

**🔁 What Is Happening in This Diagram?**

It shows **how incoming internet traffic** (from users) flows through the AWS **Elastic Load Balancer**, which distributes it to multiple EC2 servers (instances) across different **Availability Zones** to ensure **high availability** and **fault tolerance**.

**📌 Step-by-Step Breakdown**

**1. 👤 User / Client**

* This is the **person using your application** (like a website).
* They make a request using **HTTP, HTTPS, TCP, or SSL**.

**2. 🖥️ Elastic Load Balancer (ELB)**

* This is the **traffic manager**.
* It **receives the request from the user** and **distributes it** to healthy EC2 instances.
* It checks which server is healthy and less busy, and **forwards the request there**.

✅ This helps in **load distribution**, **failover**, and **better performance**.

**3. 🧱 Availability Zones (#1 and #2)**

* AWS data centers are divided into **Availability Zones (AZs)**.
* Each AZ has multiple **EC2 Instances** (your servers).
* ELB is **multi-AZ aware**, meaning if one AZ goes down, the ELB will use instances in the other.

✅ This gives **high availability**.

**4. 📈 CloudWatch**

* AWS **CloudWatch** monitors the ELB and your instances.
* It collects logs, performance metrics, health checks, etc.

✅ Useful for **troubleshooting, alerting**, and **scaling decisions**.

**5. 🔁 Auto Scaling**

* Automatically **adds or removes instances** based on traffic load.
* Works with ELB to always have the right number of servers.

✅ Saves **costs** and improves **efficiency**.

**6. 👤 ELB Owner / IAM User**

* This is **you (admin)** or a developer managing the ELB.
* You use either:
  + AWS Console (GUI)
  + CLI (Command Line)
  + API calls
* You can change load balancer settings, SSL certs, health checks, etc.

**7. 🌐 AWS Management Console**

* Web-based dashboard where you manage everything:
  + EC2, ELB, Route 53, S3, IAM, etc.

**8. 🌍 AWS DNS + Route 53 (Optional)**

* You can **connect a domain name** (like www.myapp.com) to your ELB using **Route 53** (AWS DNS service).
* The user types the domain in the browser, and it resolves to the ELB’s address.

✅ This makes your app **accessible via a friendly domain name**.

**🔄 Line Colors in the Diagram**

* **Black Solid Lines:** Represent the **actual data flow / traffic** (HTTP, HTTPS, etc.).
* **Orange Dashed Lines:** Represent **control, settings, or monitoring operations**, like:
  + Admin managing ELB
  + ELB checking CloudWatch
  + ELB settings being changed by an IAM user

**✅ Final Summary**

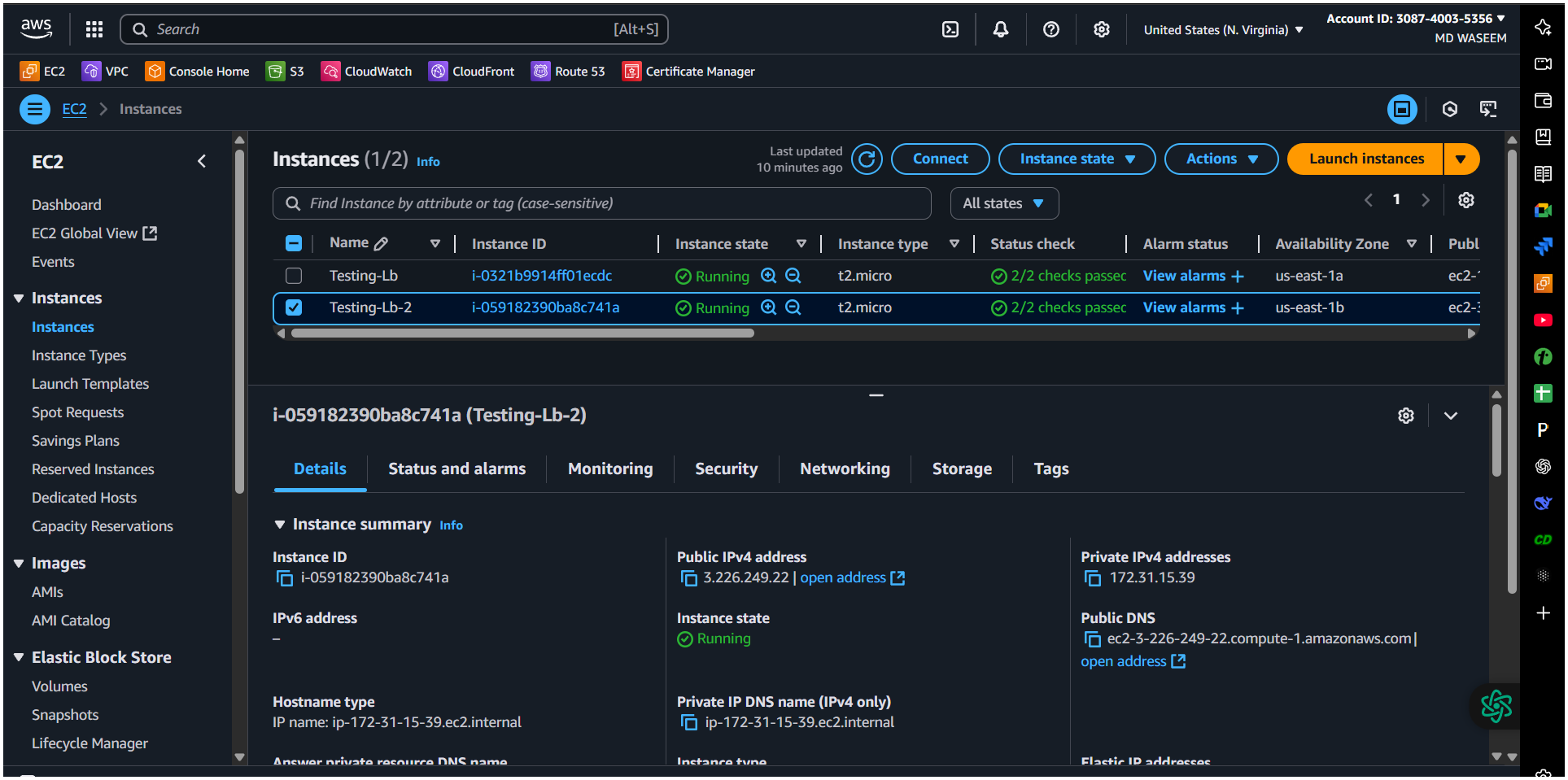
| **Component** | **Role** |
| --- | --- |
| User | Sends request (like opening a website) |
| ELB | Distributes the request to backend servers |
| Instances | EC2 servers that respond to the request |
| CloudWatch | Monitors logs, traffic, and health |
| Auto Scaling | Adds/removes servers automatically |
| Route 53 | Maps domain name to the ELB |
| IAM/Admin | You, managing the system settings |

**1) Configure Classic Load balancer.  
  
Configure Classic Load Balancer (CLB)**

**What to do:**

* **In the AWS Console, go to Load Balancers → Create Load Balancer → choose Classic Load Balancer.**
* **Give it a unique name, choose Internet-facing, add at least two subnets (one per availability zone).**
* **Attach a security group that allows HTTP (port 80) and, if needed, HTTPS (port 443).**
* **Add listeners: typically HTTP → HTTP on port 80.**
* **Register your EC2 instances as targets.**

**Why:  
Classic Load Balancer is the older version (Layer 4/7) that simple checks traffic and sends it to healthy servers to prevent overloading, improving availability**

**  
🡪Here we have create two instances to add in the Load balancer**

**🡪Here we have testing the instances is running or not we got this message / output as we have enter the user data in the instances setting**

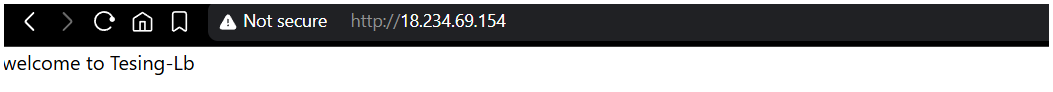
**#!/bin/bash**

**sudo yum -y install httpd**

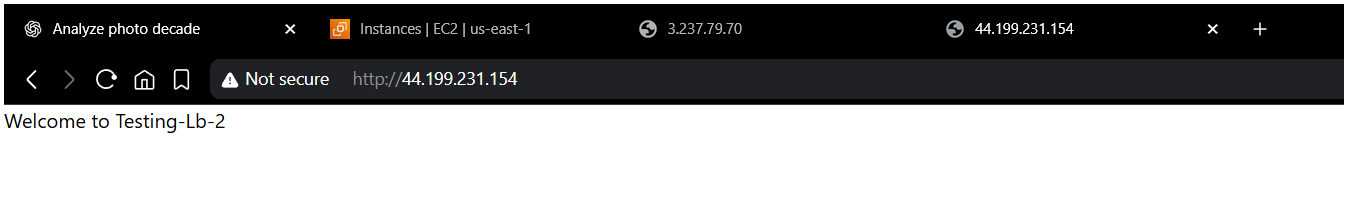
**echo "Welcome to Testing-Lb" > /var/www/html/index.html**

