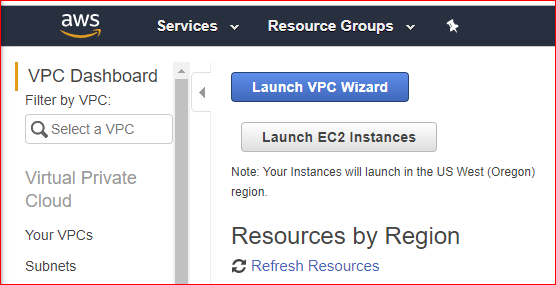
**ACIT 3640 - Cloud Computing**

**(Lab 8)**

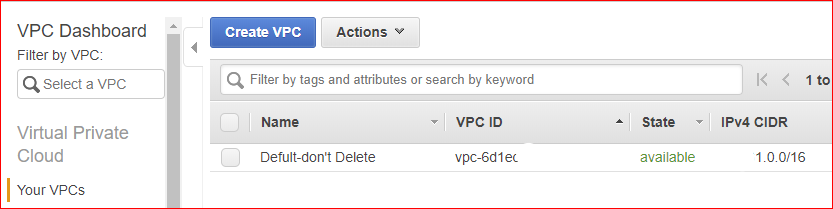
**Virtual Private Cloud (VPC), Subnets, NAT Instance, and NAT Gateway**

Note: If you are doing this lab from a Mac or Linux machine, see the note at the end of this document.

Before starting the lab, name the default VPC to “Default-don’t delete”. Go to VPC dashboard and select your VPCs on the left side.



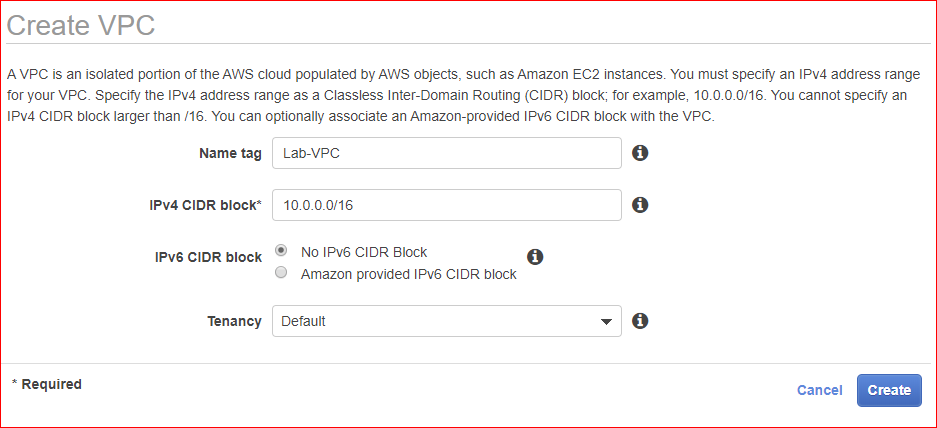
You will see one VPC, name it as “Default-don’t delete”.



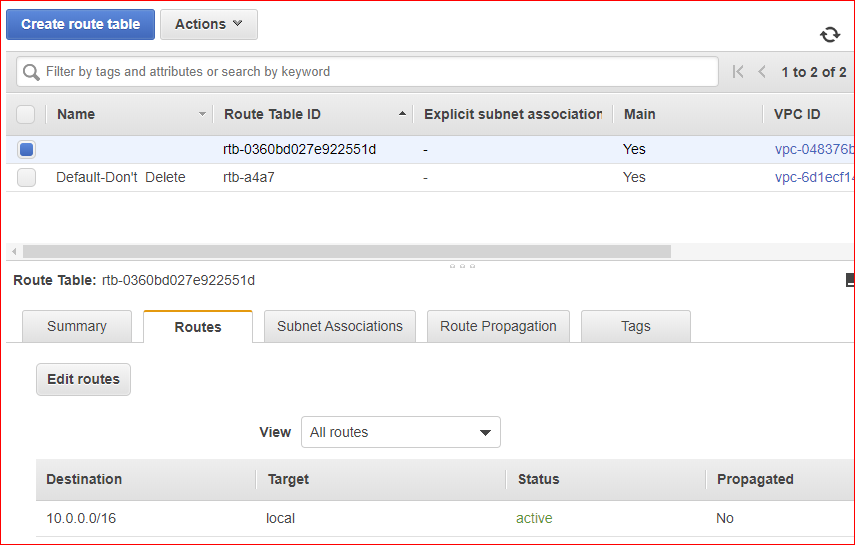
In VPC dashboard select Rout Tables and name the default routing table as “Default-don’t delete”.

Part I – VPC, Subnets, and NAT Instance

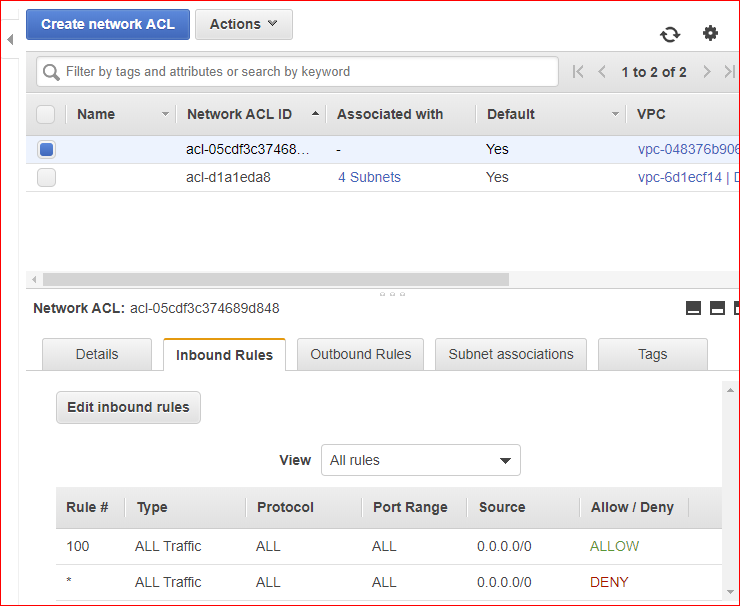
1. Log into the AWS console.
2. Under Networking & Content Delivery, select VPC.
3. Create a VPC without using the wizard. Do this by selecting “Your VPCs” on the left menu. Then, click the blue “Create VPC” button at the top of the screen.
4. In the Create VPC screen, enter a Name tag, e.g., “Lab-VPC,” a CIDR block of 10.0.0.0/16 and accept “Default” for the Tenancy option. Note: You can create a maximum of five (5) VPCs per region. This includes the default VPC.



