



# AMAZON WEB SERVICES

# Cloud Computing Platforms – Overview

- The three biggest cloud providers—
- **Amazon (AWS), Google Cloud, and Microsoft Azure**
- These platforms provide SERVICES, **computing power, storage, databases, tools** to help users deploy and manage applications efficiently.

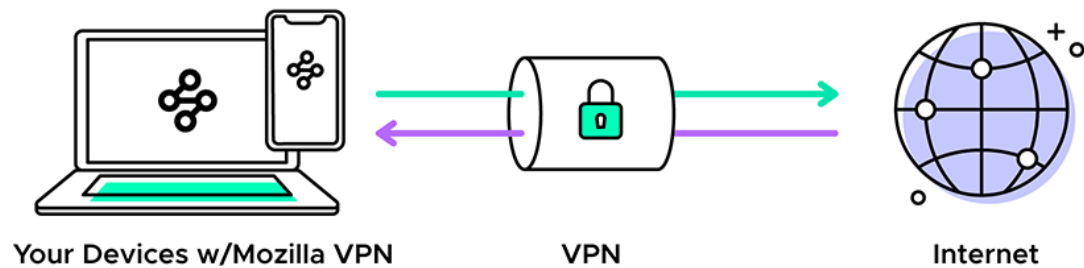
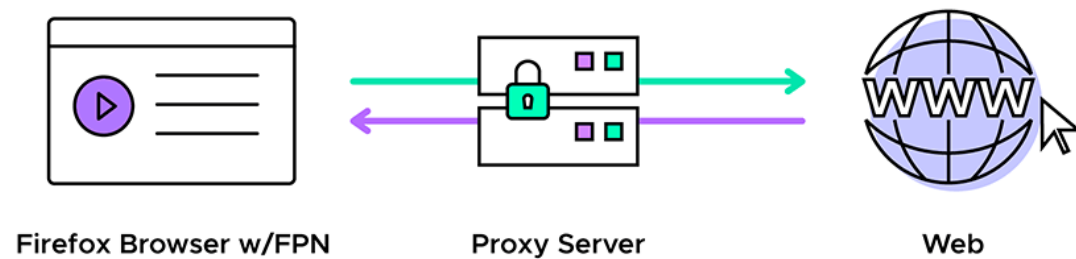
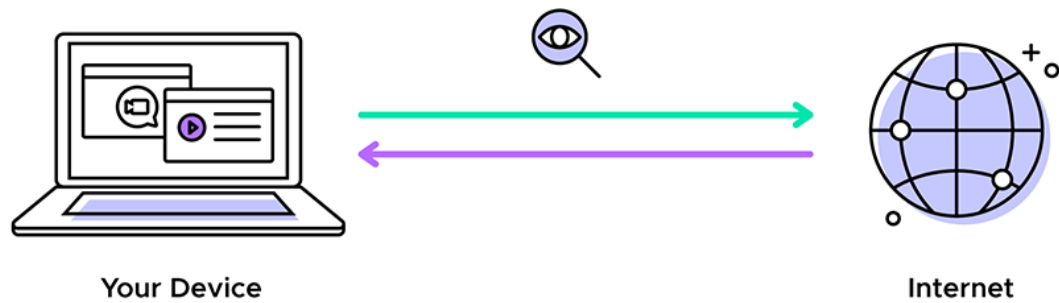
# What is a VPN?

A **Virtual Private Network (VPN)** is a **secure, encrypted connection** over the internet that allows users to access private networks **as if they were directly connected**. It ensures **privacy, security, and data integrity** by preventing unauthorized access.

## How VPN Connects to Amazon EC2 & VPC?

Amazon EC2 provides **Virtual Private Cloud (VPC)** services that allow enterprises to **connect their cloud resources (like EC2 instances) to their internal network using a VPN**. Here's how it works:

- - 1 **Users A & B access EC2 virtual machines (VM1, VM2, VM3) through a VPN.**
  - 2 **These VMs have private IPs assigned and are isolated from the public internet.**
  - 3 **Traffic between the enterprise network and cloud resources is securely encrypted.**
  - 4 **Firewalls, security policies, and proxies can be applied just like in an on-premise data center.**



# Infrastructure as a Service (IaaS) – Amazon EC2 (AWS)

- ◆ **Amazon EC2 (Elastic Compute Cloud)** is a service that provides **virtual machines (VMs)** in the cloud.
- ◆ Users can **launch, configure, and scale** servers on-demand, paying only for what they use.