**sudo systemctl start httpd**

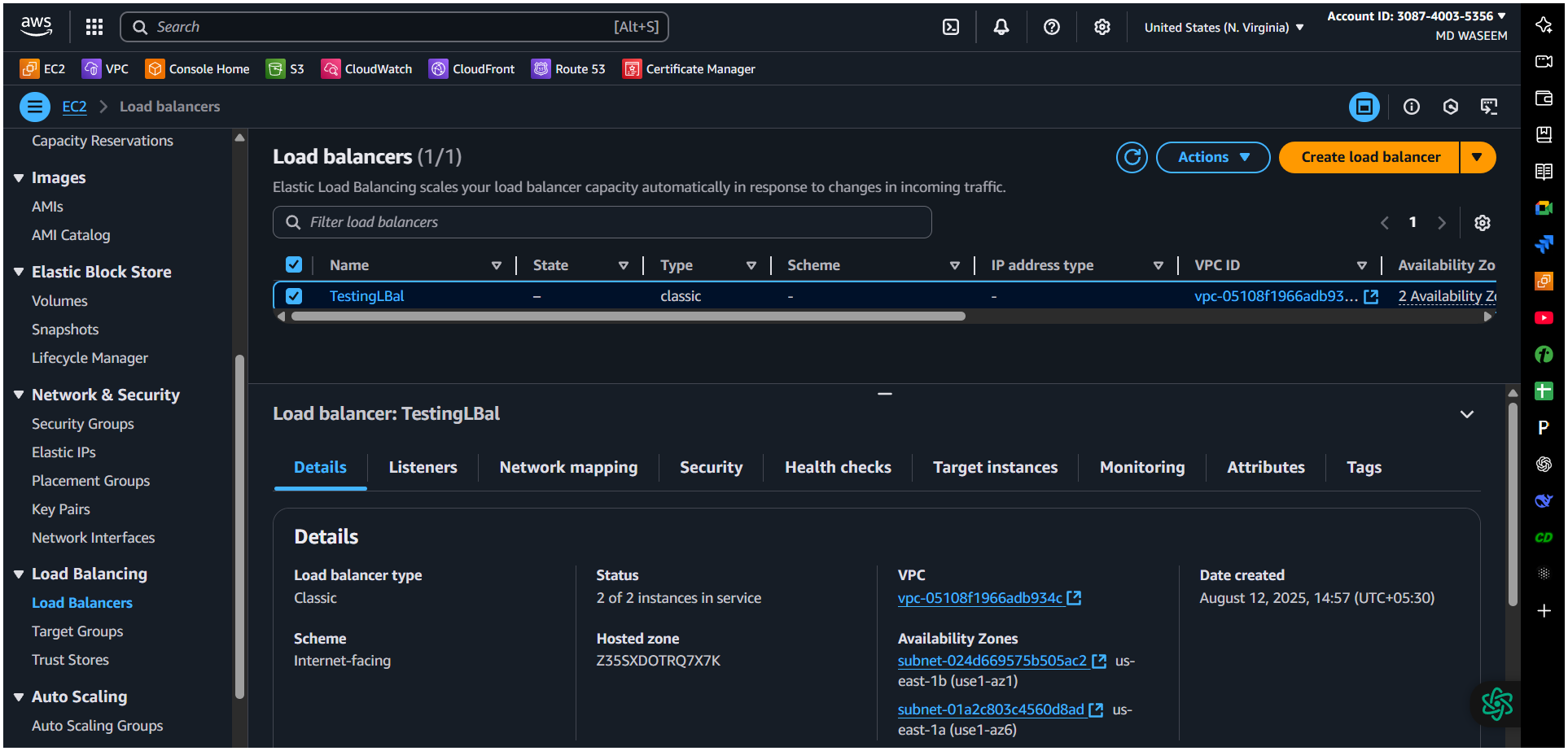
**sudo systemctl enable httpd**

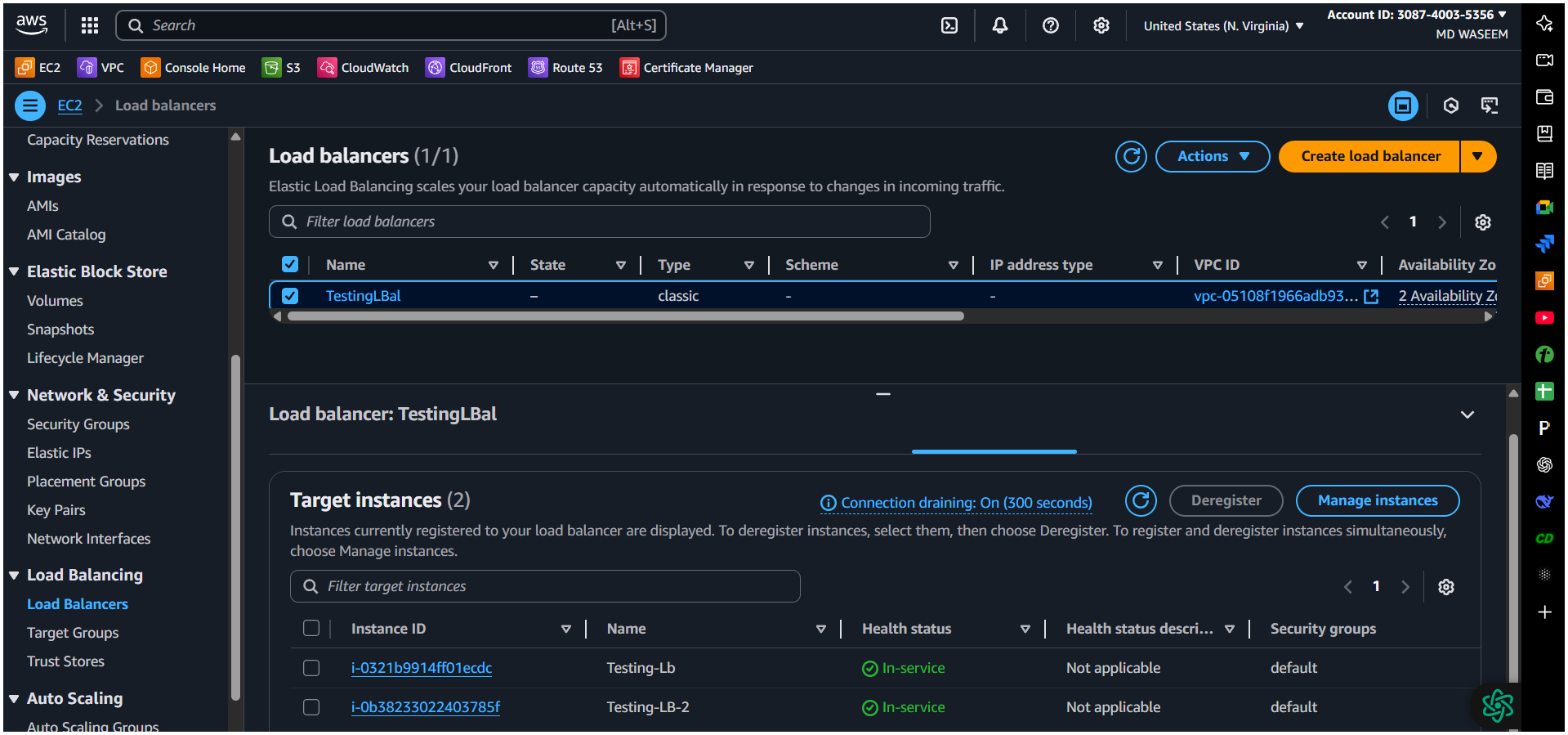
**🡪1st instances testing Running✅  
**

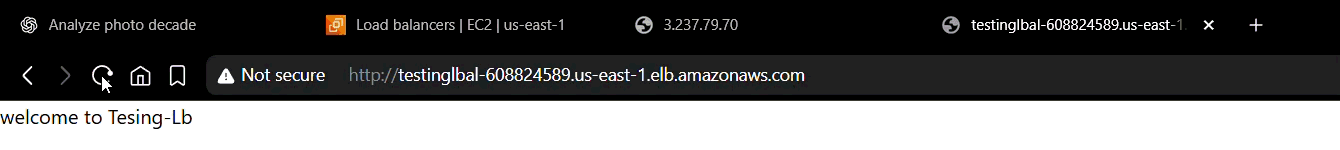
**🡪2nd instances testing Running ✅**

****

* **Here I have Created the Load Balancer and add the instances**

****

**🡪Target Instances is working ✅  
**

**🡪Result of both instances are working properly**

**Why these settings / how they help**

* **Multiple AZs / subnets** → high availability if an AZ fails.
* **Health checks** ensure CLB only sends traffic to healthy instances (prevents user errors and reduces 5xx).
* **Connection draining / deregistration** helps in-flight requests finish when you remove instances.
* **Cross-zone load balancing** evens traffic across AZs so one AZ with more instances doesn’t become underutilized