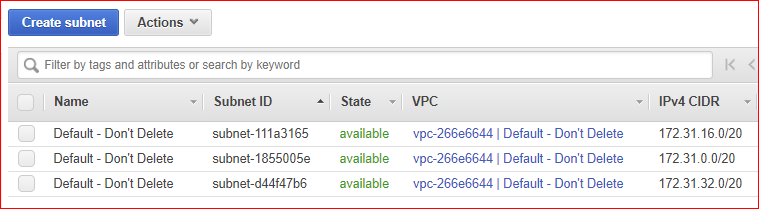
1. Examine the routing tables created automatically for the VPC. (On the left side, select Route Tables.)



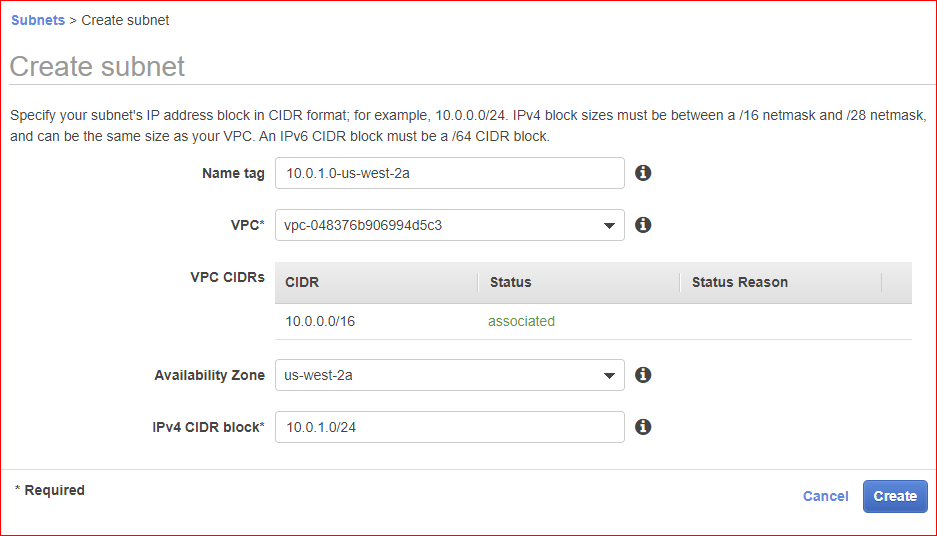
1. You should also note that a default Security Group and Network ACL (access control list) have been created for your VPC. This screen shot shows the Network ACL. Examine your Outbound Rules. What kind of traffic are you allowing into and out of our VPC?



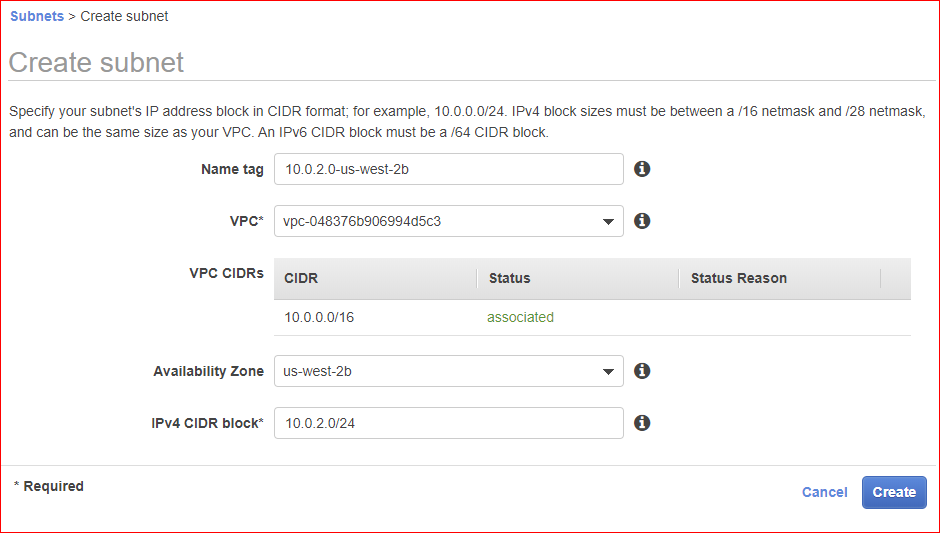
1. On the left side, select Subnets. Note: You will have 3-4 default VPCs in your region. (See the Default Subnet column on the far right.) Don’t delete them!



1. Click the blue “Create Subnet” button at the top of your screen.
2. For the name tag, I suggest a name that includes the subnet address range and Availability Zone (AZ), e.g., “10.0.1.0-us-west-2a.”
3. In the VPC field, select the VPC you created earlier (lab-vpc).
4. Select an Availability Zone. Subnets are always mapped to one AZ.
5. Set CIDR block to 10.0.1.0/24.



