

# Introduction to AWS and EC2

EC006 - AWS Cloud Services and Infrastructure

# Activity

Introduction to AWS and EC2

# Activity Output

```
ubuntu@ip-172-31-16-215: ~  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-16-215:~$
```

# Activity: Instructions

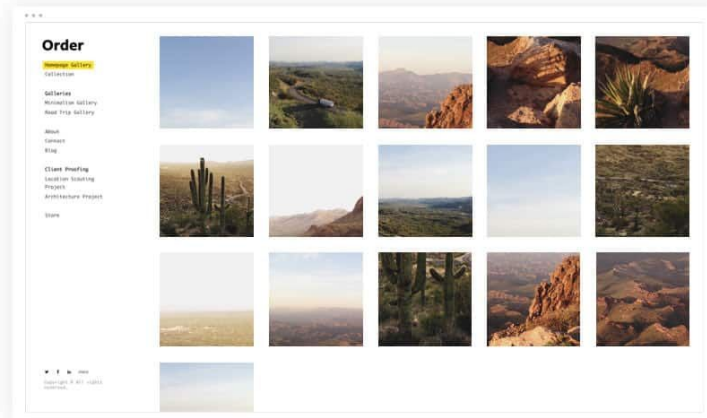
---

- Review the topics discussed in the previous session.
- Answer the quiz form about the Introduction to AWS.

# Serverful Architecture

Introduction to AWS

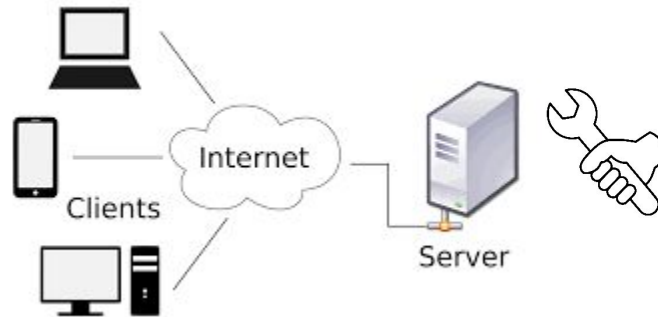
Imagine you want to build a website that allows users to upload and share photos.



Source: <https://www.format.com/magazine/resources/photography/photography-website-templates>

# Serverful Architecture

Traditionally, when you **build a software application**, you need to have a **server** to host the application. This is known as a **"serverful" architecture**.



With serverful architecture, you would have to manage the server yourself, including its maintenance, scaling, and security.

Managing a serverful architecture typically involves several processes, including:

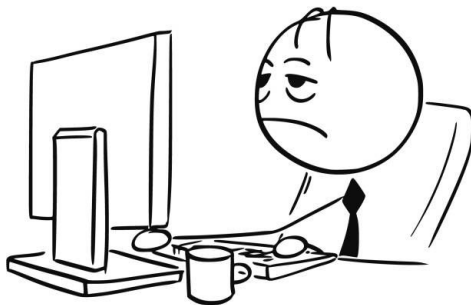
1. **Procuring and configuring hardware:** You need to purchase or lease servers and networking equipment, install operating systems and other software, and configure the hardware to meet your application's needs.
2. **Provisioning and scaling resources:** You need to manage server resources such as CPU, memory, and storage to ensure that your application can handle user traffic. This involves adding or removing servers as needed to handle changes in traffic.



3. **Monitoring and troubleshooting:** You need to monitor the servers and applications to identify and resolve issues that arise. This involves tracking metrics such as CPU usage, memory usage, network traffic, and application errors.
4. **Security:** You need to ensure that the servers and applications are secure, protecting them against threats such as hackers and malware. This involves implementing security protocols, such as firewalls and intrusion detection systems, and keeping software up-to-date with security patches.

5. **Deploy the web application:** The web application would be deployed to the server by uploading the necessary files and configurations to the server. This could be done manually, or through an automated deployment process.
6. **Configure DNS:** The domain name of the web application would need to be configured with DNS to point to the public IP address of the server.
7. **Maintain and update the server:** The server would need to be maintained and updated regularly to ensure it runs smoothly and remains secure.

**All of these processes require time and expertise to manage**, and they can be a significant drain on development resources.