**2) Configure Application Load balancer.**

**Step-by-Step: Create ALB in AWS Console**

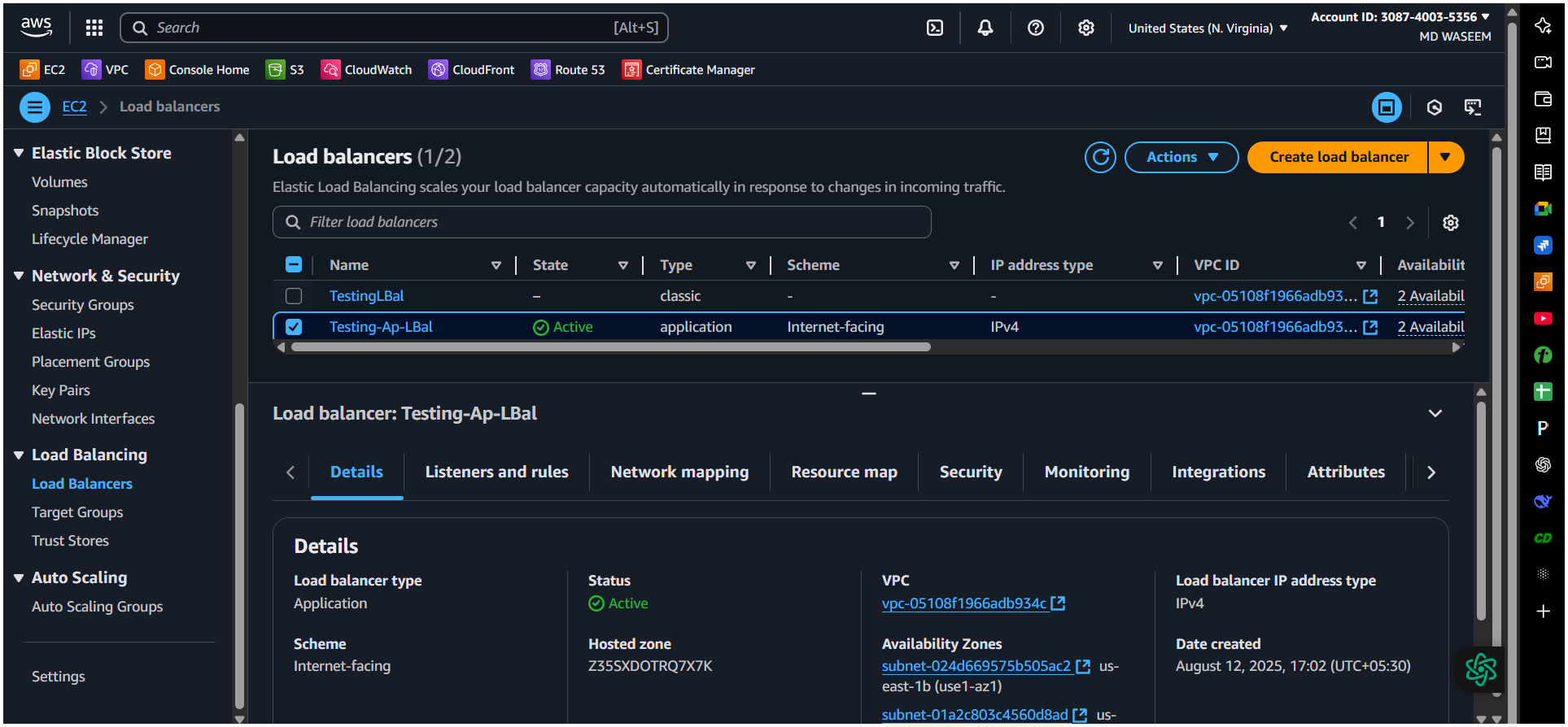
**1️⃣ Create a Target Group (the “back-end” list)**

We make the target group first because the ALB will need it when we set up listeners.

1. Go to **AWS Console** → **EC2** service.
2. In the left menu, click **Target Groups** → **Create target group**.
3. **Target type** = Instance (most common).
4. **Target group name** = my-web-tg (any name).
5. **Protocol** = HTTP, **Port** = 80 (or 443 if your servers use HTTPS).
6. **VPC** = choose your VPC where your servers live.
7. **Health checks**:
   * Protocol = HTTP
   * Path = / (or /health if your app has one)
8. Click **Next**.
9. Select the EC2 instances you want to attach → **Include as pending**.
10. Click **Create target group**.

### **2️⃣ Create the ALB**

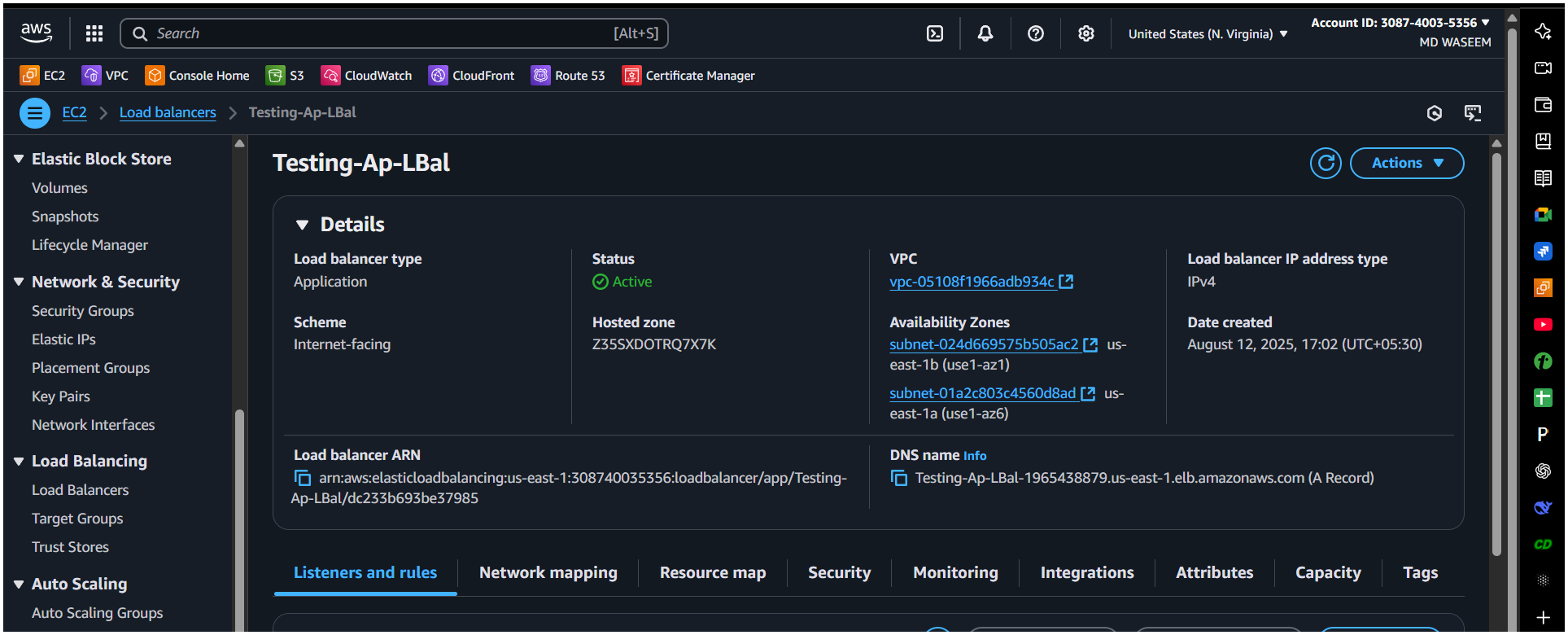
1. In the left menu, click **Load Balancers** → **Create load balancer**.
2. Choose **Application Load Balancer** → **Create**.
3. **Name**: my-alb (any name).
4. **Scheme**:
   * Internet-facing (if public website)
   * Internal (if only inside your VPC)
5. **IP address type** = IPv4.
6. **VPC** = same as your target group’s VPC.
7. **Availability Zones**:
   * Select at least **two** subnets in different AZs (for high availability).

****

### **4️⃣ Add Listener and Forwarding Rule**

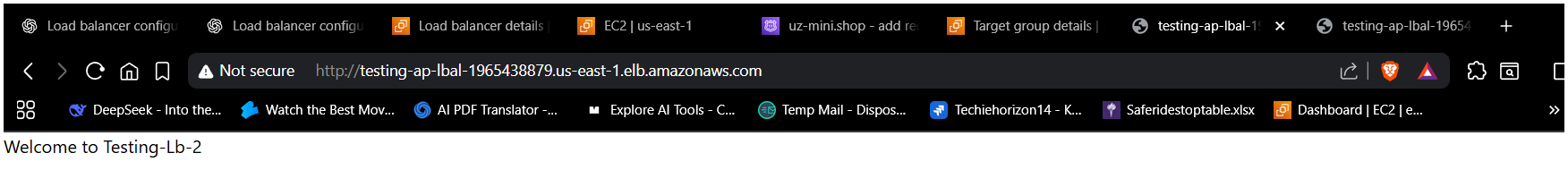
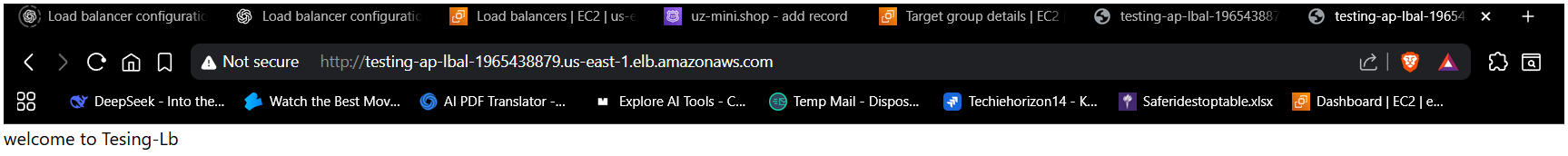
1. Under **Listeners**:
   * Protocol: HTTP, Port: 80
   * Forward to: select your **target group** created earlier.
2. You can add HTTPS listener later when you attach an SSL certificate.