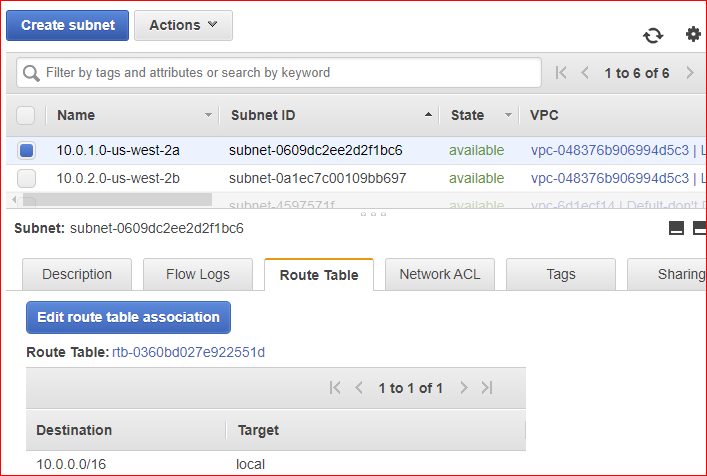
1. How many IP addresses are available in this subnet? Remember, AWS reserves five (5) IP addresses for internal use.
2. Create another subnet with the following CIDR block: 10.0.2.0/24. Select a different AZ. I selected us-west-2b. See below:



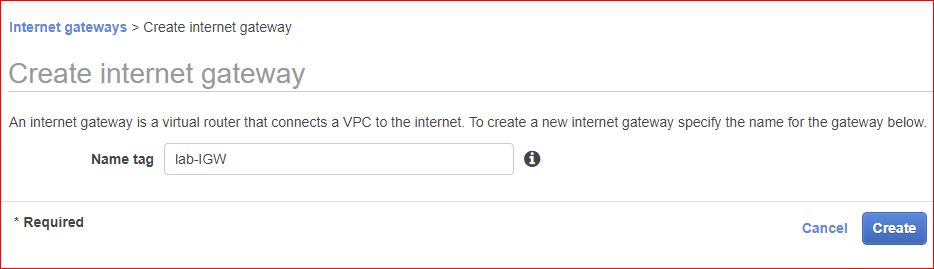
1. The first subnet will be a public subnet (10.0.1.0/24) and the second subnet (10.0.2.0/24) will be a private subnet. For the public subnet, we want to auto-assign the public IP addresses. (See the far right column on the Subnets screen.)
2. Select the public subnet (10.0.1.0/24) and click the “Subnet Actions” button at the top of your screen. Select “Modify Auto-Assign Public IP,” click the “Enable auto-assign Public IP” check box, and click Save. See below.



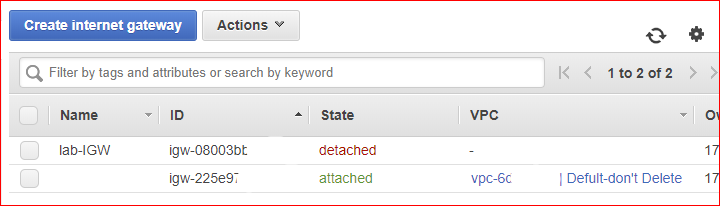
1. Now, look at the Route Table for each subnet. They should each look like the screen shot below.



1. Add an Internet Gateway (IGW) to your VPC. Do this by selecting Internet Gateways on the left side. You will see that your Default VPC already has an Internet Gateway (IGW).
2. Click the blue “Create Internet Gateway” button at the top of the screen.
3. In the “Name tag” field, enter something like “Lab-IGW.” Click “Yes, Create.”

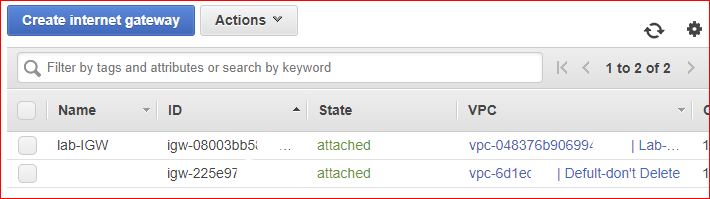


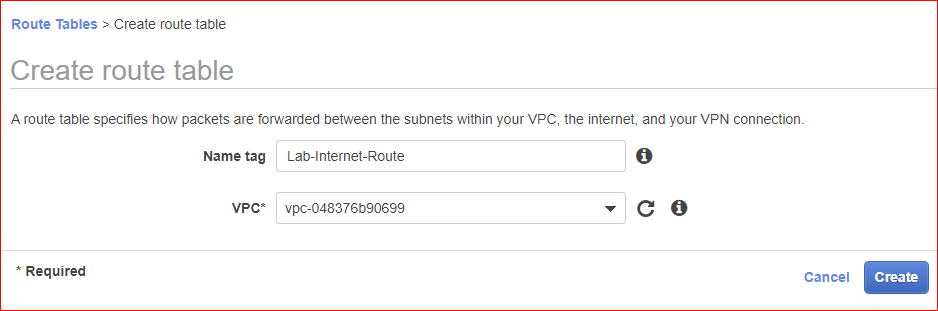
1. Attach the IGW to your VPC. Click the “Attach VPC” button at the top of the screen.) There can only be one IGW per VPC. Remember, the instances that you will create in the VPC will need access to the Internet. See the screen shots, below.

