** Block-level storage refers to a method of storing data in fixed-sized blocks or chunks. Each block is treated as an individual hard drive, with its own address and access controls. These blocks can be accessed and managed independently of each other.

Key features of AWS EBS include:

1. **High Performance**: EBS volumes offer low-latency performance, making them suitable for a wide range of applications, from transactional databases to high-performance computing workloads.
2. **Durability**: EBS volumes are designed for durability, with data automatically replicated within the same Availability Zone to protect against component failure.
3. **Snapshots**: Users can create point-in-time snapshots of their EBS volumes, which are stored in Amazon S3. Snapshots can be used to create new EBS volumes or to back up data for disaster recovery purposes.
4. **Encryption**: EBS volumes support encryption at rest, allowing users to encrypt their data using AWS Key Management Service (KMS) keys.
5. **Flexible Storage Options**: EBS offers several types of volumes optimized for different use cases, including General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), and Cold HDD (sc1).

**Difference in AWS S3 And AWS EBS:**

EBS is primarily used for block-level storage attached to EC2 instances, while S3 is used for object storage accessible over HTTP/S. EBS is suitable for applications that require low-latency, high-performance storage with the ability to attach to EC2 instances, whereas S3 is ideal for storing large volumes of data, backups, and static content with high durability and scalability.

* Amazon vpc(virtual private cloud):

AWS VPC can be referred to as the private cloud inside the cloud. It is a logical grouping of servers in a specified network. The servers that you are going to deploy in the Virtual Private Cloud(VPC) will be completely isolated from the other servers that are deployed in the Amazon Web Services. You have complete control over your VPC, from creation to customization and even deletion. With the help of security groups and network access control lists, you can protect your application more.

The basic architecture of a properly functioning VPC consists of many distinct services such as Gateway, Load Balancer, Subnets etc. Altogether, these resources are clubbed under a VPC to create an isolated virtual environment. Along with these services, there are also security checks on multiple levels.

It is initially divided into subnets, connected with each other via route tables along with a load balancer.

Virtual Private Cloud Componets:

**Subnet**

These are segments of the VPC's IP address range in which you can place groups of resources. You can define multiple subnets within a VPC to isolate different types of resources.

To reduce traffic, the subnet will divide the big network into smaller, connected networks. Up to /16, 200 user-defined [subnets](https://www.geeksforgeeks.org/introduction-to-subnetting/).

**Route Tables**

[Route Tables](https://www.geeksforgeeks.org/routing-tables-in-computer-network/) contains a set of rules, called route which helps us to route the network traffic.

A single VPC can have as many as route tables it requires. Route Table are mainly used to Define the protocol(rules) for routing network traffic between subnets, internet gateways, virtual private gateways, NAT gateways, and other network interfaces.

**Network Access Control Lists**

[Network Access Control Lists (NACL)](https://www.geeksforgeeks.org/amazon-web-services-security-group-vs-nacl/)  are stateless firewalls that control traffic entering and exiting a subnet. They allow you to create rules that apply to all traffic entering or leaving the subnet.. There will be a default NACL for each VPC that cannot be deleted.

"Stateless" refers to a system or process that does not maintain information about the previous interactions or states of users or components

**Internet Gateway(IGW)**

The I[nternet Gateway (IGW)](https://www.geeksforgeeks.org/introduction-of-gateways/) enables communication between instances in your VPC and the internet. It provides a target in your VPC route tables for internet-routable traffic.One VPC can’t have more than one IGW

**Network Address Translation (NAT)**

[Network Address Translation (NAT)](https://www.geeksforgeeks.org/network-address-translation-nat/) These components allow instances in private subnets to initiate outbound traffic to the internet while preventing inbound traffic from directly reaching those instances.

**VPC peering**

[VPC peering.](https://www.geeksforgeeks.org/amazon-vpc-concept-of-vpc-peering/)can establish the connection between two Virtual Private Clouds which enables you to route the traffic between two VPCs using the IP address. The virtual servers which are in the same network can communicate with each other with out VPC peering connections but the servers which are in the two different networks can’t communicate with each other with out VPC peering.

