AWS Introduction

What is AWS?

Amazon Web Services (AWS) is the leading cloud computing platform, offering over 200 services for computing, storage, databases, machine learning, and more.

Key Features of AWS

- Global Availability: AWS has regions and availability zones worldwide to ensure reliability.
- Pay-as-You-Use Model: Users pay only for what they consume, theoretically making it cheaper than traditional data centers.
- Scalability & Flexibility: Services scale automatically based on demand.
- Security & Compliance: AWS ensures high security, with tools for encryption, identity management, and monitoring.

History of AWS

- 2006: AWS launched with two services: S3 (Simple Storage Service) & EC2 (Elastic Compute Cloud).
- 2010: Expansion to include SimpleDB, Elastic Block Store (EBS), RDS (Relational Database Service), DynamoDB, CloudFront, CloudWatch, and more.
- Continuous Growth: AWS frequently introduces new services for cloud computing, machine learning, and analytics.

Cloud Computing Models

Cloud services fall into three categories:

Infrastructure as a Service (IaaS)

- Provides virtual machines, storage, and networking.
- Users manage the OS, applications, and configurations.
- Example: AWS EC2 (Elastic Compute Cloud)

Platform as a Service (PaaS)

- Provides a managed environment to develop, run, and manage applications.
- Users focus on application development without worrying about infrastructure.
- Example: AWS Elastic Beanstalk

Software as a Service (SaaS)

- Fully managed applications provided by a vendor.
- Users only interact with the front-end application.
- Example: Google Drive, Dropbox, or AWS WorkSpaces

AWS Shared Responsibility Model

AWS follows a shared responsibility model for security:

AWS Responsibilities (Security OF the Cloud)

- Securing physical infrastructure, data centers, and hardware.
- Maintaining hypervisors, host operating systems, and managed services.

Client Responsibilities (Security IN the Cloud)

- Configuring identity and access management (IAM).
- Protecting and encrypting data stored in AWS.
- Managing applications and network security within their Virtual Private Cloud (VPC).

AWS Global Infrastructure

AWS ensures availability and performance using:

- Regions: Geographic areas (e.g., us-east-1, us-west-1).
- Availability Zones (AZs): Data centers within regions to ensure redundancy.
- Edge Locations: Used for content delivery networks (CDN) to cache data closer to users.

AWS Core Services

Compute Services

- EC2 (Elastic Compute Cloud): Virtual machines for running applications.
- EC2 Spot Instances: Discounted EC2 instances for flexible workloads.
- ECS (Elastic Container Service): Manages Docker containers.
- EKS (Elastic Kubernetes Service): Manages Kubernetes clusters.
- AWS Lambda: Serverless computing—runs code without provisioning servers.

Storage Services

- S3 (Simple Storage Service): Object storage, highly scalable.

- EBS (Elastic Block Store): Block storage for EC2 instances.
- EFS (Elastic File System): Serverless, scalable file storage.
- AWS Backup: Automates data backup across AWS services.

Database Services

- Relational Databases: Amazon RDS, Amazon Aurora.
- Key-Value Stores: Amazon DynamoDB.
- In-Memory Databases: Amazon ElastiCache, Amazon MemoryDB.
- Document Databases: Amazon DocumentDB (MongoDB compatible).
- Graph Databases: Amazon Neptune.

Analytics Services

- Athena: SQL-based querying for data in S3.
- EMR (Elastic MapReduce): Big data analytics using Spark, Hive, and Presto.
- Glue: Data integration, preparation, and ETL (Extract, Transform, Load).
- Redshift: Data warehousing for analytical processing.
- Kinesis: Real-time data streaming and processing.
- QuickSight: Cloud-based business intelligence (BI) tool.

Machine Learning & AI Services

- SageMaker: End-to-end machine learning platform.
- Comprehend: Natural language processing (NLP).
- Rekognition: Image and video analysis.
- Textract: Extracts text from scanned documents.
- Translate: Machine translation.

AWS for Data Analytics & Engineering

AWS provides several key tools for data professionals:

- EC2 & Lambda: Compute services for processing workloads.
- S3: Scalable object storage for big data.
- RDS & DynamoDB: Relational and NoSQL databases.
- Glue & Athena: Data transformation and querying.
- EMR: Managed big data frameworks.
- Redshift: Cloud-based data warehouse for structured analytics.