## ✓ **Key Features:**

- Choose OS (Windows/Linux)
  - Flexible compute power
  - Autoscaling & load balancing
  - Secure networking
- 
- ◆ **Example:** A startup running an **e-commerce website** can use **EC2 instances** to handle traffic spikes during sales and scale down afterward, reducing costs.

# Amazon EC2 (Elastic Compute Cloud)

- Amazon EC2 (Elastic Compute Cloud) is a virtual machine service that allows businesses to run applications in the cloud without owning physical servers.
- Instead of buying hardware, users can rent virtual machines (VMs) and pay only for what they use.

## ◆ Key Features of EC2:

- ✓ Virtual Machines on Demand: Deploy multiple VMs (called instances) as needed.
- ✓ Scalability: Increase or decrease compute power based on demand.
- ✓ Security & Networking: Secure access with firewalls, VPNs, and encryption.
- ✓ Flexible Pricing: Pay-as-you-go or choose reserved instances for long-term savings.
- ✓ Multiple OS Options: Supports Windows, Linux, macOS for different workloads.

# How Amazon EC2 Works:

- ✓ Virtualized Infrastructure: Provides compute power, storage, and networking as virtual resources.
- ✓ On-Demand Provisioning: Deploy servers in minutes using a web-based AWS Management Console.
- ✓ Customizable: Choose the operating system, CPU, memory, and storage based on workload needs.
- ✓ Scalability & Flexibility: Easily scale up or down based on demand using Auto Scaling.
- ✓ Pay-as-You-Go Pricing: No upfront costs—users only pay for the resources they use.

## ◆ Example Use Cases of EC2 in IaaS

- [1] Web Hosting: Run dynamic websites and applications on scalable servers.
- [2] Big Data Processing: Perform large-scale data analysis using high-performance EC2 instances.
- [3] Machine Learning (ML) & AI: Train ML models with GPU-enabled instances.
- [4] Enterprise Applications: Deploy ERP, CRM, and database systems.
- [5] Disaster Recovery: Use EC2 instances for backup and failover solutions

# Amazon Simple Storage Service (Amazon S3)

- is a highly scalable object storage service used for storing, retrieving, and managing data in the cloud.

## ◆ Key Features of Amazon S3

- ✓ Object-Based Storage: Stores data as objects inside buckets (containers for objects).
- ✓ Scalability: Automatically scales to store unlimited amounts of data.
- ✓ Durability & Availability: Provides 99.999999999% durability, ensuring data is never lost.
- ✓ Security: Supports encryption, access control policies, and AWS Identity & Access Management (IAM).
- ✓ Cost-Effective: Pay only for what you use, with no upfront costs.

## ◆ How Amazon S3 Works

- 1 Create a Bucket – Choose a unique bucket name and select an AWS region
- 2 Upload Objects – Store files (images, videos, backups, logs, etc.) as objects.
- 3 Set Permissions – Manage access with public/private settings and IAM policies.
- 4 Retrieve & Manage Data – Access objects via REST APIs, AWS SDKs, or AWS CLI.



# Amazon SimpleDB – NoSQL Database Service

- a managed NoSQL database that offers high availability, flexibility, and automatic administration, making it easy for developers to store and query data without managing traditional databases.

## ◆ Key Features of Amazon SimpleDB

- ✓ Schema-Free: No predefined schema, allowing flexible data storage.
- ✓ Highly Available: Ensures automatic replication across multiple locations.
- ✓ Simplified Management: No need to manage servers, indexing, or scaling.
- ✓ REST & SOAP APIs: Access and query data through web service requests.
- ✓ Automatic Indexing: All data attributes are automatically indexed for fast retrieval.

