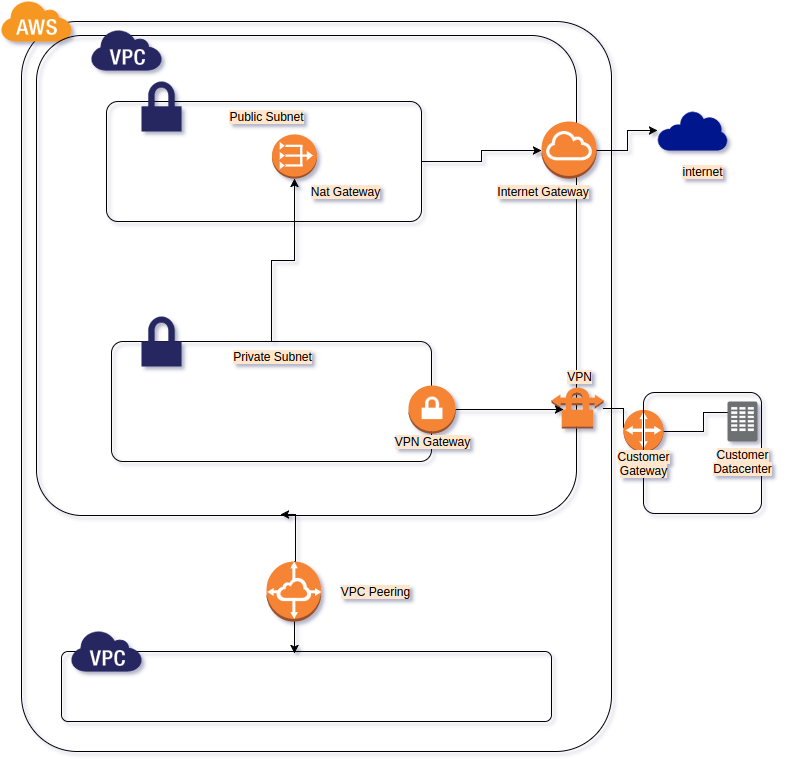
Amazon AWS VPC Introduction and Features

In this AWS VPC tutorial, we will see an introduction to AWS VPC. We will also look at some of the key features of Amazon VPC.

What is AWS VPC?

AWS VPC or Virtual Private Cloud is an Amazon service that allows you to create you own virtual network inside Amazon cloud and to use this virtual network to launch amazon resources. You can think of a VPC as your own network of machines and databases that live completely inside Amazon’s infrastructure but can be managed as if they were in your own data center. Here’s a diagram of a how a typical VPC setup looks like   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Amazon_VPC.png)

**Features of AWS VPC**

AWS VPC allows you to do the following. Not that this is not an exhaustive list, but highlights the important features:

Create multiple Virtual networks (VPC) inside Amazon cloud.

The VPC can span multiple regions and availability zones.

Create multiple subnets within each VPC. Each subnet, however, can be in only one availability zone.

The subnet can be private (not publicly accessible) or public (publicly accessible). The private subnet generally does not have public IP addresses.

Manage access to the subnet using route tables and Access control list.

Create Internet gateways to allow a subnet to be publically accessible.

Add NAT gateways to allow a private subnet to access the internet.

Create elastic IPs to attach to NAT gateways or other instances

Allow connection between two VPC using VPC peering.

Allow a secured private connection between a VPC and your own data center using a secured VPN connection. The secured connection as three parts:

A VPN gateway in VPC

The actual VPN connection

A customer gateway in the customer data center

This was just an introductory tutorial on Amazon VPC. In the next tutorial, we will look at how to create private and public subnets, add an internet gateway and add a NAT gateway.

AWS VPC Tutorial – Part I Introduction

In this AWS VPC tutorial, we will learn how to create a VPC; create public and private subnets that cover Multiple Availability zone; create CIDR blocks for each subnet; add Internet Gateways and NAT Gateways, and modify route table.

AWS VPC Tutorial – Introduction to Concepts

**What is AWS VPC**

Amazon VPC is your own private network inside Amazon’s cloud infrastructure. It is an alternative to maintaining your own data centre and is cheaper since it creates resources on demand. It is also more secure since Amazon takes care of the infrastructure security for you.

**What is a Subnet**

We will not go into much detail about subnets from a networking point of view, but for this tutorial, you should know that subnet is a part of your VPC that can contain resources that share a common subnet mask and that contain instances and resources that can normally only be accessed within that subnet except if you use an internet gateway to make them public.

**What is an Internet Gateway?**

An Internet Gateway allows you to make a subnet public by providing a route to the internet. All instances within the subnet can access the internet only through this gateway. Also, resources from the internet can access the instances in your subnet using this gateway.

**What is a NAT Gateway**

You can allow instances from your private subnet to connect to the internet using a NAT gateway. The instances in the private subnet do not have an IP address, so the NAT gateway translates the private IP to a public IP before routing the traffic out to the internet. NAT stands for Network Address Translation and it does just that – translates private IPs to public IP.

**What is a CIDR block**

CIDR or Classless Inter-Domain Routing is used to allocate IP address within a network. We will use CIDR blocks to mark a range of IP addresses for each subnet within a VPC. The VPC itself would have a CIDR block that lists all the IP addresses available with it.

**What is a Route table**

A route table contains rules for routing traffic within a subnet and from the subnet to outside world. Amongst other things, we use routing tables to add internet gateways and NAT gateways to the subnet.

**Problem Statement for AWS VPC Tutorial**

Here’s the network that we are planning to build today. It has the following components:

A VPC spanning a region

Two public and two private subnets in two Availability Zones (AZ). (one AZ contains one public and one private network)

