LAB: Create a VPC with Subnets and Routing and an IG/NAT Gateway

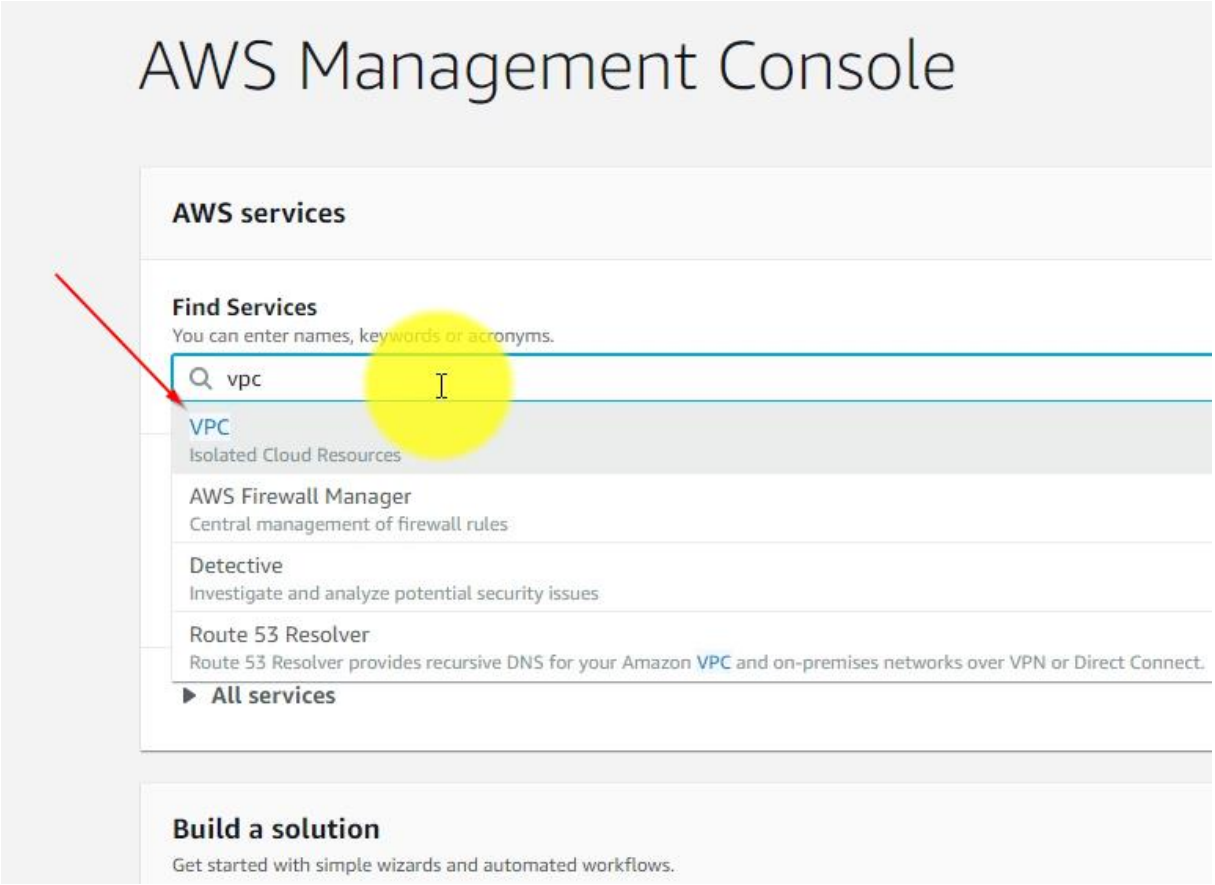
You need:

- An AWS Account

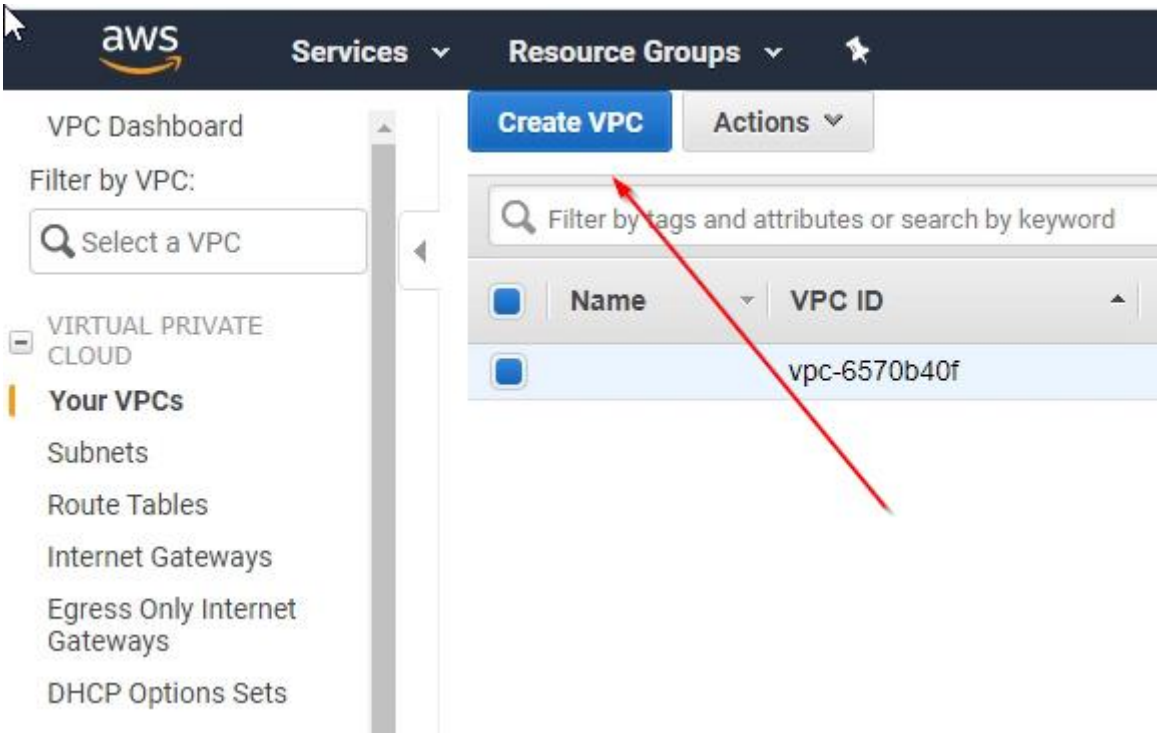
Duration of the Lab: 30 Minutes.