## ◆ How Amazon SimpleDB Works

- 1 Create a Domain – Similar to a table in relational databases.
- 2 Store Data Items – Each item (like a row) has attributes (key-value pairs).
- 3 Query Data – Use SELECT queries to retrieve data via APIs.
- 4 Scale Automatically – Amazon handles performance & replication.

# Amazon Elastic Block Store (Amazon EBS)

- is a **high-performance, scalable block storage service** designed for **Amazon EC2** instances. It provides **persistent storage** for applications running in AWS.

## ◆ Key Features of Amazon EBS

- ✓ **Durable & Persistent Storage:** Data remains even if the EC2 instance stops.
- ✓ **Scalable & Flexible:** Resize storage volumes as needed.
- ✓ **High Performance:** Optimized for **fast read/write speeds**.
- ✓ **Backup & Snapshots:** Easily create snapshots for disaster recovery.
- ✓ **Encryption & Security:** Supports encryption for data protection.

## ◆ How Amazon EBS Works

- - 1 **Create an EBS Volume** – Choose a size and performance level.
  - 2 **Attach to EC2 Instance** – Connect the storage volume to an EC2 virtual machine.
  - 3 **Format & Use** – Create a file system and store data, run databases, or use it as a raw block device.
  - 4 **Scale & Backup** – Resize storage and create snapshots for backups.

# Amazon SQS (Simple Queue Service)

- a fully managed message queuing service that enables asynchronous communication between software components, ensuring smooth message flow.

## ◆ Key Features of Amazon SQS

- ✓ Reliable Message Delivery: Ensures messages are stored and delivered without loss.
- ✓ Scalable & High Throughput: Supports millions of messages per second.
- ✓ Decouples Components: Enables independent scaling of producers and consumers.
- ✓ Secure & Encrypted: Uses AWS authentication & encryption for secure data transmission.
- ✓ Flexible Processing Speeds: Messages are stored temporarily and processed at the consumer's speed.

## ◆ How Amazon SQS Works

- Producer Sends a Message – A service or application sends a message to the queue.
- Message Stored in Queue – SQS temporarily stores the message until it's retrieved.
- Consumer Reads Message – The consumer application retrieves and processes the message.
- Message Deleted – Once processed, the message is removed from the queue.

