**Lab Exercise 9– Target Load Balancing with Two Target Servers in Apigee**

**Objective**

* Configure two target servers (mock servers)
* Create an API proxy that load balances between them
* Observe requests being distributed to both servers

**Step 1: Prepare Two Target Servers**

For demo/testing, we will use **two public mock APIs**:

* Target 1: https://httpbin.org
* Target 2: https://mocktarget.apigee.net

These are publicly accessible and return different JSON payloads.

**Step 2: Define Target Servers in Apigee**

1. Go to **Apigee Console**
2. Navigate to **Admin > Environments**
3. Select your environment (e.g., eval)
4. Go to the **Target Servers** tab
5. Create **two target servers**:

**Target Server 1**

* Name: target-server-1
* Host: httpbin.org
* Port: 80
* Enable SSL: No

**Target Server 2**

* Name: target-server-2
* Host: mocktarget.apigee.net
* Port: 80
* Enable SSL: No

Save both target servers.

**Step 3: Create a New API Proxy**

1. Go to **Develop > API Proxies**
2. Click **+ CREATE PROXY**
3. Choose **Reverse Proxy**, click **Next**
4. Fill in:
   * Name: load-balancer-demo
   * Base Path: /load-balancer-demo
   * Target URL: You can give a dummy URL (we'll override it)
5. Click through the steps and click **Create**

**Step 4: Edit TargetEndpoint to Use Load Balancing**

1. Open your proxy → **Develop** tab
2. Go to TargetEndpoints > default
3. Replace the <HTTPTargetConnection> block with the following:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<TargetEndpoint name="default">

<Description/>

<FaultRules/>

<PreFlow name="PreFlow">

<Request/>

<Response/>

</PreFlow>

<PostFlow name="PostFlow">

<Request/>

<Response/>

</PostFlow>

<Flows/>

<HTTPTargetConnection>

<Properties/>

<LoadBalancer>

<Server name="target-server-1"/>

<Server name="target-server-2"/>

</LoadBalancer>

</HTTPTargetConnection>

</TargetEndpoint>

**Explanation:**

* RoundRobin distributes requests one-by-one across the servers.
* The /get path is used for both target servers.

**Step 5: Deploy the Proxy**

1. Go to **Develop > Deployments**
2. Deploy load-balancer-demo to your environment (e.g., eval)

**Step 6: Test the API**

Run the following curl command multiple times:

curl https://your-env-group.apigee.net/load-balancer-demo

**What you'll see:**

* Alternate responses from httpbin.org and mocktarget.apigee.net
* You can also inspect headers or payload differences to confirm the backend