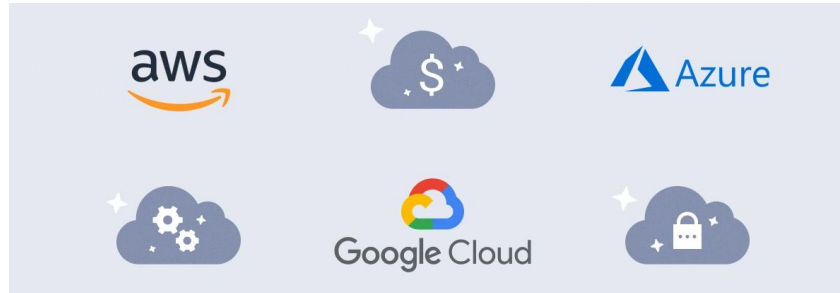
**Serverless architecture simplifies** much of this process by **outsourcing** many of these tasks to the **cloud provider**, allowing developers to focus on writing code rather than managing infrastructure.

# Serverless Architecture

Introduction to AWS

# Serverless Architecture

With **serverless architecture**, you no longer have to worry about managing the server. Instead, your application is broken down into smaller functions, and those functions are run in a cloud environment **managed by a cloud provider** such as **Amazon Web Services (AWS)**, **Microsoft Azure**, or **Google Cloud Platform**, among others.



Source: <https://sam-solutions.us/what-is-public-cloud-computing/>

So, instead of having to worry about buying and setting up a server, you can **simply upload your code and the cloud provider will take care of the rest**, only charging you for the specific resources used (e.g. storage for the photos, data transfer, number of requests to the API, etc.).

Think of it like **ordering a pizza**. When you want a pizza, you don't have to build a pizzeria, hire pizza makers, and buy all the ingredients. Instead, you just call a pizza restaurant, place an order, and they will take care of the rest, delivering the pizza to your door.

**Serverless architecture** works similarly, **allowing you to focus on what you want to build** (the pizza), **while the cloud provider takes care of the underlying infrastructure** and resources (the pizzeria and ingredients).

# What is AWS?

Introduction to AWS



# What is AWS?

---



# What is AWS?

---

**AWS (Amazon Web Services)** is a **cloud computing platform** that provides a wide range of services for building and deploying web and mobile applications. It is a **subsidiary of Amazon** that was **launched in 2006**, and has since grown to become one of the most widely used cloud computing platforms in the world.

# What is AWS?

AWS offers a number of **serverless and serverful services** that are commonly used, such as **Amazon EC2**, **AWS Lambda**, **Amazon API Gateway**, and **Amazon DynamoDB**, among others.



# What is AWS?

## Amazon Services



Amazon EC2



AWS Lambda



Amazon  
**API Gateway**



Amazon DynamoDB



Amazon S3

# Amazon Elastic Cloud Compute (EC2)

Introduction to AWS

# Amazon Elastic Cloud Compute (EC2)

## Amazon EC2 (Elastic Compute Cloud)

**EC2 is a virtual server in the cloud that allows users to run applications** on the AWS infrastructure. With EC2, users can choose from a variety of instance types and sizes, depending on the requirements of the application.



Amazon EC2

EC2 provides more flexibility, scalability, availability, and cost savings compared to a typical non-cloud setup.

- **Scalability:** EC2 instances can be easily scaled up or down depending on the application's needs. With a non-cloud setup, scaling would require physically procuring and configuring additional hardware, which can be time-consuming and costly.
- **Availability:** EC2 instances can be provisioned in different regions and availability zones, which can provide high availability and reduce the risk of downtime. A non-cloud setup would typically involve a single physical server, which would be a single point of failure.

# Amazon Elastic Cloud Compute (EC2)

EC2 provides more flexibility, scalability, availability, and cost savings compared to a typical non-cloud setup.

- **Cost:** With EC2, you only pay for the resources that you use, which can help reduce costs. With a non-cloud setup, you would need to purchase or lease hardware, which can be a significant upfront expense.
- **Management:** With EC2, Amazon takes care of the management of the underlying infrastructure, including hardware, networking, and storage. With a non-cloud setup, you would be responsible for managing and maintaining the hardware and the associated infrastructure.



# Amazon Elastic Cloud Compute (EC2)

---

AWS is known for its customer success stories, where they highlight how companies are using their services, including EC2. Some of these companies include **Netflix**, **Airbnb**, **Pinterest**, **Expedia** and **Unilever**.

# Amazon Elastic Cloud Compute (EC2)

---

**Netflix** is one of the largest users of AWS, and they **use AWS EC2 to host their streaming service**. They run thousands of EC2 instances in multiple regions around the world, which allows them to deliver high-quality streaming video to millions of customers.