Internet Gateways for each public subnet in each AZ

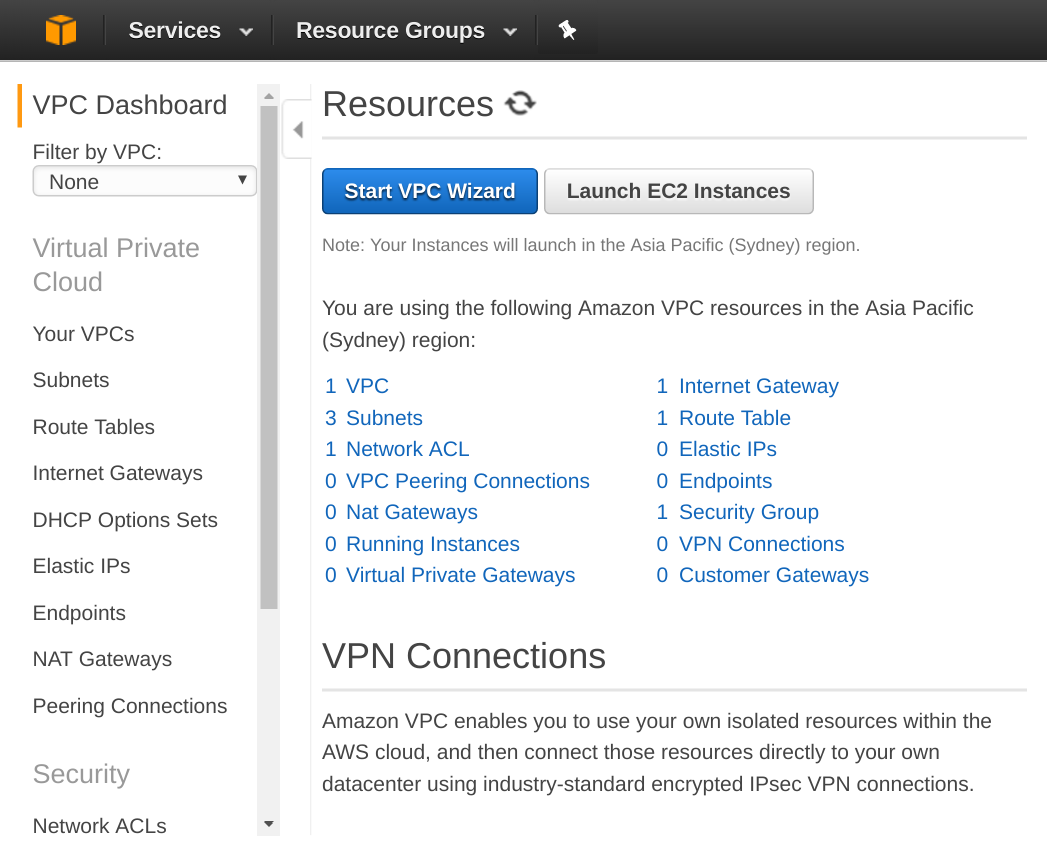
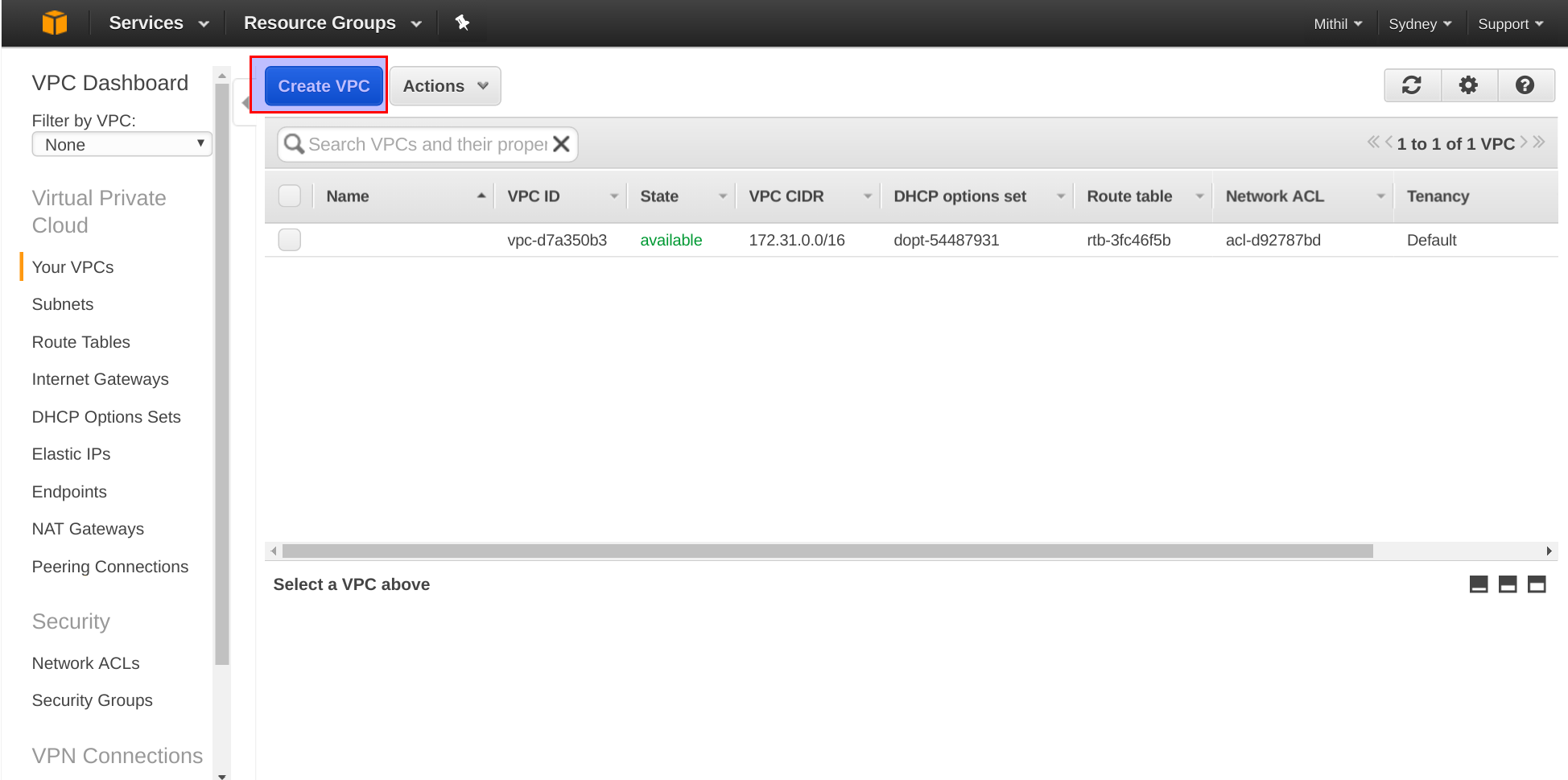
One NAT Gateway for each private subnet.

[](http://www.studytrails.com/wp-content/uploads/2016/11/VPCTutorial.png)

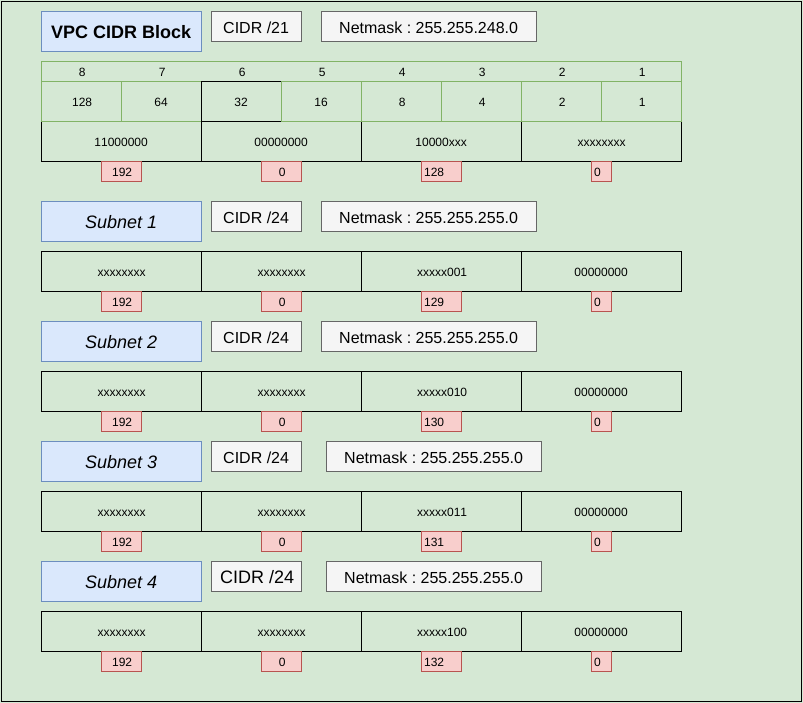
In the Next part, we will look at how to create the CIDR block for creating the subnets. We will also see how to create the VPC and the Subnets. In the third part we will see how to create internet gateways, NAT gateways and route tables.

