Provisioning a VPC

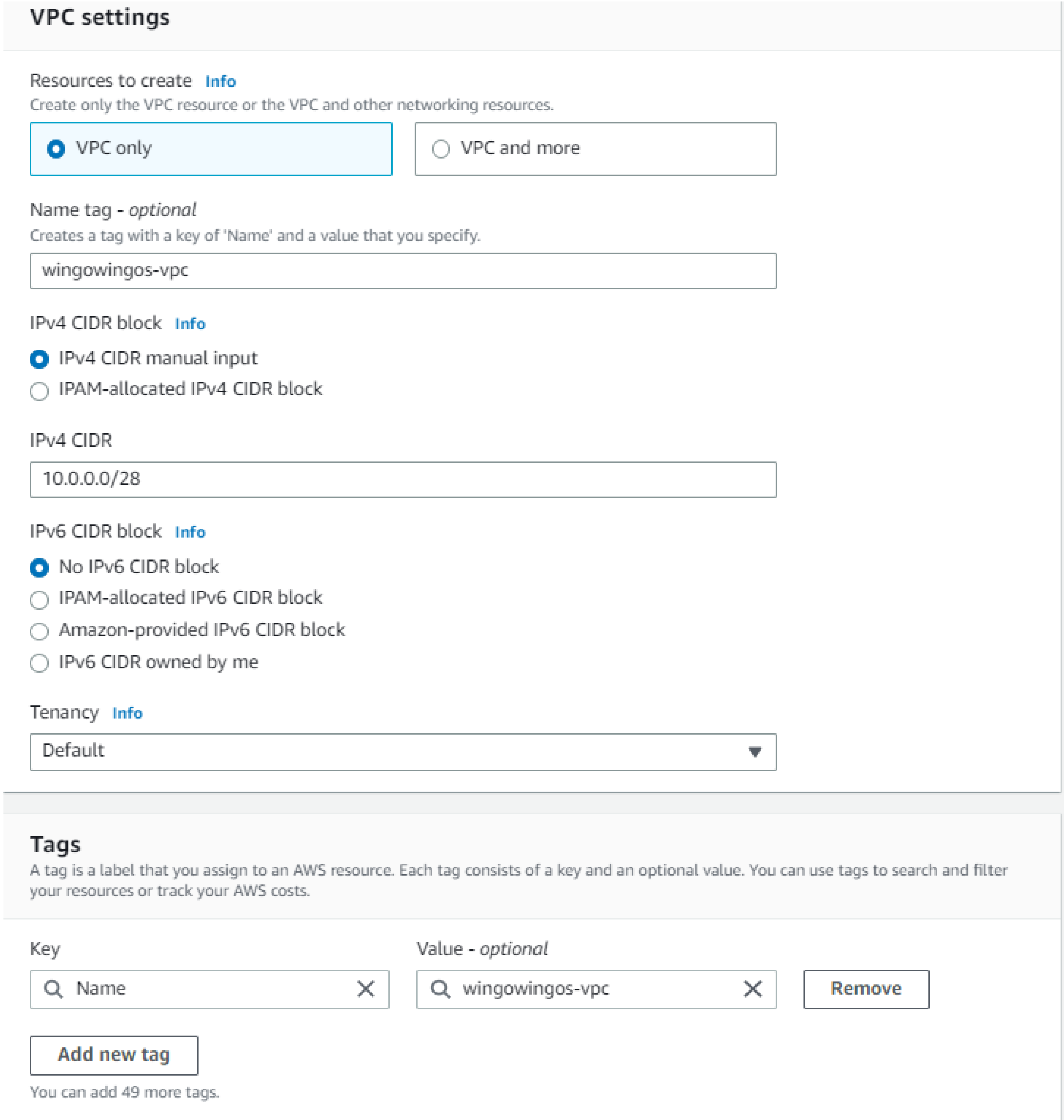
This will be a lengthy document on creating and provisioning a VPC

Create VPC

1. Go into the VPC service and click "Create VPC"
2. Select "VPC Only"
3. Name your VPC
4. Input a IPv4 CIDR address

Now when doing your IPv4 CIDR number, you must select between /16 and /28

Figure 1-1



VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

● VPC only ○ VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.

wingowingos-vpc

IPv4 CIDR block Info

● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/28

IPv6 CIDR block Info

● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

Tenancy Info

Default ▾

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	wingowingos-vpc	Remove

Add new tag

You can add 49 more tags.

Remember when created a VPC it will create 3 things by default.

- Will create a security group
- Will create a main route table
- Will create a main network ACL

5. So going into the VPC and clicking on the main route table, go to the subnet assosciations

You can see there are no subnets create that we have created. Yes we do have subnets, but that is our default subnets.

