Linux Academy

Live! Lab

Configuring VPC Peering

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Live! Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the alloted amount of time finishes

Related Courses

AWS CSA - Associate

Related Videos

Building a VPC from Scratch

VPC Networking

VPC Peering

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Introduction

In this lab, we are taking two Virtual Private Clouds within the same region and creating a peered connection, allowing you to route traffic between them using private IP addresses. This can be done between your own VPCs or those in another networks; for this lab, we are working with our own VPCs.

Log in to the AWS Console using the credentials provided on the Live! Lab page.

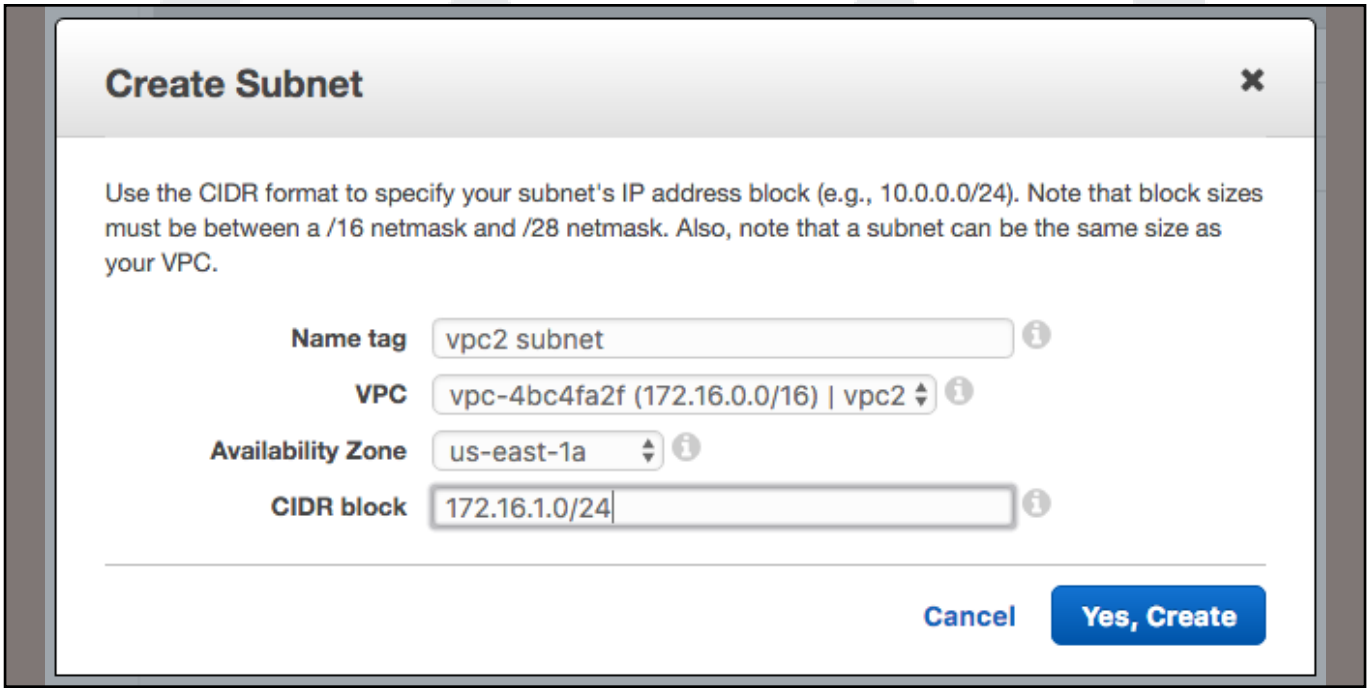
Creating a Second VPC

Navigate to the **VPC Dashboard**, located under **Networking**. There is only one VPC available. We need to create a second VPC to peer with; this is done manually in this lab for practice.

Select **Your VPCs**, **Create VPC**. For the **Name tag** we used *vpc2*, and set the **CIDR block** to *172.16.0.0/16*. Press **Yes, Create**.

Creating a Subnet

Navigate to **Subnets** in the left menu. Although there are subnets already available, we need to create one for *vpc2*. Select **Create Subnet**.