AWS VPC Tutorial – Part II subnets

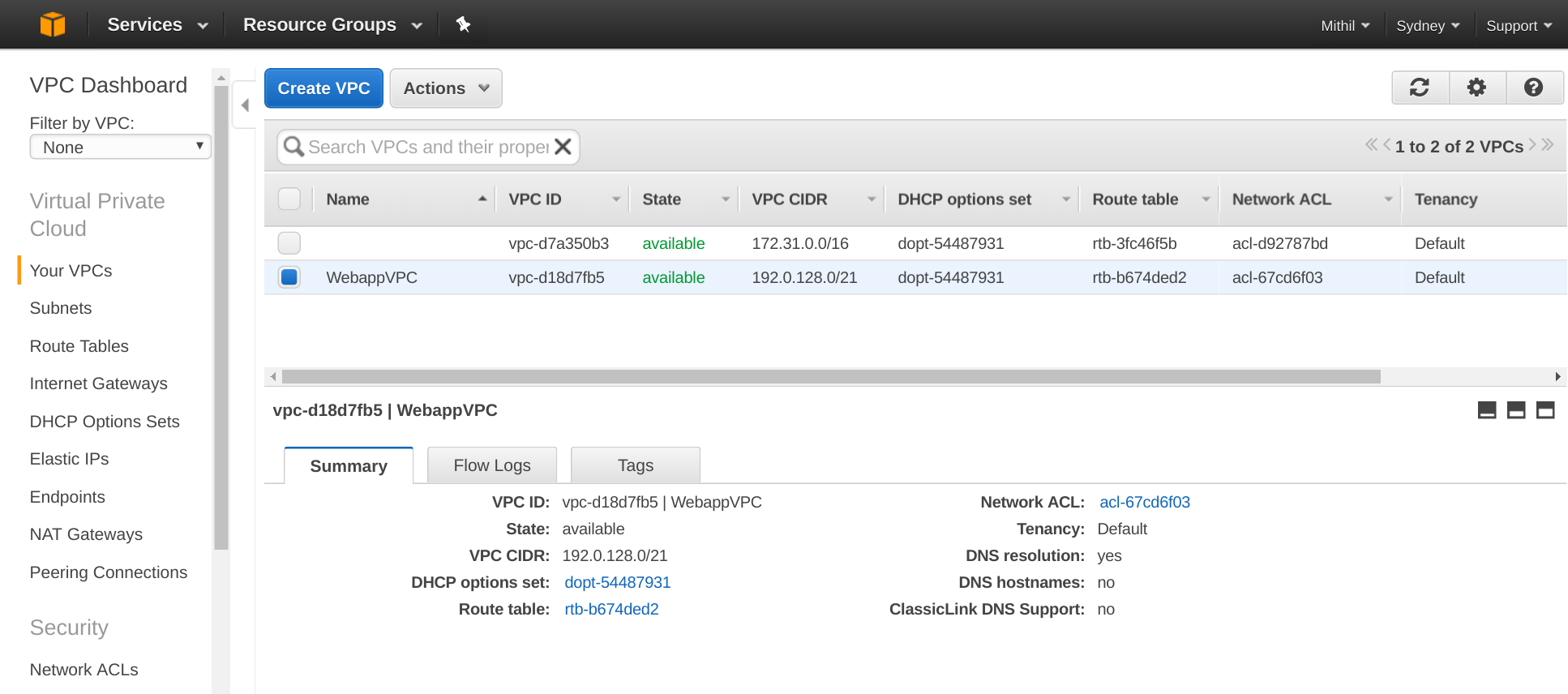
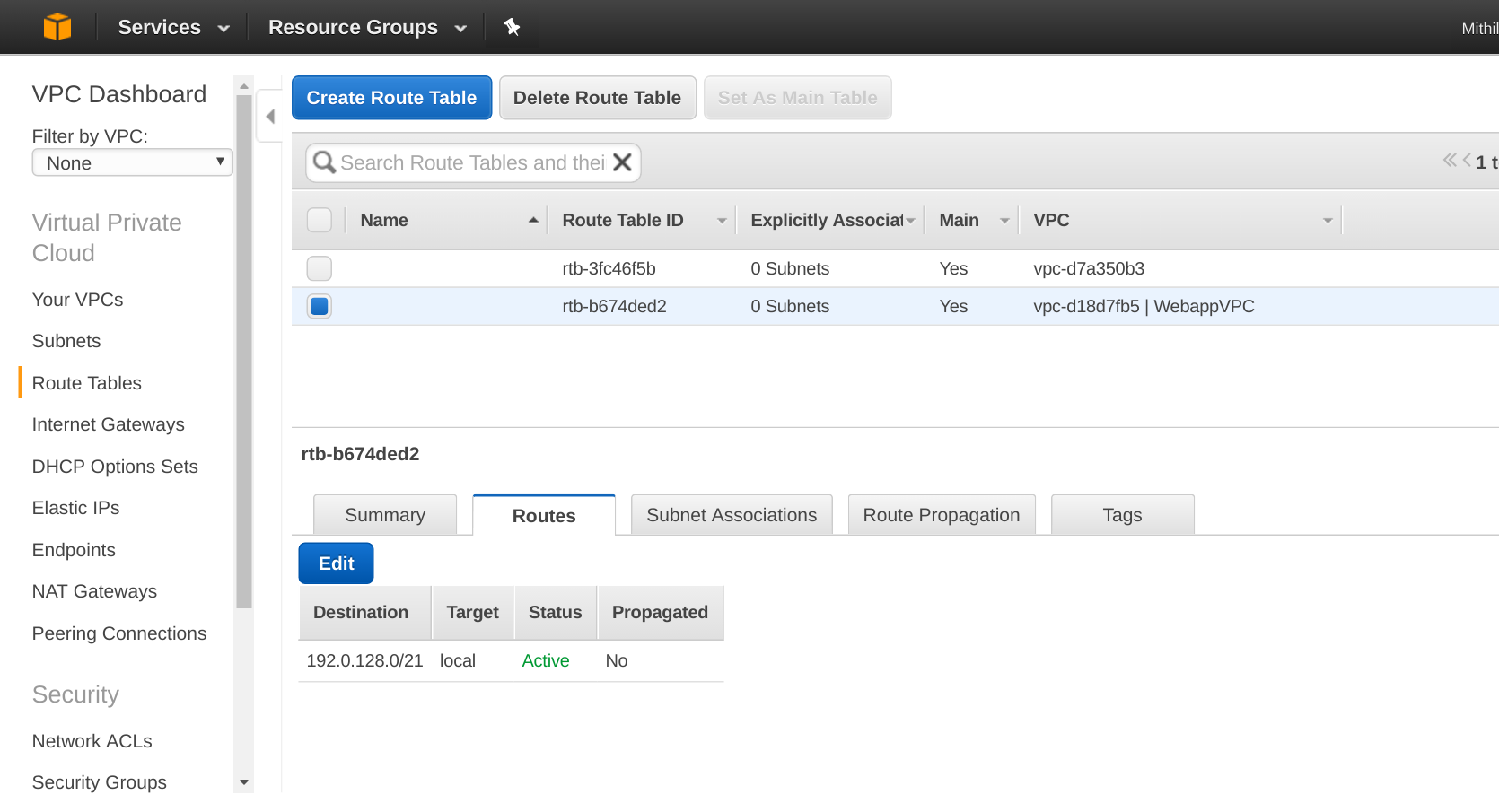
**AWS VPC tutorial – creating VPC**

We will look at how to create the VPC using the AWS management console. Login to the console and click on VPC. This is what you should see:   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_239.png)  
We will not be using the Wizard since we want to learn the inner workings of the VPC. Click on the link that says ‘1 VPC’. If you havent created a VPC before, you should still see 1 VPC which is the default that AWS creates for you. In the next screen you should see that one VPC and a button to ‘Create VPC’   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_240.png)

**Creating CIDR block for VPC and Subnets**

When you click ‘Create VPC’, you should see a popup where you can enter the VPC name and a CIDR block. This CIDR block determines the range of IP addresses that your VPC can have. It also specifies the network part of the IP addresses and the subnet mask. Here’s how our VPC and the subnets in the VPC will look like.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Subnets.png)