# NETFLIX

# Amazon Elastic Cloud Compute (EC2)

**Airbnb** is another company that relies heavily on AWS EC2 to run their website and mobile app. They **use EC2 instances to host their web servers, databases, and other backend services**, which allows them to scale their service quickly and easily as their user base grows.



# Amazon Elastic Cloud Compute (EC2)

**Pinterest** uses AWS EC2 to **run their entire web infrastructure, including their search engine, recommendation engine, and user database**. They also use EC2 to run their data processing and machine learning workflows, which help them to provide personalized recommendations to their users.



# Amazon Elastic Cloud Compute (EC2)

**Expedia**, the travel company uses AWS EC2 to power their website and mobile app. They **use EC2 instances to host their web servers, databases, and other backend services**, which allows them to provide a fast and reliable booking experience for their customers.



# Amazon Elastic Cloud Compute (EC2)

**Unilever**, a multinational consumer goods company **uses AWS EC2 to run their e-commerce platform**. They use EC2 instances to host their web servers, as well as their databases and other backend services, which allows them to provide a seamless online shopping experience for their customers.



# AWS Lambda

Introduction to AWS

## AWS Lambda

A serverless compute service that **allows developers to run their code in response to events**, such as a user uploading a file to S3 or a request coming in through an API Gateway. Developers simply write their code and upload it to AWS Lambda, and the service takes care of running and managing the computing resources required by that code.



AWS Lambda



As an example, a **financial company** may **use AWS Lambda to process incoming data from financial institutions and transform it** into a format that can be analyzed by the company's data scientists.

Here's how it might work.

1. Integration with financial institutions: The financial company sets up an integration with the financial institutions that will be sending data. This could involve **creating an API** or other integration method **that allows the institutions to securely transmit the data** to the company's AWS environment.
2. Data ingestion: When the data is transmitted, **AWS Lambda functions are triggered to ingest the data**. Lambda functions are event-driven, which means **they can be triggered automatically by specific events** or changes, such as new data being uploaded to a designated data storage bucket.

3. Data transformation: Once the data is ingested, **the Lambda functions perform data transformation operations** to convert the raw data into a format that can be analyzed by the company's data scientists. This might involve cleaning the data, converting data types, or combining multiple data sources into a single format.
4. Data storage: Once the data has been transformed, **the Lambda functions store it in a database** or data lake. This allows the data to be easily accessed and analyzed by the company's data scientists.
5. Analysis: With the data now in a usable format, **the company's data scientists can analyze it to gain insights** and make data-driven decisions that can benefit the company and its customers.

# Amazon API Gateway

Introduction to AWS

## Amazon API Gateway

A fully managed service offered by Amazon Web Services (AWS) that makes it easy to **create, deploy, and manage APIs for web applications.**

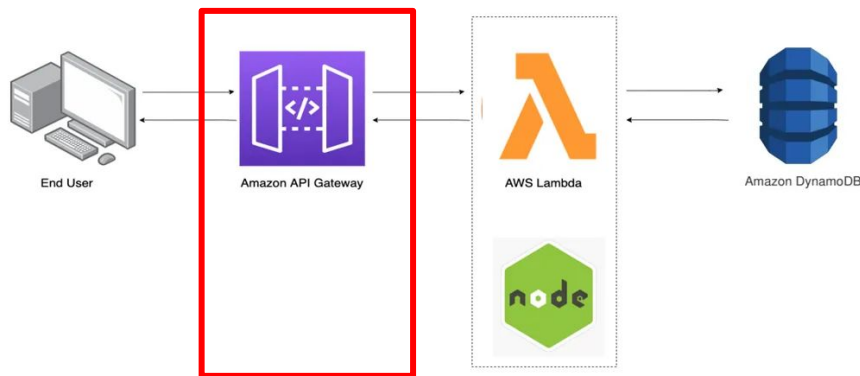
It provides **a way for developers to create RESTful APIs** that can be used to access back-end services or to integrate with other applications.



A **gateway** is a **service that allows different systems to communicate with each other.**

# Amazon API Gateway

For example, **Amazon API Gateway** is a service that **acts as a gateway between your web application and other AWS services**, like Lambda or DynamoDB. It provides a way for your application to send requests to those services and receive responses back.



Think of **gateways** as a middleman that **handles the communication between your application and other services**, allowing you to **focus on building your application** instead of worrying about the technical details of how to connect to these services.