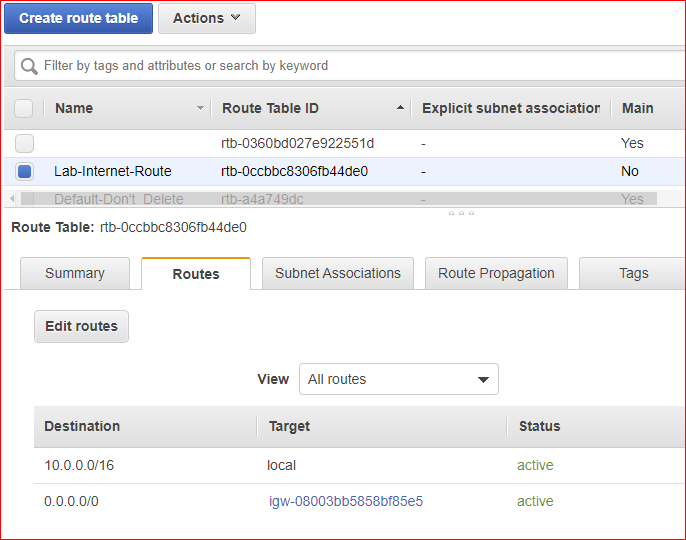


From the Actions menu, select Attach to VPC. Then select the Lab-VPC.

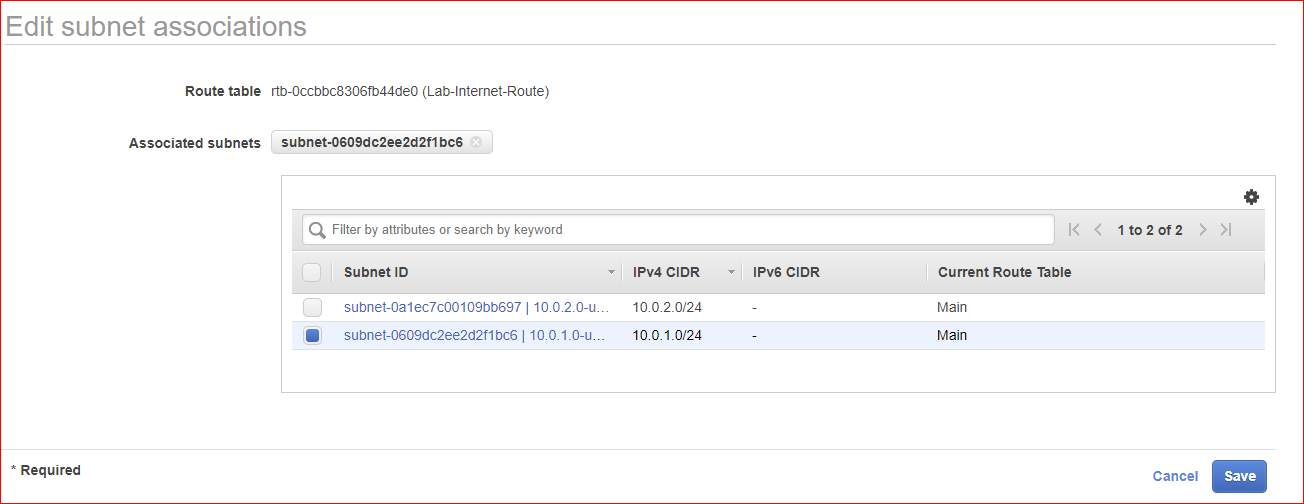
This shows on IGW attached to our VPC.



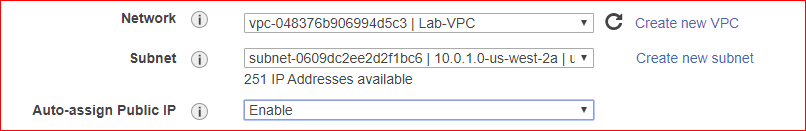
1. Click Route Tables on the left hand side. You are going to create a route table that will give the IGW access to the Internet.
2. Click “Create Route Table” at the top of the screen. You can name it something like “Lab-Internet-Route.” Make sure it attaches to the VPC you created. 
3. With the route table selected, click on the Routes *tab* below the route table.
4. Click “Edit routes” button. You want to create a route out to the Internet.
5. Click the “Add route” button.
6. In the Destination field enter “0.0.0.0/0” and for Target select the IGW you created. (All traffic can go out.) Click the “Save” button you are done.



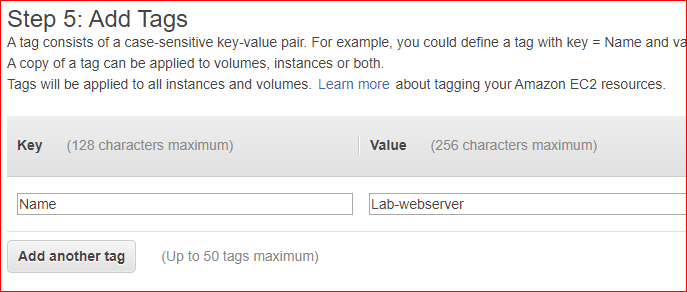
1. How would you describe how the two Destination CIDR blocks work?
2. Click on the “Subnet Associations” *tab* for your route table. No subnets should be currently connected to your new Route table. Now, you need to decide which subnet you want to give access to the Internet.
3. Click on “Edit subnet association” button. From the list of two subnets, select the 10.0.1.0/24 subnet and click “Save.” This means the 10.0.1.0/24 subnet is now the web facing DMZ (demilitarized zone). A DMZ is also called a perimeter network. See the screen shots, below.