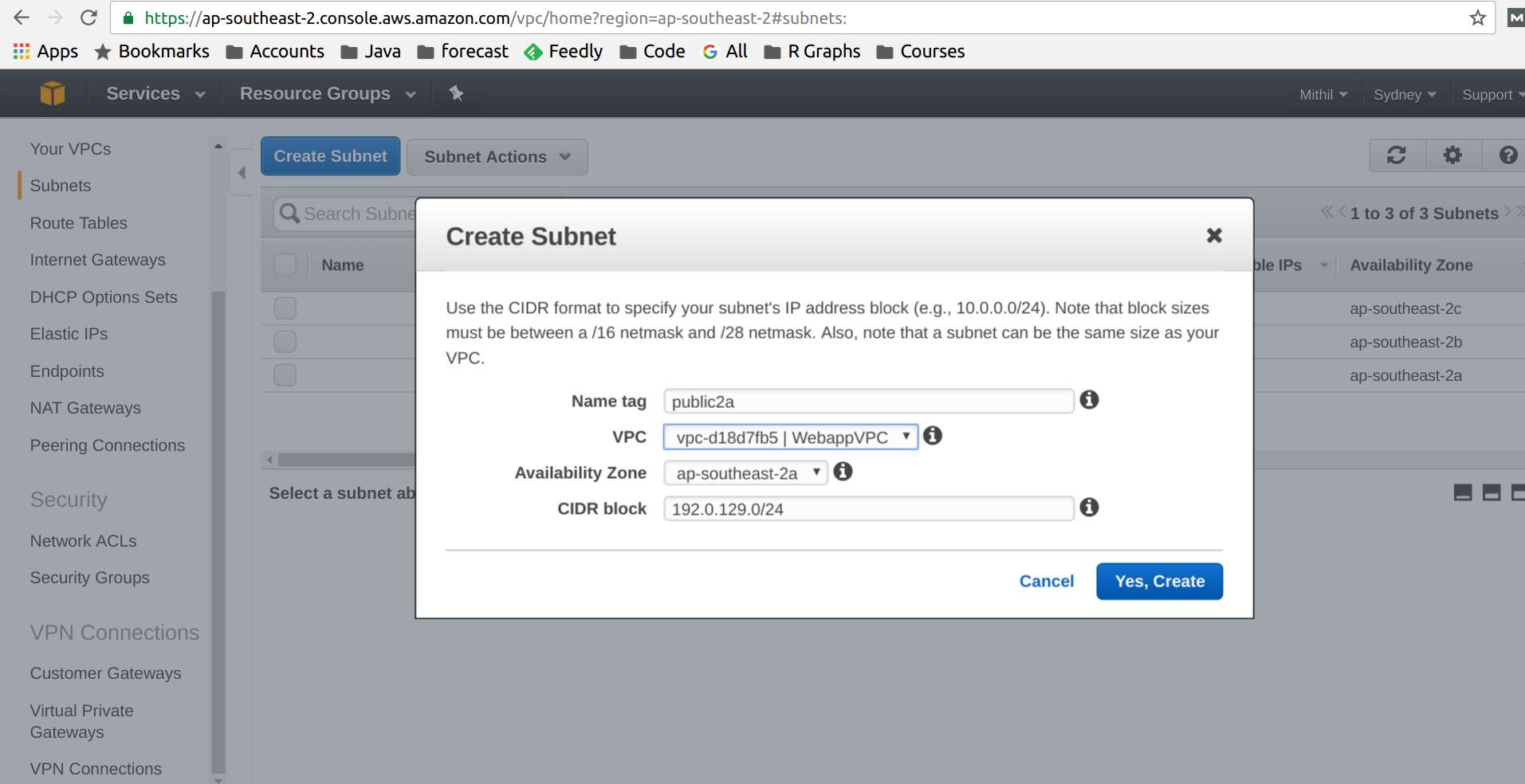
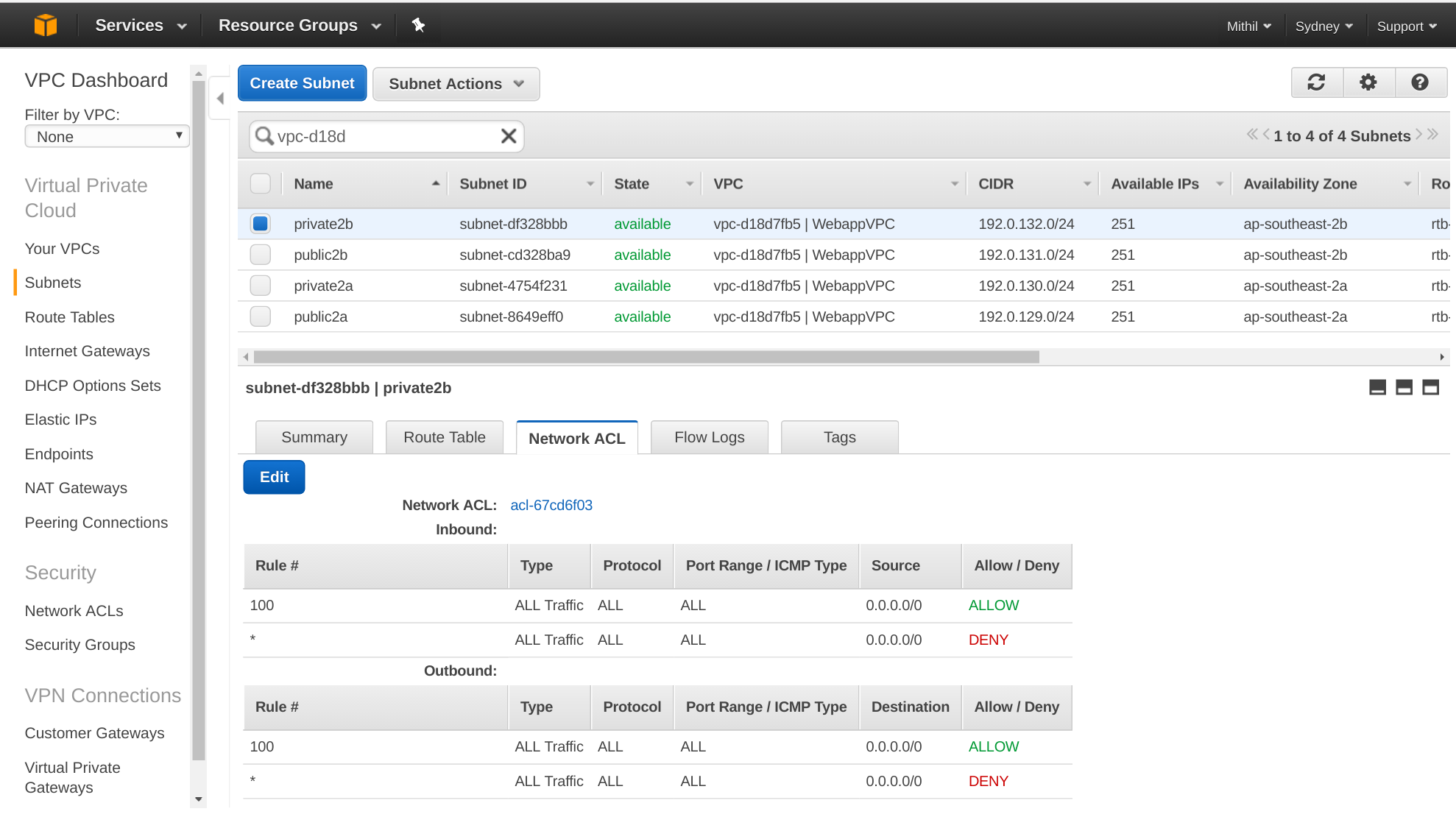
*CIDR block for the AWS VPC*

We create a CIDR block keeping in mind the number of IP addresses that we want in our VPC. In this example, we have chosen a CIDR block of 192.0.128.0/21 . What this means is that the first 21 bits of the 32 bits that form the IP address are part of the network. The remaining bits (11 ) are for the IP addresses in the VPC. Hit ‘yes, Create’ and it should create the VPC for you. When it creates the VPC it also creates a default route table and a default network ACL.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_242.png)  
The default Route table allows access to instances within the VPC. It does not allow access to instances outside the VPC.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_243.png)

