

## What is ELB Stickiness (Session Stickiness / Sticky Sessions)?

**ELB Stickiness**, also known as **Session Stickiness** or **Sticky Sessions**, is a feature in AWS Elastic Load Balancer that **binds a user's session to a specific target (like an EC2 instance)** behind the load balancer for the duration of the session.

This means that **all requests from the same client** during a session are **always routed to the same backend instance** instead of being distributed across multiple targets.

---

## Why Use Stickiness?

In some applications (e.g., user login systems or shopping carts), session data is stored **in-memory** on the server (not shared between instances). Without stickiness, requests might hit different instances, causing the session to break.

---

## How It Works

ELB uses a special **cookie** to track the client session:

### 1. Load Balancer-Generated Cookie (**AWSSELB** or **AWSALB**)

- ELB creates and manages a cookie.
- Automatically inserted into the client's browser.
- The load balancer uses this cookie to **route requests to the same instance**.

### 2. Application-Controlled Cookie

- Your app generates its own session cookie.
  - ELB can be configured to use this cookie to maintain stickiness.
-

## Load Balancer Type Support

Load Balancer Type	Stickiness Support	Cookie Type	How to Enable
Application (ALB)	✔ Yes	AWSALB, custom	Target group attribute
Classic (CLB)	✔ Yes	AWSELB, custom	Listener policy
Network (NLB)	✗ No	N/A	Not supported
Gateway (GWLB)	✗ No	N/A	Not supported

---

## How to Enable (Example for ALB)

You can enable stickiness in the **Target Group settings** for ALB:

- Set `stickiness.enabled` to `true`
- Set `stickiness.type` to `lb_cookie`
- Set `stickiness.lb_cookie.duration_seconds` (default: 1 day)

## AWS CLI Example:

```
aws elbv2 modify-target-group-attributes \  
--target-group-arn <your-target-group-arn> \  
--attributes Key=stickiness.enabled,Value=true \  
             Key=stickiness.type,Value=lb_cookie \  
             Key=stickiness.lb_cookie.duration_seconds,Value=3600
```

---

## Stickiness Duration

You can define how long the session stickiness should last:

- **In seconds** (e.g., 3600 = 1 hour)
  - After the cookie expires or is deleted, a new target can be selected
- 

### **Considerations & Limitations**

- **Scalability:** May cause uneven traffic distribution if one instance becomes sticky to many clients.
  - **High Availability:** If a sticky instance goes down, the session is lost unless the app supports session replication.
  - **Not supported** by Network Load Balancers or Gateway Load Balancers.
- 

### **When to Use Stickiness**

- Applications that store user session data locally
  - No session-sharing mechanism like Redis or Memcached
  - Simple web apps without backend session persistence
-