Difficulty: medium

Open the VPC Dashboard

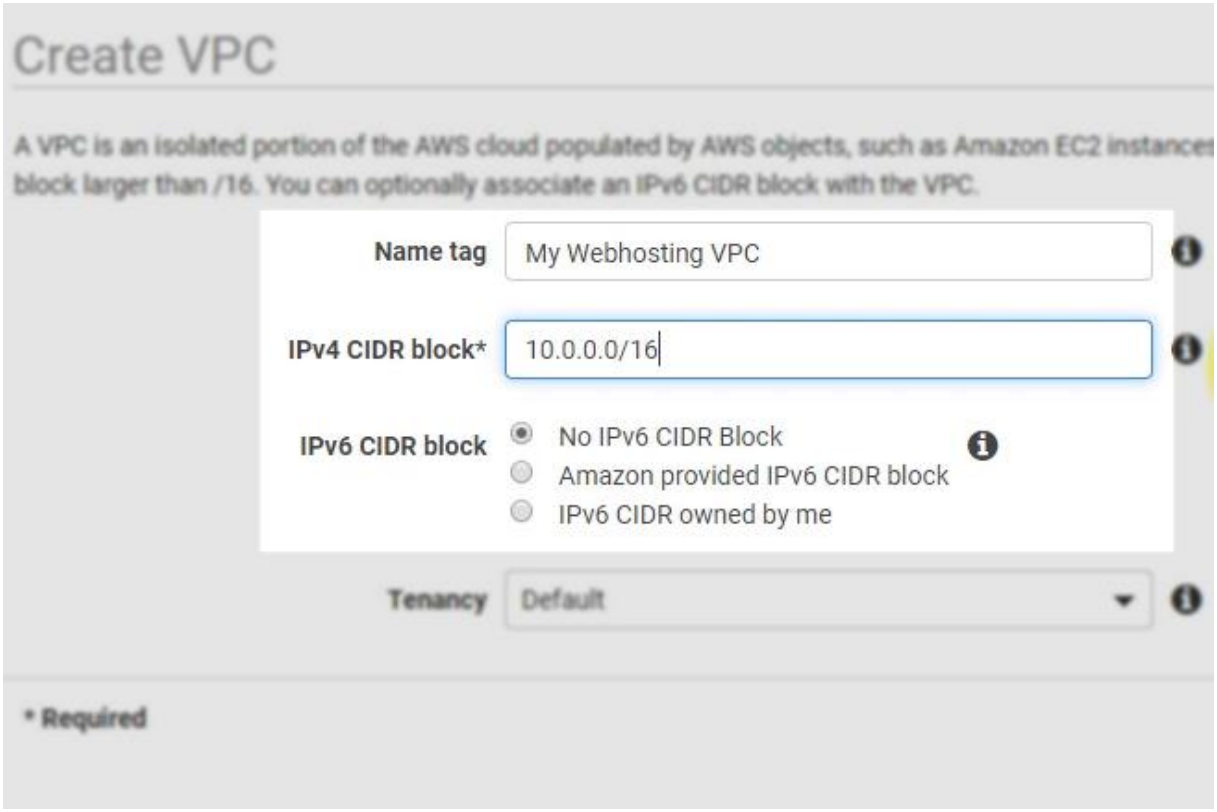


Create a new VPC



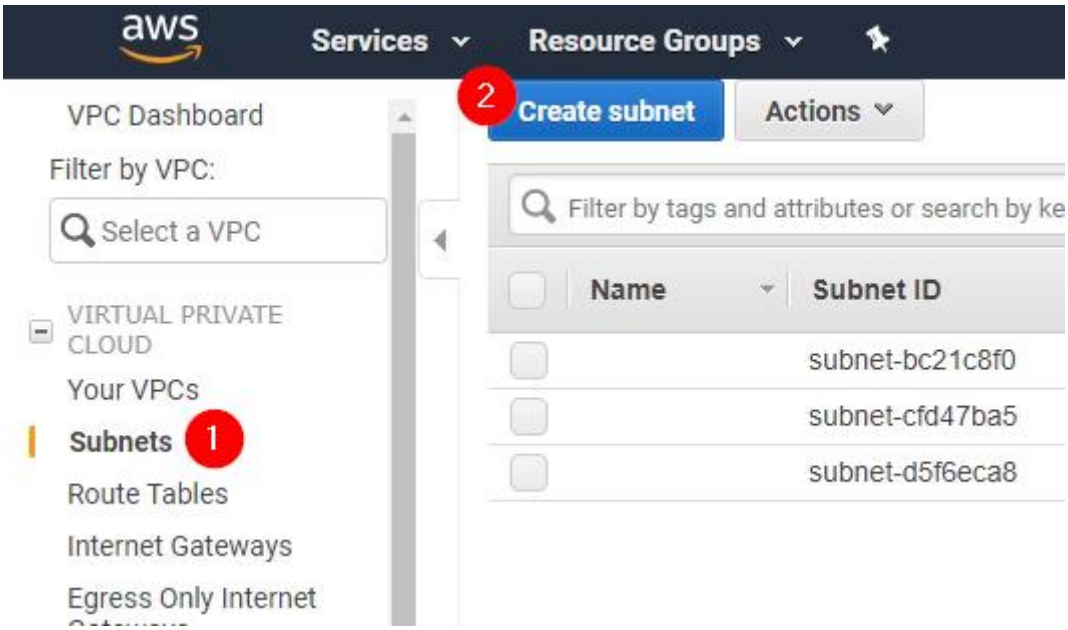
Enter a Name and a CIDR Block, for example 10.0.0.0/16.

This CIDR Block will give you 10.0.X.X IPs, which corresponds to a Class B Network with 65536 IP Addresses (256*256).



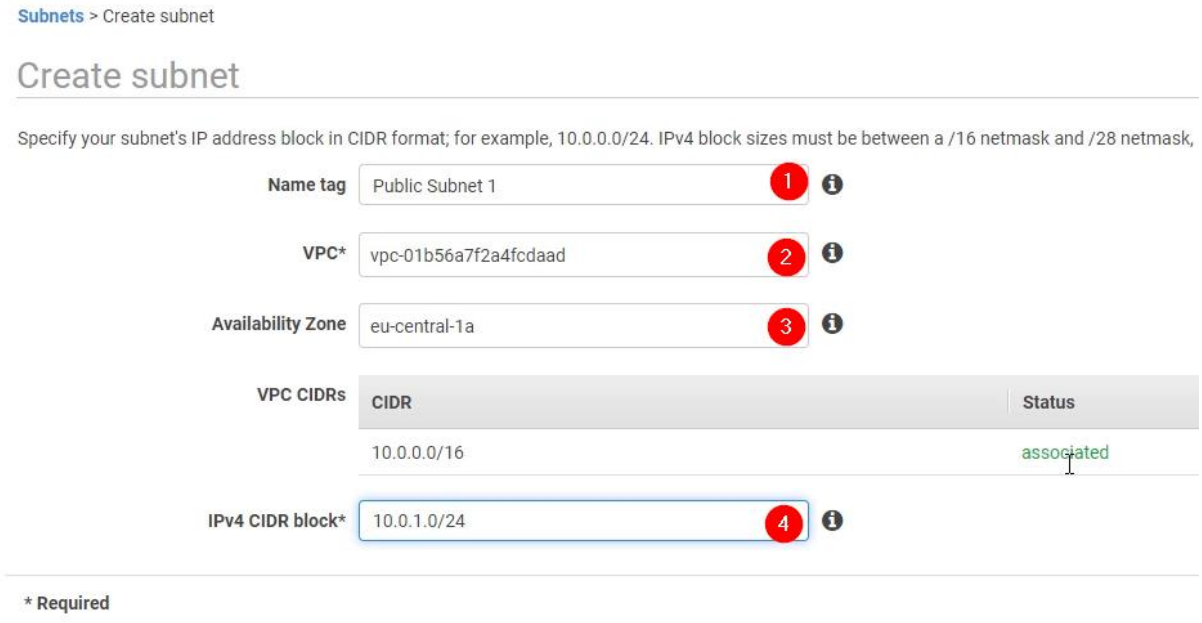
Create Subnets

Create three Subnets:



Create three subnets:

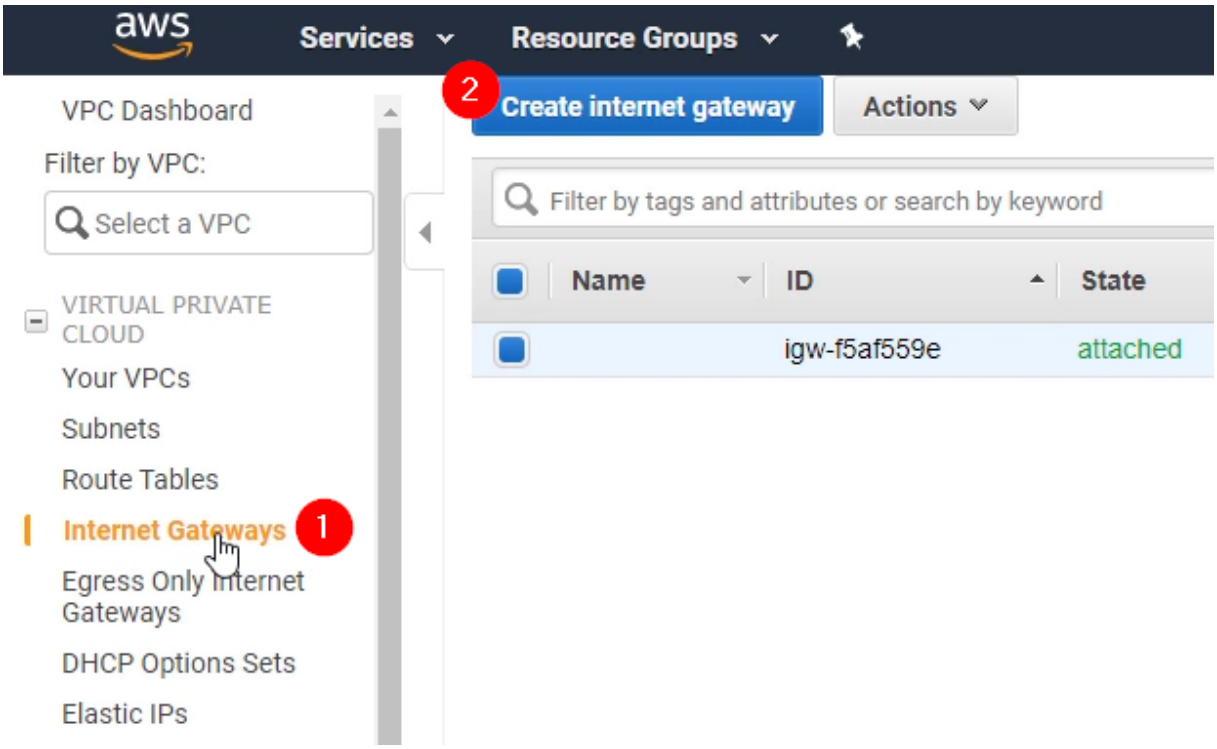
1) Public/Private Subnet 1 and 2
2) Select the VPC you created earlier
3) Select two different AZ for the public subnets and a single one for your private subnet
4) For the public subnets set 10.0.1.0/24 and 10.0.2.0/24 as the CIDR Block, for the private one set 10.0.10.0/24 as the CIDR Block. This gives you 256 IP Addresses in the Subnets, corresponding to a Class C network.