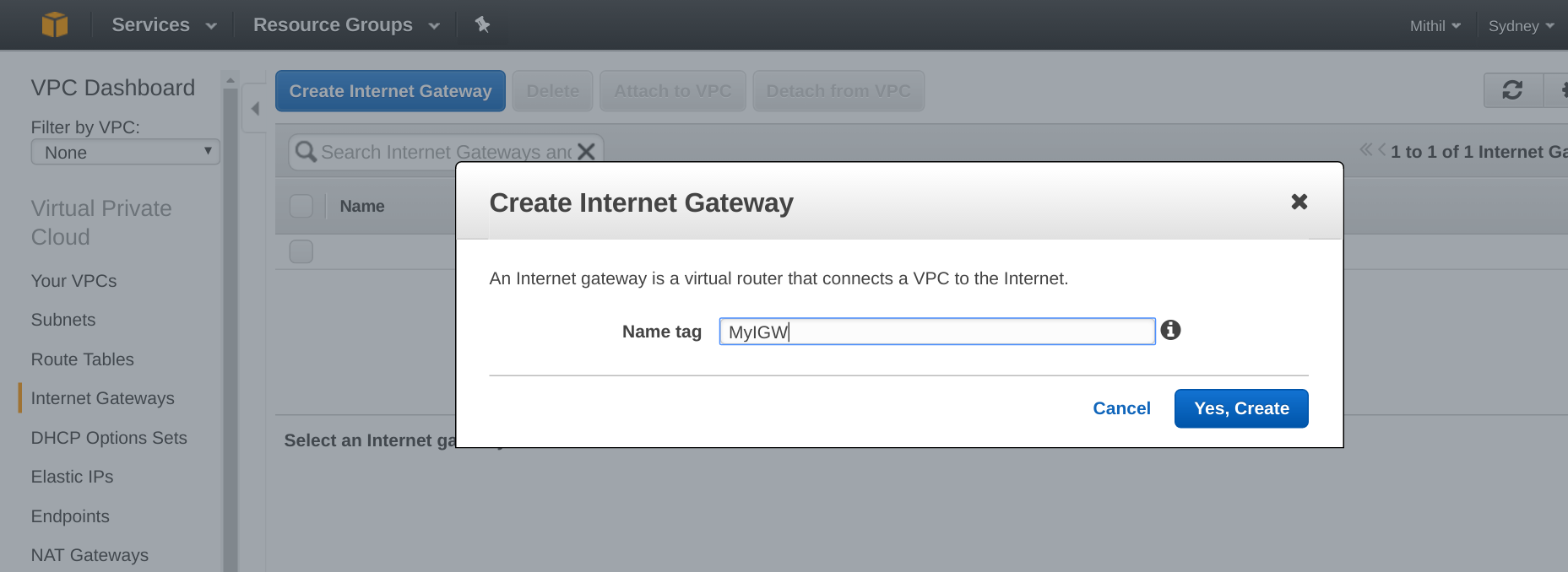
What is network ACL

VPC has two layers of security: security groups and network ACLs. Security Group can be allowed to modify permission any instance that it is attached to. ACLs, on the other hand, are applicable for the whole subnet that they are attached to. Also, ACL’s are stateless so the rules for inbound and outbound traffic are separate. Amazon recommends using security groups as the first choice. The screenshot below shows the default ACL that allows all inbound traffic within the subnet.

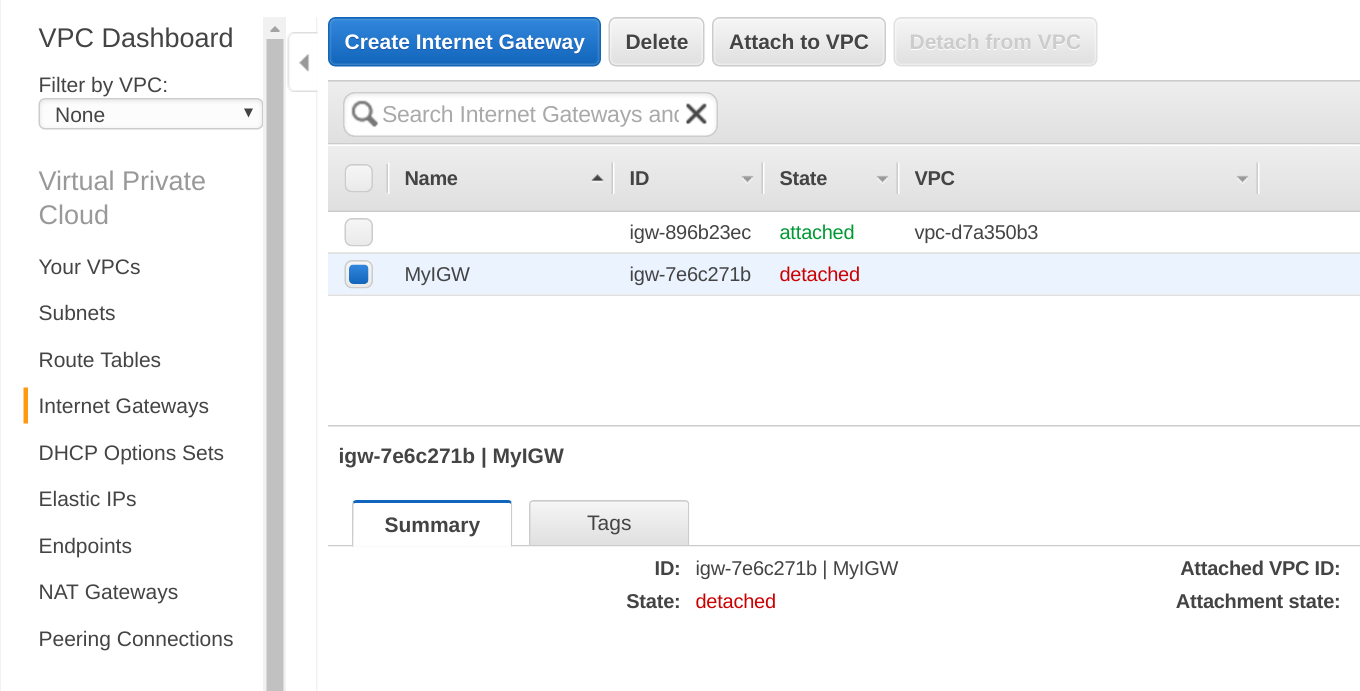
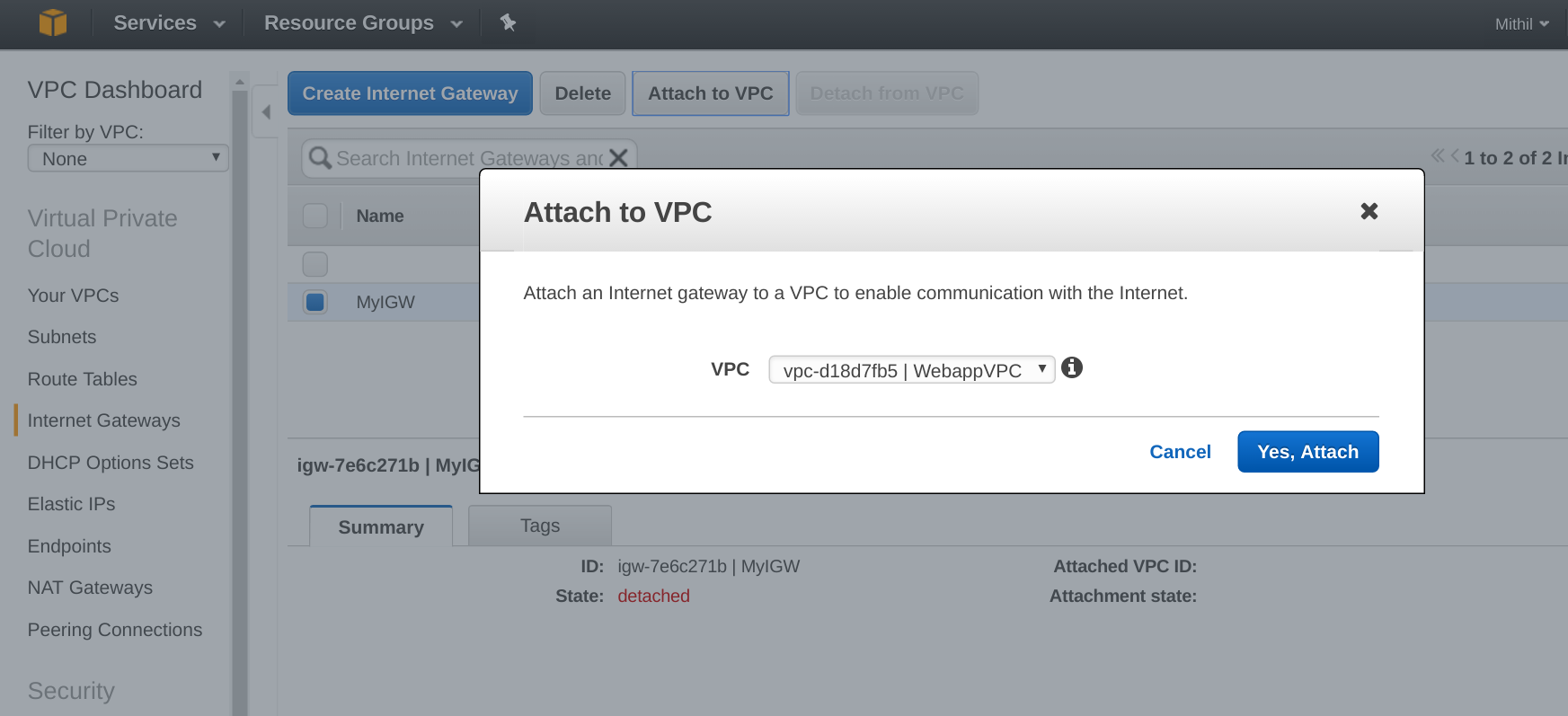
**Creating an AWS VPC Subnet Group**