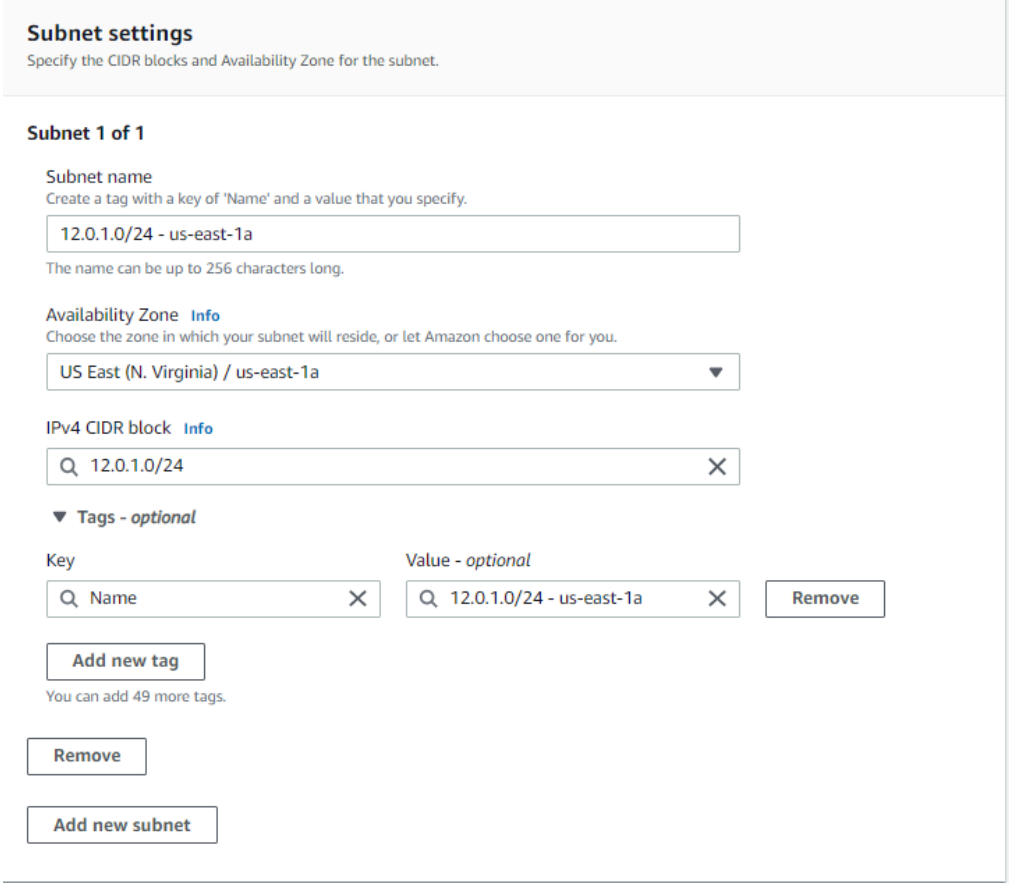
*Remember subnets are basically virtual firewalls. It can be either public or private facing.

Creating Subnets

6. So go to VPC > Subnets > Create Subnet
7. Once there select the custom VPC we just created.

For the subnet settings you can select the AZ zone first and then go back to the name. Might be a good naming convection to use.

Figure 2-1



Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
12.0.1.0/24 - us-east-1a
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info
12.0.1.0/24

Tags - *optional*

Key	Value - *optional*
Name	12.0.1.0/24 - us-east-1a

You can add 49 more tags.

*Public Subnet

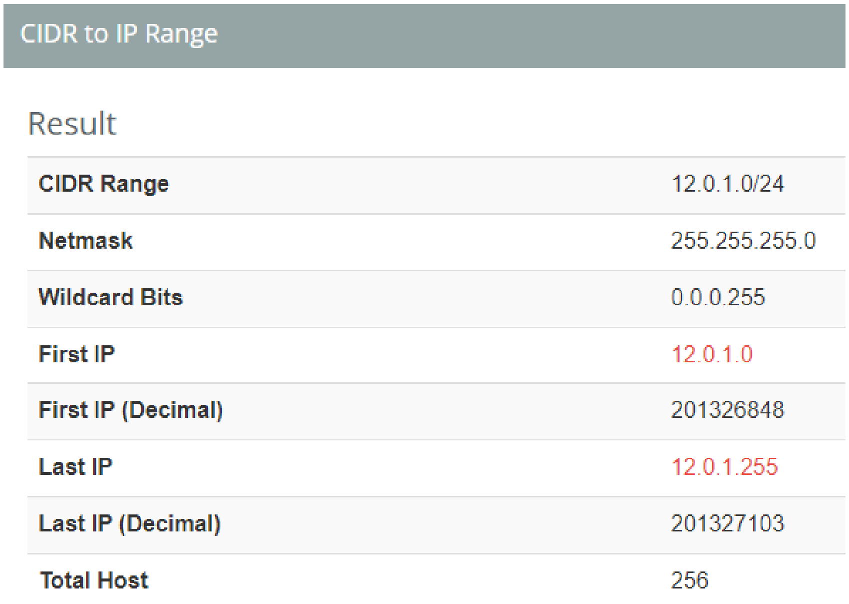
Above you may be thinking why input this the way I did?