# Amazon Dynamo DB

Introduction to AWS

## Amazon DynamoDB

A **fully managed NoSQL database service** provided by Amazon Web Services (AWS). It provides fast and predictable performance with seamless scalability.



Amazon DynamoDB

**NoSQL** stands for "**not only SQL**" and refers to a type of database that is designed to **handle large volumes of unstructured or semi-structured data**, such as documents, images, and videos. Examples include MongoDB and Amazon DynamoDB, among others.

## Amazon DynamoDB

With DynamoDB, you can **create database collections that can store and retrieve any amount of data and serve any level of request traffic.**

You can scale up or scale down your collection's throughput capacity without downtime or performance degradation.



Amazon DynamoDB

A **fully managed database service** means that **AWS takes care of all the administrative tasks**, such as hardware provisioning, software patching, and maintenance. This is **in contrast to a self-managed or on-premises database**.

With a **self-managed database**, the **responsibility for managing** the underlying infrastructure, including hardware, software, and maintenance, **falls on the user or organization**, rather than being managed by a third-party service provider.

## Differences between NoSQL and Relational Databases

Criteria	NoSQL	Relational
Data Model	<p>Can store all types of data, including things like photos, videos, and social media posts that don't fit neatly into tables.</p> <p>Examples:</p> <ul style="list-style-type: none"><li>• MongoDB</li><li>• Amazon DynamoDB</li><li>• Redis</li></ul>	<p>Good for storing data that fits into tables, like customer names and addresses or inventory levels.</p> <p>Examples:</p> <ul style="list-style-type: none"><li>• Oracle</li><li>• MySQL</li><li>• Microsoft SQL Server</li><li>• PostgreSQL</li></ul>
Data Storage	<p>Examples:</p> <p>(Unstructured)</p> <ul style="list-style-type: none"><li>• Text documents</li><li>• Audio and video</li></ul> <p>(Semi-structured)</p> <ul style="list-style-type: none"><li>• JSON and XML data</li><li>• Email Messages</li></ul>	<p>Examples:</p> <ul style="list-style-type: none"><li>• Financial transactions in a banking application</li><li>• Customer orders and payment information in a retail application</li></ul>

# Amazon S3

Introduction to AWS

## Amazon S3 (Simple Storage Service)

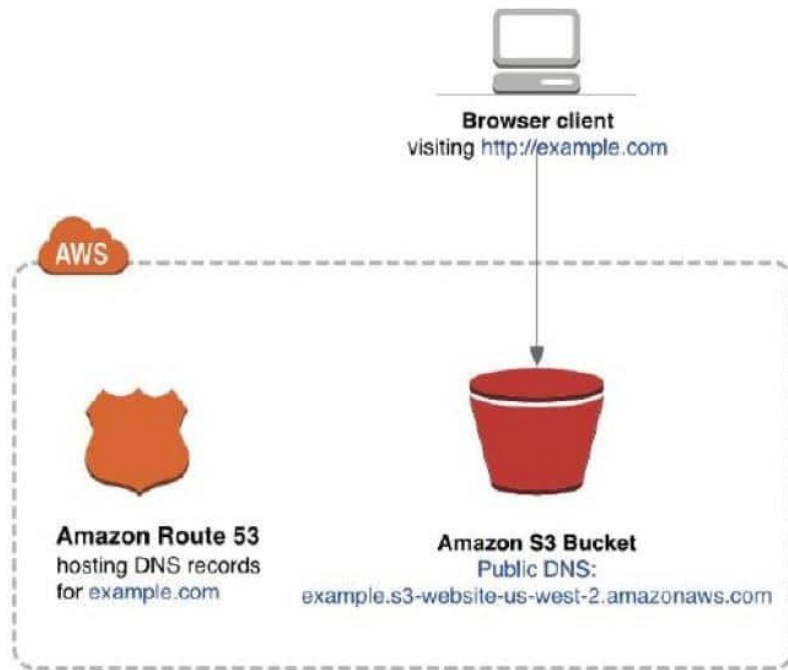
A **cloud-based object storage service** provided by Amazon Web Services (AWS) that **allows you to store and retrieve any amount of data**, at any time, from anywhere on the web.

With S3, you can store and retrieve any type of data, such as text and binary data, images, videos, and even entire websites.





# Amazon S3



# Quiz Form

Introduction to AWS and EC2

# Quiz Form

---

Click on this [link](#) to navigate to the quiz form for the session on Introduction to AWS and EC2.

Make sure to **tick the checkbox and record your email with your response.**