As shown in our network diagram, we will create four subnets spanning two Availability Zones(AZ). Each AZ will have one private and one public subnet. The idea is that if one AZ goes down, our system still works. Let’s say, you are creating a web server, an application server and an RDS instance. We will have the RDS instance and the application server in the private subnet and the web server in the public subnet. This setup will be replicated in both AZ. We use a CIDR block of 129.0.12.0/24 for the first subnet. This would give us 251 usable IPs. Amazon reserves 5 addresses.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_245.png)  
We similarly create the other three subnets.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_246.png)

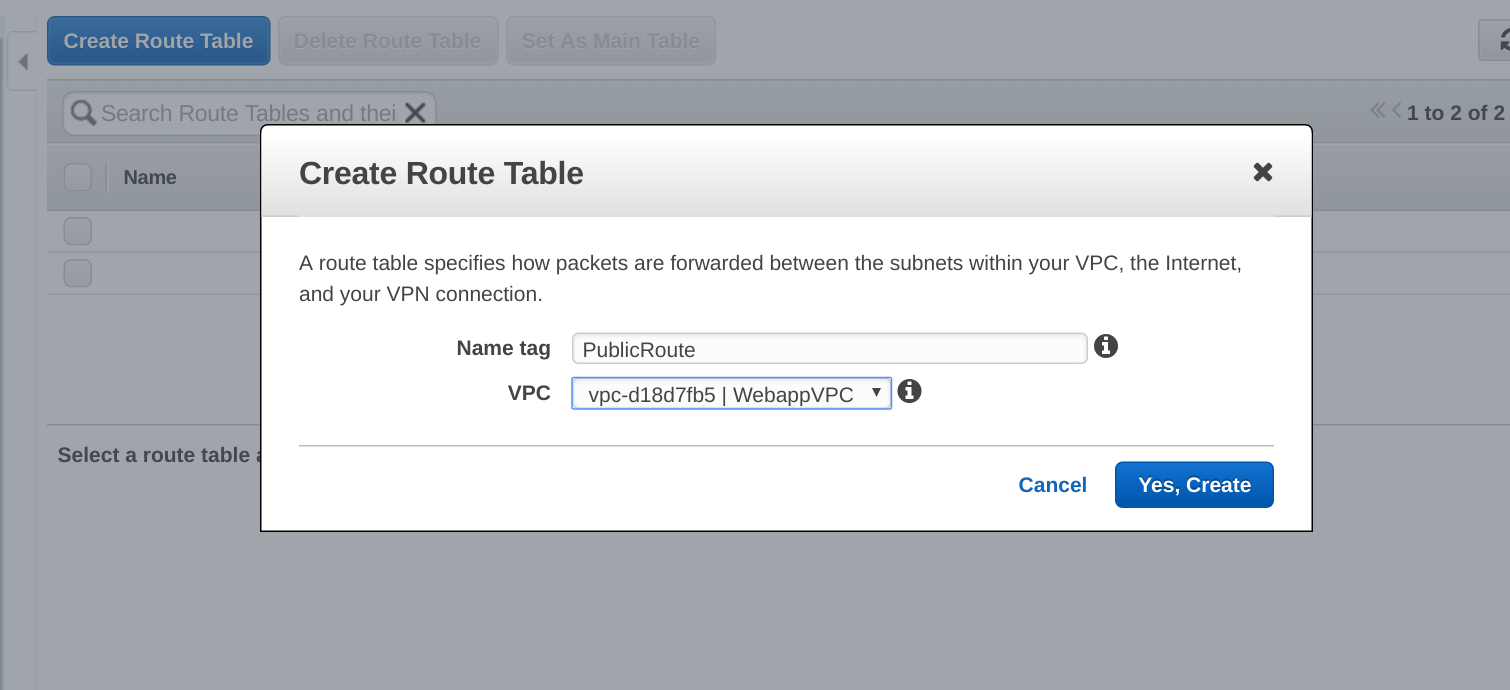
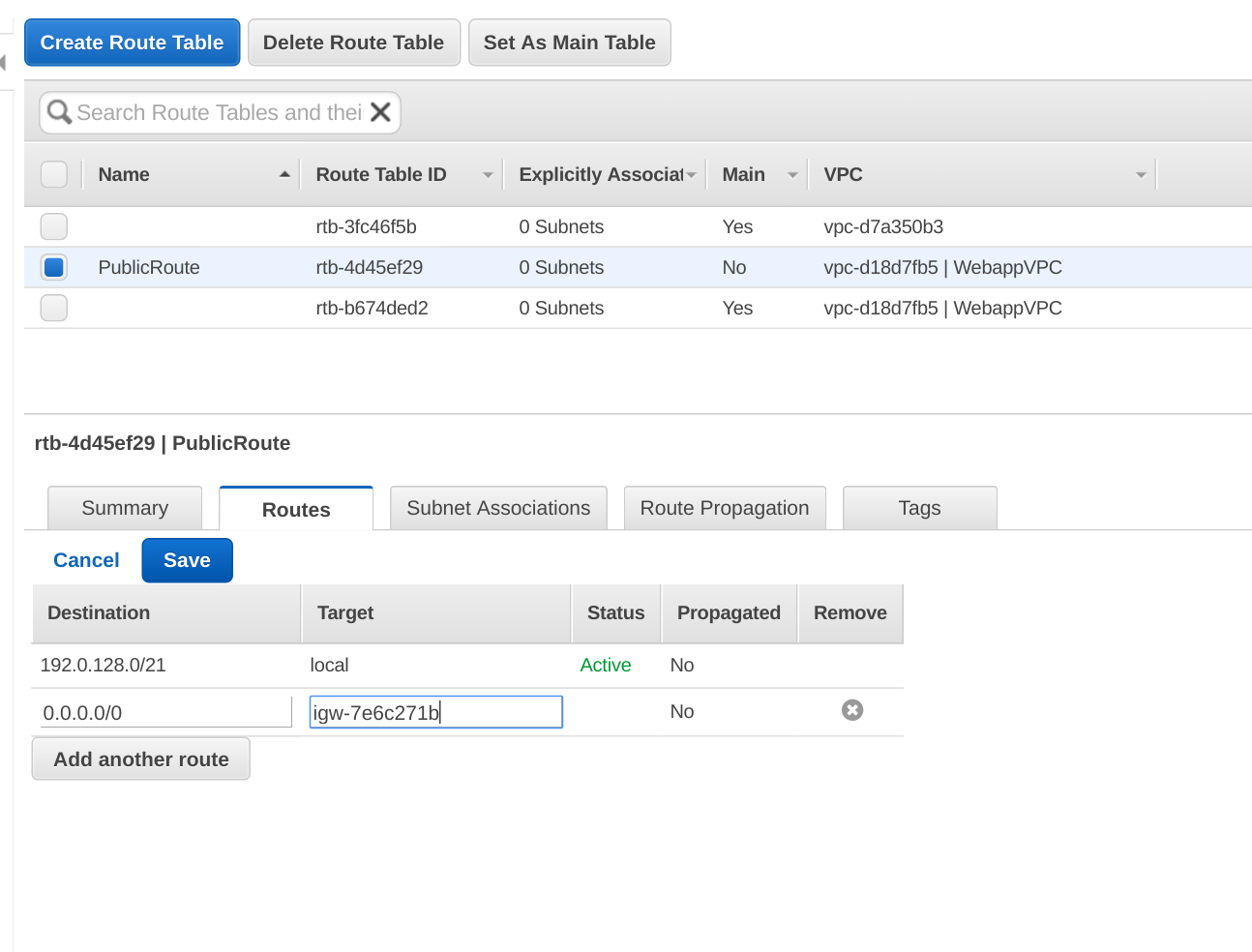
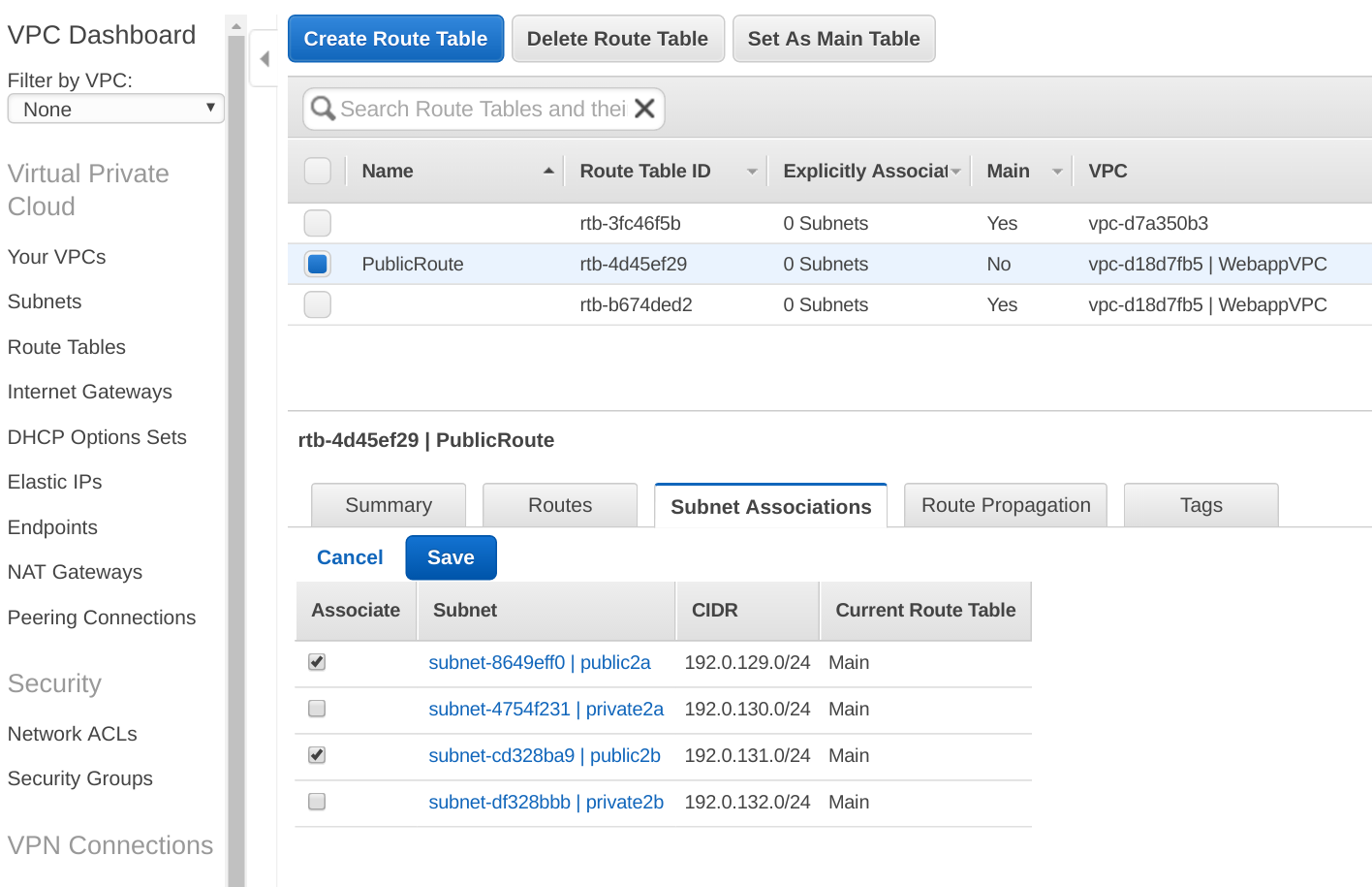
**Creating an AWS Internet Gateway**