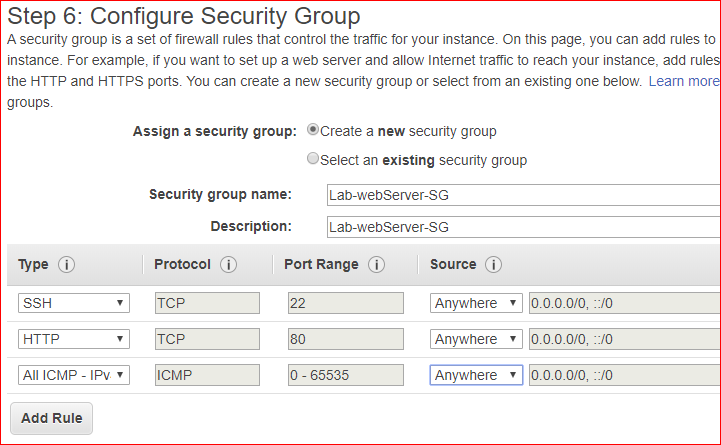
1. Now you are going to deploy two Amazon Linux EC2 instances. One instance, a web server, will be deployed into the public subnet (10.0.1.0/24). The second instance, a database server, will be deployed into the private subnet (10.0.2.0/24). Both instances will be located in the VPC you created.
2. Go to your EC2 Dashboard. Launch an Amazon Linux AMI (Free Tier). This will be the web server.
   1. In “Step 3: Configure Instance Details,” make sure you have selected your VPC for the Network field.
   2. For the subnet, select the public subnet (10.0.1.0/24).
   3. The “Auto-assign Public IP” field should be set to “Enable.”



* 1. If you want to create a bootstrap script to launch Apache, you can. It is not required for this exercise.
  2. Label the instance “Lab-WebServer.”

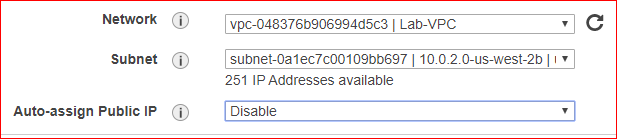


* 1. When you get to Step 6: Configure Security Group, click “Create a new security group.” Because this is a new VPC, you will have to create new security groups.
  2. For the webserver, create a new security group, and add protocols for SSH, HTTP and “All ICMP - IPv4.” For each port the source should be Anywhere. Call the security group “Lab-WebServer-SG.”

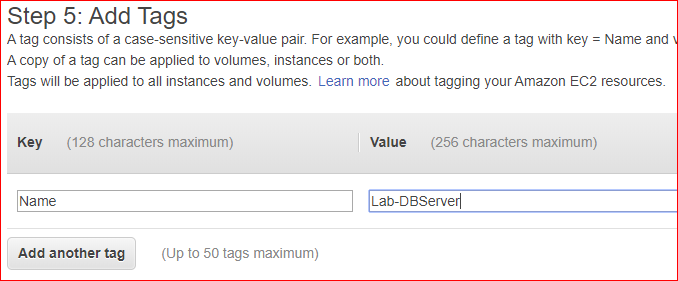


* 1. Select your Key Pair before launching.

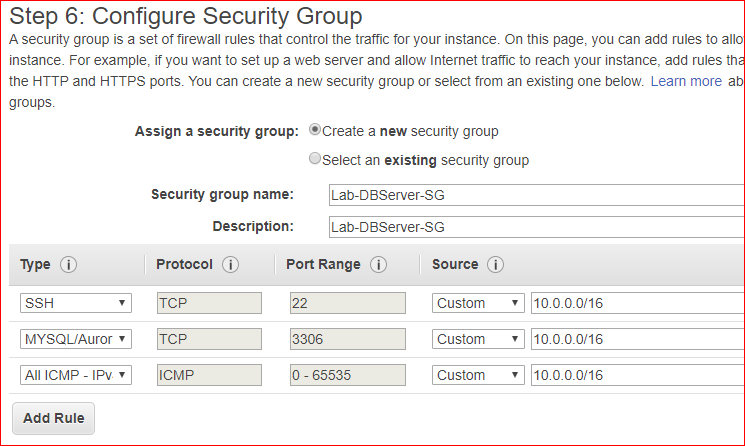
1. Create another new, Amazon Linux AMI. This will be the database server. This instance will be deployed to the private subnet (10.0.2.0/24).
   1. In “Step 3: Configure Instance Details,” make sure you have selected your VPC for the Network field.
   2. For the subnet, select the private subnet (10.0.2.0/24).
   3. The “Auto-assign Public IP” field should be set to “Disable.”



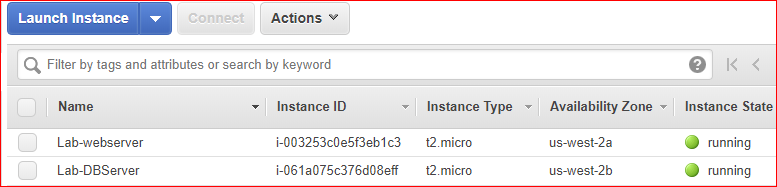
* 1. Label the instance something like “Lab-DBServer.”



* 1. Create a new security group that has SSH, MYSQL/Aurora, and “All ICMP – IPv4” traffic. Enter the CIDR address for the VPC (10.0.0/16) in the source field for all three protocols.



* 1. Select your Key Pair and launch your instance.
  2. You should now have your two “Lab” instances running. The WebServer should have a public IP address, but the DBServer should not have a public IP address. They should also be in different Availability Zones.

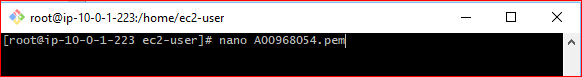


1. From your laptop, connect (SSH) into the *webserver* instance using the Public IP address. Run the following commands. These commands test whether or not the webserver has Internet access.
   1. “sudo su”
   2. “yum update –y”
   3. Ping the database server instance using it’s *private* IP address. The database server instance should not have a public IP address, only a Private IP.