AWS Free Tier

The AWS Free Tier allows users to explore AWS services for 12 months with limits:

- EC2: 750 hours/month (limited instance types).
- S3: 5GB of storage, 20,000 GET requests/month.
- RDS: 750 hours/month of database use.

AWS Free Tier enables users to experiment without upfront costs, making it great for learning cloud computing.

Amazon EC2 & AWS Lambda

Amazon EC2 (Elastic Compute Cloud)

EC2 is a cloud-based virtual computing service that allows users to run applications on virtual machines (VMs). It provides scalable computing capacity with a pay-as-you-go pricing model.

Key Features of EC2

EC2 provides elasticity, allowing instances to scale up or down dynamically based on demand. It offers multiple instance types optimized for general compute, memory, storage, and GPU workloads. Users can create custom AMIs (Amazon Machine Images) to launch instances with predefined software and configurations. EC2 integrates with AWS services such as S3, RDS, and Lambda.

EC2 Lifecycle

When using EC2, users can launch an instance with a chosen configuration, start or stop an instance without deleting it, reboot an instance without losing data, and terminate an instance permanently, deleting all associated data unless backed up.

Where Can You Store Data on EC2?

Instance store is high-speed, temporary storage tied to the instance lifecycle. Elastic Block Store (EBS) provides persistent, block-level storage. Elastic File System (EFS) is scalable file storage shared across multiple instances. Amazon S3 is scalable object storage for backup and large datasets.

Common EC2 Use Cases

EC2 is commonly used for web hosting to run websites and applications, data processing for handling workloads using Python, R, or Spark, machine learning for training ML models on GPU-optimized instances, and disaster recovery for backing up workloads in case of failure.

Setting Up an EC2 Instance

To set up an EC2 instance, users need to launch an instance via the AWS Management Console, choose an AMI (Amazon Machine Image) based on the OS and software stack, select an instance type such as t2.micro for free-tier eligibility, configure networking and security by assigning security groups and SSH keys, and connect via SSH using a key file and public IP address.

To install necessary packages on an EC2 instance, users should update the system and install software such as Python using the following commands:

```
sudo apt update && sudo apt upgrade
```

```
sudo apt install python3-pip
```

Running Applications on EC2

For deploying applications, users can install MiniConda on EC2. To do this, they need to download the MiniConda installer, execute the installation script, and verify the installation.

Another use case is deploying a Streamlit web app on EC2. To do this, users should install dependencies such as Streamlit and scikit-learn, create a directory for the web app, write a Python script for the Streamlit app, and then run the app using the Streamlit command.

AWS Lambda (Serverless Computing)

AWS Lambda is a serverless computing service that runs code without requiring users to manage servers. It executes functions only when triggered, making it cost-effective.

Key Features of AWS Lambda

AWS Lambda is event-driven and runs in response to AWS events such as S3 uploads or API Gateway requests. It supports multiple programming languages, including Python, Node.js, Java, and Go. The pricing model is based on execution time rather than infrastructure usage. Lambda automatically scales up or down based on demand.

How AWS Lambda Works

Users upload code via the AWS console, CLI, or AWS SDK, define a trigger event such as API Gateway or DynamoDB updates, and AWS Lambda executes the function when the trigger occurs. The results can be stored in AWS services like S3 or DynamoDB.

Creating an AWS Lambda Function

To create an AWS Lambda function, users need to navigate to AWS Lambda in the AWS Console, click “Create Function,” select "Author from Scratch," choose a runtime such as Python 3.8, write code in the inline editor or upload a ZIP file, configure triggers such as S3 or API Gateway, deploy the function, and test it.

Example AWS Lambda Function in Python

```
import json


def lambda_handler(event, context):
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from AWS Lambda!')
    }
```

After writing the function, users should deploy it and create a test event to verify execution.

Differences Between EC2 and Lambda

EC2 provides full control over the virtual machine, while Lambda is a fully managed serverless execution environment. EC2 requires manual or auto-scaling, whereas Lambda automatically scales based on demand. EC2 pricing is based on running instances, while Lambda charges based on execution time. EC2 is best for persistent applications and databases, whereas Lambda is ideal for short-lived tasks and event-driven execution.

Summary

Amazon EC2 provides virtual machines in the cloud, allowing full control over computing resources. AWS Lambda enables serverless execution, reducing infrastructure management and costs. EC2 is best suited for persistent workloads, while Lambda is ideal for event-driven applications. Both services integrate seamlessly with the AWS ecosystem, including S3, RDS, and API Gateway.