Well for the subnet name, it may be best to go ahead and put in the range you want to use. My original IP that I used was 10.0.0.0/28 and the range I will go to will be /32. That being said, we can input the 10.0.1.0/32 in the subnet name. Same IP range goes to the CIDR Block.

This may make things easier so when I am selecting the subnet, not only do we now know what the CIDR address range is, but also know what AZ it is going to go in.

8. Click on the subnet ID to bring up all the details about the subnet

Figure 2-3



CIDR to IP Range

Result

CIDR Range	12.0.1.0/24
Netmask	255.255.255.0
Wildcard Bits	0.0.0.255
First IP	12.0.1.0
First IP (Decimal)	201326848
Last IP	12.0.1.255
Last IP (Decimal)	201327103
Total Host	256

So, the first four IP address and the last IP address in each subnet CIDR block are not available for you to use and cannot be assigned to an instance.

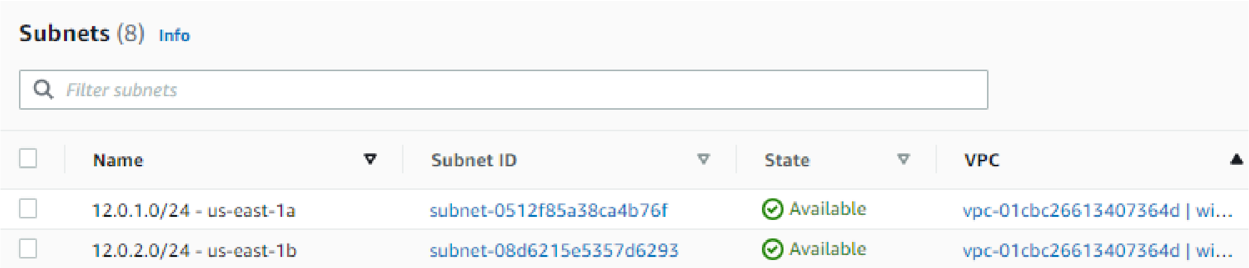
https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html

So you will basically lose 5 IP addresses from a traditional /24 network

Scroll down to the “Subnet sizing”