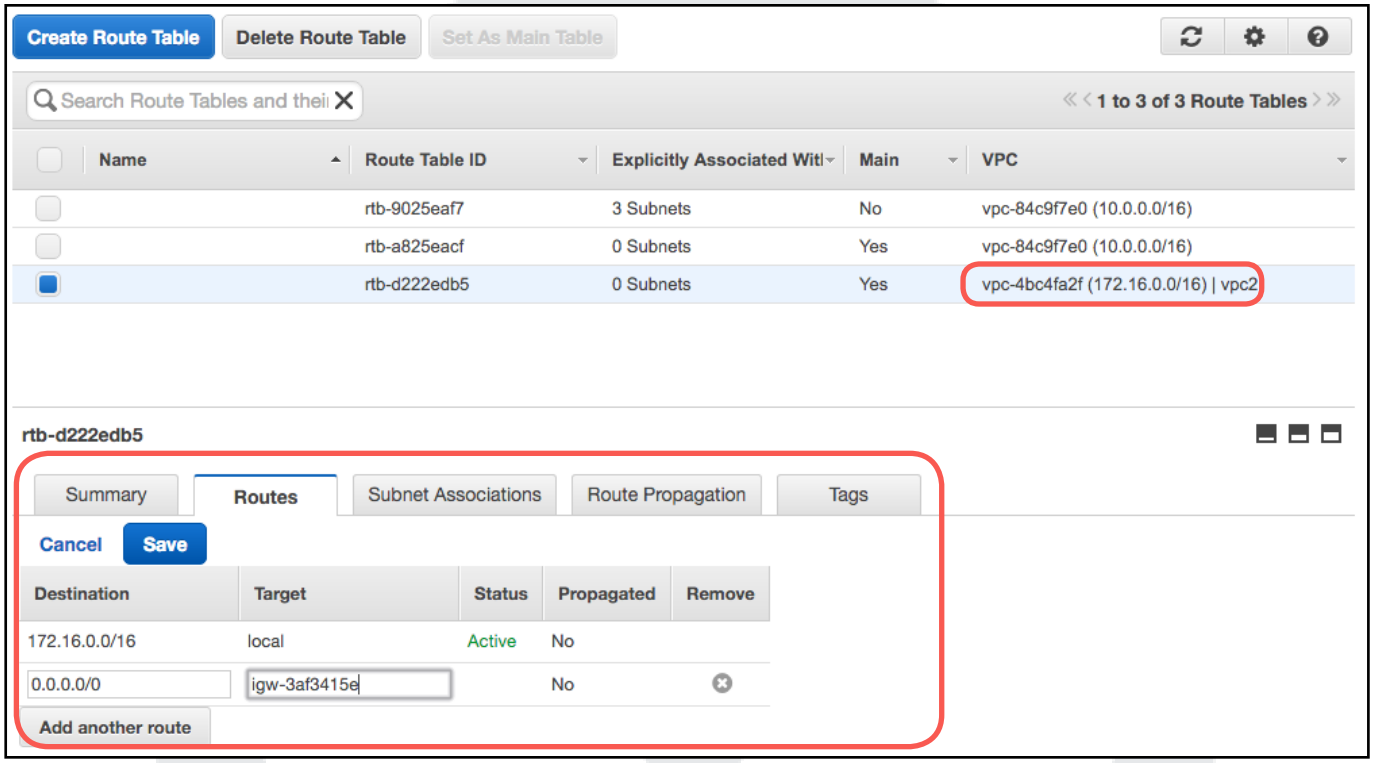
Set the **Name tag** to *vpc2 subnet*, the **VPC** to *vpc2*, the **Availability Zone** to any within the region, and the **CIDR block** to *172.16.1.0/24*. Press **Yes, Create**.

Creating an Internet Gateway

To demonstrate VPC peering, we need two instances with which to connect. Because our second VPC does not have an Internet gateway, we need to add one now.

Click **Internet Gateways** on the left menu, then **Create Internet Gateway**. For the **Name tag** we choose *vpc2* again. Press **Yes, Create**.

Select **Attach to VPC** with your newly-created Internet gateway selected. Attach it to *vpc2*.



Now go to your **Route Tables**. Select *vpc2* from the list, and move to the **Routes** tab, below. Click **Edit**, **Add another route**. Set the **Destination** to *0.0.0.0/0* and the **Target** to your Internet gateway. **Save**.

Verifying VPC Communication

Navigate to your **EC2 Dashboard**, and select **Launch Instance**; we need to create an instance for each VPC.

Select the *Amazon Linux AMI*, and leave the instance type set to *t2.micro*. Press **Next: Configure Instance Details**, and set the **Network** to the 10.0.0.0 VPC, and *Enable* **Auto-assign Public IP**. Click **Review and Launch**, then **Launch** on the next page.

When the key pair prompt occurs, select **Create a new key pair**. We named ours *peering*. **Download Key Pair**, then press **Launch Instances**.

Now we need to create a second instance for our *vpc2* VPC. Follow the steps above, but set the **Network**

to *vpc2*. When asked about key pairs, select **Choose an existing key pair** and ensure that your *peering* key pair is selected.

View your instances. As their instance state goes from *pending* to *running*, open your terminal and navigate to the directory where you saved the key pair. Change the permissions:

```
[user@Penguinbook]# chmod 400 peering.pem
```

Once your instances are running, select an instance and **Connect**. Copy the sample SSH string to your terminal to access the first server.

Once connected, open another terminal tab or window and log in to the second instance.

If, from one of your instances, you attempt to use the private IP address to connect to the other, you will be unable to connect. We need to create a peering connection for this to work.

Configuring VPC Peering

We now need to make our VPC peering connection. Return to your VPC Dashboard, and select **Peering Connections**, **Create VPC Connection**.

In this lab, we are giving our connection the **Name tag** of *peering*, although in actual practice you want to supply a more descriptive name.

For **Local VPC to peer** select *vpc2*. The **Account** is *My account*, and the **VPC ID** should be set to the *10.0.0.0* VPC. Press **Create**, **OK**.

At the bottom area of the page, select **Accept request**, **Yes, Accept**. However, we still need to configure the route table for VPC peering to work correctly.

Select **Route Tables** *from the left menu*, and choose your *vpc2* VPC. Navigate to the **Routes** tab, and add a new route to your 10.0.0.0/16 VPC. To do this set the **Destination** to *10.0.0.0/16*, and the **Target** to your peering VPC. **Save**.

Move to your other subnet's route tables, this time adding a route to *172.16.0.0/16*, the peering VPC. **Save**.

Return to your terminals and test the connection from each instance using the private IP addresses of the other. You are now connected!