Module 01 – Fundamentals

**Table of Content**

[AWS Regions 1](#_Toc198298960)

[AWS Regions 1](#_Toc198298961)

[Availability Zones (AZs) 1](#_Toc198298962)

[AWS EC2 2](#_Toc198298963)

[AWS EC2 (Elastic Compute Cloud) 2](#_Toc198298964)

[Security Groups 2](#_Toc198298965)

[Introduction to Security Groups 2](#_Toc198298966)

[Security Groups: A Closer Look 2](#_Toc198298967)

[Important Things to Know 2](#_Toc198298968)

[Pro Tips 3](#_Toc198298969)

[Elastic IP 3](#_Toc198298970)

[Introduction to Elastic IPs 3](#_Toc198298971)

[Important Things to Know 3](#_Toc198298972)

[Should You Use Elastic IPs? 4](#_Toc198298973)

[EC2 User Data 4](#_Toc198298974)

[Introduction to EC2 User Data 4](#_Toc198298975)

[What Can You Do with User Data? 4](#_Toc198298976)

[Key Points to Remember 4](#_Toc198298977)

[EC2 Instance Launch Types 4](#_Toc198298978)

[On-Demand Instances 5](#_Toc198298979)

[Reserved Instances (RIs) 5](#_Toc198298980)

[Spot Instances 5](#_Toc198298981)

[Dedicated Instances 6](#_Toc198298982)

[Dedicated Hosts 6](#_Toc198298983)

[Spot Fleet 6](#_Toc198298984)

[EC2 Instance Families (By Use Case) 7](#_Toc198298985)

[Burstable Instances (T2/T3) 7](#_Toc198298986)

[Elastic Load Balancers 8](#_Toc198298987)

[Auto Scaling Groups 9](#_Toc198298988)

[Elastic Block Storage 10](#_Toc198298989)

[Elastic File System 11](#_Toc198298990)

[RDS – Relational Database Service 12](#_Toc198298991)

[Amazon Aurora 13](#_Toc198298992)

[ElastiCache 14](#_Toc198298993)

[Amazon S3 15](#_Toc198298994)

[Route S3 16](#_Toc198298995)

# AWS Regions

## AWS Regions

Think of AWS Regions as neighborhoods of data centers spread across the globe. Each Region is a collection of physical locations where AWS builds and maintains its infrastructure. These are labeled with names like us-east-1 (Virginia, USA) or eu-west-1 (Ireland).

Most AWS services are *region-scoped*, which means the data and resources you create in one Region stay in that Region unless you intentionally replicate them elsewhere. So, if you launch a server in us-west-2, it won’t automatically show up in us-east-1—they’re completely independent environments.

## Availability Zones (AZs)

Now, let’s zoom into a Region. Each AWS Region is divided into smaller parts called Availability Zones, or AZs. These are like separate buildings within the same neighborhood. A Region typically has 3 AZs, but it can have anywhere from 2 to 6.

Each AZ consists of one or more physically distinct data centers, each with its own power supply, networking, and connectivity, built to be highly reliable.

Even though AZs within a Region are geographically separated—to protect against issues like power outages or natural disasters—they’re also closely connected through high-speed, low-latency networking. This means you can design systems that are both resilient and fast, using multiple AZs within the same Region.

# AWS EC2

## AWS EC2 (Elastic Compute Cloud)

AWS EC2 is one of the core services in the AWS ecosystem. It allows you to run virtual machines in the cloud, giving you flexible computing power without needing to buy physical hardware.

Here are the main things you can do with EC2:

* **Rent Virtual Machines (EC2 Instances)**: You can launch servers—called instances—on demand. These act like traditional computers, but they run in the cloud and can be started, stopped, resized, or terminated whenever you need.
* **Store Data on Virtual Drives (EBS)**: EC2 works with Elastic Block Store (EBS) to provide storage. Think of EBS volumes like external hard drives that attach to your EC2 instances, allowing you to store data persistently—even after the instance shuts down.
* **Distribute Traffic Across Multiple Machines (ELB)**: With Elastic Load Balancing (ELB), you can automatically spread incoming traffic across several EC2 instances. This helps your applications stay responsive and available, even during high demand.
* **Automatically Scale with Demand (Auto Scaling Group - ASG)**: An Auto Scaling Group (ASG) monitors your instances and automatically adds or removes them based on traffic or other conditions. This way, your system can handle traffic spikes and save costs during low activity.

## Security Groups

### Introduction to Security Groups

Security Groups are a core part of network security in AWS. You can think of them as virtual firewalls that protect your EC2 instances. They control what kind of traffic is allowed in or out of your cloud servers.