9. Now we can create our second subnet.

Figure 2-4



Subnets (8) Info

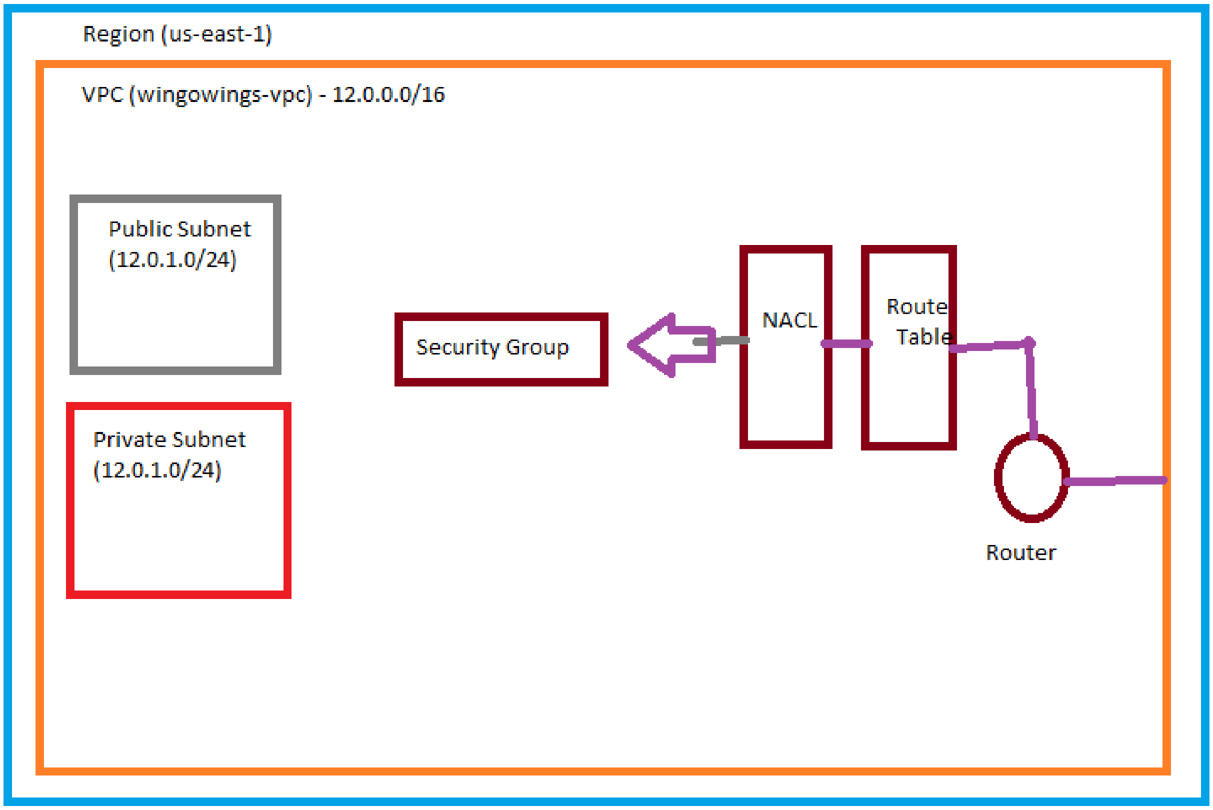
Filter subnets

□	Name	Subnet ID	State	VPC
□	12.0.1.0/24 - us-east-1a	subnet-0512f85a38ca4b76f	✓ Available	vpc-01cbc26613407364d | wi...
□	12.0.2.0/24 - us-east-1b	subnet-08d6215e5357d6293	✓ Available	vpc-01cbc26613407364d | wi...

After creating the second subnet, you can see I included a number 2 in the 3rd octet and this will be our private subnet going to us-east-1b AZ

So far we have built this below

Figure 2-5


Region (us-east-1)
VPC (wingowings-vpc) - 12.0.0.0/16
Public Subnet (12.0.1.0/24)
Private Subnet (12.0.1.0/24)
Security Group
NACL
Route Table
Router

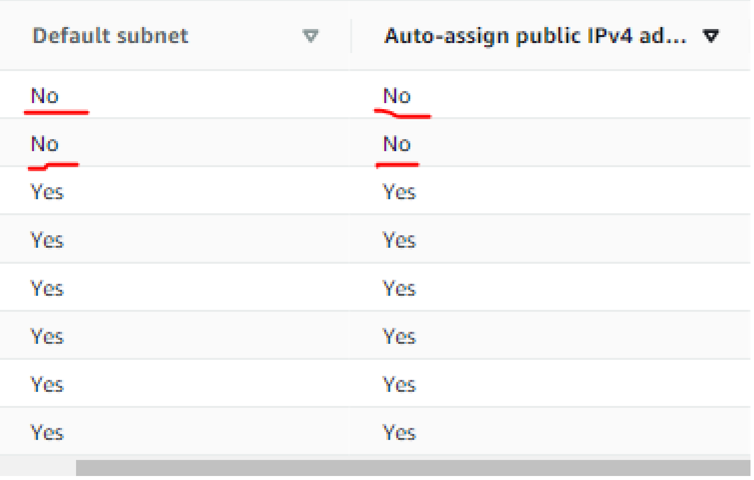
Yes this was done in Paint…

Assigning Auto-assign public IPv4 address

10. Go back into VPC > Subnets

a. Look at the custom ones we created compared to the default subnets

Figure 3-1

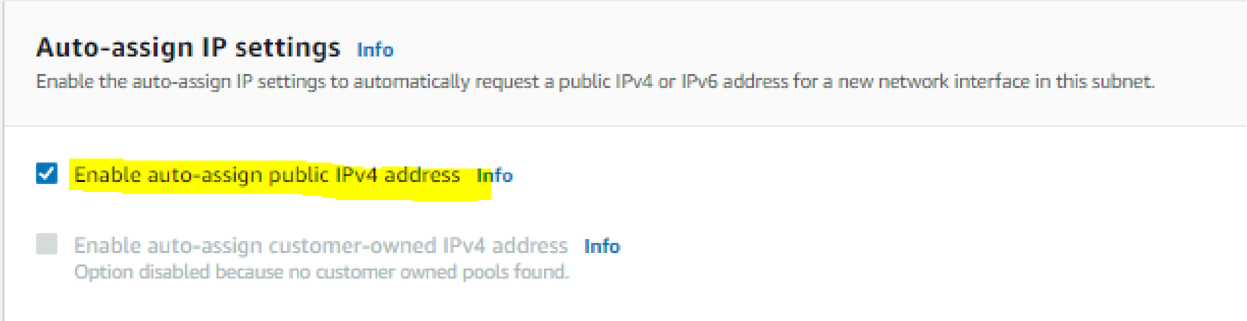


Default subnet	Auto-assign public IPv4 ad...
No	No
No	No
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes

You can see that the 2 custom ones we created have no on the "Auto-assign.." column compared to the default subnets. So we want to enable that to assign the IPv4 address to any EC2 Instance(s) in the subnet (The Public subnet). So when we deploy our EC2 instances we want them to be publicly accessible.

11. Click on the public subnet that we want to change > Actions > Edit subnet settings
12. When in the settings select the "Enable auto-assign…"

Figure 3-2



Auto-assign IP settings Info

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

☑ Enable auto-assign public IPv4 address Info

□ Enable auto-assign customer-owned IPv4 address Info
Option disabled because no customer owned pools found.