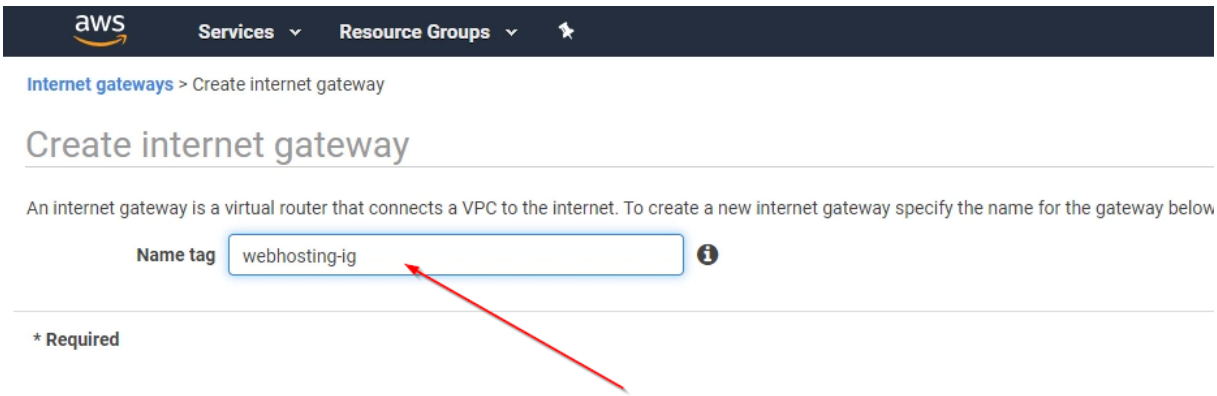
QUESTION: Is this is a High Availability Setup? Why yes, why not?

Create Internet Gateway

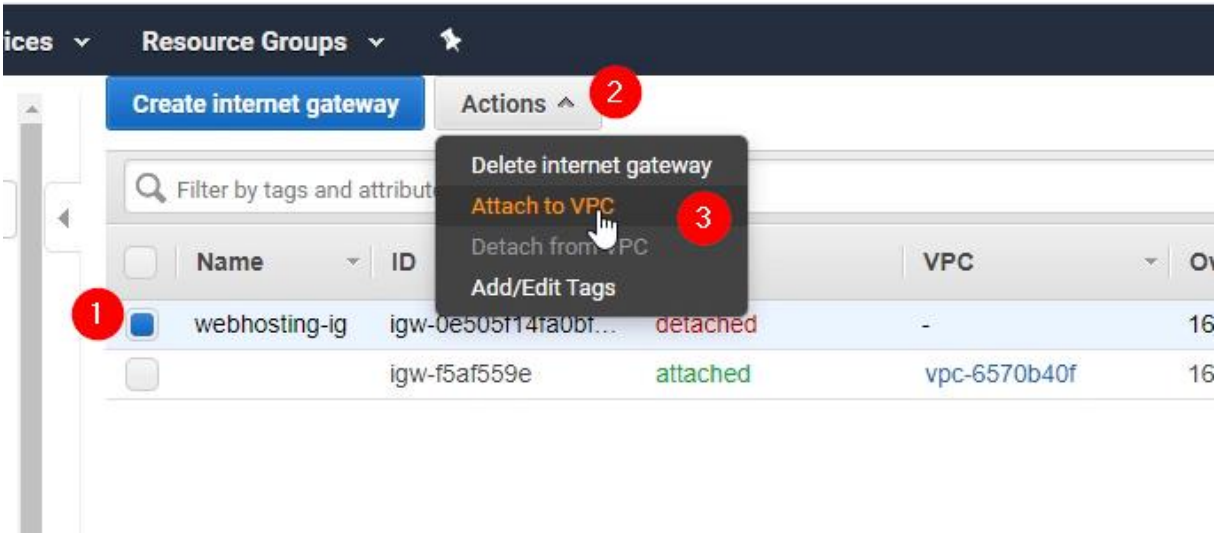
Open the Internet Gateway section of the VPC Dashboard and create an internet Gateway:



Give it a name:

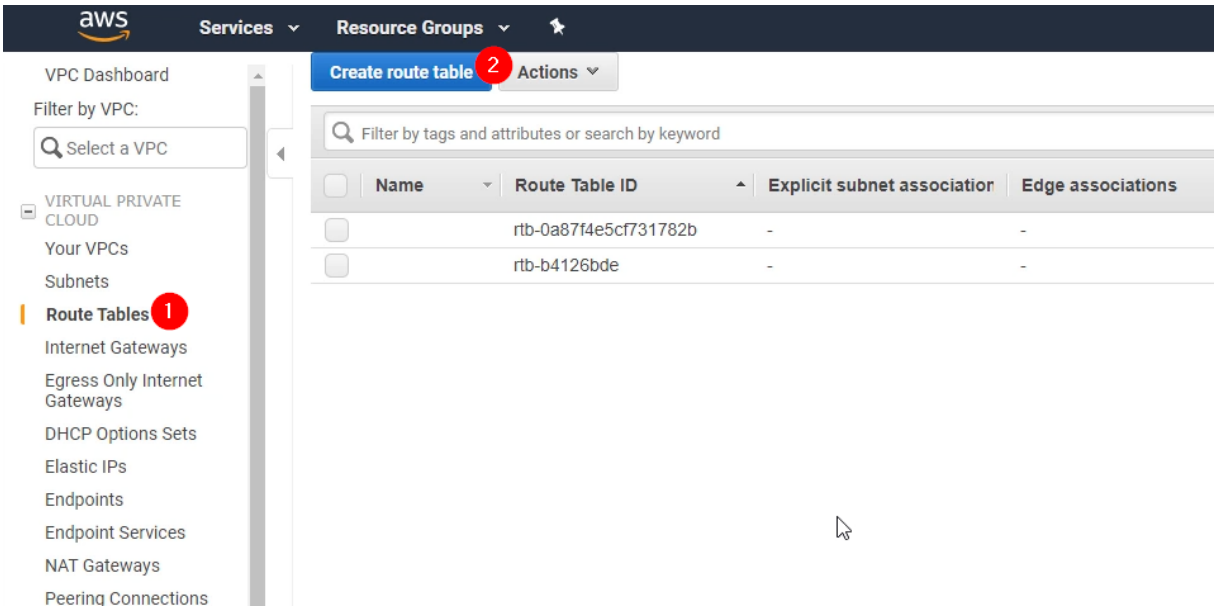


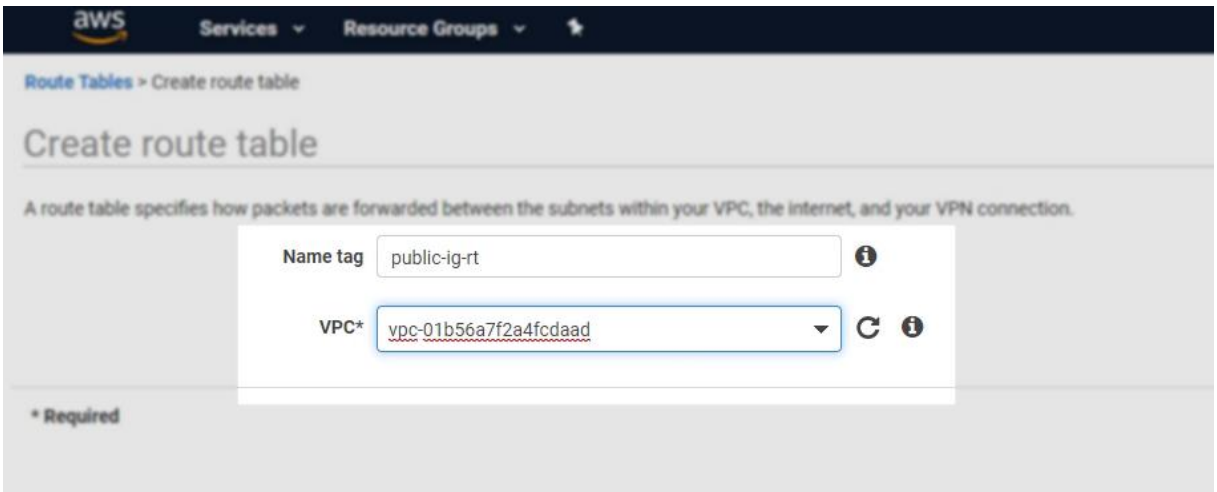
Then attach it to your VPC:



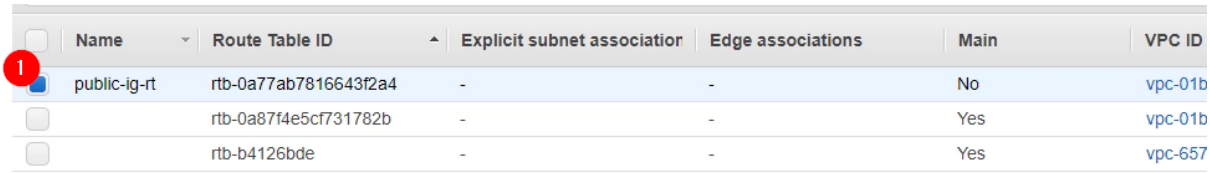
Create a Route Table

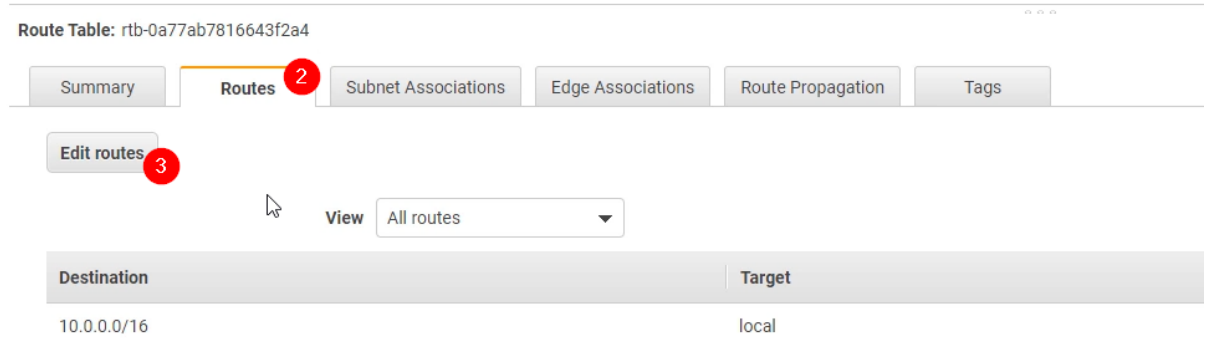
To route traffic from your public subnet to the Internet Gateway you have to create a new Route table:



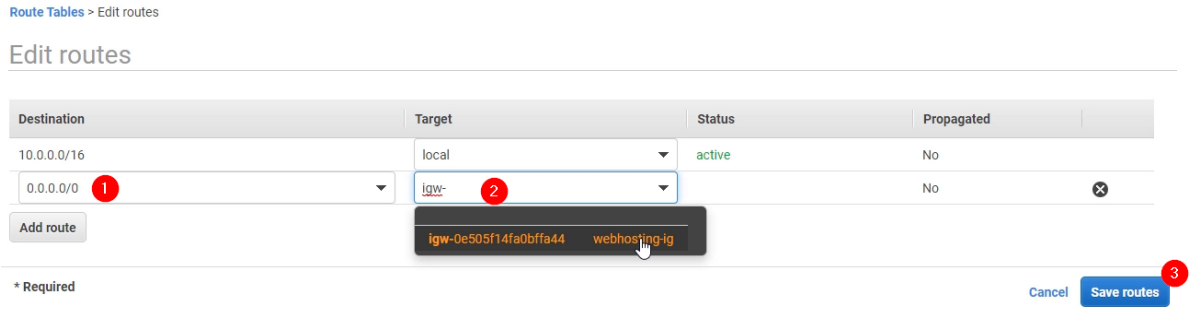


Edit the new Route Table:



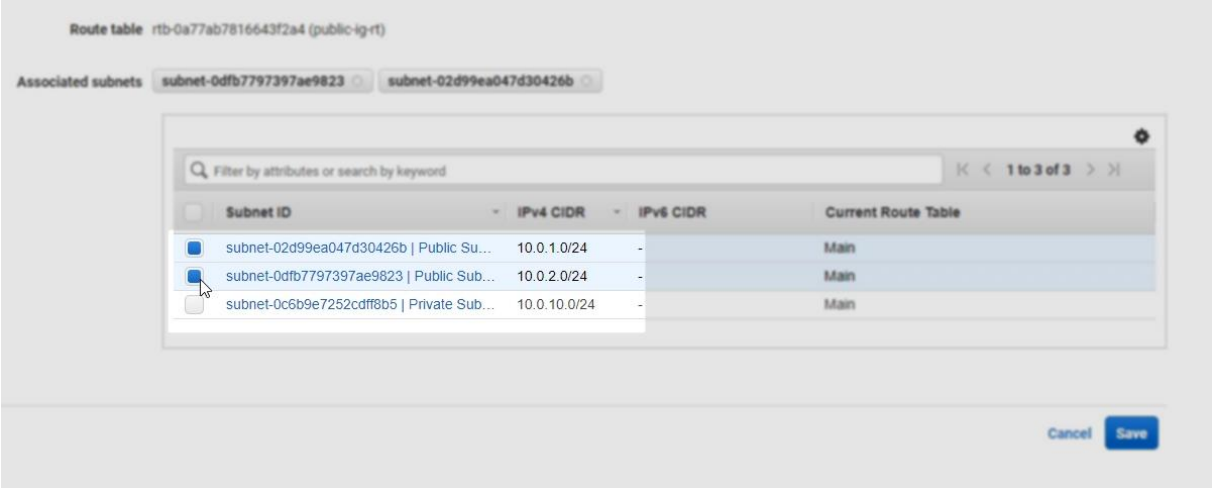


Select 0.0.0.0/0 for the destination and the newly created Internet Gateway for the Target:



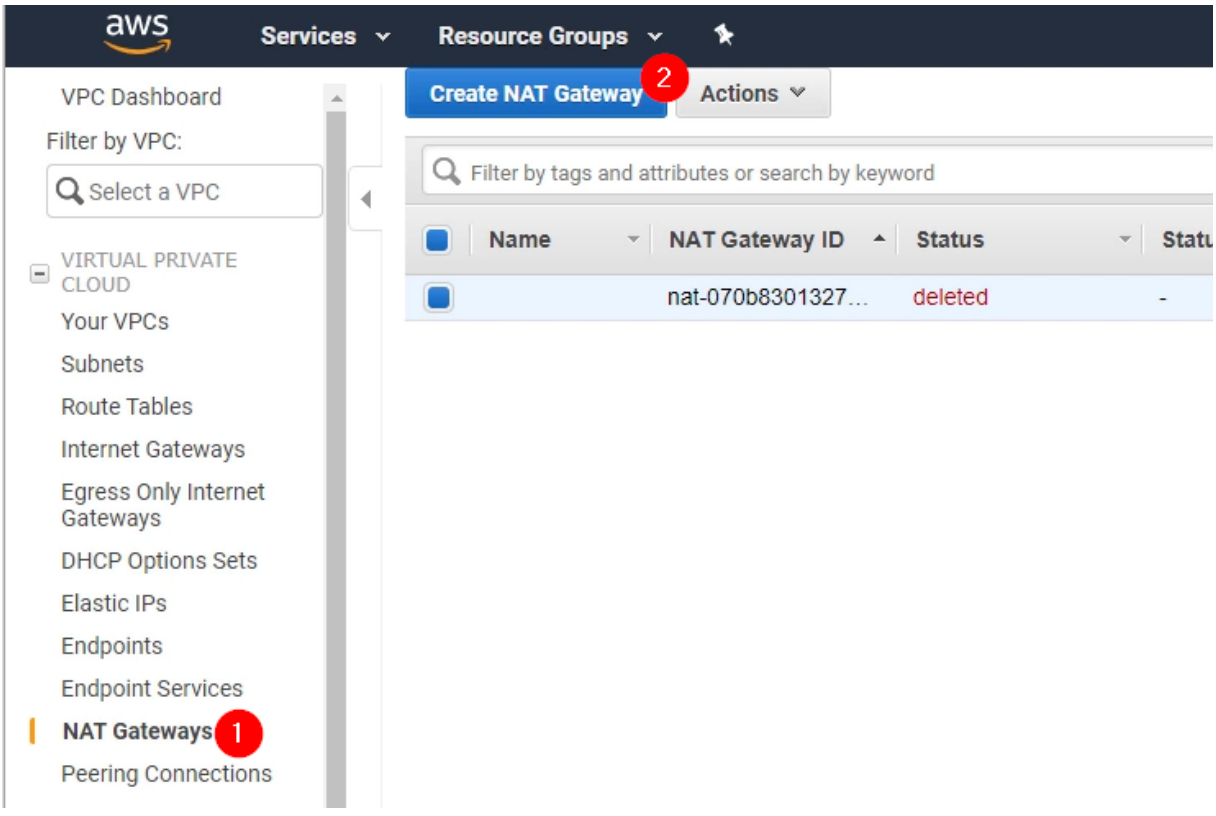
Associate the right subnet with the new route table:

Select the public subnets:

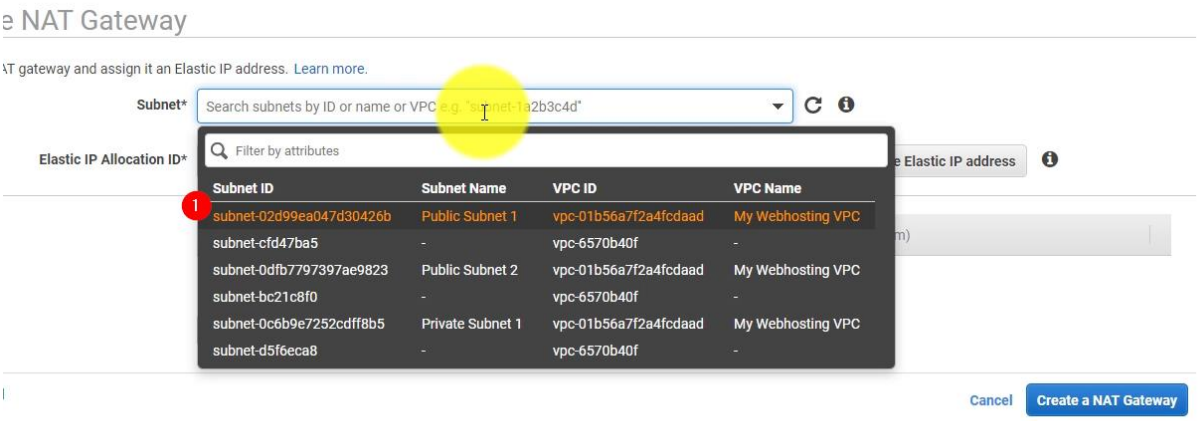


Create a NAT Gateway

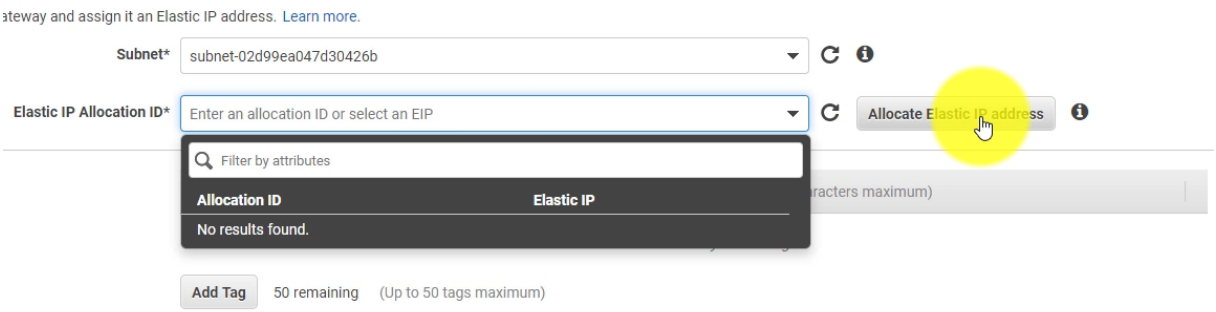
For your private subnet to get internet accessibility create a NAT Gateway:



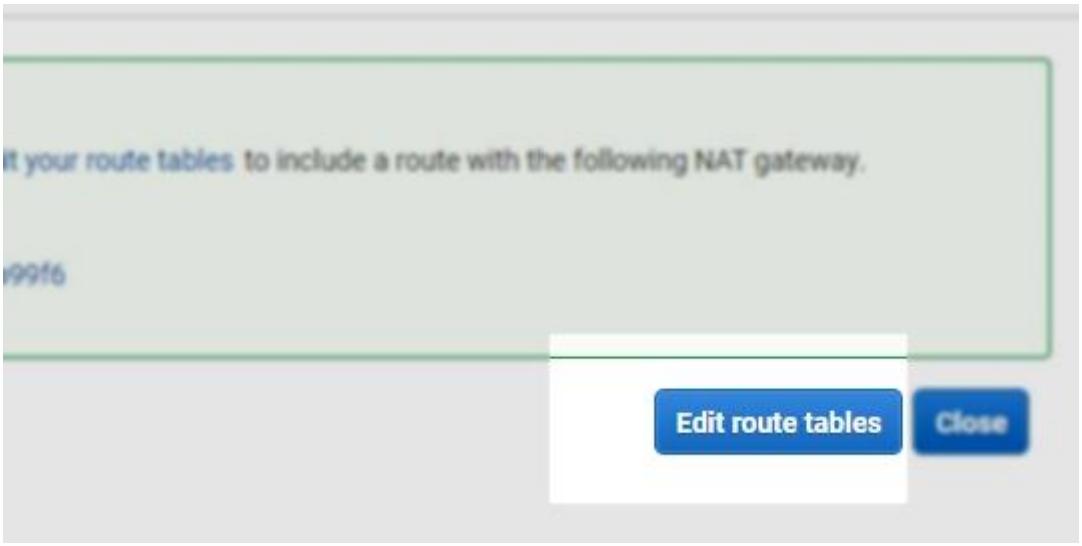
You need to place your subnet in a *public* subnet, because the NAT Gateway needs internet access:



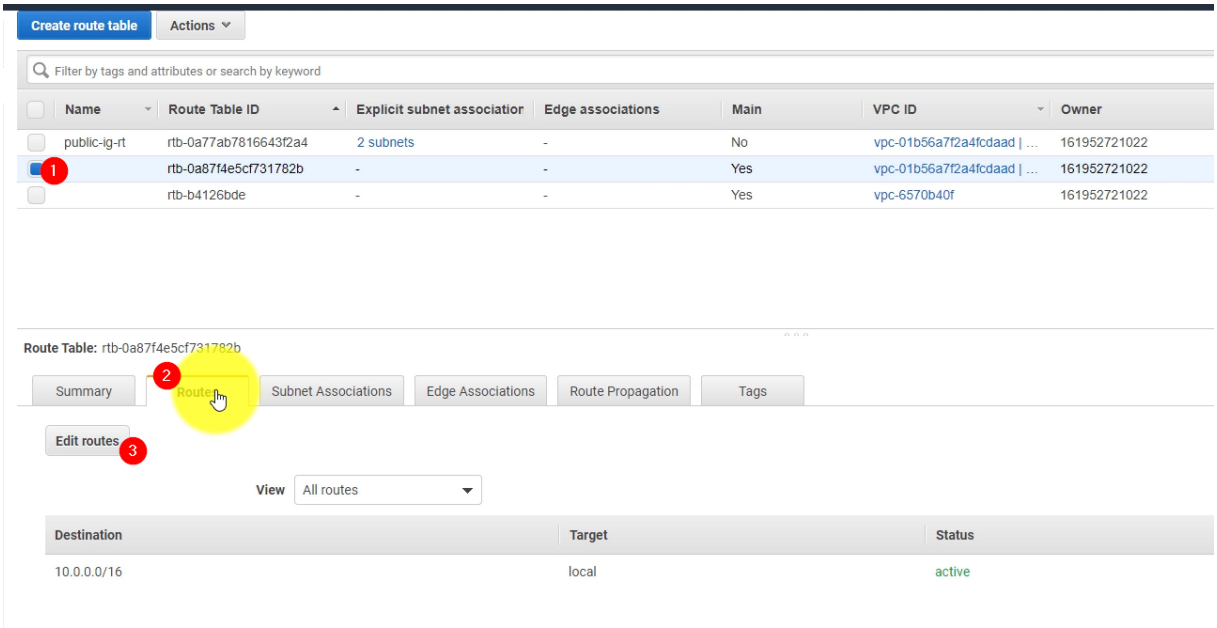
Allocate a new Elastic IP Address:



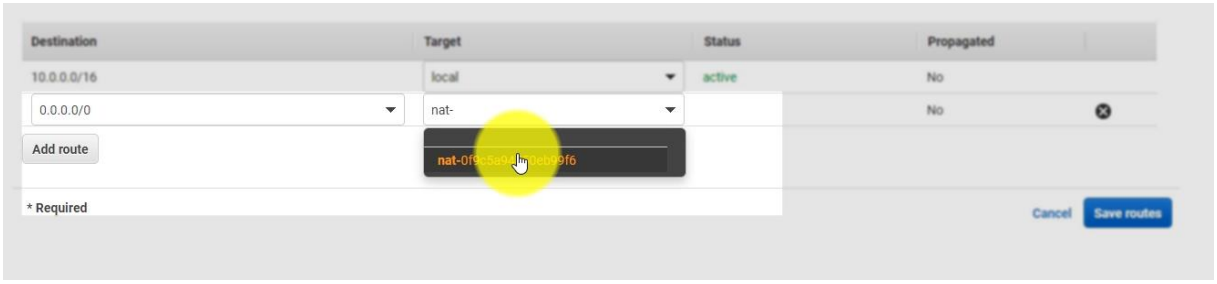
Edit the Route Tables:



Edit the Main Route Table for your new VPC:

