Date: 11.06.2025



Copyright @ 2024 PibyThree.com All Rights Reserved

Amazon Web Services

# 1. AWS and its pricing fundamentals

Amazon Web Services (AWS) is Amazon’s cloud provider service.

AWS has three pricing fundamentals, following the pay-as-you-go pricing model:

1. **Compute** – Pay for compute time
2. **Storage** – Pay for data stored in the cloud
3. **Data Transfer** – Pay for data transferred out of cloud. Data transferred in is free.

# 2. AWS Cloud History

|  |  |
| --- | --- |
| **2002** | Launched internally |
| **2004** | **Simple Queue Service (SQS) –** First AWS service launched for public use. |
| **2006** | **Simple Storage Service (S3) –** Launched in March 2006.  **Elastic Cloud Compute (EC2)** – Launched in August 2026 and is the backbone to everything as it is the most used service till date. |
| **2007** | Launched in Europe |
| **2010** | All of amazon.com’s retail sites had migrated to AWS |
| **2013** | Started offering a certification program for computer engineers to support industry-wide training and skill standardization. |
| **Present** | AWS now offers 200+ services. |

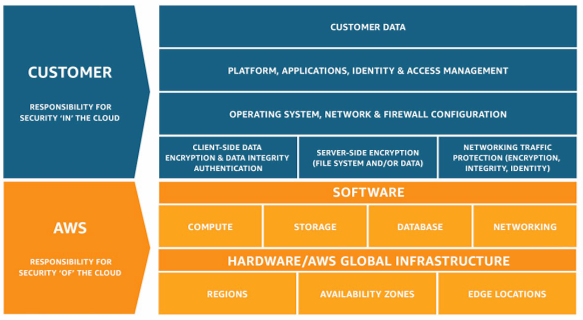
# 3. AWS’ Global Infrastructure

|  |  |
| --- | --- |
| **Region** | * AWS has regions all around the world * A region is a cluster of data centers * Most AWS services are region-scoped * us-east-1, eu-west-3, ap-south-1 |
| **Availability Zone** | * Each region has many availability zones (usually 3, min 3, max 6) ap-southeast-2a, ap-southeast-2b, ap-southeast-2c * Each AZ is one or more discrete data centers with redundant power, networking and connectivity * They’re separate from each other, so they’re isolated from disasters * They’re connected with high bandwidth, ultra-low latency networking |
| **Points of Presence or Edge Locations** | * Edge locations are data centers specifically designed to deliver content to users with minimal latency * Amazon has 400+ Points of Presence (400+ Edge Locations & 10+ Regional Caches) in 90+ cities across 40+ countries |

# 4. AWS Services

|  |  |
| --- | --- |
| **Global Services** | **Region-scoped Services** |
| * Identity and Access Management (IAM) * Route 53 (DNS service) * CloudFront (Content Delivery Network) * WAF (Web Application Firewall) | * Amazon EC2 (Infrastructure as a Service) * Elastic Beanstalk (Platform as a Service) * Lambda (Function as a Service) * Rekognition (Software as a Service) |

# 5. Shared Responsibility Model Diagram



# 6. Creating AWS Account

1. Go to [AWS Console](https://signin.aws.amazon.com/signup?request_type=register) for creating an account.
2. Enter the **root email address** and appropriate **AWS account name**, then verify your email address.
3. Enter your verification code, then click on **Verify.**
4. Enter a strong password for your root user, confirm it, and then choose **Continue**.
5. Choose **Business** or **Personal**. Personal accounts and business accounts have the same features and functions.
6. Enter your company or personal information.
7. Read and accept the [AWS Customer Agreement](https://aws.amazon.com/agreement/).
8. Choose **Continue**. At this point, you'll receive an email message to confirm that your AWS account is ready to use. You can sign in to your new account but you can't use any AWS services until you finish activating your account.
9. Enter the information about your payment method, and then choose **Verify and Continue**.
10. Select one of the available AWS Support plans.
11. Choose **Complete sign up**. A confirmation page appears that indicates that your account is being activated.
12. Check your email and spam folder for an email message that confirms your account was activated. Activation usually takes a few minutes but can sometimes take up to 24 hours.