Quick and easy task

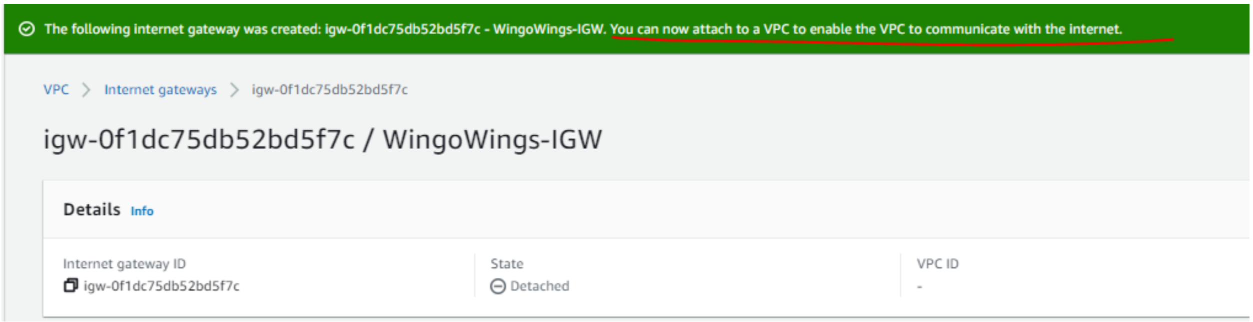
13. So now that we have done that, we want this subnet publicly accessible. So that is were we have to create an IGW (Internet Gateway)

Creating Internet Gateway

14. On the left pane, below route tables, click Internet gateways > and select "Create internet gateway".

15. Give it a name > Create

__Figure 4-1__



The following internet gateway was created: igw-0f1dc75db52bd5f7c - WingoWings-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet.

VPC > Internet gateways > igw-0f1dc75db52bd5f7c

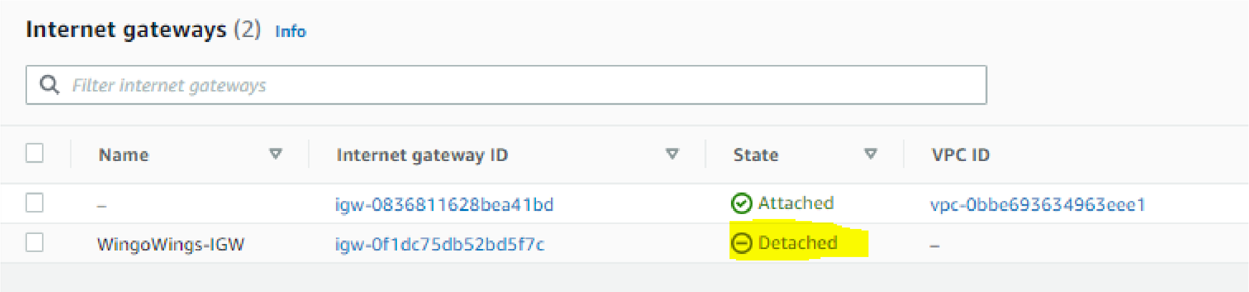
igw-0f1dc75db52bd5f7c / WingoWings-IGW

Details Info

Internet gateway ID	State	VPC ID
igw-0f1dc75db52bd5f7c	Detached	-

You can see that we can now attach to a VPC to enable the VPC to communicate with the internet

__Figure 4-2__



Internet gateways (2) Info

Filter internet gateways

□	Name	Internet gateway ID	State	VPC ID
□	–	igw-0836811628bea41bd	Attached	vpc-0bbe693634963eee1
□	WingoWings-IGW	igw-0f1dc75db52bd5f7c	Detached	–

Going back to the Internet Gateway we can see the IGW is in the "Detached" state. So the next step is to attach it to our VPC. Note from Figure 4-1.

*You can only have 1 IGW per VPC

Now that we have our IGW attached to our VPC. We need a route out to the internet.