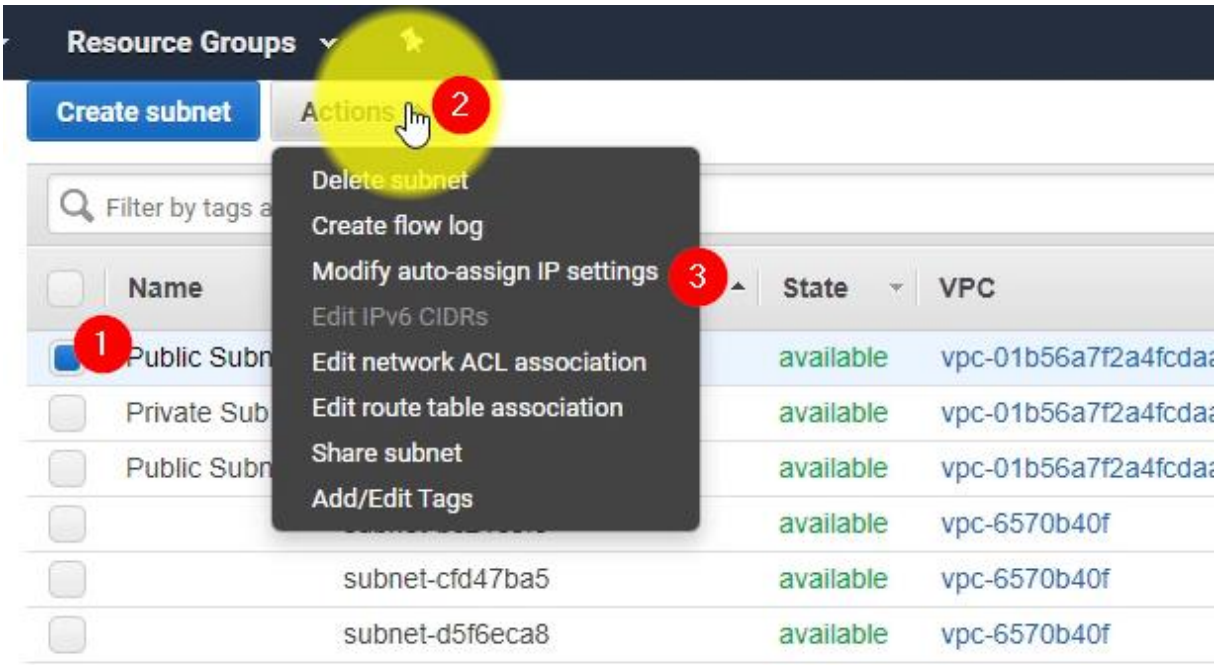


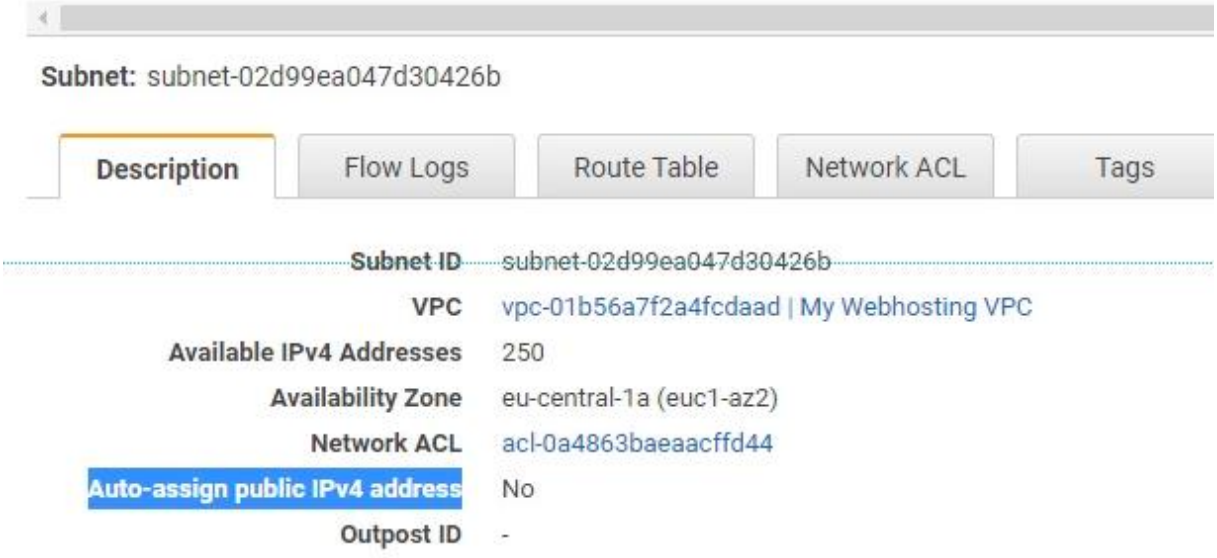
Select the traffic destination 0.0.0.0/0 with the NAT Gateway as your target:



Auto-Assign a Public IP in public Subnets

Modify *both* public subnets and activate that IP Addresses are automatically assigned:



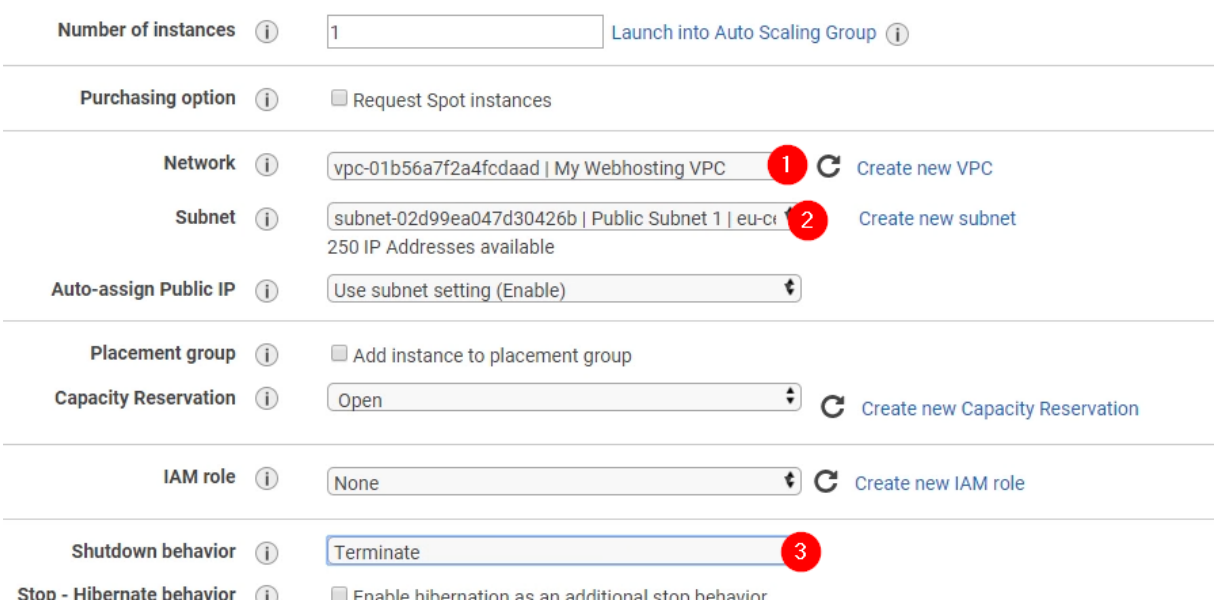


Launch a Bastion Host Architecture

Head over to the EC2 Dashboard and launch two instances. One in the Private Subnet and one in the public subnet with user-data.

1. Select the Amazon Linux 2 AMI
2. Select the t2.micro

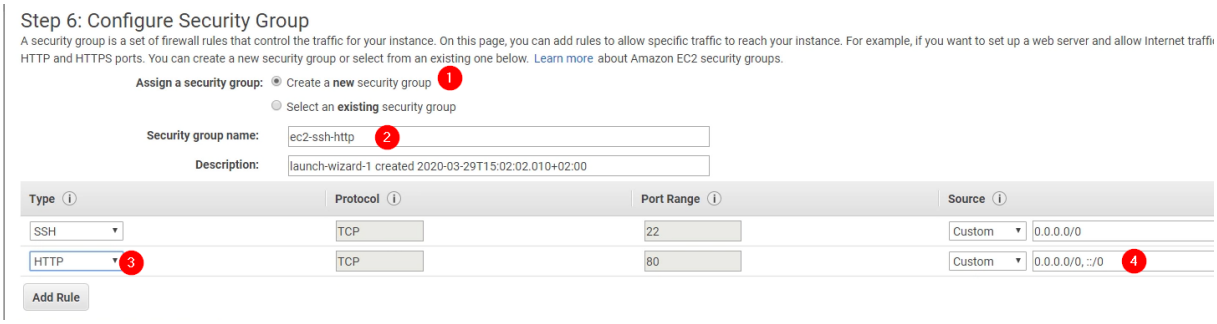
Select the new VPC (1) and the public subnet for one instance, and the private subnet for another instance (2). Also select that instances should terminate on shutdown (3):



As User-Data enter the following (for both instances):