* Aws iam(identity and access management):

IAM stands for Identity and Access Management, which is a service provided by Amazon Web Services (AWS) for securely managing access to AWS resources. IAM enables you to control who can access your AWS resources (authentication) and what actions they can perform on those resources (authorization). The account we created to sign in to Amazon web services is known as the root account and it holds all the administrative rights and has access to all parts of the account.

IAM identities are the entities (users, groups, and roles) that you use to manage access to AWS resources. Each identity has its own set of security credentials, such as a username and password for users, or an access key and secret key for roles and applications.

**Users** are individual identities that you create in AWS, to access the AWS Console and their administrative permissions differ from those of the Root user  each with its own set of security credentials.

**Groups** are collections of users,  and a single person can be a member of several groups. With the aid of groups, we can manage permissions for many users quickly and efficiently.

**Roles** are similar to users, but they are meant to be assumed by entities such as AWS services, applications, or users from other AWS accounts.

IAM (Identity and Access Management) is a key component of AWS security, allowing you to control access to your AWS resources and services securely. IAM identities are the entities within IAM that you use to manage access permissions.

* Aws ebs(elastic beanstalk):

Elastic Beanstalk is a Platform as a Service (PaaS) offering from AWS, which simplifies the process of deploying, managing, and scaling web applications and services developed with various programming languages, frameworks, and containers. Elastic Beanstalk provides a convenient and cost-effective way to deploy and manage web applications on AWS, allowing developers to focus on building great software without worrying about the underlying infrastructure..

The need for Elastic Beanstalk arises from the complexity involved in managing the underlying infrastructure of web applications. Traditionally, deploying and managing web applications required expertise in server provisioning, configuration, scaling, and monitoring. With Elastic Beanstalk, AWS handles much of this complexity, allowing developers to focus more on writing code and less on managing infrastructure.

**The components of Elastic Beanstalk include:**

1. **Application**: This is the logical collection of Elastic Beanstalk components, which consists of your application code and its associated configurations. Elastic Beanstalk directly takes in our project code. So Elastic Beanstalk application is named the same as your project home directory.
2. **Environment**: An environment is an instance of your application running on Elastic Beanstalk. You can have multiple environments (e.g., development, testing, production) for the same application. Each environment has its own set of AWS resources, such as EC2 instances, databases, and load balancers.
3. **Environment Health**: One of the most lucrative features of running applications on AWS or most of the other cloud platforms is automated health checks. AWS runs automatic health checks on all EC-2 deployments (Elastic Beanstalk is a managed EC-2 service) which can be monitored from the AWS console.
4. **Version**: A version represents a specific deployment of your application code and configurations within an environment. You can deploy multiple versions of your application and easily roll back to previous versions if needed.

**Health check responses**:

* **Red:** The application failed all health tests.
* **Yellow:** The application failed some of the health tests.
* **Grey:**The application is updating.
* **Green:**The application passed the health check successfully

1. **Configuration**: Elastic Beanstalk allows you to configure various aspects of your environment, such as instance types, auto-scaling settings, environment variables, and more, through configuration files or the AWS Management Console.
2. **Resources**: Elastic Beanstalk automatically provisions and manages the underlying AWS resources required to run your application, such as EC2 instances, load balancers, and databases.
3. **Elastic Load Balancing:** All the web requests to the application are not directly relayed to application instances. They first hit the Elastic Load Balancer (ELB), which, as the name suggests, balances the load across all the application instances.
4. **Language support:**Elastic Beanstalk supports the applications developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

* Aws Cloudformation:

"CloudFormation is an Infrastructure as Code (IaC) service provided by Amazon Web Services (AWS) for automating the provisioning and management of AWS resources. It allows users to define their infrastructure in a declarative template format, enabling consistent and repeatable deployments."