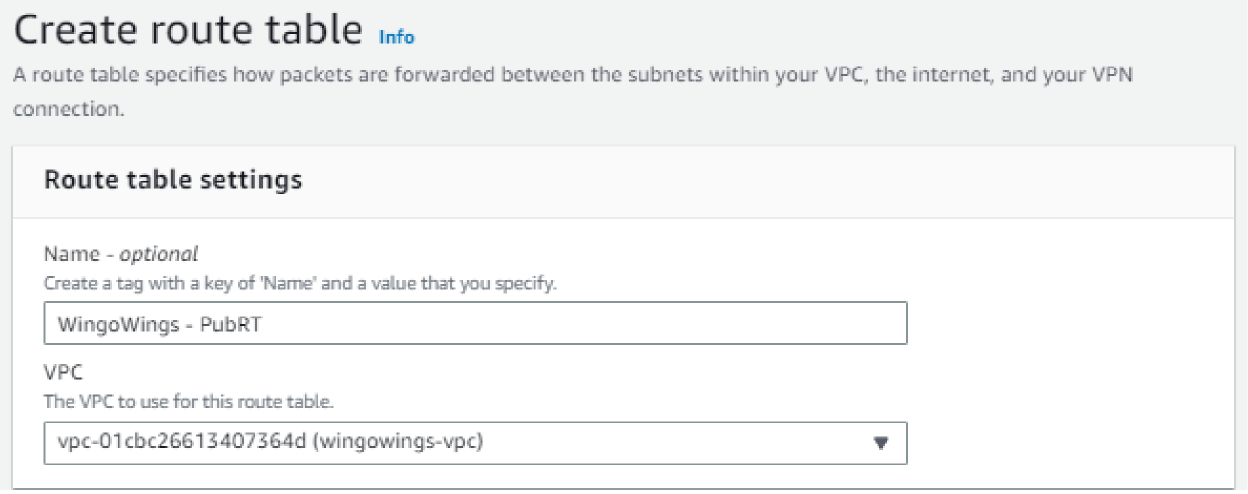
Creating a Route Table

When going to the route table, we are only interested in the one going to our VPC. Click on the hyperlink in the Route Tables column to view the details.

If we create a route out to the internet from our main route table, every time we create a new subnet, it is by default associated with our main route table. Basically, if we have our main route table out to the internet, every time we create a new route, that subnet is going to be public. Which isn't a good idea from a security side of things

16. So go back to Route Tables > Create route table

__Figure 5-1__



Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

WingoWings - PubRT

VPC

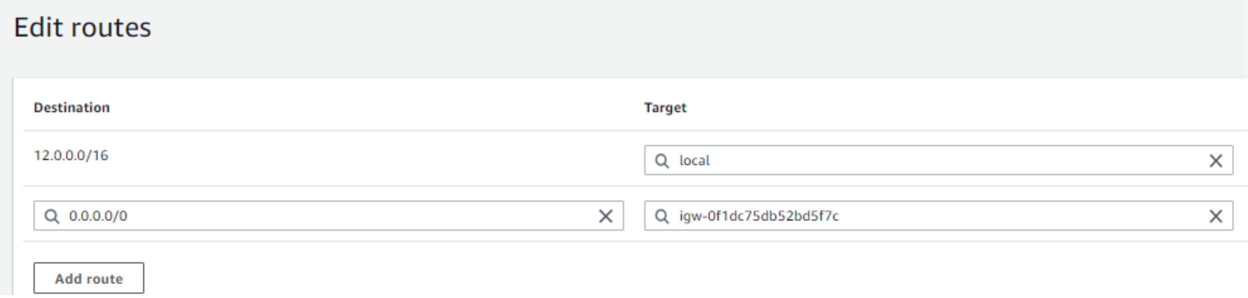
The VPC to use for this route table.

vpc-01cbc26613407364d (wingowings-vpc)

I named it PubRT because this will be our public route table (RT)

17. Now that is completed we need a route out to the internet.
18. Go back into the newly created route table > Click edit routes

Figure 5-2



Edit routes

Destination	Target
12.0.0.0/16	local
0.0.0.0/0	igw-0f1dc75db52bd5f7c

Add route

Above I added a new route, which was a public destination address along with our IGW VPC.

So every time we provision a new subnet it will be provisioned to our default/main route table which does not have any route out to the internet.

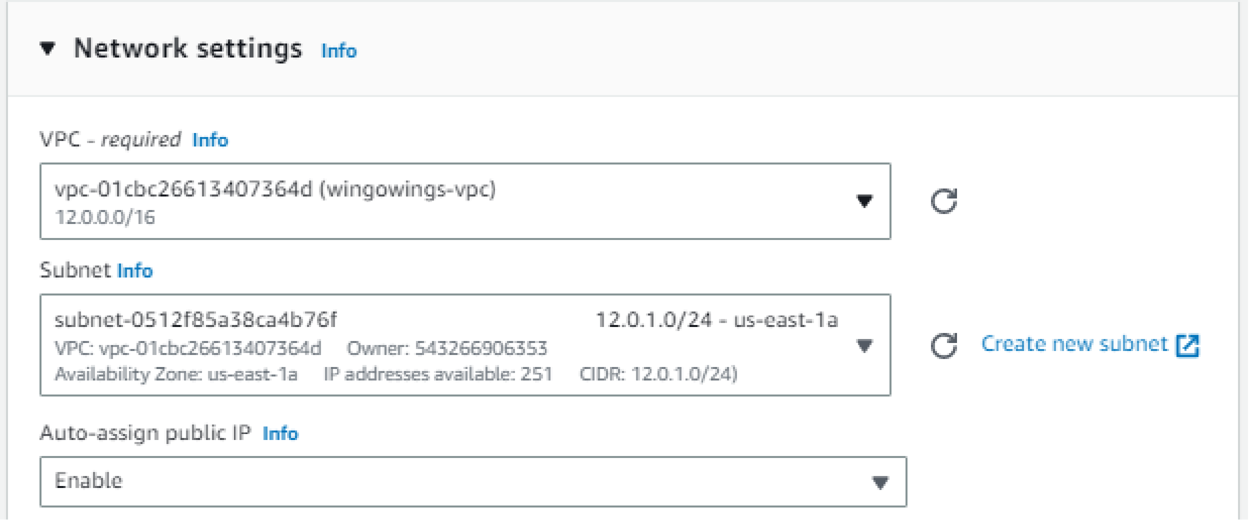
19. Go back into the Route table and click on “Subnet Associations” tab and under the explicit subnet associations click “Edit subnet associations”
20. Select the correct public VPC which in this case is the 12.0.1.0/24 > Click “Save associations”.