After you receive the activation message, you have full access to all AWS services.

# 7. Interacting with AWS

## AWS Management Console

* Web Interface
* A user-friendly, web-based graphical interface to interact with AWS services.
* Ideal for manual operations, service monitoring, and quick setup tasks.

## AWS CLI

* Command Line Interface
* A terminal-based tool that lets users control AWS services via public APIs
* Great for automation, scripting, and managing services in bulk

## AWS SDK

* Software Development Kit
* A collection of language specific libraries available in languages like Python, JavaScript, Java, Go, and more to interact with AWS services.
* Ideal for manual operations, service monitoring, and quick setup tasks.

## AWS CloudShell

* Web Interface
* A user-friendly, web-based graphical interface to interact with AWS services.
* Ideal for manual operations, service monitoring, and quick setup tasks.

# 8. Identity and Access Management

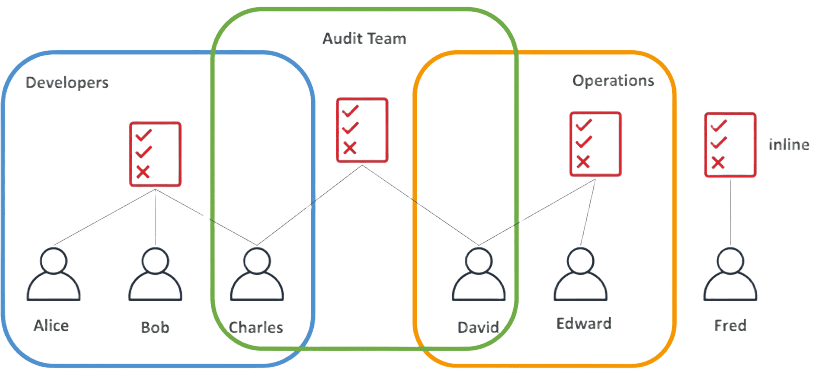
## Users and Groups

* IAM is a global service provided by the Amazon Web Services.
* A **root** account is created by default and shouldn’t be used or shared.
* **Users** are people within your organization, and can be grouped logically.
* **Groups** can only contain users, not other groups.
* A user can belong to multiple groups.
* Users don’t have to belong to a group, but this is not advised.
* Users must be created to interact with and use AWS.



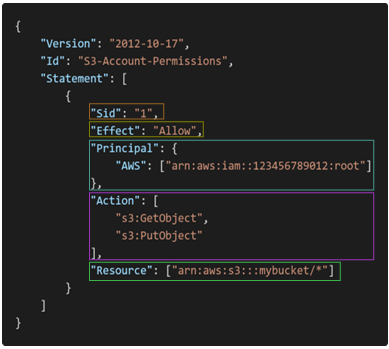
## IAM Permissions

* **Users or Groups** can be assigned **JSON** documents called **policies**.
* These policies define the **permissions** of the users
* In AWS you apply the **least privilege principle**:   
  *Don’t give more permissions than a user needs.*



## Policies Structure

* Consists of
  + Version: policy language version; Recommended to use the latest “2012-10-17”
  + Id: an identifier for the policy (optional)
  + Statement: one or more individual statements (required)
* Statements consists of
  + Sid: an identifier for the statement (optional)
  + Effect: whether the statement allows or denies access (Allow, Deny)
  + Principal: account/user/role to which this policy applied to
  + Action: list of actions this policy allows or denies
  + Resource: list of resources to which the actions applies to
  + Condition: conditions for when this policy is in effect (optional)



## Password Policy

* Stronger passwords == higher security for your account
* In AWS, you can setup a password policy:
  + Set a minimum password length
  + Require specific character types: including
    - uppercase letters
    - lowercase letters
    - numbers
    - non-alphanumeric characters
* Allow all IAM users to change their own passwords
* Require users to change their password after some time (password expiration)
* Prevent password re-use