### **🔹 What is an *instance* in AWS?**

An **instance** in AWS is like a **virtual computer** that runs in the cloud.

* 🖥️ Just like a computer has a CPU, memory (RAM), and storage, an **AWS instance** has those too—but instead of being on your desk, it's on a server inside Amazon's data center.
* 💡 You can use it to run websites, apps, databases, or anything you’d normally do on a regular computer.

Perfect! You're asking about **EC2 Tenancy types** — specifically:

* **Shared Instances**
* **Dedicated Instances**
* **Dedicated Hosts**

Let me explain each one **very simply**, then tell you what’s best for your **eCommerce** project.

### **🧩 1. Shared Instances (Default — most common ✅)**

* Your server runs on the **same physical machine** as other AWS customers.
* You still get your **own private virtual server** (no one sees your data).
* ✅ **Cheapest** and good performance.
* 👌 Ideal for: **Startups, small-medium apps**, most general usage.

### **🏠 2. Dedicated Instances**

* You get your **own physical server**, but **only the instance level** is isolated.
* Costs **more** than Shared.
* Used if you have **security/compliance rules** or need **better isolation**.

### **🏢 3. Dedicated Hosts**

* You rent the **entire physical server**.
* You can see and control the **hardware layout** (licensing, CPU control).
* 🟥 Very expensive.
* Needed for **special compliance needs** or if you bring your own licenses (BYOL).

## **✅ Which One for eCommerce?**

| **Tenancy Type** | **Cost** | **Security** | **Best for eCommerce?** |
| --- | --- | --- | --- |
| Shared Instance | 💸 Low | ✅ Good | ✅✅✅ Yes (Recommended) |
| Dedicated Instance | 💰 Medium | ✅✅ Higher | ❌ Only if needed |
| Dedicated Host | 🟥 High | 🛡️ Highest | ❌ Overkill |

✅ **Recommendation:** Go with **Shared Instances** — perfect for most eCommerce sites unless you're in **finance/healthcare with strict data rules**.

Sure! Let’s break this image down **very simply** — you're looking at **Amazon EC2 pricing options** for the instance type **t3.small** using Linux. These are **ways to pay** for using a cloud server (EC2 = virtual computer). Here's what each section means:

## **💡 1. On-Demand**

* **What it is:** Pay **only when you use** it. No long-term commitment.
* **Price shown:**
  + 0.022 USD/hour
  + ~16.06 USD/month if used 100% (24/7).
* **Best for:** Testing, learning, or short-term projects.

## **💸 2. Spot Instances**

* **What it is:** Cheapest option. AWS gives you **unused servers**.
* **BUT:** AWS **can stop it anytime**, so not reliable for critical work.
* **Price:** Around 0.022/hour, but this **fluctuates**.
* **Best for:** Experiments, batch jobs, or apps that can handle interruptions.

## **🛡️ 3. Reserved Instances**

* **What it is:** You **reserve** a server for **1 or 3 years**, and you get a **discount**.
* You choose:  
  + ✅ 1 year or 3 year plan
  + ✅ Pay all now, some now, or monthly

### **🧮 Example in the Image:**

* **3-Year Plan with Partial Upfront**:  
  + Pay $142 upfront
  + Then $3.94/month
  + Works out to ~$8.88/month total
* **Why do it?** Cheaper if you know you'll run the server for a long time.

### **📊 Summary for You:**

| **Option** | **Monthly Cost (approx.)** | **Can stop anytime?** | **Cheap?** | **Good for** |
| --- | --- | --- | --- | --- |
| On-Demand | $16.06 | ✅ Yes | ❌ Normal | Short-term use, testing |
| Spot Instance | ~$16.22 (varies) | 🚫 No (can stop) | ✅ Cheapest | Flexible/backup tasks |
| Reserved (3 yr) | ~$8.88 | 🚫 No | ✅ Very good | Long-term use (cheapest) |

## **🗺️ What is an "AZ"?**

**AZ = Availability Zone** It's a **data center** inside a region (like us-east-1 in Virginia, USA).  
 Each AWS region has **multiple AZs** — isolated buildings with power/network redundancy.

## **🌐 Deployment Options:**

### **✅ 1. Single-AZ**

* Your database is hosted in **just one data center (AZ)**
* ✔️ **Cheaper**
* ❌ If that AZ goes down (rare), your DB will also go down

👉 Best for **non-critical, dev, small apps**

### **🔁 2. Multi-AZ**

* Your database is **replicated across 2 AZs**
  + One is **primary** (active)
  + One is **standby** (passive copy)
* ✅ Automatic failover if one AZ fails
* ✅ High availability + durability
* ❌ Costs more (because you're using 2 DB instances)

👉 Best for **production apps**, eCommerce, anything that needs uptime

### **📊 Example Pricing Impact:**

| **Type** | **Uptime** | **Cost** | **Use Case** |
| --- | --- | --- | --- |
| Single-AZ | ~99.5% | 💸 Lower | Small/demo apps |
| Multi-AZ | ~99.99% | 💰 Higher | Real users/live app |