Now we have to drop some EC2 instances into these subnets in both public and private.

EC2 into subnets

21. So go into EC2 and select “Launch EC2 Instance”.
22. After creating the name and key pair go to the network settings and click edit.
23. Once there you need to make sure the VPC is changed and also make sure the subnet information is to your public subnet.

Figure 6-1



▼ Network settings Info

VPC - *required* Info

vpc-01cbc26613407364d (wingowings-vpc)
12.0.0.0/16

Subnet Info

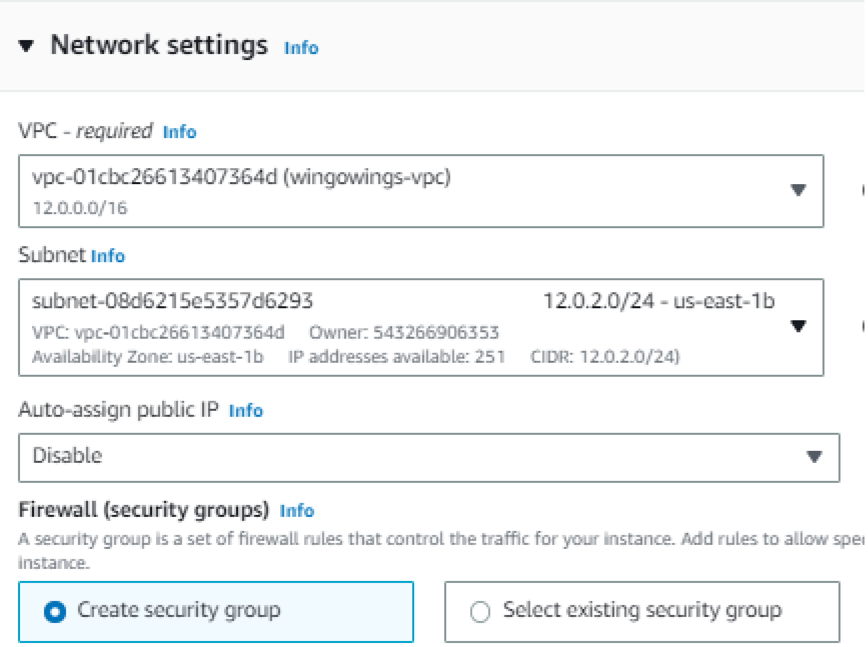
subnet-0512f85a38ca4b76f 12.0.1.0/24 - us-east-1a
VPC: vpc-01cbc26613407364d Owner: 543266906353
Availability Zone: us-east-1a IP addresses available: 251 CIDR: 12.0.1.0/24)

Auto-assign public IP Info

Enable

24. Include the Security group (Make one if you haven't done so by simply naming it)
25. Include another security group with HTTP > Source type: Anywhere
26. Launch Instance
27. Go back into the EC2 dashboard and click on the newly created instance ID
28. And look around the Details, Security and Networking tab making sure everything looks good.
29. So now that the status checks out, lets create another instance.
    a. This one will be our private EC2
30. When creating it, notice how in figure 6-1 we can see that was pointing to the public subnet. This time we want private.

Figure 6-2



▼ Network settings Info

VPC - *required* Info

vpc-01cbc26613407364d (wingowings-vpc)
12.0.0.0/16

Subnet Info

subnet-08d6215e5357d6293 12.0.2.0/24 - us-east-1b
VPC: vpc-01cbc26613407364d Owner: 543266906353
Availability Zone: us-east-1b IP addresses available: 251 CIDR: 12.0.2.0/24)

Auto-assign public IP Info

Disable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow spe
instance.

