2.1 AWS Global Infrastructure (Regions, Availability Zones, Edge Locations)

Introduction

Amazon Web Services (AWS) operates one of the **largest cloud infrastructures** in the world. Its global network is designed to provide:

- **High availability** (apps remain online even if one data center fails).
- **Low latency** (services are delivered quickly, closer to end users).
- **Fault tolerance** (multiple backup systems prevent outages).
- **Compliance** with local data regulations (storing data within a country/region if required).

The AWS Global Infrastructure consists of:

1. **Regions**
2. **Availability Zones (AZs)**
3. **Edge Locations**

2.1.1 AWS Regions

Definition

- A **Region** is a **geographical area** where AWS has multiple data centers.
- Each region consists of **at least 2 Availability Zones (AZs)**.
- Each region is **independent** in terms of services and pricing.

Examples of AWS Regions (with codes):

- **Asia Pacific (Mumbai, India)** → ap-south-1
- **US East (N. Virginia)** → us-east-1
- **Europe (Frankfurt, Germany)** → eu-central-1

Why Regions are Important

1. **Latency** – Choosing a region close to users improves performance.
    - Example: Indian customers get faster response when hosted in **Mumbai region** instead of U.S. region.
2. **Compliance** – Some governments require data to stay within national borders.

    - Example: EU data stored in Frankfurt region for GDPR compliance.
3. **Cost** – AWS pricing varies by region.
    - Example: EC2 instance in Mumbai may cost more than in N. Virginia.
4. **Service Availability** – Not every service is available in all regions.
    - Example: New services usually launch first in U.S. regions.

Regions

AWS Global Infrastructure

├— Region: Asia Pacific (Mumbai) [ap-south-1]

├— Region: US East (N. Virginia) [us-east-1]

├— Region: Europe (Frankfurt) [eu-central-1]

...

2.1.2 Availability Zones (AZs)

Definition

- An **Availability Zone (AZ)** is a **physically isolated data center (or cluster of data centers)** within a region.
- Each AZ has **independent power, cooling, and networking**.
- AZs within a region are connected with **low-latency, high-speed links**.

Examples:

- **Mumbai Region (ap-south-1)** has 3 AZs → ap-south-1a, ap-south-1b, ap-south-1c.
- **N. Virginia (us-east-1)** has 6 AZs → us-east-1a, us-east-1b, etc.

Why AZs are Important

1. **High Availability** – Deploy apps across multiple AZs. If one fails, another serves users.
2. **Disaster Recovery** – Data replicated across AZs reduces downtime risk.
3. **Fault Tolerance** – Independent infrastructure prevents a single point of failure.

Example:
Netflix deploys servers across multiple AZs. If one AZ goes down, streaming continues from another.

Region with AZs

Region: Mumbai (ap-south-1)

├── Availability Zone 1: ap-south-1a

├── Availability Zone 2: ap-south-1b

└── Availability Zone 3: ap-south-1c

2.1.3 Edge Locations

Definition

- **Edge Locations** are AWS data centers distributed in many cities to cache content closer to users.
- They are part of **Amazon CloudFront (CDN – Content Delivery Network)**.

Why Edge Locations are Important

1. **Low Latency** – Users get data from the nearest edge location instead of a faraway region.
2. **Improved Performance** – Videos, images, and web files load faster.
3. **Global Reach** – Over **450+ edge locations** worldwide (as of 2025).

Example:
A student in Chennai accessing a U.S.-hosted video may actually get it from an **edge location in Mumbai**, reducing delay.

Use Cases of Edge Locations

- **Content Delivery** – Streaming (Netflix, Hotstar).
- **Web Acceleration** – Websites load faster via CloudFront.
- **Security** – Edge locations integrate with AWS WAF (Web Application Firewall) and Shield (DDoS protection).

Real-World Analogy

Imagine **Amazon delivery network**:

- **Regions** → Amazon's big distribution hubs in different countries.
- **AZs** → Separate warehouses in each city.

- **Edge Locations** → Local pickup points close to neighborhoods.

This way, customers always get products faster, reliably, and from nearby locations.