“ping 10.0.2.224” [your IP address will be different]

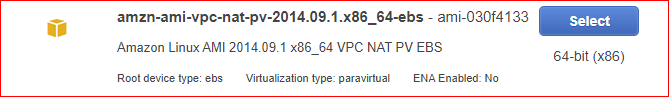
* 1. Do not close this SSH connection to the web server!

1. Now, you want to access the database server directly from the webserver instance. You need to do this because the database server does not have a public IP address. (Make sure you are still SSHed into the web server.) To do that, you need to create a .pem file on the webserver. Find your .pem file that is stored on your laptop or workstation. Open the .pem file using a simple text editor like Notepad. Copy the contents of the file into a copy buffer. Now, from within the Webserver instance, do the following:
   1. Run “nano <yourKeyPair.pem>”

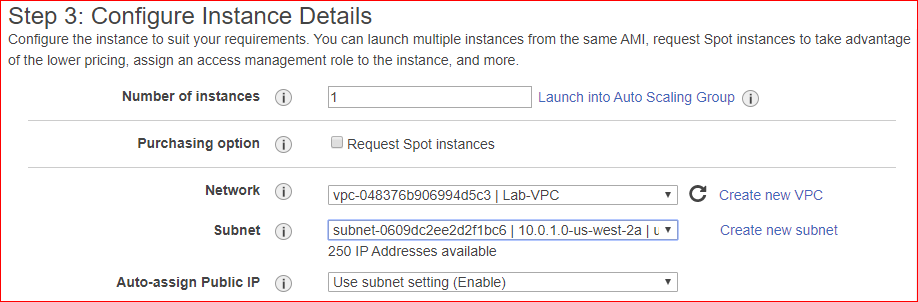


* 1. Paste the contents of your copy buffer into this file and save it.
  2. Run “chmod 400 <yourKeyPair.pem>” This changes the file permissions.

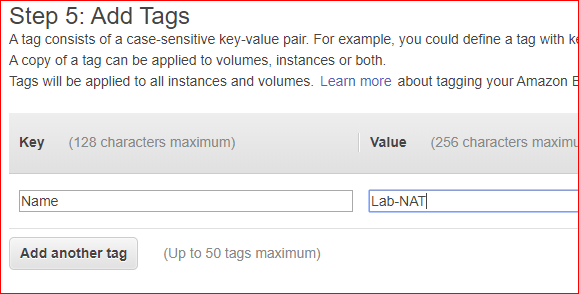
1. Next, from *within the webserver,* SSH into the database server instance using the Private IP address of the database server Here’s the command to do that: “ssh –i <yourKeyPair.pem>ec2-user@<private IP of database server >” You should now be attached to the database server. Run these commands:
   1. “sudo su”
   2. “yum update –y”
   3. Was the update successful? Should it be? Why or why not? Quiz question.
2. LEAVE this SSH session active. Don’t terminate!
3. Now, you are going to give the database server access to the Internet. You are going to do this by creating a NAT instance. NAT is an acronym for “Network Address Translation.” The NAT instance will be create in the public subnet (10.0.1.0/16).
4. Go to the EC2 dashboard. You are going to create NAT instance. Click “Launch Instance.”
   1. On the left side of your screen, click “Community AMIs.”
   2. Enter “nat” in the search field.
   3. Select the instance with a name like “amzn-ami-vpc-nat-pv-xxxxx.”. This will be an Amazon AMI.



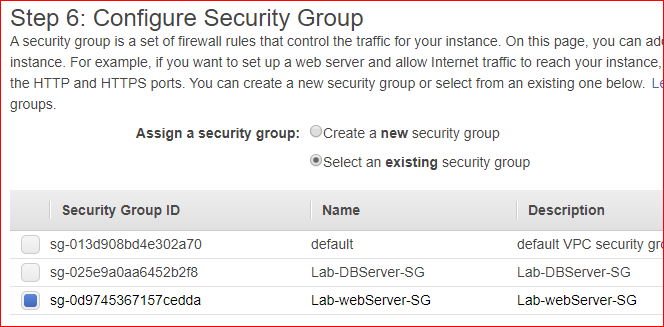
* 1. Select the Free Tier eligible t1.mico. You may not have a choice for another instance type. Note: There is more network throughput if you choose a larger instance type.
  2. In Step 3, place the instance in your VPC and place it in the *public* subnet (10.0.1.0/24).



* 1. Label the instance something like “Lab4-NAT.”

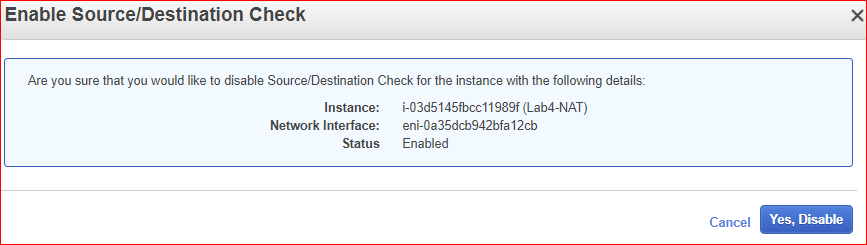


* 1. Select the security group you used for the webserver (SSH, HTTP, ICMP).



* 1. Select your Key Pair and launch.
  2. Wait for the NAT instance to pass its two status checks.