**5️⃣ Review & Create**

1. Review all settings.Click **Create load balancer**.  
   Wait until status changes to **Active**.

**6️⃣ Test**

1. Copy the **DNS name** of your ALB from the console.
2. Paste in browser → should show your app from the EC2 targets.

****

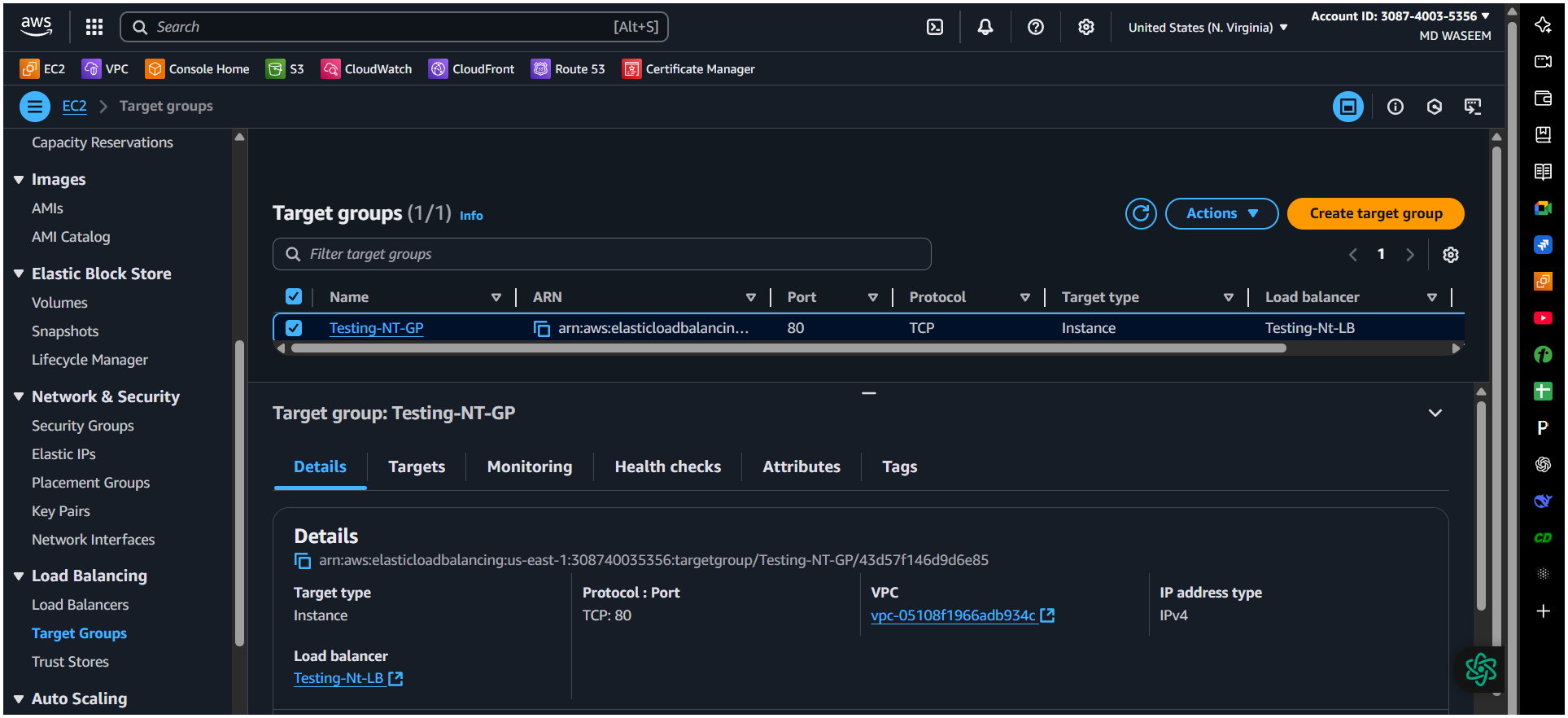
**3) Configure Network Load balancer.**

**What an NLB does (short)**

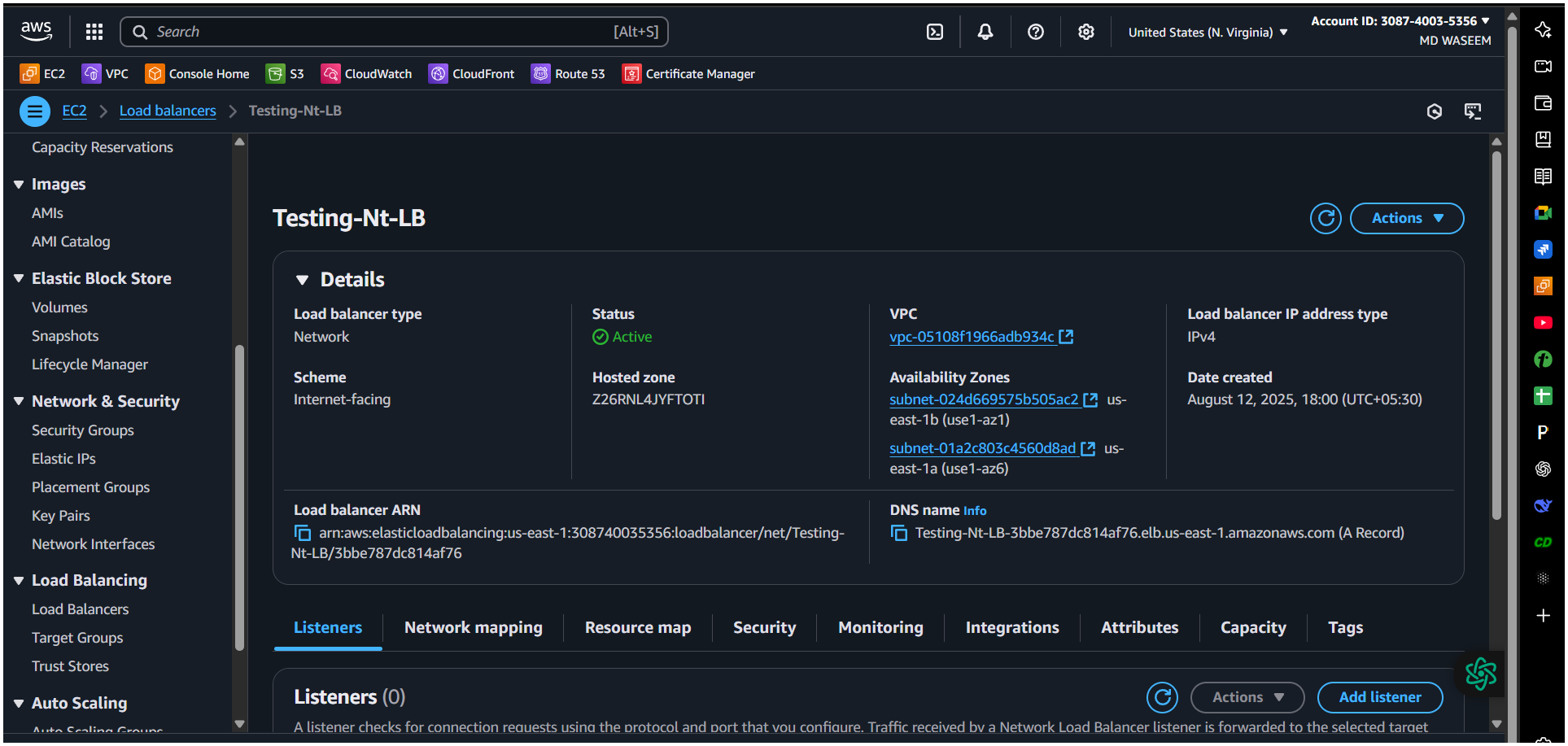
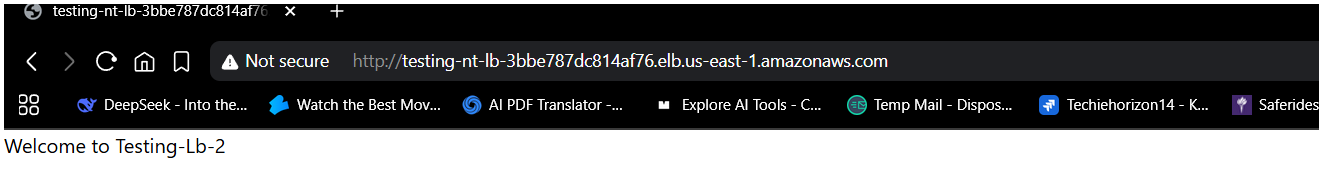
An NLB is an ultra-fast Layer-4 load balancer (TCP/UDP) that can handle very high throughput and preserve the client’s source IP. It can also provide one static IP per AZ (you can optionally attach Elastic IPs). Use NLB when you need very low latency, high throughput, TCP/UDP load balancing, source-IP preservation, or static IPs for firewall whitelisting

**Before you start (prereqs)**

* VPC with public subnets (for internet-facing) and an Internet Gateway.
* EC2 targets (or IP targets) running the service (e.g., a TCP server on port 80/443).
* Elastic IPs (optional) if you want fixed public IPs per AZ.
* Security groups / NACLs that allow traffic to the target instance ports.  
  (If you’ll preserve client IP, your targets must permit client source IPs in their SGs.)

