**AWS Application Load Balancer (ALB) - Complete Step-by-Step Guide**

**Table of Contents**

1. Introduction to ALB
2. ALB vs NLB Comparison
3. Lab Architecture
4. Step-by-Step Implementation
   * Step 1: VPC Setup
   * Step 2: Security Groups
   * Step 3: Launch EC2 Instances
   * Step 4: Create Target Groups
   * Step 5: Configure ALB
   * Step 6: ACM Certificate
   * Step 7: Route 53 DNS
   * Step 8: Path-Based Routing
   * Step 9: HTTP to HTTPS Redirect
5. Testing & Validation
6. Cleanup
7. Real-World Use Cases

**1. Introduction to ALB**

**Application Load Balancer (ALB)** is a **Layer 7** load balancer that supports:  
✔ **HTTP/HTTPS traffic**  
✔ **Path-based routing** (e.g., /movies, /shows)  
✔ **Host-based routing** (e.g., api.example.com, app.example.com)  
✔ **SSL/TLS termination**  
✔ **WebSockets & HTTP/2**  
✔ **Integration with AWS WAF**

**2. ALB vs NLB Comparison**

| **Feature** | **ALB** | **NLB** |
| --- | --- | --- |
| **Layer** | Layer 7 (HTTP/HTTPS) | Layer 4 (TCP/UDP) |
| **Routing** | Path/Host-based | No path-based routing |
| **SSL Termination** | Yes | No |
| **Static IP** | No (DNS-based) | Yes |
| **Use Case** | Web apps, APIs | Gaming, VoIP, high-performance apps |

**3. Lab Architecture**

We will deploy:

* **3 Private Subnets** (for EC2 instances)
* **3 Public Subnets** (for ALB nodes)
* **ALB** with:
  + /homepage → homepage-target-group
  + /movies → movies-target-group
  + /shows → shows-target-group
* **Route 53** DNS (www.cloudvishwakarma.in)

**4. Step-by-Step Implementation**

**Step 1: VPC Setup**

1. **Create VPC** (alb-vpc, 10.0.0.0/16)
2. **Create Subnets**:
   * **Private Subnets** (for EC2):
     + 10.0.1.0/24 (us-east-1a)
     + 10.0.2.0/24 (us-east-1b)
     + 10.0.3.0/24 (us-east-1c)
   * **Public Subnets** (for ALB & NAT):
     + 10.0.4.0/24 (us-east-1a)
     + 10.0.5.0/24 (us-east-1b)
     + 10.0.6.0/24 (us-east-1c)
3. **Set Up NAT Gateway** (for private instances to access updates).

**Step 2: Security Groups**

* **ALB Security Group** (alb-sg):
  + Allow **HTTP (80)** & **HTTPS (443)** from 0.0.0.0/0
* **EC2 Security Group** (ec2-sg):
  + Allow **HTTP (80)** from alb-sg
  + Allow **SSH (22)** from your IP

**Step 3: Launch EC2 Instances**

**Deploy 3 instances (1 per AZ) with User Data:**

bash

#!/bin/bash

yum update -y

yum install -y nginx

systemctl start nginx

echo "<h1>$(hostname) - $(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)</h1>" > /usr/share/nginx/html/index.html

* **Instance 1**: homepage (us-east-1a)
* **Instance 2**: movies (us-east-1b)
* **Instance 3**: shows (us-east-1c)

**Step 4: Create Target Groups**

1. **Homepage Target Group** (homepage-tg):
   * Protocol: HTTP
   * Port: 80
   * Health Check: /
   * Register homepage instance
2. **Movies Target Group** (movies-tg):
   * Same as above, register movies instance
3. **Shows Target Group** (shows-tg):
   * Same as above, register shows instance

**Step 5: ACM Certificate**

1. **Request Certificate** (\*.cloudvishwakarma.in)
2. **Validate via Route 53** (Automated DNS validation)

**Step 6: Route 53 DNS**

1. **Create Hosted Zone** (cloudvishwakarma.in)
2. **Add A Record**:
   * Name: www
   * Alias: **Yes** → Point to ALB DNS

**Step 7: Configure ALB**

1. **Go to EC2 → Load Balancers → Create Application Load Balancer**
   * **Name:** alb-demo
   * **Scheme:** Internet-facing (for public access)
   * **IP address type:** IPv4
2. **Network Mapping:**
   * **VPC:** Select your alb-vpc
   * **MUST select all 3 PUBLIC subnets** (ALB nodes need public IPs):
     + us-east-1a (Public)
     + us-east-1b (Public)
     + us-east-1c (Public)
3. **Security Groups:**
   * Attach the alb-sg (allows HTTP/80 and HTTPS/443 from anywhere)
4. **Listeners:**
   * **First Listener: HTTPS (443)**
     + **Default action:** Forward to homepage-tg (critical for root path /)
     + **SSL Certificate:** Select your ACM certificate (\*.cloudvishwakarma.in)
   * **Second Listener: HTTP (80)**
     + **Action:** Redirect to HTTPS (443) with status code HTTP 301
5. **Routing Configuration (HTTPS Listener):**
   * **Default Rule:** / → homepage-tg (handles all unmatched traffic)
   * **Add Rule 1:**
     + **Condition:** Path is /movies\*
     + **Action:** Forward to movies-tg (Priority 10)
   * **Add Rule 2:**
     + **Condition:** Path is /shows\*
     + **Action:** Forward to shows-tg (Priority 20)
6. **Final Review:**
   * Verify all 3 public subnets are selected
   * Confirm ACM certificate is attached to HTTPS listener
   * Ensure homepage-tg is set as the default target group

**Step 8: HTTP to HTTPS Redirect**

1. **Edit ALB Listener Rules** (HTTP:80):
   * **Redirect to HTTPS (443)**
   * Status Code: **HTTP 301**

**5. Testing & Validation**

1. **Test Path-Based Routing**:

bash

curl https://www.cloudvishwakarma.in/movies

curl https://www.cloudvishwakarma.in/shows

1. **Verify HTTPS Redirect**:

bash

curl -v http://www.cloudvishwakarma.in *# Should return 301*

1. **Check Traffic Distribution**:

bash

watch -n 1 'curl -s https://www.cloudvishwakarma.in | grep "Server"'

**6. Cleanup**

1. **Delete ALB**
2. **Terminate EC2 Instances**
3. **Delete Target Groups**
4. **Release Elastic IPs**
5. **Delete VPC**

**7. Real-World Use Cases**

| **Scenario** | **Why ALB?** |
| --- | --- |
| **Microservices** | Path-based routing (/api, /app) |
| **Multi-tenant Apps** | Host-based routing (client1.app.com, client2.app.com) |
| **Web Apps** | SSL termination, WAF integration |