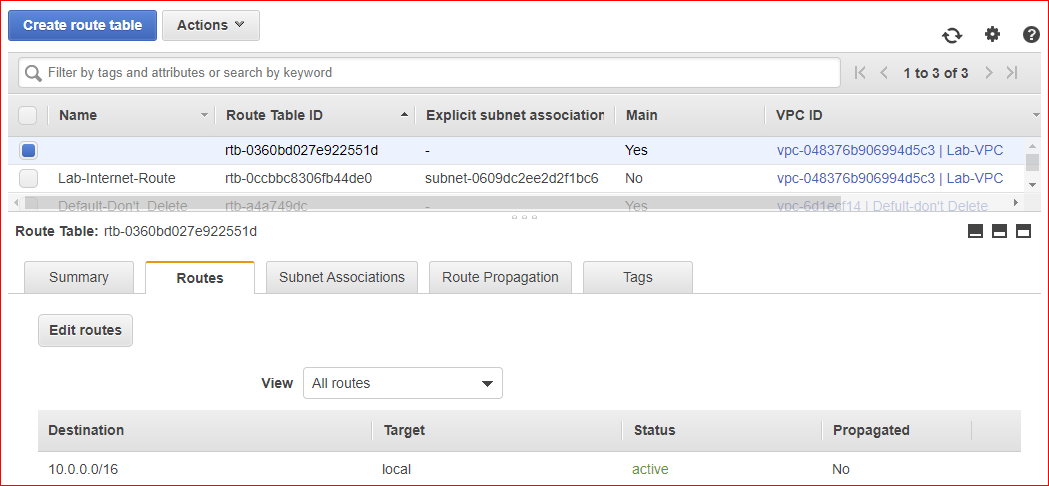
1. From the EC2 dashboard, select the NAT instance. Click on the “Actions” button and go to “Networking -> Change Source/Dest. Check.”



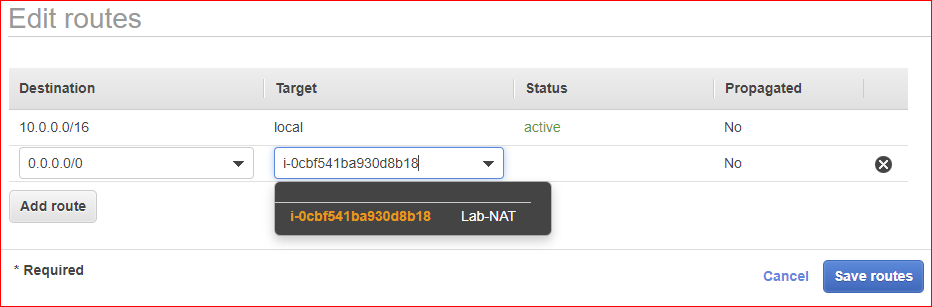
1. Click “Yes, disable.”

You must do this before you can change your route table. You don’t want the NAT instance to be the source or destination of network traffic. You only want to route traffic to go through the NAT instance.

1. You need to enable servers in the *private* subnet to communicate with the Internet through the NAT instance. To do this, you need to create a new route.
2. Go back to your VPC and click on the Route Tables. You currently have an Internet Gateway route table (Lab-Internet-Route) and a few others.
   1. Select the 10.0.0.0/16 route table that is *not* connected to the IGW. It will still be in the VPC.



* 1. Click the “Edit routes” button. Click “Add route.”
  2. Add another route and make the destination 0.0.0.0/0. Select your NAT instance as the target.
  3. Click Save. See the screen shot below.

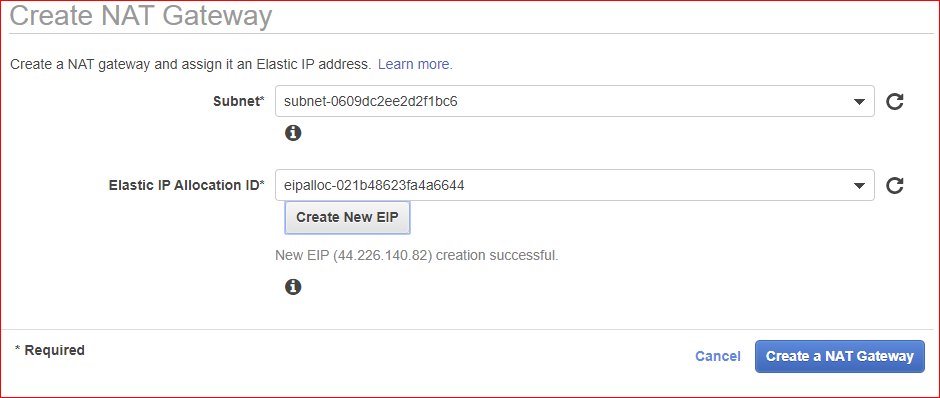


1. Go back to your SSH session. Remember, you are still SSHed into the database server through the webserver!
   1. Try running “yum update –y” again.
   2. Should this be successful? Why or not?
2. Leave everything running! Leave your SSH convection running!
3. STOP!!!! Get instructor approval. Show your instructor the terminal widow of your SSH connection attached to the database instance.