**🡪Created Target Group (added instances in it)  
**

* **Added the Network - Load Balancer**

**  
  
  
  
  
🡪 Tested and it working ✅**

**Notes & gotchas (short)**

* **Cross-zone load balancing is disabled by default** for NLBs; enable it if you need even distribution across AZs. Enabling may change performance and costs. [AWS Documentation](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/network-load-balancers.html?utm_source=chatgpt.com)
* **Client IP preservation vs proxy protocol:** if you need the real client IP in your app logs, choose preservation (if supported) or enable Proxy Protocol v2 and parse it on the server. [AWS Documentation](https://docs.aws.amazon.com/elasticloadbalancing/latest/network/edit-target-group-attributes.html?utm_source=chatgpt.com)
* **Security groups for NLB:** as of 2023 AWS added support to attach SGs to NLBs — you can use that to centrally filter allowed clients. If you create the NLB without SGs you may not be able to attach them later.

**4) Attach SSL for application load balancer.**

**✅ Checklist to Resolve ACM Certificate Stuck in “Pending Validation”**

**1. Check if DNS Records Were Created Successfully**

* Go to **ACM > Certificates**, click your certificate.
* In the **Domains** section, check the **Validation status**.
* It will show something like “Pending validation” and tell you the required **CNAME record**.
* Make sure this CNAME record exists in **Route 53 (or your DNS provider)**.

**2. If Using Route 53:**

* Did you click **“Create records in Route 53”** when requesting the certificate?
  + If **yes**: Go to **Route 53 > Hosted Zones > Your domain**, and confirm the CNAME record exists.
  + If **no**: Go back to ACM, and click **“Create records in Route 53”** now.
* Ensure you're modifying the correct **hosted zone** for the domain (e.g. example.com).

**3. If Using an External DNS Provider:**

* You must **manually copy and paste** the CNAME validation record from ACM into your DNS settings.
* Be very careful with:
  + **CNAME name** (should not include the domain again if your DNS provider auto-appends it)
  + **No typos**, no extra spaces
  + **TTL** can be left as default (e.g., 300)

**4. Wait for DNS Propagation**

* It may take **a few minutes to several hours** (usually under 30 minutes).
* Use tools like https://dnschecker.org to check if the CNAME record is globally visible.

**5. Domain Spelling and Wildcards**

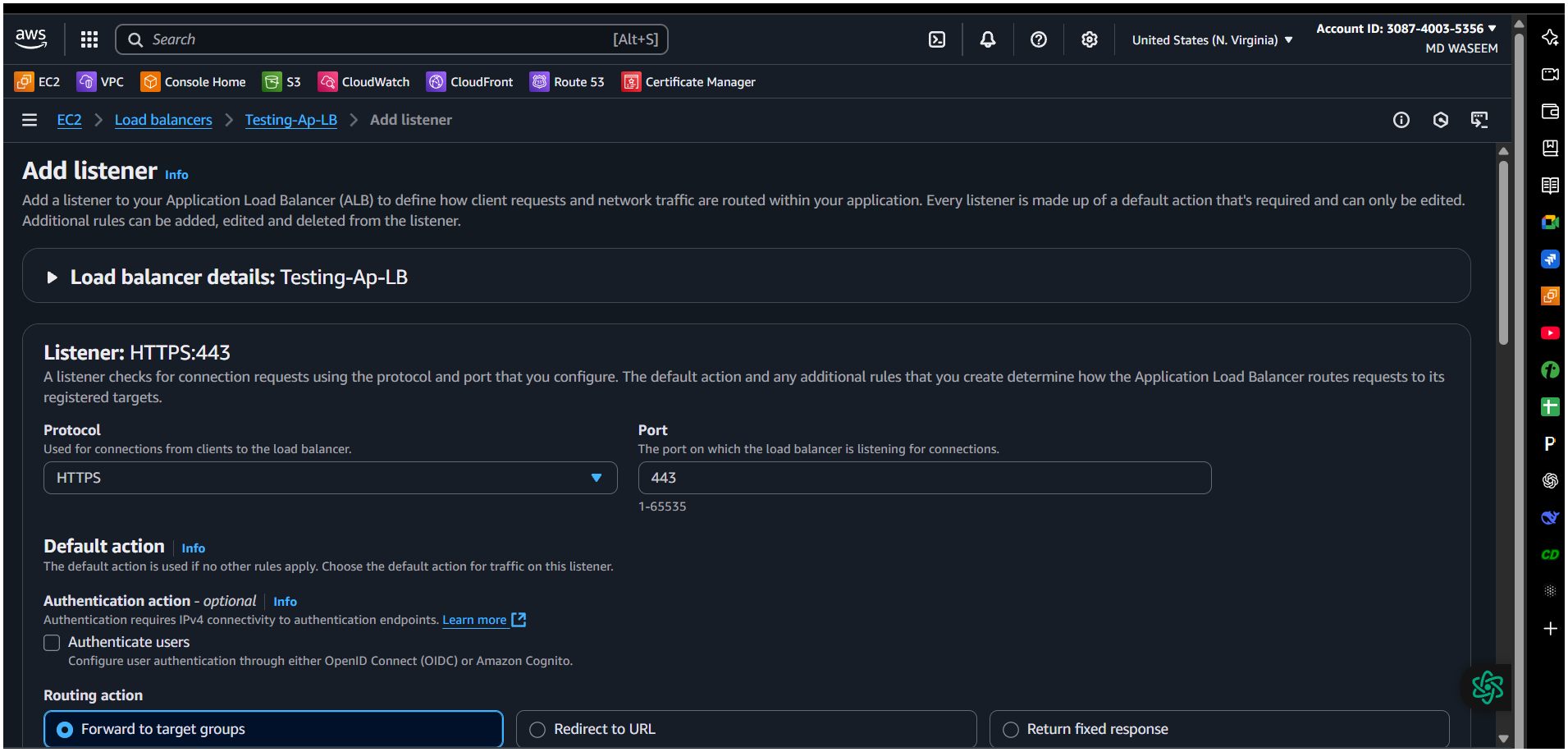
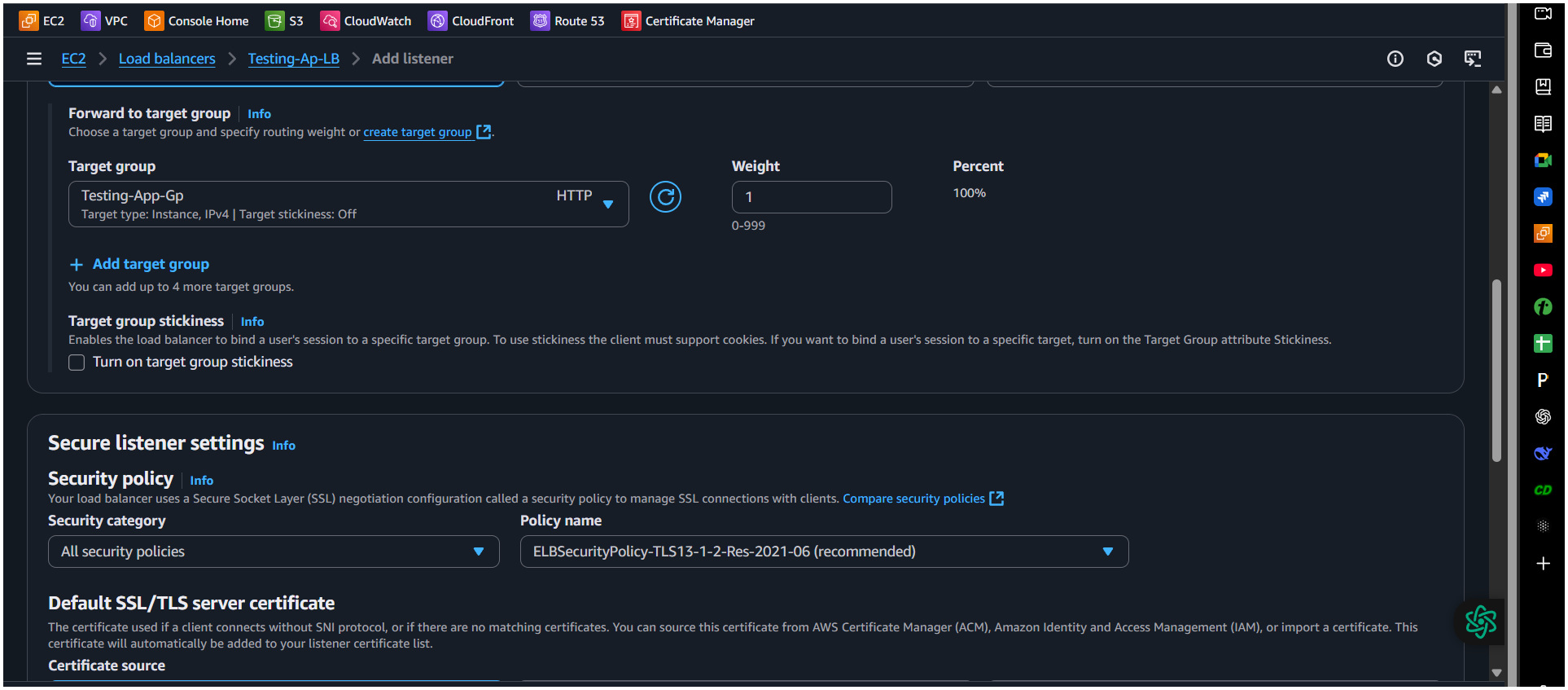
* Ensure you requested both example.com and www.example.com (if needed).
* You can also use a wildcard: \*.example.com for subdomains.