In simpler terms: they help answer the question, *"Who can talk to my EC2 instance, and on which port?"*

### Security Groups: A Closer Look

Security Groups work by setting rules that define how network traffic flows to and from your EC2 instances. Here's what they do:

* **Control Access to Specific Ports:** For example, port 22 for SSH, port 80 for HTTP, or port 443 for HTTPS.
* **Allow or Deny IP Ranges:** You can specify who gets access using IPv4 or IPv6 address ranges.
* **Manage Inbound and Outbound Traffic:** Decide what traffic can come in (inbound) or go out (outbound) of your EC2 instance.

### Important Things to Know

* **Reusable Across Instances:** One security group can be attached to multiple EC2 instances, so you don’t need to create a new one for every server.
* **Scoped to Region and VPC:** Security Groups are limited to a specific Region and Virtual Private Cloud (VPC). You can’t share them across regions.
* **External to EC2:** Security Groups operate outside the EC2 instance. If a request is blocked, the EC2 instance won’t even know it happened—like a bouncer turning someone away at the door.

### Pro Tips

* **Create a Separate SG for SSH:** It’s a good practice to isolate SSH access in its own security group. This makes it easier to manage and audit access.
* **Diagnosing Access Issues:**
* If your application request times out, it’s likely that traffic is being blocked by a Security Group rule.
* If you get a “connection refused” error, it usually means the traffic made it through, but the application itself isn't responding correctly.
* **Default Rules:**
* By default, all inbound traffic is denied.
* All outbound traffic is allowed—unless you explicitly block it.
* **Referencing Other Security Groups:** You don’t have to use IP addresses. A security group can allow traffic from another security group directly, which is especially helpful when working with multiple tiers (like web and database servers).

## Elastic IP

### Introduction to Elastic IPs

When you launch an EC2 instance, AWS assigns it a public IP address so it can communicate with the internet. But here’s the catch: if you stop and restart that instance, it will likely get a new public IP. This can be a problem if you need the instance to always be reachable at the same IP address.

That’s where Elastic IPs come in.

An Elastic IP is a static public IP address that you can own and control within your AWS account. Once you allocate an Elastic IP, it’s yours until you release (delete) it. You can associate it with any EC2 instance in your account, and even move it between instances if one fails—helping you quickly recover from outages.

Think of it as your permanent street address in the cloud, even if you change the house (the EC2 instance) it points to.

### Important Things to Know

* AWS gives you a soft limit of 5 Elastic IPs per region by default.
* Elastic IPs are not free if they’re allocated but not actively used (i.e., not associated with a running instance). AWS encourages efficient use of resources.

### Should You Use Elastic IPs?

In most cases, it's better to avoid Elastic IPs unless you have a specific need for a static IP. Here's why:

* They can indicate a rigid or outdated architecture, especially in modern, scalable cloud environments.
* A better practice is to use a random public IP for your EC2 instance and point a DNS name (like a domain name) to it. This way, even if the IP changes, you can just update the DNS record—keeping your setup more flexible and cloud-native.

## EC2 User Data

### Introduction to EC2 User Data

When you launch an EC2 instance, you often want it to be ready to go—without having to log in and set things up manually. That’s where EC2 User Data comes in.

User Data is a way to automate the initial setup of your instance by running a script at launch.

### What Can You Do with User Data?

You can use a User Data script to perform common setup tasks, such as:

* Installing system updates
* Installing necessary software (like web servers or databases)
* Downloading files from the internet
* Running custom startup commands—basically, anything you’d normally do right after logging in

These scripts run automatically when the instance starts for the first time, and they’re executed with root (administrator) privileges, so they can make system-level changes.

### Key Points to Remember

* **Runs Only Once:** The User Data script runs only during the first boot of the instance (unless you explicitly configure it to run every time).
* **Great for Automation:** It helps you save time, avoid manual errors, and create consistent setups for multiple instances.
* **Ideal for Launching Pre-Configured Servers:** For example, you can spin up a web server that’s fully installed and ready to serve traffic in just a few minutes.

## EC2 Instance Launch Types

When launching EC2 instances, AWS offers several pricing and deployment options—each suited for different use cases. Here's an overview of the main types:

### On-Demand Instances

* **Pay-per-use model**: You’re billed by the second after the first minute.
* No upfront payment required.
* **Best for:** Short-term, unpredictable workloads or development and testing environments.
* **Pros:** Flexibility and no commitment.
* **Cons:** Highest cost compared to other options.

### Reserved Instances (RIs)

Reserved Instances offer significant savings in exchange for committing to a specific instance type over a long term.

**Types of Reserved Instances:**

* **Standard Reserved Instances**
  + Save up to 75% compared to On-Demand.
  + Fixed commitment for 1 or 3 years.
  + Recommended for: Steady, long-term workloads (e.g., databases, production apps).
* **Convertible Reserved Instances**
  + Save up to 54%.
  + Allows changing instance types, OS, or tenancy.
  + Best for: Long-term workloads that may change over time.
* **Scheduled Reserved Instances** *(less common)*
  + Run during specific time windows (e.g., 9 AM–5 PM weekdays).
  + Useful when workloads only run at scheduled times.