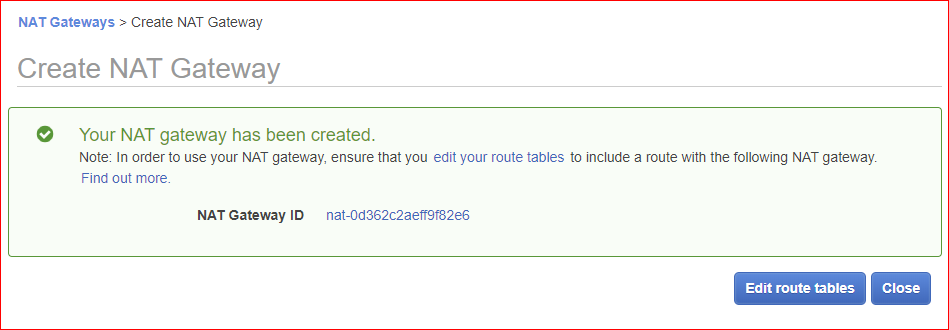
**Instructor’s initials: \_\_\_\_\_**

Part 2 – Create a NAT Gateway

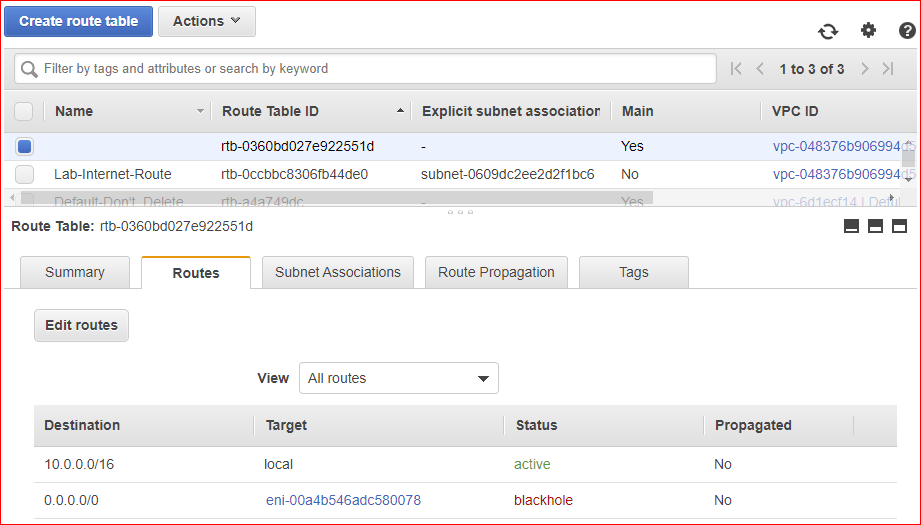
1. A NAT gateway is used instead of creating a NAT instance. Basically, NAT gateways are easier to configure than NAT instances. NAT gateways cost money. See VPC pricing here: https://aws.amazon.com/vpc/pricing/
2. From the EC2 Dashboard, terminate your NAT instance. Wait until the instance has a status of Terminated before proceeding.
3. From within the terminal window connected to the database server, run the following command: “ping 8.8.8.8”
   1. Were you successful? Why or why not?
4. Go to the Networking/VPC area of the Amazon console.
5. On the left side, select “NAT Gateways.”
6. Click the blue “Create NAT Gateway” button.
7. In the Subnet field, select the *public* subnet (10.0.1.0/24).
8. In the “Elastic IP Allocation ID” field, click the “Create New EIP” button.
9. Finish by clicking “Create a NAT Gateway.”



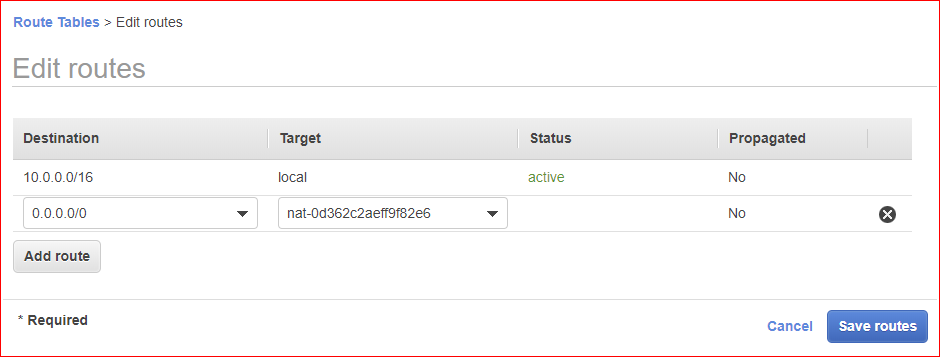
1. On the next screen, click “Edit Route Tables.”



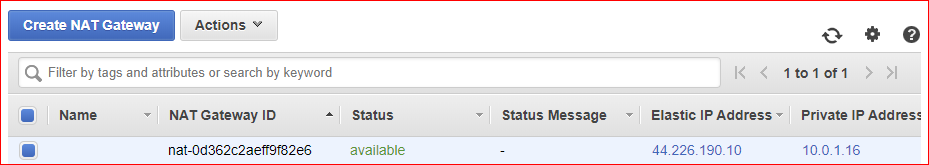
1. On the Route Tables screen, select the 10.0.0.0/16 route table that is *not* the IGW.
   1. Click the Routes tab and click the Edit routes button.



* 1. You will see the status of our previous route is labeled “Black Hole.” Why does it say Black Hole? You need to *delete* that route and create a new one.
  2. Make the destination 0.0.0.0/0 and select the NAT gateway as the target.
  3. Click Save. See screen shot below.



1. Click “NAT Gateways” on the left to view the status of your new NAT Gateway.

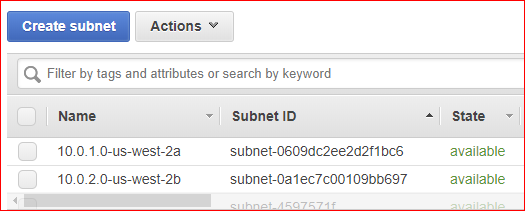


1. Go back to the terminal window containing the database server.
2. Install MySQL on the database server by running “yum install mysql –y”
3. Should this command work?
4. Show your instructor the terminated NAT instance and new NAT gateway. Show your instructor the terminal widow of your SSH connection attached to the database instance.

**Instructor’s initials: \_\_\_\_\_**

**Clean up:**

1. Terminate your EC2 instances.
2. Delete two subnets, public and private.



1. Delete Nat gateway and your VPC (lab-VPC). Don’t delete default VPC.

Note to Mac and Linux Users

In order for your pem file (key pair) to work, you are going to have to change the ssh\_config file located in the /etc/ssh directory on your Linux/Mac machine. The parameter “PasswordAuthentication” must be changed from Yes to No. You are going to have to make this change on each AWS instance as you tunnel from one instance to another.