CloudFormation mainly used to provision the service in the AWS like [EC2](https://www.geeksforgeeks.org/what-is-elastic-compute-cloud-ec2/), [S3,](https://www.geeksforgeeks.org/introduction-to-aws-simple-storage-service-aws-s3/) [Autoscaling](https://www.geeksforgeeks.org/amazon-web-services-scaling-amazon-ec2/), l[oad balancing](https://www.geeksforgeeks.org/elastic-load-balancer-in-aws/) and so on you can provision all the service automation with the Infrastructure as a code (IAC), instead of managing all of them manually.

AWS CloudFormation that helps you create and manage the resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS.

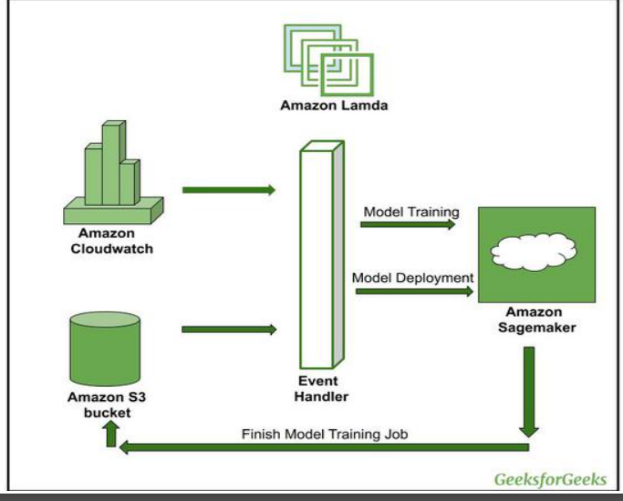
You just have to create a template thatbased on describes all the resources you require, then AWS Cloudformation will take care of managing and provisioning all the resources. AWS provides a Cloudformation designer for designing the template wherein you can put all the resources. You can also define the dependencies of all the resources that are needed. You can also reuse your templates to replicate your infrastructure in multiple environments and regions.

CloudFormation simplifies and streamlines the process of managing AWS infrastructure, improving efficiency, consistency, and reliability.

* Aws SageMaker:

Amazon SageMaker is a fully managed service provided by AWS that simplifies the process of building, training, and deploying machine learning (ML) models at scale. It provides a comprehensive set of tools and infrastructure to streamline the entire ML lifecycle, from data preprocessing and model training to deployment and monitoring. the other benefit is that you can use other AWS services in your model such as S3 bucket, amazon Lambda for monitoring the performance of your ML model you can use AWS Cloudwatch which is a monitoring tool.

Overall, SageMaker simplifies the end-to-end process of building and deploying ML models, enabling faster experimentation, efficient resource utilization, and seamless integration with other AWS services.



**SageMaker's key modules and workflow include:**

1. **Data Preparation**: SageMaker facilitates data preprocessing and feature engineering tasks, allowing users to collect, clean, transform, and prepare their datasets for training.
2. **Model Building**: Once the data is prepared,Users can choose from a variety of built-in algorithms(including TensorFlow, PyTorch, and MXNet) or bring their own custom algorithms to train ML models using SageMaker.
3. **Model Training**: Once the data is prepared, After the model is built, the next step is to train it using the prepared data. SageMaker provides a range of options for training, including distributed training on multiple instances for faster results.
4. **Model Optimization(Evaluation)**: Once the model is trained, the next step is to optimize it for performance.SageMaker enables users to optimize model performance by tuning hyperparameters and evaluating model metrics. It automates the process of experimenting with different configurations to find the best-performing model.
5. **Model Deployment**: Once a model is trained and evaluated, SageMaker makes it easy to deploy it into production with just a few clicks or through automated pipelines. SageMaker provides options for deploying models to various endpoints, including Amazon EC2 instances, Lambda functions, and API Gateway. Models can be deployed as real-time endpoints for inference or as batch transformations for large-scale data processing.
6. **Model Monitoring**: SageMaker provides built-in monitoring capabilities to track model performance and detect drift in data distributions over time.
7. **Model Management:** Finally, once the model is in production, it’s important to manage it over time. This includes tasks such as updating the model with new data, retraining the model periodically, and ensuring that it remains performant over time. It also offers tools for managing deployed endpoints, scaling resources based on demand, and integrating with AWS services for logging and security.