2.2 AWS Free Tier & Use Cases

Introduction

One of the biggest challenges for students, startups, and small businesses is **cost** when exploring cloud computing. To encourage learning and experimentation, AWS offers a **Free Tier**—a collection of services that are **free to use (with limits)** for a certain period or permanently.

The AWS Free Tier helps:

- **Students** practice AWS without cost.
- **Startups** test new ideas before investing.
- **Developers** prototype apps.
- **Organizations** explore cloud migration risk-free.

Types of AWS Free Tier

The AWS Free Tier has **three categories**:

1. Always Free

- Services that are free **indefinitely**, with limits.
- Examples:
    - **AWS Lambda** – 1 million requests/month.
    - **Amazon DynamoDB** – 25 GB storage.
    - **Amazon CloudWatch** – 10 custom metrics.

2. 12-Month Free

- New customers get **12 months of free usage** from the date of account creation.
- Popular services:
    - **Amazon EC2** – 750 hours/month of t2.micro or t3.micro instance.
    - **Amazon S3** – 5 GB storage.
    - **Amazon RDS** – 750 hours of db.t2.micro usage.

3. Trial Offers

- Short-term free trials for specific services.
- Examples:
    - **Amazon SageMaker** – Free for 2 months.
    - **Amazon Lightsail** – 3 months free trial.

Why AWS Free Tier is Useful

1. **Learn by Doing** – Students can explore EC2, S3, Lambda, etc. without cost.
2. **Prototyping** – Developers can test MVPs (Minimum Viable Products).
3. **Experimentation** – Startups can experiment with databases, AI/ML.
4. **Cost Awareness** – Teaches cloud cost monitoring and budget alerts.

Popular Free Tier Services

Amazon EC2 (Elastic Compute Cloud)

- 750 hours/month (about 1 instance running full-time).
- Instance type: t2.micro or t3.micro (1 vCPU, 1 GB RAM).
- Use Cases: Host a small website, run Linux/Windows server.

Amazon S3 (Simple Storage Service)

- 5 GB standard storage.
- 20,000 GET requests, 2,000 PUT requests.
- Use Cases: Store documents, images, project files.

Amazon RDS (Relational Database Service)

- 750 hours/month of db.t2.micro.
- 20 GB of storage.
- Use Cases: Practice MySQL/PostgreSQL databases.

AWS Lambda

- 1 million requests/month.
- 3.2 million seconds of compute time.
- Use Cases: Run serverless apps, event-driven tasks.

Amazon DynamoDB

- 25 GB storage, 25 read/write capacity units.
- Use Cases: NoSQL database practice, student projects.

Amazon CloudFront

- 50 GB data transfer, 2 million requests/month.
- Use Cases: Distribute images, videos globally.

Cautions While Using Free Tier

- **Monitor usage**: Going beyond free limits → charges.
- **Billing Alerts**: Enable AWS Budgets and Billing Alarms.
- **Shut Down Resources**: Stop/terminate instances when not in use.
- **One-Year Limit**: After 12 months, charges apply for 12-month free services.

Use Cases for Students

1. **Personal Portfolio Website**
    - Host a website using **EC2 + S3 + Route 53**.
2. **Database Project**
    - Use **RDS MySQL** for student records.
3. **Serverless Application**
    - Create a Lambda function to resize images automatically.
4. **IoT Project**
    - Use AWS IoT Core (trial) + DynamoDB to collect sensor data.
5. **AI/ML Mini Project**
    - Try Amazon SageMaker free trial to train a small ML model.

Real-World Startup Stories (Free Tier to Success)

- **Airbnb**: Initially ran experiments on AWS Free Tier before scaling globally.
- **Dropbox**: Used AWS storage services to test their file-sharing app.
- **Zomato (India)**: Early adoption of cloud helped scale rapidly without upfront costs.

Diagram: AWS Free Tier Categories (Text Representation)

AWS Free Tier

├── Always Free

│ └── Lambda, DynamoDB, CloudWatch

├── 12-Month Free

│ └── EC2, S3, RDS

└── Trial Offers

└── SageMaker, Lightsail

2.3 AWS Management Console Walkthrough

Introduction

The **AWS Management Console** is the **primary web-based interface** for interacting with AWS services.
It is designed for:

- **Students and beginners** → Easy-to-use GUI.
- **Developers** → Quick testing of services.
- **Administrators** → Managing resources, monitoring, and billing.