Two out of four of our Subnets are public. We need a gateway that allows the instances and services from the public subnet to access the internet. Here’s how we create the gateway: click on the link on the left that says ‘Internet Gateways’   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_247.png)  
We call it ‘MyIGW’.

*Attaching an internet gateway to a VPC*

When you create a new gateway it is in a detached state.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_251.png)  
An Intenet Gateway needs to be attached to a VPC. We attach it to our VPC   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_252.png)

*Adding route to a VPC*

We will now create a new route table that allows instances inside a subnet to direct all traffic to the Internet gateway so that the gateway can direct it out to the internet. Click on ‘Route Table’ link on the left and then click on ‘Create Route Table’   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_254.png)  
Add a new route that redirects all traffic (0.0.0.0/0) to the internet gateway that we created   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_255.png)  
As the last step we assign this route table to the subnets that we want to be public   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_256.png)

This finishes the second part of the tutorial. In the third and the last part we will look at how to create an Elastic IP address and assign that address to a NAT gateway so that instances in the private subnet can talk to the internet.

AWS VPC Tutorial Part III Elastic IP and NAT

In this AWS VPC Tutorial, we will learn how to create an Elastic IP address, a NAT Gateway, and accessing the internet from private subnet using the NAT gateway.   
This is the third in the 3-series tutorial, the [first](http://www.studytrails.com/amazon-aws/aws-vpc-tutorial/) tutorial introduced the key concepts and tutorial problem and the second tutorial looked at creating VPC, subnets, and internet gateway.

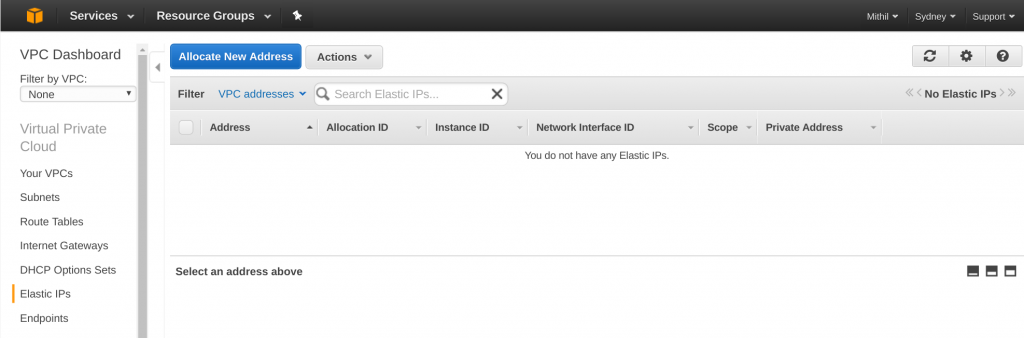
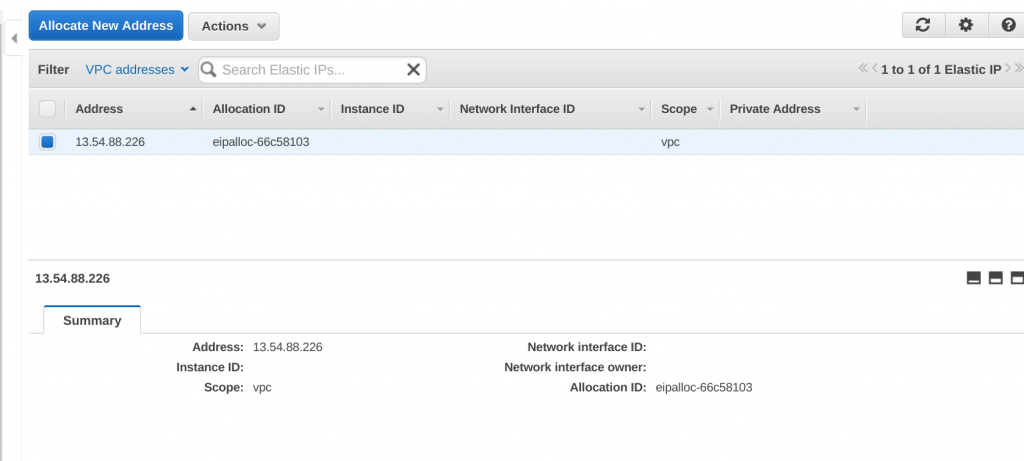
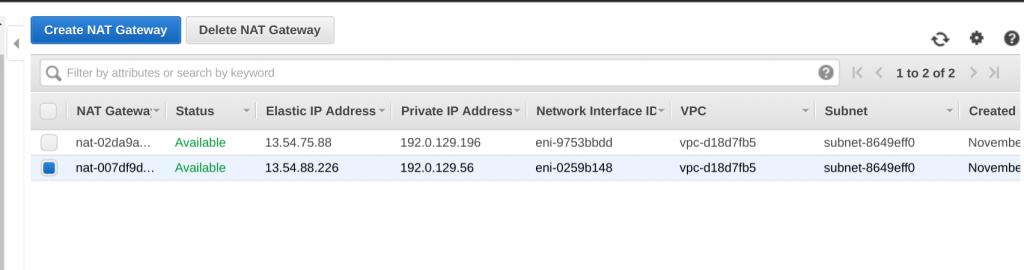
**Why a NAT Gateway?**

Before we explain why we need a NAT Gateway, here’s a network diagram of what we are trying to accomplish. [](http://www.studytrails.com/wp-content/uploads/2016/11/VPCTutorial.png)  
The two private networks need to talk to the internet for things like updating the operating system or installing software. Since the instances in the private network do not have a public IP, they need a NAT (Network Address Translation) Gateway that can convert the private IP to public IP for routing traffic to the internet and back. AWS provides two kinds of NAT resources – NAT instances and NAT gateways. The gateways are completely managed by AWS and so they are preferred over NAT instances.