**6. Avoid Modifying or Deleting Records**

* If you edit or delete the CNAME record, validation will fail — leave it until the certificate is issued.

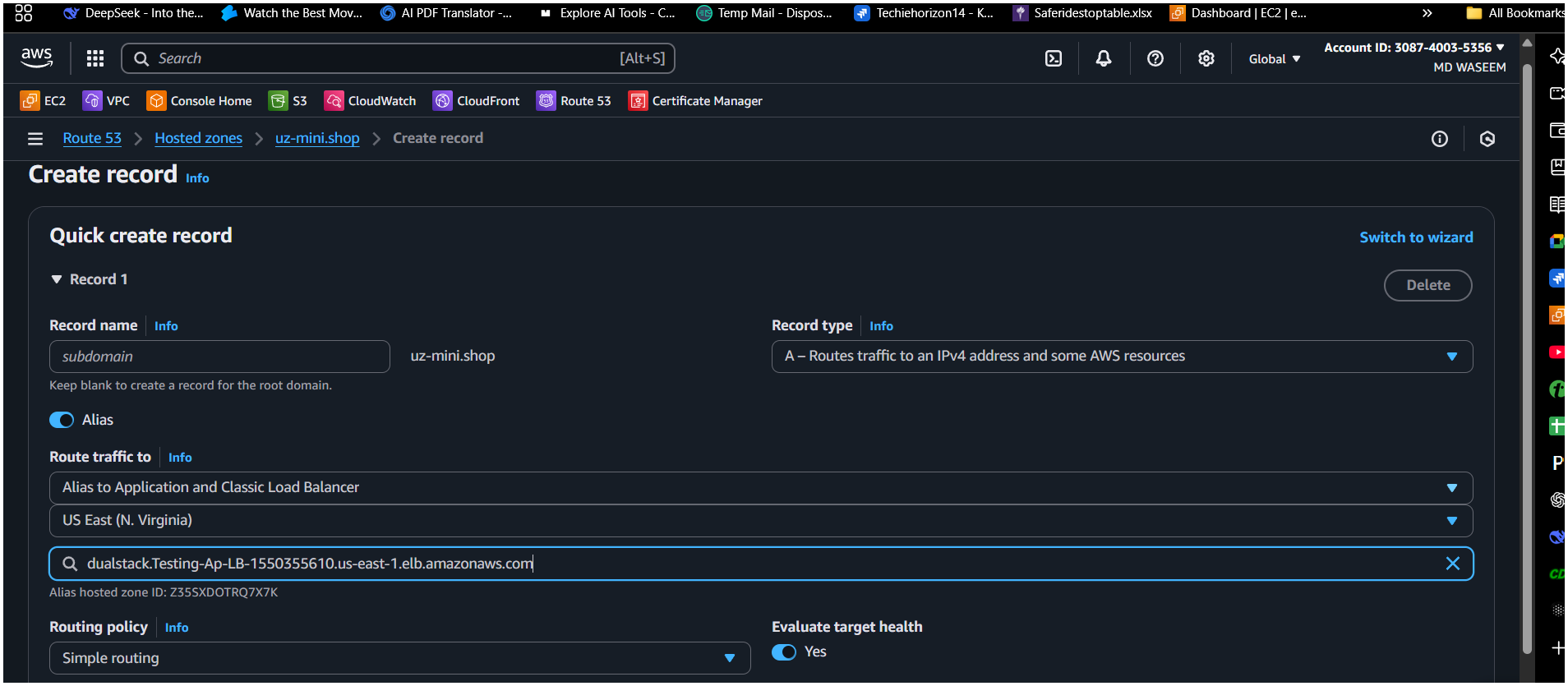
**📌 What To Do Now?**

* **Check Route 53 (or your DNS) for the CNAME record**
* If missing, go back to ACM and **manually create** or click **“Create records in Route 53”**
* Wait up to 30 minutes, and refresh the ACM page

****

**5) Map Application load balancer to R53.**

Get your ALB’s DNS name   
 Go to the EC2 console, select your ALB in the Load Balancers section, and copy its DNS name.   
   
➢ Open the Route 53 console and navigate to your Hosted Zones.   
   
➢ Create a new record:   
   
➢ Click Create record.   
   
➢ Choose record type A (IPv4 address).   
   
➢ Enable Alias.   
   
➢ Set "Alias to Application and Classic Load Balancer".   
   
➢ Select the region where your ALB resides.   
   
➢ Pick your ALB from the dropdown (if in the same AWS account), or paste its DNS name if in   
another account.   
   
➢ (Optional) Turn on Evaluate Target Health so Route 53 can route traffic only to healthy targets.   
   
➢ Save the record. DNS updates typically propagate within about a minute.

****

**6) Push the application load balancer logs to s3**

Steps to Enable ALB Access Logs to S3   
1. Create or choose an S3 bucket   
   
○ Ensure it's in the same AWS Region as your ALB.   
   
○ Enable Server-Side Encryption with S3-managed keys (SSE-S3).   
   
2. Attach the correct bucket policy   
   
Permit the ELB service account to put objects into your bucket. Here’s a simplified example{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"AWS": "arn:aws:iam::127311923021:root"