# Amazon Machine Images (AMI)

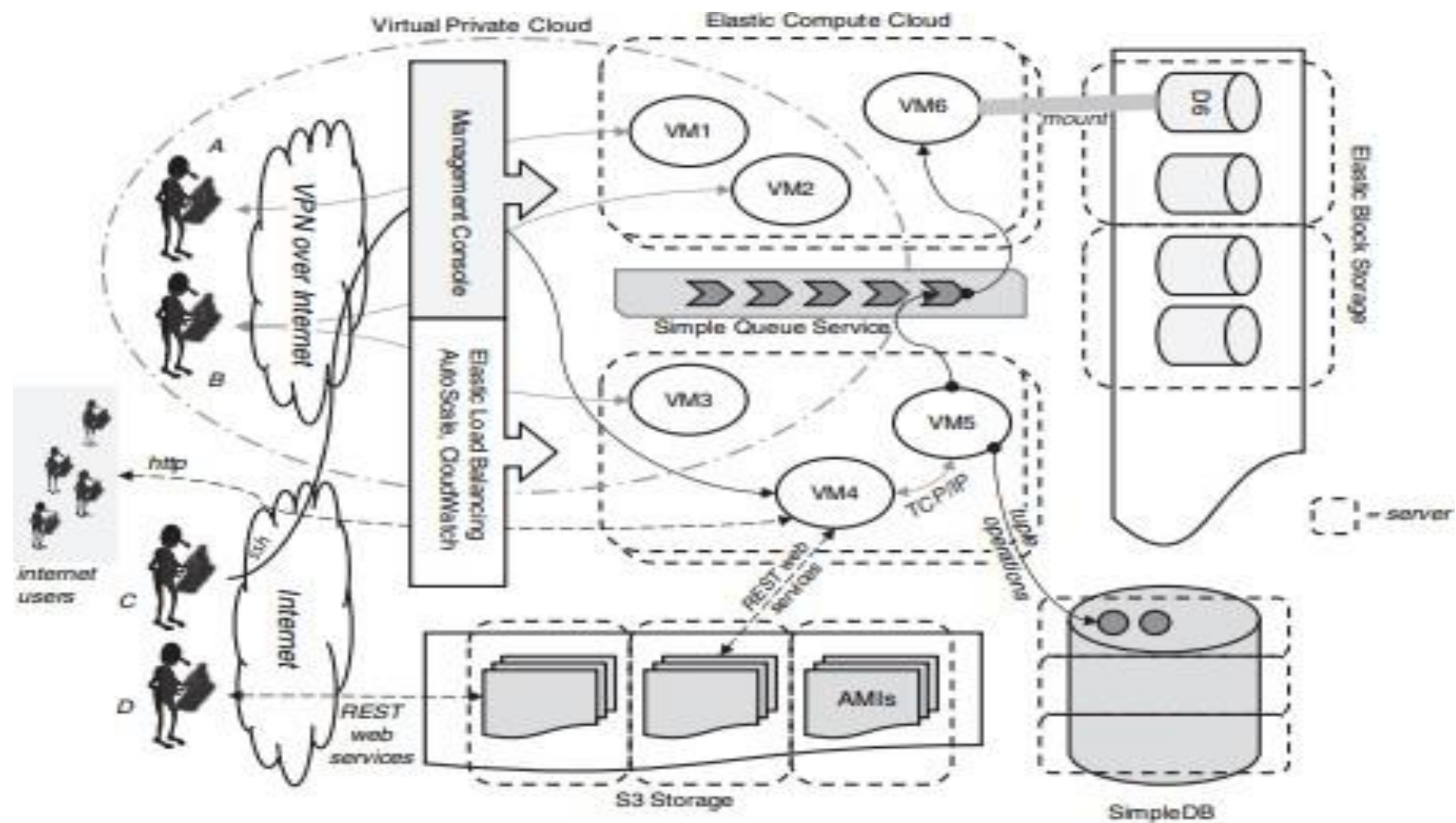
- a **pre-configured template** that contains the software, settings, and permissions required to **launch virtual servers (EC2 instances) in AWS**.
- It includes the **operating system (OS), application server, and applications**, making deployment **faster and more efficient**.

## ◆ Types of AMI Storage

- **1 EBS-backed Instances** – Uses **Amazon Elastic Block Store (EBS)** for persistent storage. The data is retained even after stopping the instance.
- **2 Instance Store-backed Instances** – Uses **temporary storage**. Data is **lost** when the instance is stopped or terminated.

## ◆ How AMI Works

- **1 Choose an AMI** – Select from AWS Marketplace, Community AMIs, or create a custom AMI.
- **2 Launch an EC2 Instance** – Use the AMI to deploy a server.
- **3 Customize & Save as New AMI** – Modify an instance and create a new AMI for future use.



**FIGURE 5.1. Amazon infrastructure cloud**

- These services are implemented on a **very large network of servers**, shown as **dashed boxes** in the figure.
- The **Elastic Compute Cloud** service provides users access to dedicated virtual machines of a desired capacity that are provisioned on these physical servers, with details of the actual physical server, such as its location, capacity, etc. being transparent to the end-user.
- Through the **management console** users generate PKI1 key- pairs using which they can securely **login** to these virtual servers over the internet.
- In Figure, user C provisions the virtual server **VM4** through the **management console**, and accesses it using **ssh**, for a Linux server, or via 'remote desktop' for a Windows server.
- Users have a choice of **virtual machine images** (called Amazon machine images, or AMIs) to choose from when provisioning a server. All AMIs are stored in common storage in the **Amazon S3** storage service (which we shall return to below), and **used to boot the desired configuration of virtual server**.

