



Linux Academy

Hands-On Training

Creating Complex
Routing Policies

Contents

Introduction.....	1
Getting Started.....	1
Building the Policies.....	1



Introduction

In this lab, we create complex DNS-level routing policies. We use four EC2 instances to mimic a multi-region environment, and a weighted routing policy, in which we define how much traffic we want each server to handle. Although not in the scope of this lab, this can also be easily amended for latency- or geolocation-based routing.

Getting Started

Log into your AWS Console with the given credentials.

From here, navigate to **Route 53**, ignoring the permissions warning. Go to your **Hosted zones**. You should see a domain name of *linuxacademylab#.com*, with the **#** replaced with a number. This is the domain you are using for this lab. Make note of this, and replace the **#** with the appropriate number, as needed.

Move to the **EC2 Dashboard**, and select **Instances**. You have four instances: **Region A**, **Region A-1**, **Region B**, and **Region B-1**. These will each have separate public IP addresses, and should you navigate to each instance, you should see a simple page that simply states the name of the region. This is to make testing the routing policies easier.

Building the Policies

Our goal is to have our top-level domain (*linuxacademylab#.com*) go to either Region A or B, based upon weighted DNS. The region then determines which instance to navigate. Once the request is made, it is cached until the TTL runs out. You can adjust the TTL within the domain's record sets.

When building the routing policy, we need to work from the bottom up: First building the region-independent DNS, moving until we reach the top-level domain.

Navigate to **Route 53**, select your *linuxacademylab#.com* zone, and press **Go to Record Sets**. Press **Create Record Set**, naming the set *regiona.linuxacademylab#.com*. Inside of this, input the *public IP of any Region A instance* as the value. Be sure to select **Weighted** for the **Routing Policy**. Put a **Weight** of **2**, and set the **Set ID** to denote which Region A instance you are using (*a* or *a-1*). Press **Create**.

The screenshot shows the AWS Route 53 console. On the left, a table lists existing record sets for the domain *linuxacademylab4.com*:

Name	Type	Value
linuxacademylab4.com.	A	52.73.19.137
linuxacademylab4.com.	NS	ns-1369.awsdns-43.org. ns-2036.awsdns-62.co.uk. ns-524.awsdns-01.net. ns-220.awsdns-27.com.
linuxacademylab4.com.	SOA	ns-1369.awsdns-43.org. awsdns-43.net.

On the right, the 'Create Record Set' form is shown with the following details:

- Name:** *regiona.linuxacademylab4.com.*
- Type:** A - IPv4 address
- Alias:** ☐ Yes ☒ No
- TTL (Seconds):** 300 (with buttons for 1m, 5m, 1h, 1d)
- Value:** 54.175.7.210
- Routing Policy:** Weighted

Repeat this with the second Region A instance, also naming the set `regiona.linuxacademylab#.com`, but inputting the *second Region A instance's public IP* in as the **Value**, and setting the weighted **Set ID** to *A* or *A-1* depending upon the instance you are using.

Now, what happens should the region request go to Region B? Like in Region A, we need to create Region B record sets. Follow the same process as above, but substituting the **Name** for `regionb`, and the **Values** and **Set IDs** to *those relevant for the Region B servers*.

Should we navigate to `regiona.linuxacademylab#.com` or `regionb.linuxacademylab#.com`, we receive one of the two servers accessed through this link.

Note

It may take time for the DNS to propagate before you can check this. If it is not working, wait for a few minutes, and then try again.

We still need to send the traffic from the top level (`linuxacademylab#.com`) domain to either Region A or Region B. We do this by creating two additional record sets.

Press **Create Record Set** and do not add anything to the name field. We will, however, add an **Alias Target**, set to `regiona.linuxacademylab#.com`. (with the trailing period as part of the alias). Set the **Routing Policy** to *Weighted*, as we did with the others, then give the **Set ID** a value of `regiona`. **Create**.

Name	Type	Value
linuxacademylab4.com.	A	ALIAS regiona.linuxacademylab4.com.
linuxacademylab4.com.	NS	ns-1369.awsdns-43.com, ns-2036.awsdns-62.com, ns-524.awsdns-01.net, ns-220.awsdns-27.co.uk
linuxacademylab4.com.	SOA	ns-1369.awsdns-43.com
regiona.linuxacademylab4.com.	A	54.175.7.210
regiona.linuxacademylab4.com.	A	54.172.18.62
regionb.linuxacademylab4.com.	A	54.210.136.191
regionb.linuxacademylab4.com.	A	54.88.165.51

A second record set for Region B now needs to be created. Leave the name as-is, with the **Alias** set to `regionb.linuxacademylab#.com`. Set the **Routing Policy** as before. Press **Create**.

Now, should you navigate to `linuxacademylab#.com`, there is now a 25% chance of you ending up in each instance. Please note these percentages can be adjusted through the Weight option in your record sets.