**Lesson 7 Demo 5**

**Restrict Access using VPC Endpoint**

**Objective:** To provide access to public and private subnets using the VPC endpoint

**Tools required:** AWS Lab

**Prerequisites:** None

**Steps to be followed:**

1. Create a custom VPC and enable DNS hostname
2. Create an Internet Gateway
3. Create 2 subnets
4. Create a Route table and attach them to subnets
5. Launch 3 instances
6. Launch public and private VM
7. Install DB on a private DB server
8. Create a bucket and upload a document
9. Create an Endpoint

**Step 1: Create a custom VPC and enable DNS hostname.**

* 1. In the AWS console, click on the **Services** and select **VPC.**

A screenshot of a computer

Description automatically generated

* 1. Click on the **Create VPC** button. In the **Name tag** enter the name.

Graphical user interface, text, application, email

Description automatically generated

* 1. In the **IPv4 CIDR** enter the **10.0.0.0/16** then click on the **Create VPC** button.

Graphical user interface, text, application

Description automatically generated

* 1. Click on the **Actions** and select the **Edit DNS hostnames** option.

A screenshot of a computer

Description automatically generated

* 1. Select **Enable** and click on the **Save changes** button.

Graphical user interface, text, email

Description automatically generated

**Step 2: Create an Internet Gateway**

* 1. Select the **Internet gateways** and click on the create button.

A screenshot of a computer

Description automatically generated

* 1. In the **Name tag,** enter the name and click on the **Create internet gateway** button.

Graphical user interface, text

Description automatically generated

* 1. Select the internet gateway, click on the **Actions,** then select the **Attach to VPC** option.

A screenshot of a computer

Description automatically generated

* 1. Select the VPC created previously and click on the **Attach internet gateway** button.

Graphical user interface, text, application, Word, email

Description automatically generated

**Step 3: Create 2 subnets**

* 1. Select the **Subnets** and click on the **Create subnet** button.

A screenshot of a computer

Description automatically generated

* 1. Select the **VPC ID**.

Graphical user interface, text, application

Description automatically generated

* 1. Enter the name and **IPv4 CIDR,** then click on the **Create subnet** button.

Note:

Enter IPv4 10.0.1.0/24

Graphical user interface, text, application, Word

Description automatically generated

* 1. Follow the above same procedure for creating a **private subnet.**

Note:

Enter Ipv4 10.0.2.0/24

Graphical user interface, text, application, Word

Description automatically generated

* 1. Select the **public-subnet,** click on the **Actions** tab and select **Edit subnet settings.**

Graphical user interface, text

Description automatically generated

* 1. Select the **Enable auto-assign public IPv4 address** and click on the **Save** button.

Graphical user interface, text, application

Description automatically generated

**Step 4: Create a Route table and attach them to subnets.**

* 1. Select the **Route** **table** from the left pane and click on the **Create route table**.
  2. Enter the name and VPC then click on the create button.

Graphical user interface

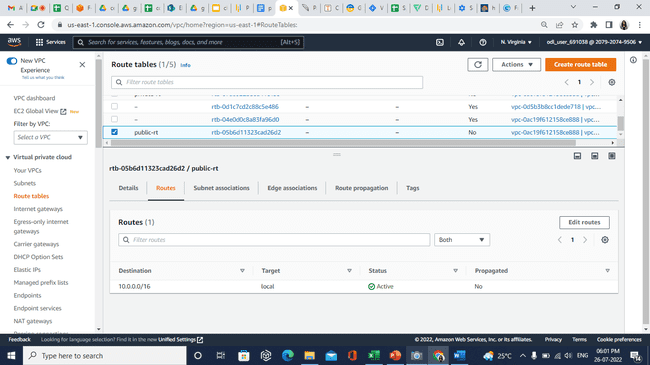
Description automatically generated

* 1. Follow the same above procedure for creating **private-rt.**

Graphical user interface, text, application

Description automatically generated

* 1. Select the public route and click on the **Routes** tab.



* 1. Click on the **Edit routes** button.
  2. Click the **Add route** button, select the **Internet Gateway,** then click on the **Save changes** button.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. Once the route to the Internet is set, click on the **Subnet associations,** and select the **Edit subnet associations** button.

Graphical user interface, text, email

Description automatically generated

* 1. Select the **public-subnet** and click on the **Save associations**.

Graphical user interface, text

Description automatically generated

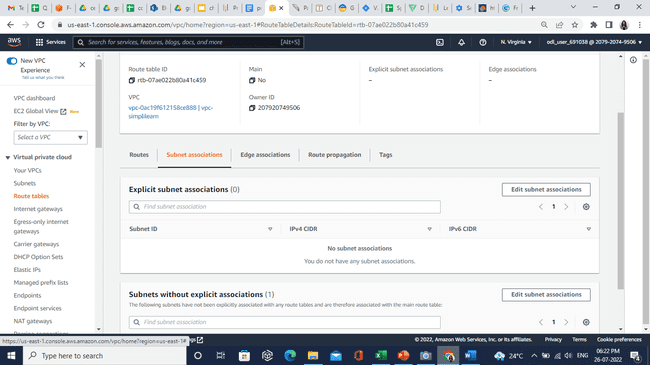
* 1. Attach the **private route table** to the **private subnet** using the previous steps.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



A screenshot of a computer

Description automatically generated

**Step 5: Launch 3 instances**

* 1. In the dashboard, search and select the **EC2**.
  2. Click on the **Launch instance**.
  3. Enter the name.

