Elastic Load Balancers & Health Checks

Server-01

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo chown ec2-user:ec2-user /var/www/html/
echo "Hello from Server-01 $(hostname)" > /var/www/html/index.html
sudo systemctl start httpd
sudo systemctl enable httpd
```

Server-02

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo chown ec2-user:ec2-user /var/www/html/
echo "Hello from Server-02 $(hostname)" > /var/www/html/index.html
sudo systemctl start httpd
sudo systemctl enable httpd
```

Server-03

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo chown ec2-user:ec2-user /var/www/html/
echo "Hello from Server-03 $(hostname)" > /var/www/html/index.html
sudo systemctl start httpd
sudo systemctl enable httpd
```

What is an Elastic Load Balancer (ELB)?

AWS **Elastic Load Balancer (ELB)** distributes incoming traffic across multiple EC2 instances to improve **availability, fault tolerance, and scalability**.

Types of Load Balancers

1. **Application Load Balancer (ALB)** → Works at **Layer 7 (HTTP/HTTPS)**
   - Best for: Web applications, microservices, content-based routing
2. **Network Load Balancer (NLB)** → Works at **Layer 4 (TCP/UDP)**
   - Best for: Low latency, high-throughput applications
3. **Classic Load Balancer (CLB)** → Works at **Layer 4 & 7** (Legacy)
   - Best for: Basic load balancing (not recommended for new applications)

What is a Health Check in ELB?

A **health check** is a mechanism ELB uses to determine whether an instance is **healthy** (able to serve traffic).

- If an instance fails the health check, ELB **stops sending traffic** to it.
- Health checks are based on **protocol (HTTP, HTTPS, TCP), port, and path**.

Hands-On Guide: Deploy an ELB with EC2 Instances

Step 1: Launch EC2 Instances

1. Go to **AWS Console → EC2 → Instances**
2. Click **Launch Instances** and select **Amazon Linux**
3. Choose **t2.micro** (Free Tier)
4. Create a **Security Group**:
   - Allow **SSH (22)** from your IP
   - Allow **HTTP (80)** from **anywhere (0.0.0.0/0)**
5. User Data (Optional) → Automatically install a web server:
6. #!/bin/bash
7. yum update -y
8. yum install -y httpd
9. echo "Hello from Server-01 $(hostname)" > /var/www/html/index.html
10. systemctl start httpd
11. systemctl enable httpd

Launch **at least 2 instances:**
Repeat above steps :
echo "Hello from Server-02 $(hostname)" > /var/www/html/index.html
echo "Hello from Server-03 $(hostname)" > /var/www/html/index.html

Step 2: Create an Elastic Load Balancer (ELB)

1. Go to **AWS Console → EC2 → Load Balancers**
2. Click **Create Load Balancer**
3. Choose **Application Load Balancer** (ALB)
4. Set a **Name** (e.g., my-alb)
5. **Select Internet-facing**
6. Choose at least **2 Availability Zones (AZs)**
7. Create a **Security Group**:
   - Allow **HTTP (80)**
8. Click **Next**

Step 3: Create a Target Group

1. Select **"Create a new target group"**
2. **Target Type**: Instance
3. **Protocol**: HTTP & **Port**: 80
4. Click **Next** → Register your EC2 instances
5. Click **Create Target Group**

Step 4: Attach Target Group to Load Balancer

1. In **Load Balancer settings**, select the **Target Group** you created
2. Click **Create Load Balancer**
3. Wait until the **status** changes to **Active**

Step 5: Test Load Balancer & Health Checks

1. Get the **DNS name** of the ALB (from AWS Console)
2. Open a browser and go to:
   http://<your-alb-dns-name>

How to Check & Modify Health Check Settings

1. Go to **EC2 → Target Groups**
2. Select your target group
3. Click **Health checks**
4. Modify settings:
   - **Protocol**: HTTP
   - **Port**: 80
   - **Path**: /index.html
   - **Thresholds**:
     - Healthy: 2 (Number of successful checks before instance is marked healthy)
     - Unhealthy: 2 (Failed checks before marking instance unhealthy)
5. Click **Save changes**

Summary

- **ELB** distributes traffic across multiple EC2 instances.
- **Health checks** ensure only healthy instances receive traffic.
- **Hands-on setup** includes launching EC2 instances, creating an ALB, configuring a target group, and testing.