**What are Elastic IP addresses**

We will build a NAT gateway, however, the gateway needs an IP address. AWS provides Elastic IP addresses, that you can create on demand. These IP addresses can be attached to instances and resources and detached when not required.

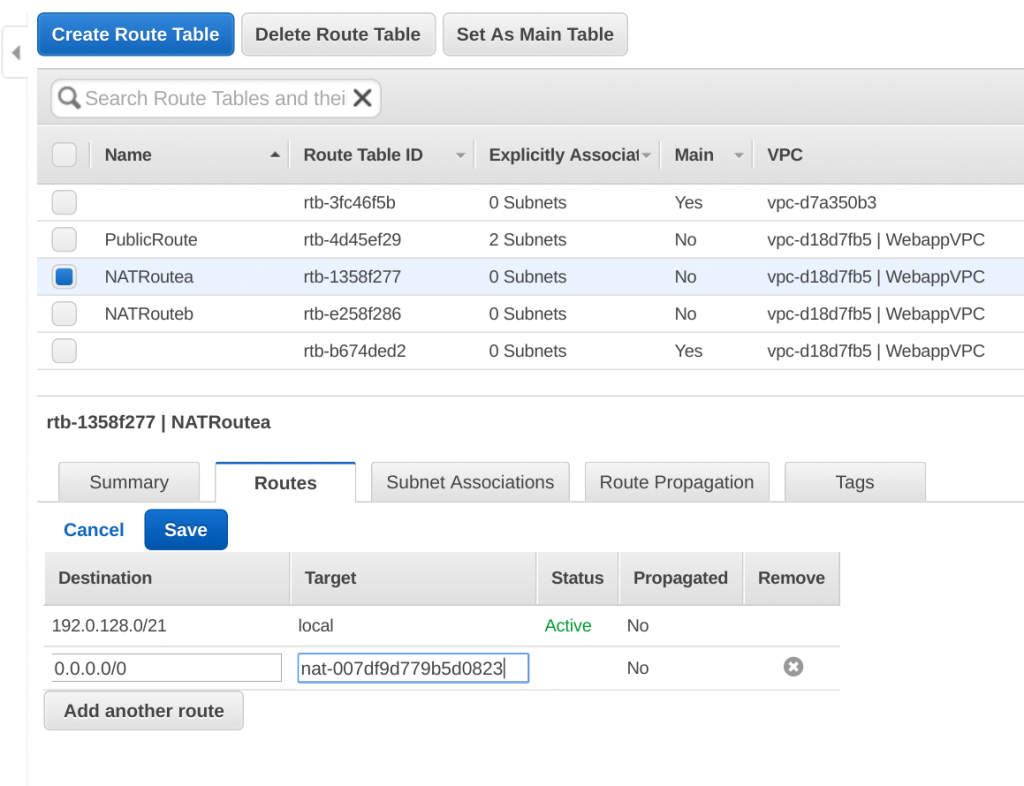
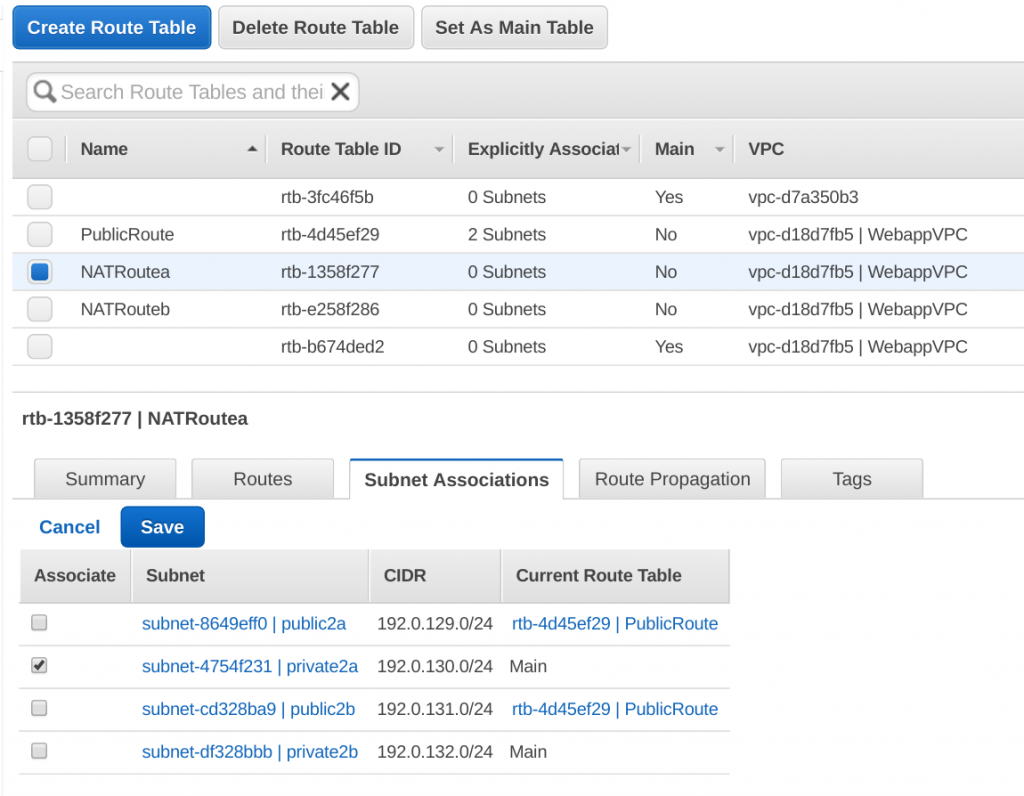
*Steps to create AWS Elastic IP Address*

Click on Elastic IPs in the VPC console of AWS.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_257.png)  
If you haven’t created an IP address before you should see an empty table. Click on ‘Allocate New Address’ to add a new IP address. AWS will start creating the IP address for you.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_258.png)  
Create two Elastic IPs for the two Gateways in each AZ.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_263.png)

*Steps to create the NAT gateway*

Once the IP address is created, click on NAT Gateway to open up the NAT gateway homepage. Click on ‘Create NAT Gateway’ to create a new NAT Gateway. In The field that says ‘Elastic IP Allocation ID’ select the new IP that you just created and then hit ‘Create NAT Gateway’. This will create the NAT Gateway.

*Create a Route Table for NAT*

The next step is to create the route table that will direct all traffic in the private subnet through the NAT Gateway. Click on the ‘Route Table’ section and click on ‘Create Route Table’. In the Routes section, add a route that maps all traffic (0.0.0.0/0) to the NAT gateway that we just created.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_264.png).   
The Route table would need to be created for both the NAT Gateways.   
[](http://www.studytrails.com/wp-content/uploads/2016/11/Selection_265.png)

**Recap of the AWS VPC Tutorial**

Our first VPC setup is done. To recap, we created a VPC with four subnets. Two of the subnets were made public by attaching an internet gateway to it and the other two subnets were private. However, the private subnets were allowed to connect to the internet using a NAT gateway. The instances in the public subnet will have public IP addresses. We can configure the subnet to assign a public IP address to all instances that are created in the subnet, OR, during instance creation we can specify the instance to have an auto-assigned public IP.   
This completes our AWS VPC Tutorial.