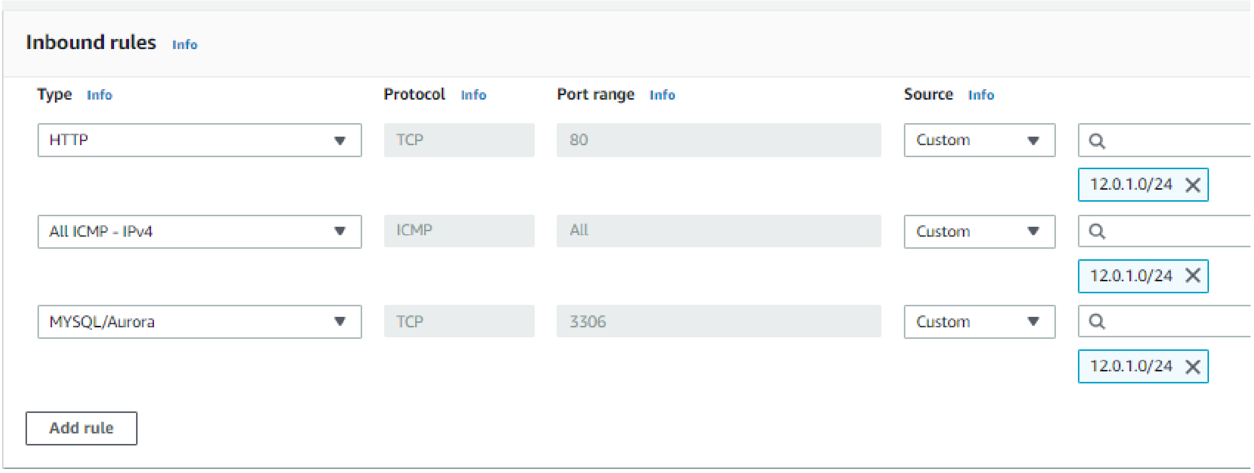
● Create security group ○ Select existing security group

31. Select the same security group, since this is a lab. It may not be best to do this in production to have it under the same security group.
32. Launch Instance, so that is launching our DB server into our private subnet into our custom VPC.

So remember back on step 31 lets go ahead and create that new security group now.

Security Groups

Figure 7-3

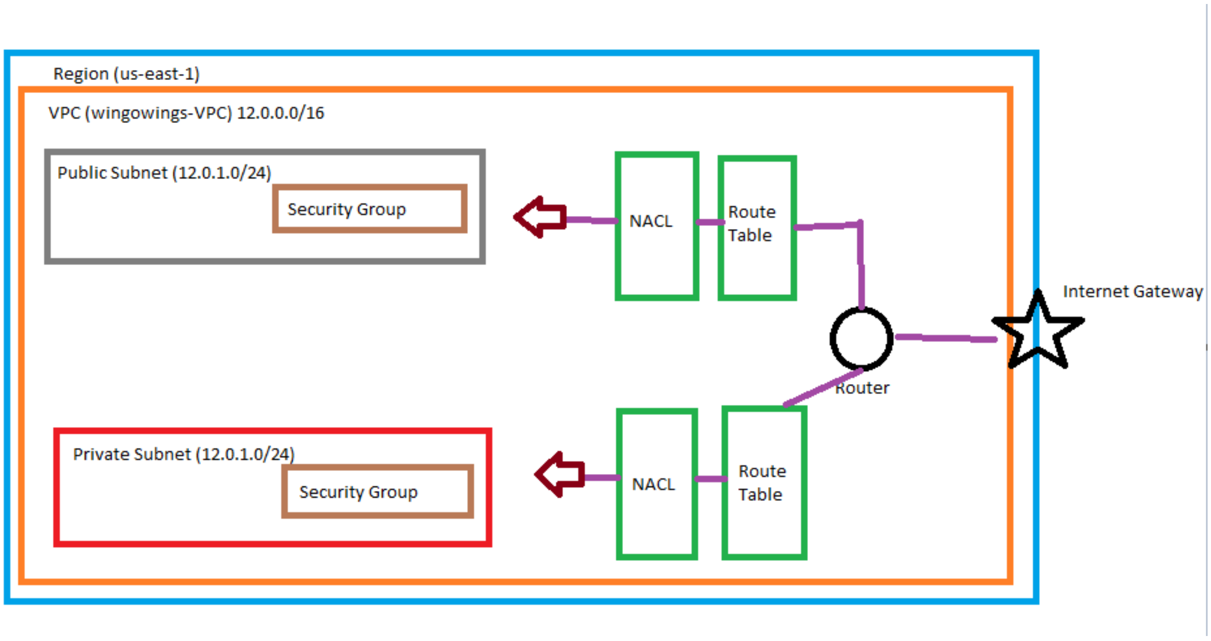


Inbound rules Info

Type	Protocol	Port range	Source	
HTTP	TCP	80	Custom	12.0.1.0/24
All ICMP - IPv4	ICMP	All	Custom	12.0.1.0/24
MYSQL/Aurora	TCP	3306	Custom	12.0.1.0/24

I did not add in SSH because if we were say to get attacked they can’t SSH in.

33. And then create our security group
34. Now go back to our instances table and select the private instance > Actions > Security > Change Security Groups
35. Remove the old group and add in the new one.



https://github.com/CamCam919

So here is our final project for provisioning and securing our VPCs