- The user's account is **charged on an hourly basis based on actual consumption**, i.e. time that the server is up. Charges vary depending on the AMI used and capacity of server chosen while provisioning.
- Cloud users have **root/administrator** access to these servers, and therefore control them completely. For example, they can deploy applications and make them publicly accessible over the internet.
- **Static network addresses** required for such purposes can also be provisioned through the **management console**. Thus, **VM4 is also accessible by internet users at large over HTTP**.
- Such publicly available static IP addresses are **charged on a fixed monthly basis**.

- Users can **provision and access many servers** that can communicate with each other over the fast internal network within the Amazon cloud.
- For example, user C in Figure has provisioned **VM5 and VM6 in addition to VM4**. If VM4 is a **web server**, VM5 may be a **database server**, and these two **communicate** over the internal cloud network using **TCP/IP**.
- Since VM5 is a database server, it needs to store and retrieve data. The **Amazon SimpleDB** service provides an object store where key-value pairs can be efficiently stored and retrieved.
- It is important to note that **SimpleDB is not a relational database**.
- It is important to understand that virtual servers **do not have any persistent storage**; so any user data on file system (i.e. whatever is not part of the AMI) **is lost when the server shuts down**. In order to store data persistently, such in a relational database, **Elastic Block Storage** needs to be mounted on a virtual server.



- In the **figure VM6 might be an archival server** where VM5 sends logs of whatever updates it makes to the SimbleDB data store.
- Note that **VM6 has mounted a logical volume D6**, where it possibly maintains archived data.
- Now notice that VM5 sends data to VM6 not over TCP/IP, but using the **Amazon Simple Queue Service (SQS)**. The SQS is a reliable persistent message queue that is useful for temporarily storing data.
- Usage of the SQS is charged based on **data volumes and how long data resides in the queue**.
- SQS is normally used for managing such **asynchronous transfer of data between processing servers in a batch-oriented workflow**.

- The **Amazon S3 Storage Service** provides a different storage model.
- Data in S3 can be files of **any type**, and in general any **blob (binary large object)**.
- Users access and modify S3 objects via **URIs, using REST web services** where as Persistent storage in the EBS as described above can be accessed **only if it is attached to a running virtual server**.
- S3 objects are accessible over the **internet as well as from virtual servers** within the Amazon cloud.
- S3 is especially useful for reliably storing **large collections of unstructured data** that need to be accessed by many client applications.
- It is important to note that all data in S3 is automatically replicated **at least three times for fault tolerance**.

- An important goal of any cloud service is insulating users from variable demand by automatically **managing scaling up and down of the resources** allocated to a cloud application.
- In an infrastructure cloud, such as Amazon EC2, the user needs to explicitly define an architecture that enables scalability **using tools** provided by Amazon to manage **elasticity**: Runtime performance parameters, such as CPU and I/O utilization, of a user's virtual servers can be monitored in real-time by **Amazon Cloud Watch**; this data can be used by **Amazon Auto Scale** to **add (or remove) virtual servers from an application cluster and automatically provision them with predefined machine images**.
- **Elastic Load Balancing** allows a group of servers to be configured into a set across which **incoming requests (e.g. HTTP connections) are load balanced**. The performance statistics of the load-balanced requests can also be monitored by **Cloud Watch** and **used by Auto Scale to add or remove servers from the load balanced cluster**.

- Enterprises seeking to adopt cloud computing also need to address the **security concerns** of corporate IT.
- Network security is an important element of these concerns: An enterprise's computing resources are usually protected by **firewalls, proxy servers, intrusion detection systems etc.**
- Amazon EC2 provides a Virtual Private Cloud service, whereby virtual servers can be connected to an enterprise's internal network using a **VPN**. (virtual private network).
- For example, users A and B in Figure access virtual servers VM1, VM2 and VM3 through a **VPN running over the public internet**. These servers then have **private IP addresses controlled by the enterprise's network operations center**. Access to and from this network to the outside world can be controlled by the same set of policies, firewalls and proxies as for other resources in the enterprise's own data center.