### Spot Instances

* Get up to 90% discount compared to On-Demand.
* AWS uses unused capacity, but instances can be interrupted at any time.
* **Best for:** Fault-tolerant and stateless workloads (e.g., batch jobs, data analysis).
* **Not recommended for:** Critical applications or databases.
* **Max Price:** You define the maximum price you're willing to pay.
* **Termination Notice:** AWS gives a 2-minute warning if your instance is being reclaimed.
* **Spot Block:** Lock a Spot instance for 1–6 hours without interruptions (rarely reclaimed).
* **Spot Request:** Specify pricing, instance type, number of instances, and request type:
  + **One-time:** Fills once, then disappears.
  + **Persistent:** Keeps trying to maintain your desired capacity.
* **Canceling Spot Requests:** Cancel the request before terminating the instance, or it may relaunch.

### Dedicated Instances

* EC2 instances that run on hardware dedicated to your AWS account.
* Other AWS customers won’t share the same physical host, but instances from your account can.
* Per-instance billing, but no control over where instances are placed.

### Dedicated Hosts

* You get an entire physical server.
* Allows full visibility into the underlying hardware, including sockets and cores.
* Required for software licenses tied to physical hardware (e.g., Oracle, Microsoft).
* 3-year reservation required.
* Best for: Compliance-heavy environments and software with strict licensing rules.

## Spot Fleet

A Spot Fleet is a group of Spot Instances (with optional On-Demand instances) that AWS uses to meet your desired capacity at the lowest possible cost.

* AWS selects from multiple launch pools (combinations of instance types, OS, and Availability Zones).
* You define:
  + Capacity goal
  + Pricing constraints
  + Instance configurations

**Allocation Strategies:**

* **Lowest Price**: Choose instances from the cheapest pool.
* **Diversified**: Spread instances across multiple pools for redundancy.
* **Capacity Optimized**: Focus on pools with more available capacity to reduce interruptions.

## EC2 Instance Families (By Use Case)

|  |  |  |
| --- | --- | --- |
| **Type** | **Best For** | **Example Use Case** |
| R | Memory-intensive | In-memory databases (e.g., Redis) |
| C | Compute-optimized | High-performance compute, CPU-bound tasks |
| M | General purpose | Balanced CPU and memory – ideal for web apps |
| I | Storage-optimized | High local storage throughput – databases |
| G | GPU-based | Machine learning, video rendering |
| T2/T3 | Burstable performance | Spiky workloads like development environments |

## Burstable Instances (T2/T3)

Burstable instances offer baseline performance with the ability to temporarily burst to higher CPU power when needed.

* Use CPU credits to burst.
* Earn credits during idle time; spend credits during bursts.
* If credits run out, performance drops to the baseline.
* Monitor usage with CloudWatch.

**T2/T3 Unlimited**

* Allows unlimited bursting without being throttled, even if you run out of credits.
* You’ll be charged extra for CPU usage beyond earned credits.
* Ideal for unpredictable workloads where performance consistency is important.

# Elastic Load Balancers

…

# Auto Scaling Groups

…

# Elastic Block Storage

…

# Elastic File System

…

# RDS – Relational Database Service

…

# Amazon Aurora

…

# ElastiCache

…

# Amazon S3

…

# Route 53

## Part I

### Route 53

Amazon Route 53 is a managed DNS (Domain Name System) service provided by AWS. If you think of the internet as a giant phone book, DNS is what helps computers look up phone numbers (IP addresses) for domain names like example.com.

### Domain Name Service

DNS translates domain names into IP addresses so that your browser can find and connect to the right server. Route 53 manages these DNS records for you. In AWS, you’ll commonly work with the following types of DNS records:

* **A record**: Maps a domain name to anIPv4 address
* **AAAA record**: Maps a domain name to an IPv6 address
* **CNAME record**: Points one domain name to another domain name
* **Alias record**: Like a CNAME, but specifically designed to map a domain to AWS resources like CloudFront distributions, S3 buckets, or ELBs

### Public Domain vs Private Domains

Route 53 can manage both:

* Public domain names (like mywebsite.com) that you own
* Private domain names that are only accessible within your VPC (Virtual Private Cloud) — for example, internal services in your application that EC2 instances can resolve

### Route 53 Features

Route 53 isn’t just about directing traffic—it can make smart decisions about *how* to direct that traffic. Some of its advanced capabilities include:

* **DNS-based load balancing**: Also called **client-side load balancing**, where the DNS response provides multiple IPs and lets the client pick one
* **Health checks**: Automatically monitor your resources and redirect traffic if something goes down
* **Routing policies**: Choose how to route users with options like:
* **Simple**: One record, one response
* **Failover**: Redirect traffic if a resource becomes unhealthy
* **Geolocation**: Send users to resources based on their location
* **Latency-based**: Route to the fastest responding region
* **Weighted**: Distribute traffic in specific proportions
* **Multi-value**: Return multiple healthy IPs and let the client decide

### TTL

…