To launch an AWS load balancer for your Spring Boot application, follow these steps:

Step 1: Set Up Your Environment

Clone the Repository: Clone your app from GitHub to a local machine or EC2 instance.

bash

Copy code

git clone https://github.com/vickydevo/springboot-hello.git

Build the Application: Build the Spring Boot application using Maven or Gradle.

bash

Copy code

cd springboot-hello

./mvnw clean package

Step 2: Launch an EC2 Instance

Log in to the AWS Console and go to EC2 Dashboard.

Launch a New EC2 Instance:

Choose an Amazon Linux or Ubuntu image.

Configure security group settings to allow HTTP (port 80), HTTPS (port 443), and the port your Spring Boot app uses (typically 8080).

Choose an instance type and key pair.

Install Java on the EC2 Instance:

bash

Copy code

sudo yum install java-11-openjdk -y

Deploy Your Spring Boot Application:

Copy the built .jar file to the instance.

Start the Spring Boot application.

bash

Copy code

java -jar springboot-hello/target/your-app-name.jar

Step 3: Create a Target Group

In EC2 Console, navigate to Target Groups and click Create target group.

Choose Instances as the target type, set a name, and choose HTTP protocol on port 8080 (or your app’s port).

Register your EC2 instance with the target group.

Step 4: Create an Application Load Balancer

Go to the Load Balancers section in the EC2 Console and click Create Load Balancer.

Choose Application Load Balancer.

Configure load balancer settings:

Choose Internet-facing and select the relevant VPC and subnets.

Add listeners on HTTP (port 80) and HTTPS (port 443) if needed.

Assign the security group with HTTP and HTTPS access.

Under Target Groups, select the target group you created.

Step 5: Test the Setup

Access the application using the DNS name of the load balancer in a browser:

arduino

Copy code

http://<load-balancer-dns-name>

This setup will distribute incoming traffic across the EC2 instances in the target group, ensuring load balancing for your Spring Boot application.