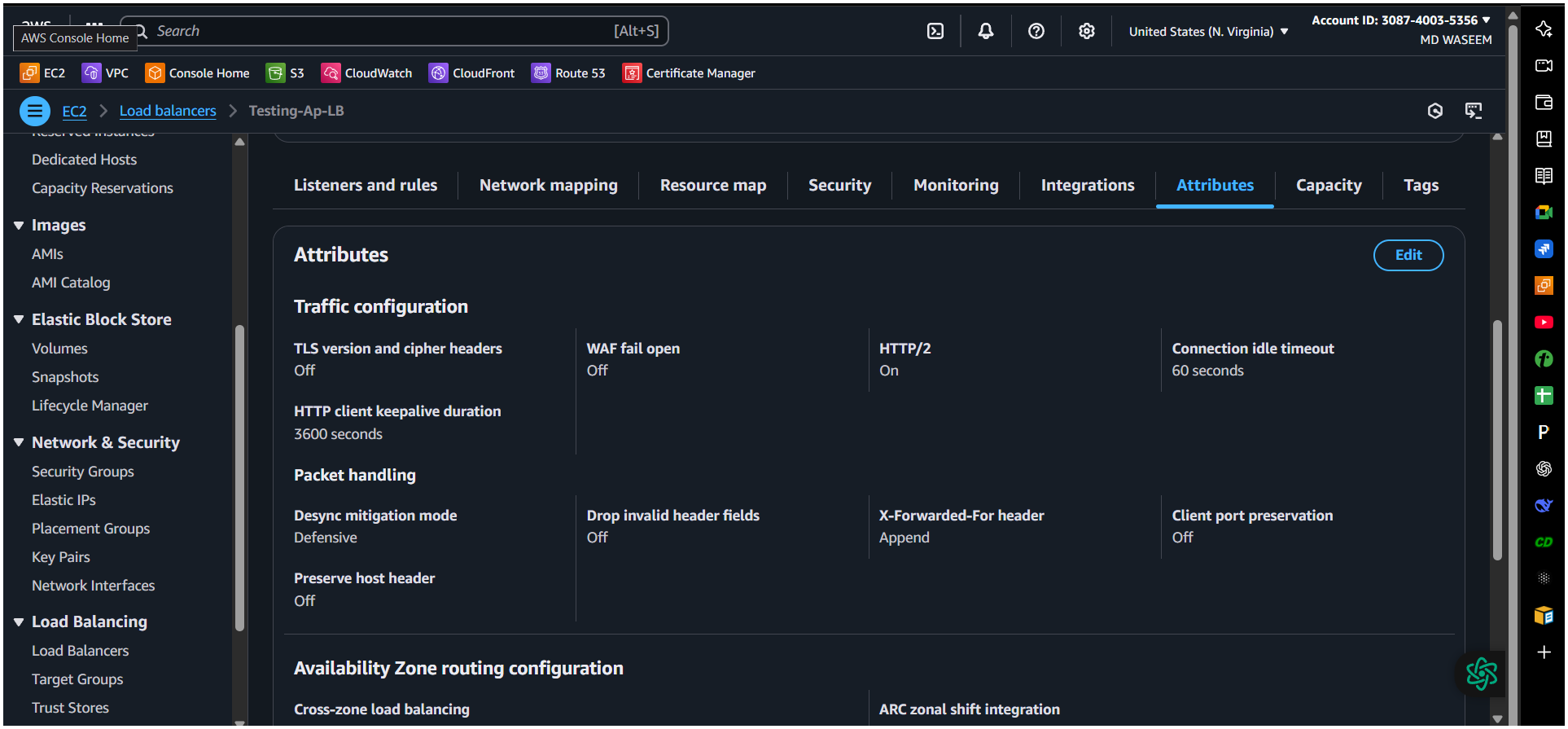
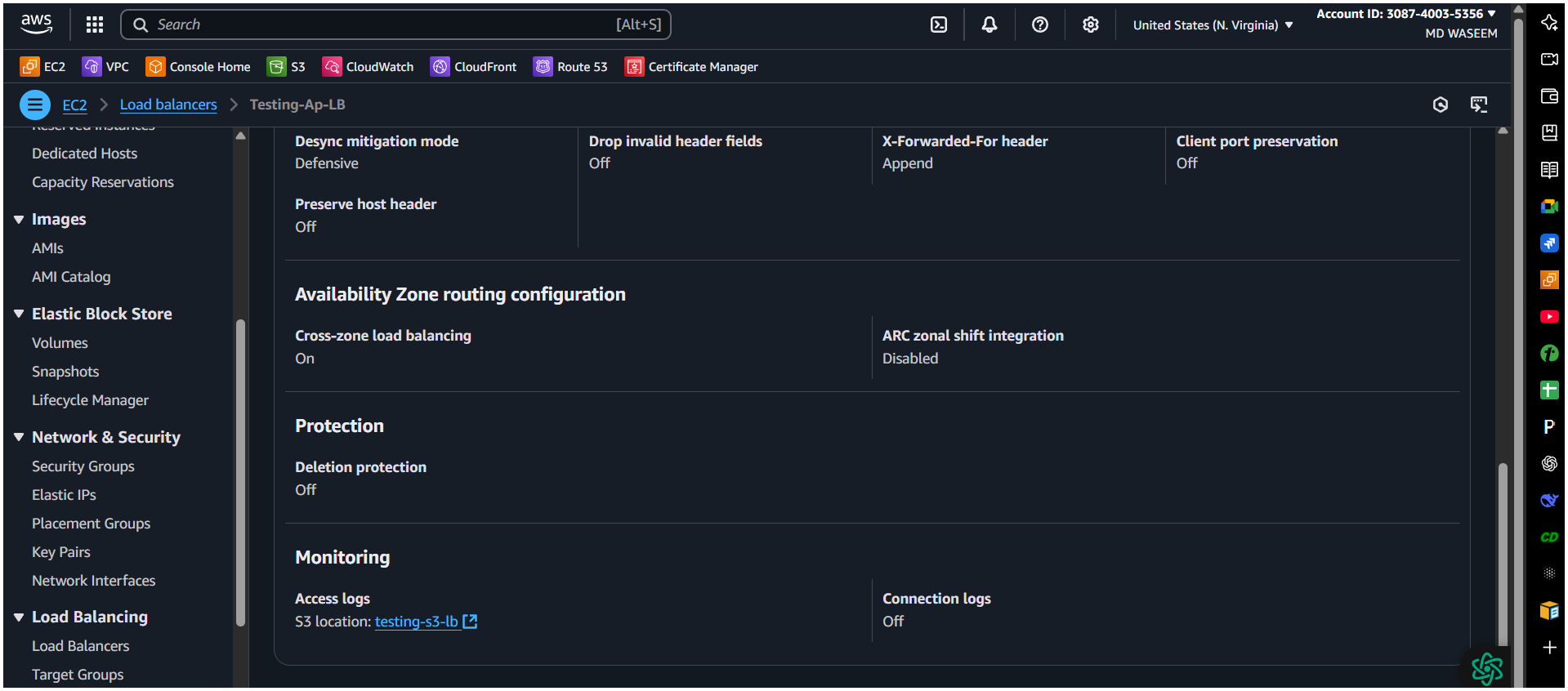
},

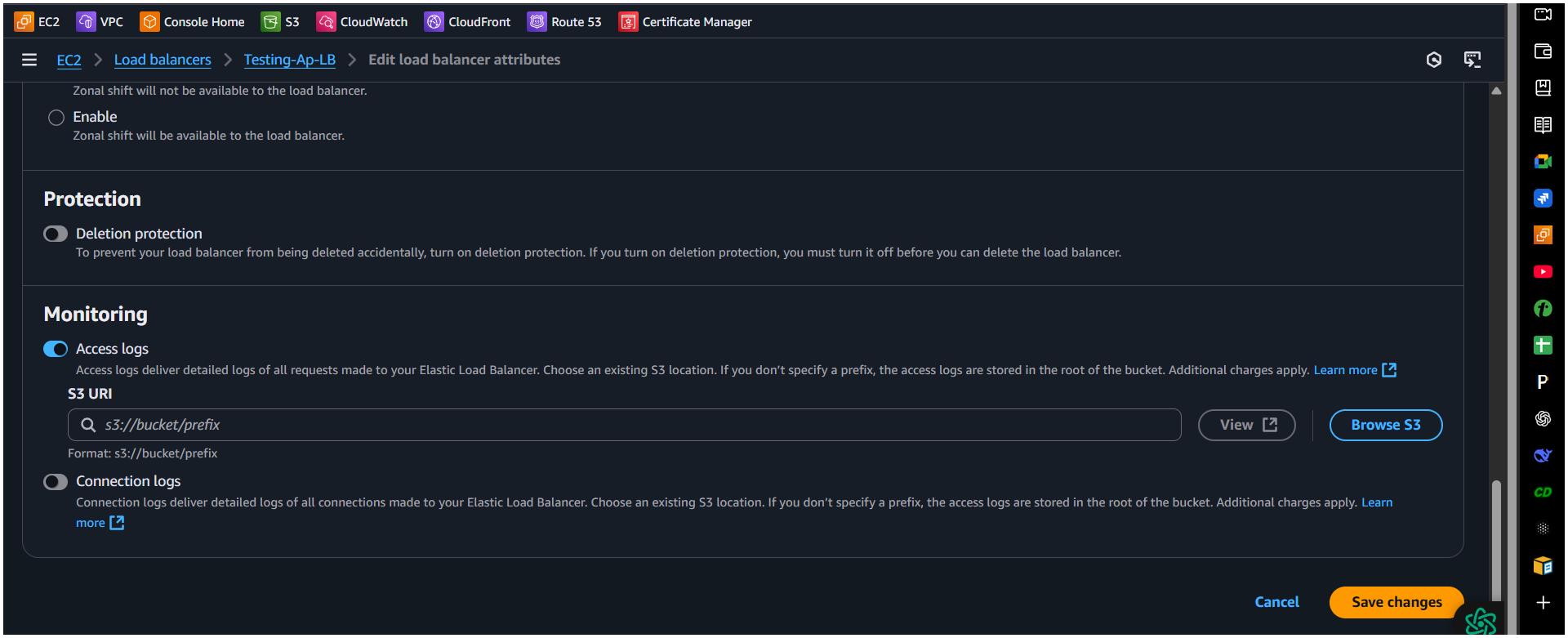
"Action": "s3:PutObject",

"Resource": "arn:aws:s3:::testing-s3-lb/AWSLogs/308740035356/\*"