Graphical user interface, text, application, email

Description automatically generated

#### For the OS search and select the **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type option.**

Graphical user interface, text, email

Description automatically generated

* 1. Click on the **Create new key pair.**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, email

Description automatically generated

* 1. In the **Network settings** select the information as shown in the images below:

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* 1. Click on the **Launch instance** button.
  2. Create a new instance for the private subnet.
  3. Enter the name and select the key pair created previously.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. Create a NAT instance using the previous steps.

Graphical user interface, text, application, email

Description automatically generated

* 1. Select **amzn-ami-vpc-nat-hvm-2018.03.0.20181116-x86\_64-ebs** option.

Graphical user interface, text, application, email

Description automatically generated

* 1. For network setting, refer to the following images:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. For the security group, refer following images and launch instance:

Enter the security group SSH, HTTP, and HTTPS enter source IP-10.0.1.0/24 and again create **Custom TCP, HTTP, and HTTPS** using the following details:

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Select the Nat-instance, click on the Actions, select the Networking, and click on the Change source/destination check.

Graphical user interface, text, email

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Go to the VPC route table, select the private route and choose the **Routes** tab.

A screenshot of a computer

Description automatically generated

* 1. Click on the Edit routes button and select the **NAT instance created previously.**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 6: Launch public and private VM**

* + 1. Go to the EC2 console and select the public instance.
    2. Copy the public IP and paste it into the **CloudShell** using the following command:

**chmod 400 demo.pem**

Note:

If you are getting an error, then in **CloudShell** in the **Actions** tab, click on the upload file and upload the **.pem** file.

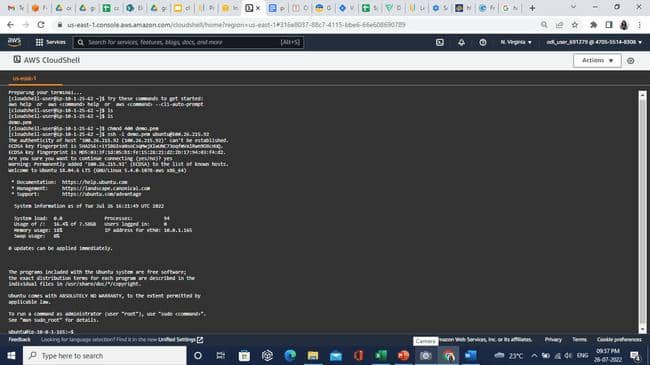
Replace .pem name.

Graphical user interface, text, application, email

Description automatically generated

* 1. Copy and paste the following command in CloudShell:

**ssh -i demo.pem** [**ubuntu@100.26.215.92**](mailto:ubuntu@100.26.215.92)(replace the public ipv4 address accordingly):



* 1. Enter the following command for ssh to private VM using public VM.

**sudo su**

**ping 10.0.2.92**

**Note:**

Replace 10.0.2.92 with private instance ip.

Graphical user interface, text

Description automatically generated

**Step 7: Install DB on a private DB server**

* 1. To install DB, enter the following command:

**wget https://storage.googleapis.com/skl-training/aws-codelabs/mattermost/install\_mysql.sh**

**chmod 700 install\_mysql.sh**

**sudo ./install\_mysql.sh**

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

**Step 8: Create a bucket and upload a document.**

* 1. Create a bucket.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* 1. Upload the document, **Object Actions** and then click on the **Make public using ACL**.
  2. Click on the **Make public** button.

A screenshot of a computer

Description automatically generated

**Step 9: Create an Endpoint**

* 1. Open the VPC, select the Endpoints and click on the **Create endpoint**.

A screenshot of a computer

Description automatically generated

* 1. Enter the name.

Graphical user interface, text, application, Word

Description automatically generated

* 1. Search and select the **com.amazonaws.us-east-1.s3** option.

Graphical user interface, text, application, Word

Description automatically generated

* 1. Select the **VPC** and default **Route table** then click on the create button.

A screenshot of a computer

Description automatically generated

* 1. Launch the private VM using the following command:

**ping 10.0.2.11**

Note:

Replace 10.0.2.11 with the private instance IPv4.

Graphical user interface

Description automatically generated

* 1. Copy and paste the Object URL of the bucket document using the following command:

wget <https://demobucketsimplilearn.s3.amazonaws.com/bucketdoc.txt>

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

* 1. Now, in bucket policy enter the following code:

{

"Id": "VpcSourceIp",

"Version": "2012-10-17",

"Statement": [

{

"Sid": "VpcSourceIp",

"Action": "s3:\*",

"Effect": "Deny",

"Resource": [

"arn:aws:s3:::DOC-EXAMPLE-BUCKET",

"arn:aws:s3:::DOC-EXAMPLE-BUCKET/\*"

],

"Condition": {

"NotIpAddress": {

"aws:VpcSourceIp": [

"10.1.1.1/32",

"172.1.1.1/32"

]

}

},

"Principal": "\*"

}

]

}

Note:

The above example of bucket policy blocks traffic to the bucket unless the request is from a specified private IP address (aws: VpcSourceIp).

After this edit, only wget in VMS in this VPC will work. VMS in any other VPC, for example, the default VPC, will not be able to access this bucket.