While AWS also offers **CLI (Command Line Interface)** and **SDKs (Software Development Kits)**, the **Console** is the most visual and beginner-friendly way to explore cloud services.

Key Features of AWS Management Console

1. **Web-Based Access**
    - URL: https://console.aws.amazon.com/
    - Accessible from any browser.
2. **Service Search Bar**
    - Search for services (e.g., “EC2”, “S3”, “RDS”) quickly.
3. **Global Navigation Bar**
    - Select AWS **Regions**.
    - Access account settings, billing, and support.
4. **Service Dashboard**
    - Displays categorized AWS services: Compute, Storage, Database, Networking, AI/ML, etc.
5. **Resource Groups**
    - Group multiple services (e.g., EC2 + S3 + RDS for one project).
6. **Tagging Support**
    - Assign tags to resources for cost tracking (e.g., “Project=StudentPortal”).
7. **AWS Cost Explorer & Billing Dashboard**
    - View usage and cost in real time.
8. **IAM Integration**
    - Secure login with different roles (Admin, Developer, Student).

Step-by-Step Walkthrough

Step 1: Login

- Go to: https://console.aws.amazon.com/
- Use AWS account email + password (with MFA if enabled).

Step 2: Explore the Dashboard

- The **Home page** shows recently used services.
- Services are grouped into categories:
    - Compute → EC2, Lambda, ECS
    - Storage → S3, EBS
    - Databases → RDS, DynamoDB
    - Networking → VPC, Route 53
    - Security → IAM, KMS

Step 3: Select a Region

- On the top right, choose a **Region** (e.g., Mumbai ap-south-1).
- Note: Some services are **region-specific** (e.g., EC2, RDS).

Step 4: Launching Services

- Example: **EC2**
    - Click "EC2" → Launch Instance → Choose AMI (OS) → Select instance type → Configure → Launch.
- Example: **S3**
    - Click "S3" → Create Bucket → Name + Region → Upload file.
- Example: **Lambda**
    - Click "Lambda" → Create function → Write code → Test.

Step 5: Monitoring

- **CloudWatch** → Monitor CPU, memory, logs.
- **Cost Explorer** → Check cost trends.
- **Billing Dashboard** → See free tier usage and prevent unexpected bills.

Step 6: IAM (Identity and Access Management)

- Create IAM Users and Groups for students.
- Assign least-privilege permissions.
- Enable MFA for security.

Step 7: Logging Out

- Always sign out when using public/shared computers.

Shortcuts for Beginners

- **Pin Favorite Services** → Add EC2, S3, RDS, Lambda to favorites.
- **Search Instead of Browsing** → Type directly in search bar.
- **Resource Explorer** → See all running resources across services.

AWS CLI vs AWS Console

Feature	**AWS Console (GUI)**	**AWS CLI (Command Line)**
Ease of Use	Beginner-friendly	Advanced users
Speed	Slower (click-based)	Faster (scriptable)
Automation	Manual	Supports automation via scripts
Best For	Students, demos	DevOps, production automation

Common Mistakes

1. **Forgetting to stop instances** → Results in charges after Free Tier limit.
2. **Not setting correct region** → Creates resources in unintended regions.
3. **Confusing IAM roles and users** → Giving too many permissions.
4. **Ignoring billing alerts** → Leads to surprise costs.

Real-World Example

- **Flipkart Dev Team**: Uses AWS Console for quick setup, but automates production using AWS CLI + SDK.
- **Students**: Use AWS Console to practice free tier services like EC2, S3, RDS, before moving to automation.

TASKS

1. **AWS Console Exploration**
    - Log in → Check default region on top-right.
    - List at least 2 other available regions from dropdown.
2. **AZ Identification**
    - Go to **EC2 → Launch Instance (don't launch)**.
    - Note how many AZs are available in your region.
3. **Global Infrastructure Website**
    - Visit: AWS Global Infrastructure
    - Write down 5 regions and their codes.
4. **Edge Location Research**
    - Find the nearest Edge Location to your city.
    - Explain why that improves user experience.
5. **Mini Case Study**
    - *Flipkart wants to serve Indian users.*
    - Which AWS region should they choose? Why not U.S. regions?