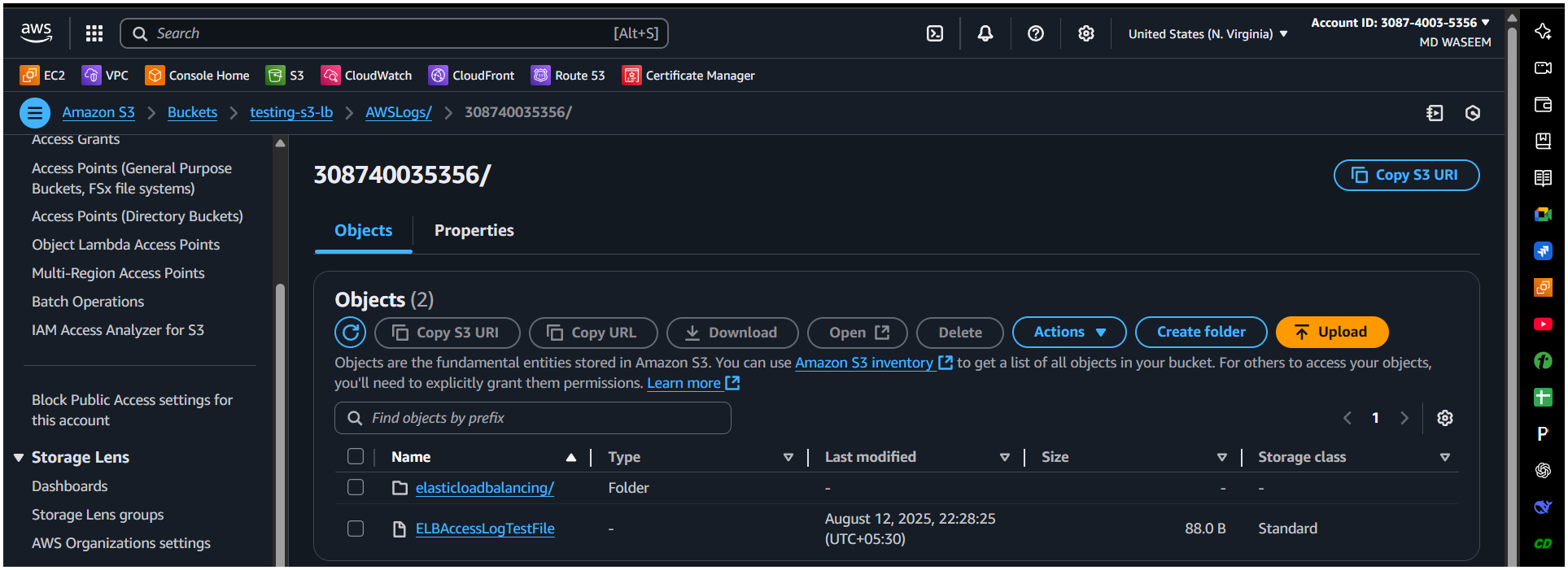
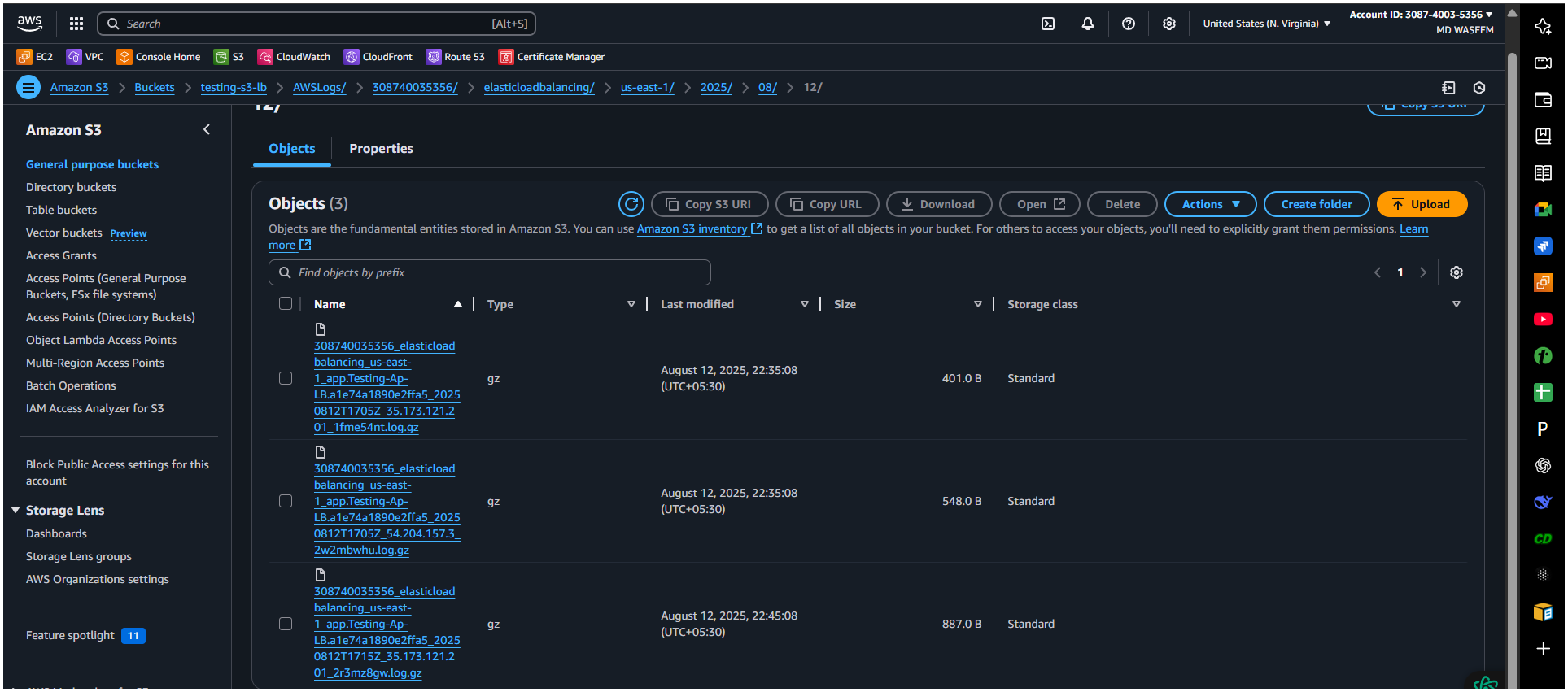
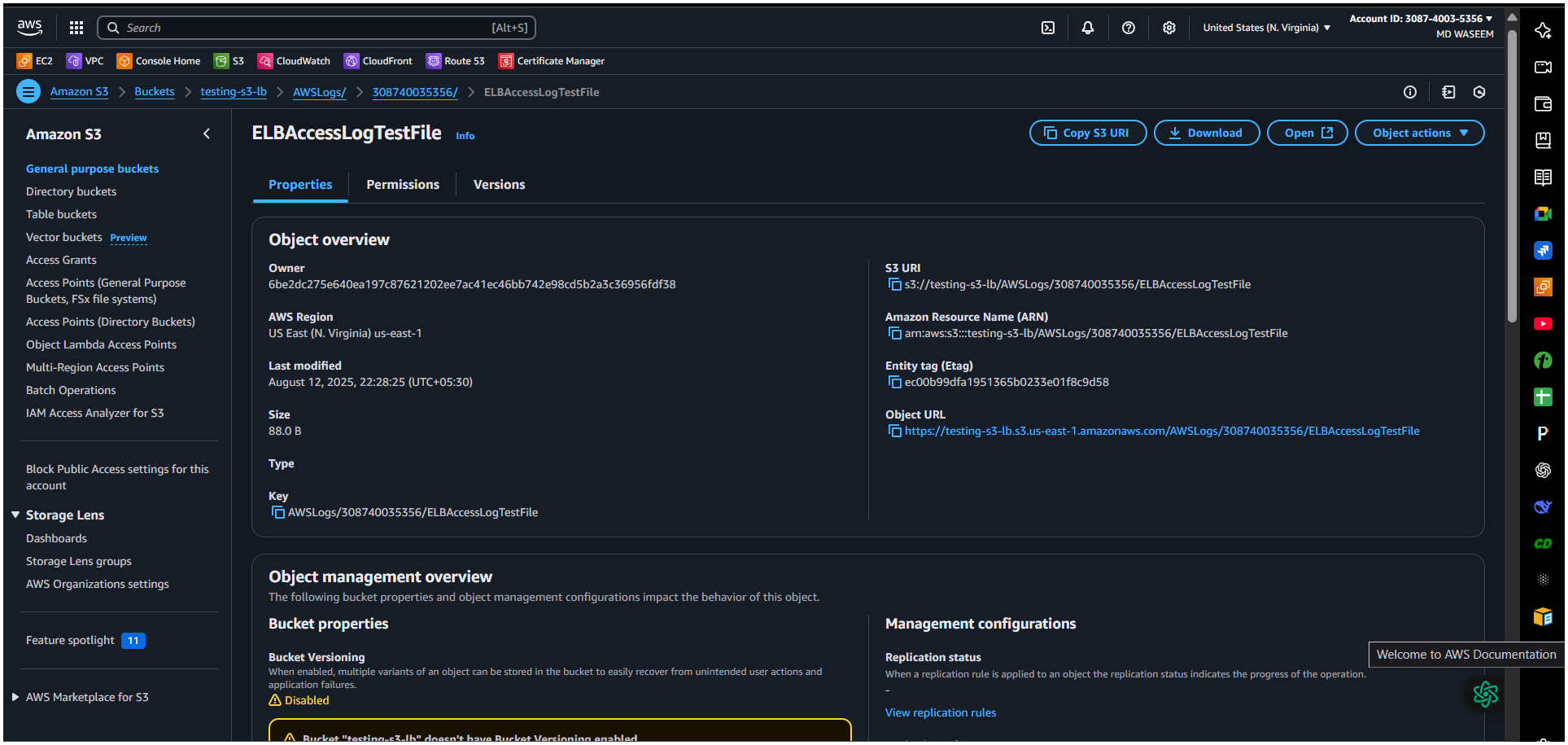
}

]

****}

****Use the correct regional ELB account ID—not your own AWS account. This is critical   
to avoid "Access Denied" errors   
   
Enable access logs in the ALB console   
   
○ Open the EC2 console, go to Load Balancers, then select your ALB.   
   
○ In the Attributes tab, click Edit.   
   
○ Turn on Access logs and enter the S3 bucket URI (e.g.,

s3://your-bucket-name/optional-prefix  
 — Wait & Verify  
•Logs are delivered every 5 minutes

****