Linux Academy

Hands-On Training

Configuring VPC S3 Endpoints and NAT Gateways

Contents

Introduction

Previously, to use a NAT gateway, you needed to launch an AMI that contained NAT software. With AWS's new VPC NAT service this can be done with the AWS Console. In this lab, we are learning how to configure this new NAT gateway feature, as well as creating additional S3 endpoints for security purposes. With the application layer of our infrastructure on a private subnet, we need to send all traffic through a NAT as a proxy for the private subnet. Traditionally, this made scaling difficult because higher network throughput either meant creating a larger NAT or adding more NAT instances. With the new VPC NAT service, your NAT gateway will be able to scale. Additionally, using S3 endpoints, we can increase security by working solely with AWS resources through the AWS Cli.

Getting Started

Using the credentials given on the Live! Lab page, log into your **AWS Console**.

Navigate to the **EC2 Dashboard**, and view your running **instances**: *Private* and *Public*. *Private* does not have an Internet gateway, and this is the instance in which we are working. To access this instance, we need to use our *Public* server.

The Academy suggests:

Changing the password of both linuxacademy and root users upon logging into your lab servers.

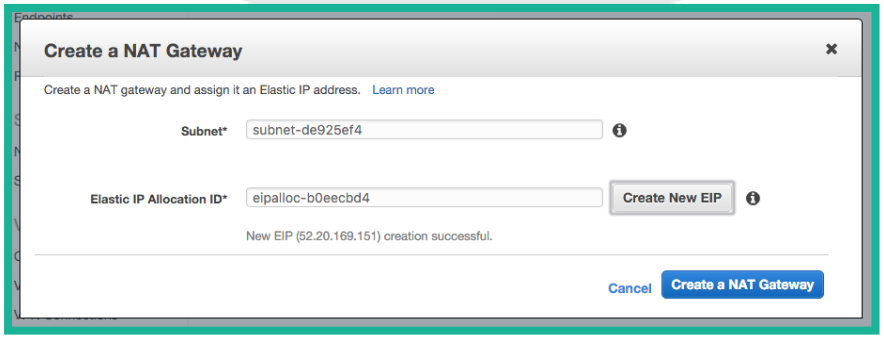
Copy the **Public IP address** of the **Public** image, and `ssh` into the instance using your terminal. Login information is on the Live! Lab page.

Now, select the **Private IP** of the *Private* image. From the *Public* instance, `ssh` into the *Private* instance. Try to `ping` any website. Because we are in a private instance, `ping` should not be able to contact outside websites.

Configuring a NAT Gateway

From the **AWS Console**, under **Network**, go to your **VPC Dashboard**. From here, select **NAT Gateways**, **Create NAT Gateway**.

The **subnet** the NAT gateway uses needs to be one with access to the Internet. As such, select your *public subnet* from the list. For the **Elastic IP Allocation ID**, press **Create New EIP**. Press **Create NAT Gateway**. View your NAT gateways.



Refresh the page and wait until the NAT gateway's **Status** transitions from *Pending* to *Available*. You may need to refresh the page to see changes.

Navigate to the **Subnets** page in the **VPC Dashboard**. View the **Route Table** for the *Private* subnet, and click on the *current Route Table*. Press **Edit**, then add another route to the **Destination** *0.0.0.0/0*. Set the **Target** to the newly-created *NAT gateway ID*. **Save**. Our traffic now has a route out, but no incoming route to the Internet.

Return to your *Private* instance in your terminal. Run `apt-get update` then `apt-get install python-pip` to test if the connection worked. You need the `python-pip` package for the next portion of the lab.

Configuring S3 Endpoints

Using the newly-downloaded `pip`, install the **AWS Cli**:

```
pip install awscli
```

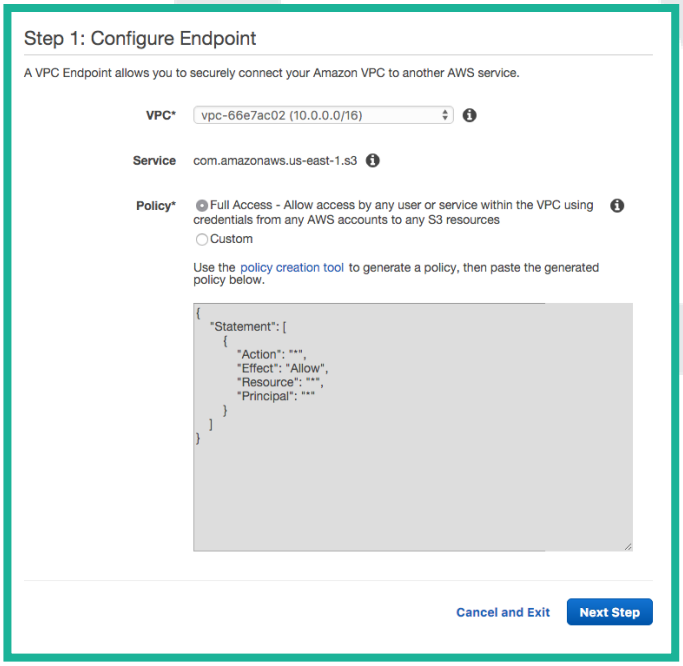
Configure:

```
aws configure
```

Leave the default `[None]` for the **AWS Access Key ID** and the **AWS Secret Access Key**. Set the **Default region name** to `us-east-1`. Leave the **Default output format** set to `[None]`.

Now, create a bucket:

```
aws s3 mb s3://linuxacademy-lab-123
```



Change `123` to any series of numbers not yet taken. Remember, S3 buckets must all have unique names.

All of this is possible through the use of the NAT gateway. However, there are some issues with this: We are still accessing the open Internet for something that can be more secure, and we are relying on the bandwidth limitations of the NAT gateway. Would it not then be ideal to use, instead, AWS's backend architecture to not have to use the open Internet to communicate with Amazon S3?

From the **VPC Dashboard**, move to **Endpoints**, and press **Create Endpoint**. Select your *Private VPC* from the drop-down list, press **Next Step**. We now need to associate it with the subnet from which we are making requests. However, here is where things

begin to diverge from the NAT gateway system: This requires a route from the *private* subnet. Select the private subnet, and then click **Next**. View **Endpoints**.

Any uploading or downloading done through the API will now not have to go through the open Internet, but will instead communicate through these endpoints.

To prove that this is working, remove the NAT gateway, then run `aws s3 ls` from your terminal to list your S3 buckets. This should work even without the NAT gateway because it is running internally through the endpoints. Please note this will not allow you to upload to different regions outside of the VPC's region.