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
echo "hello apache" > /var/www/html/index.html
```

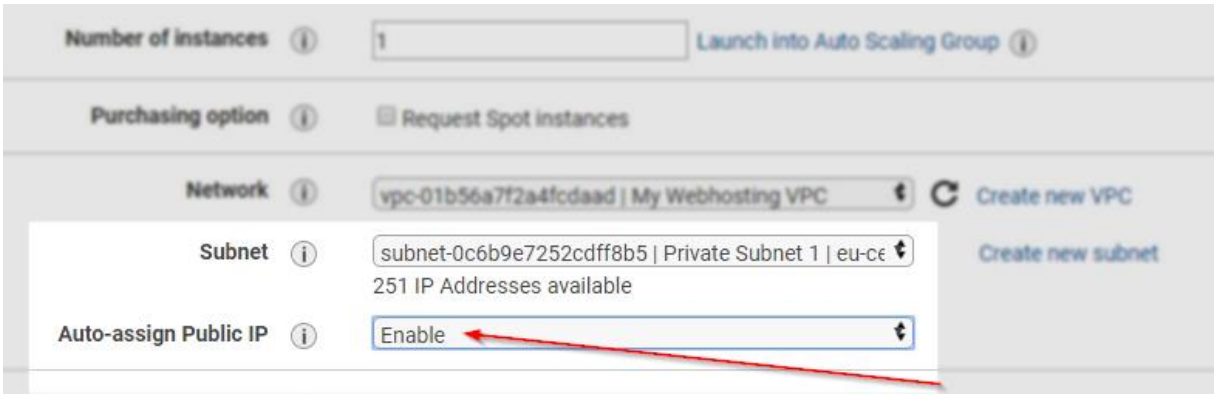
For the first instance add a new security Group which allows HTTP Access from anywhere:



Then launch the instance.

Launch the instance in the private subnet

For the second instance, do exactly the same as for the public instance, just launch it into the private subnet, but still enable public IP address:

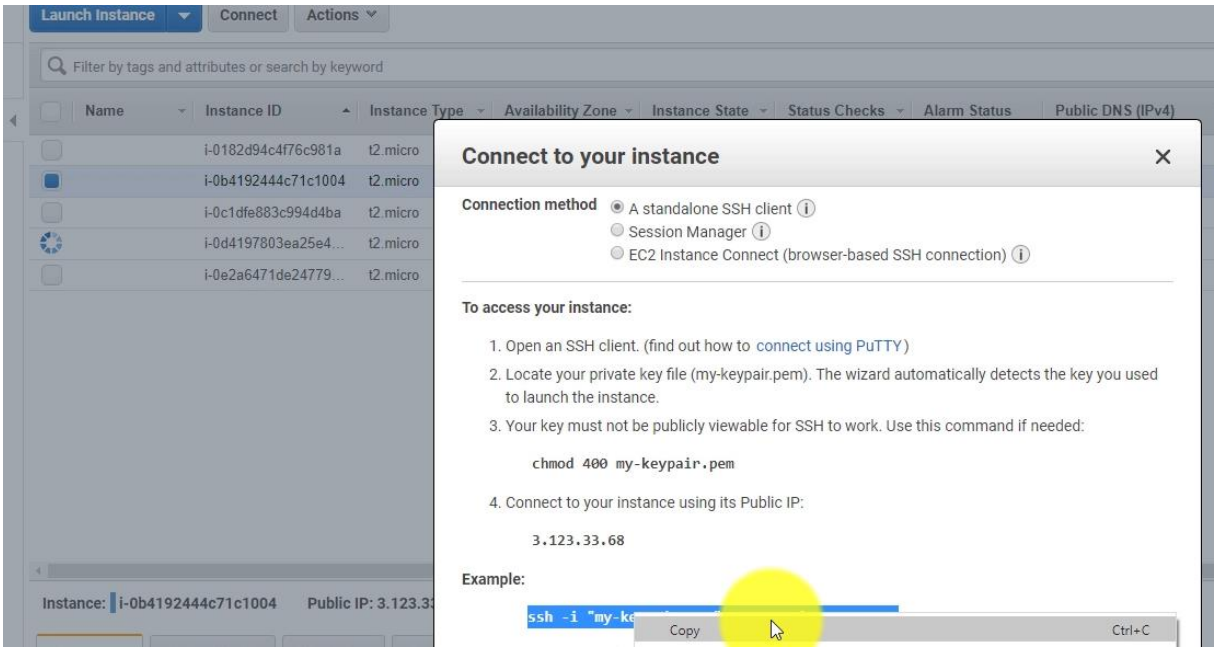


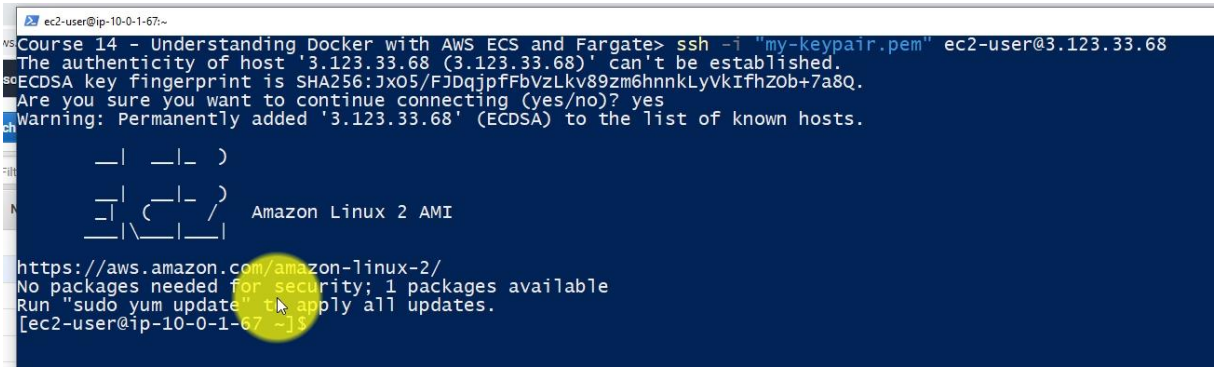
Add the same user-data, select the same Security group we created for the previous instance.

Then launch the instance.

Access Instance in Public Subnet

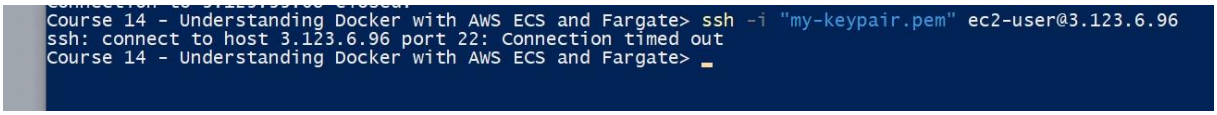
SSH Into the Instance in the public subnet:





Try to Access into the EC2 Instance in the private Subnet

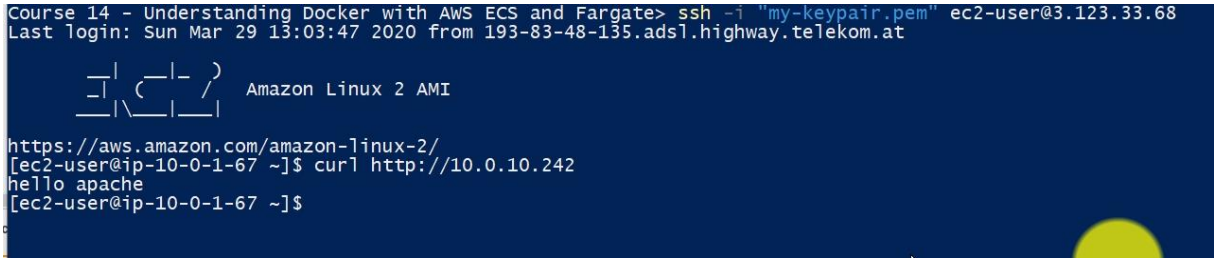
Observe a connection timeout when you try and connect to the instance in the private subnet:



Access the private instance via the bastion host

Our Instance in the public subnet acts as a bastion host. SSH into the public instance and then from there connect to the private instance:

1. SSH Into the instance in the public subnet
2. Curl from there to the private IPv4 Address of the instance in the private subnet
3. You should see the output from Apache.
4. That means you can connect via the bastion host



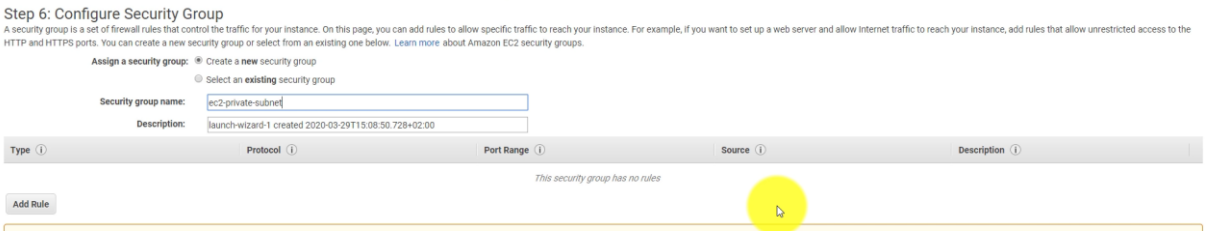
You can safely terminate your instances now to save Free-Tier credits.

Use a Load Balancer to connect to Instances in private Subnets

Launch a private EC2 Instance

Launch again an EC2 Instance with the same AMI, same Instance type as before, same User-Data.

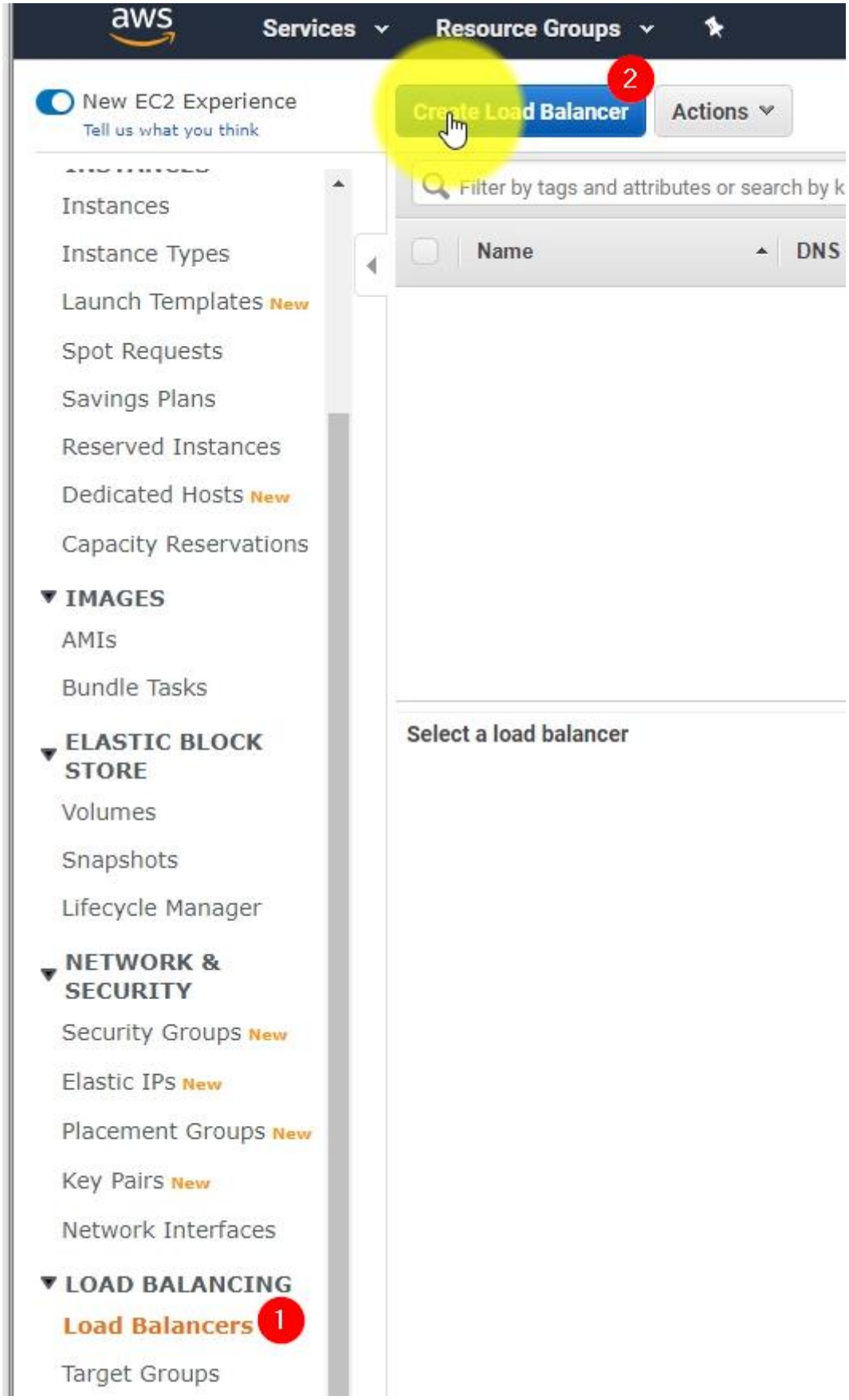
For the security group, create a new security group and remove *all* rules:



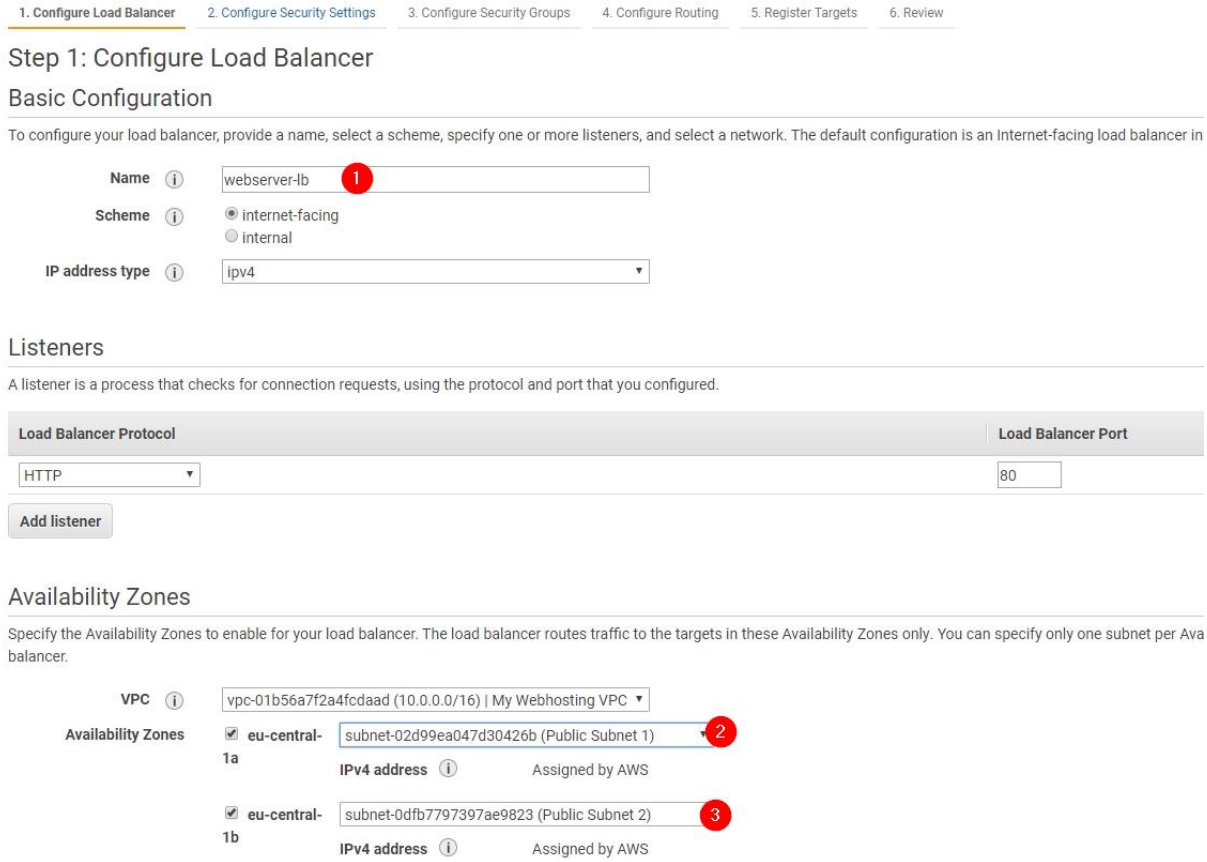
Then launch your instance.

Create an Application Load Balancer

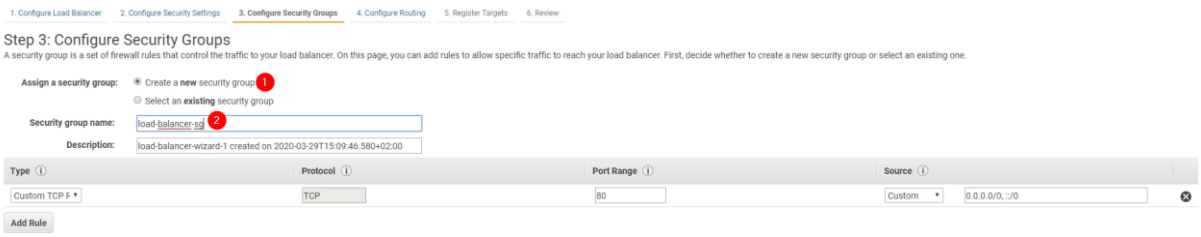
In the EC2 Dashboard select Load Balancer and hit “Create Load Balancer”



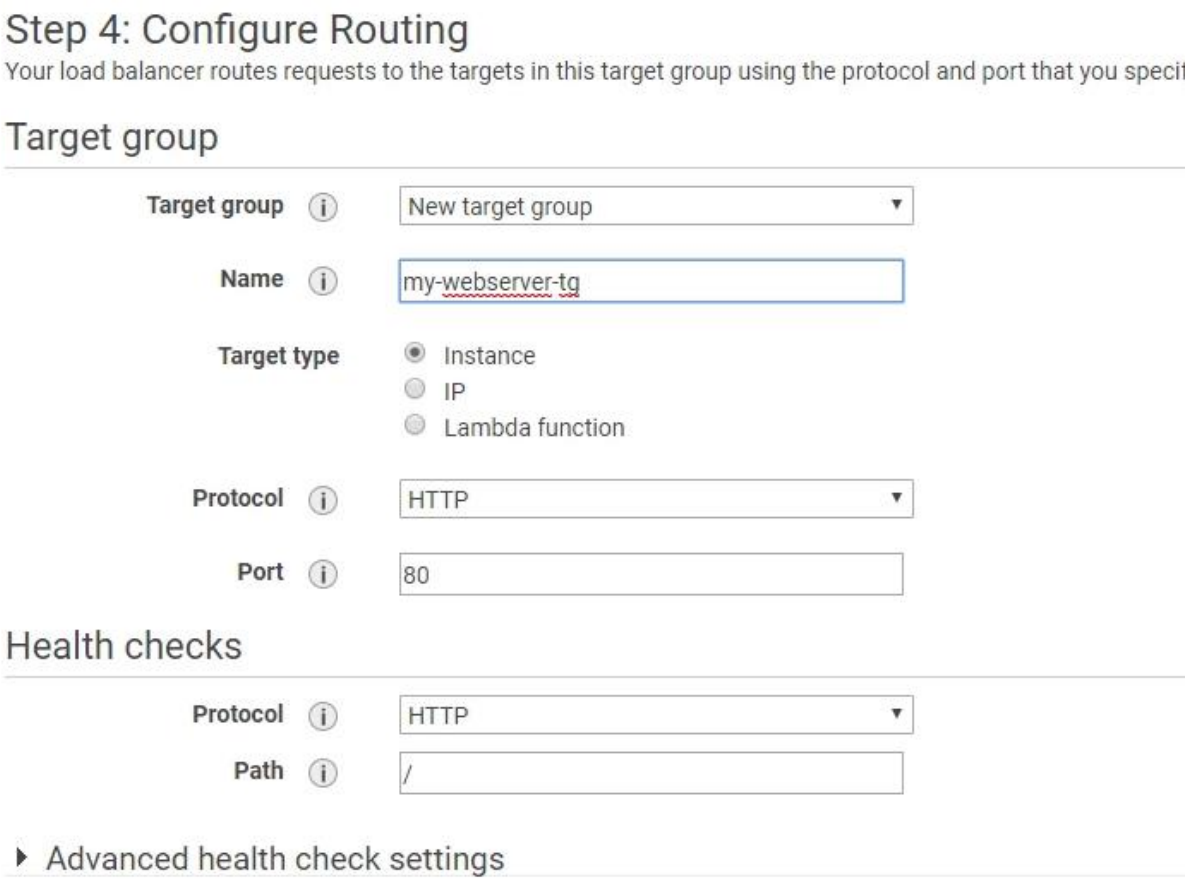
Select an Application Load Balancer. Give the Load Balancer a name (1) and place it into your two *public* subnets (2) and (3):



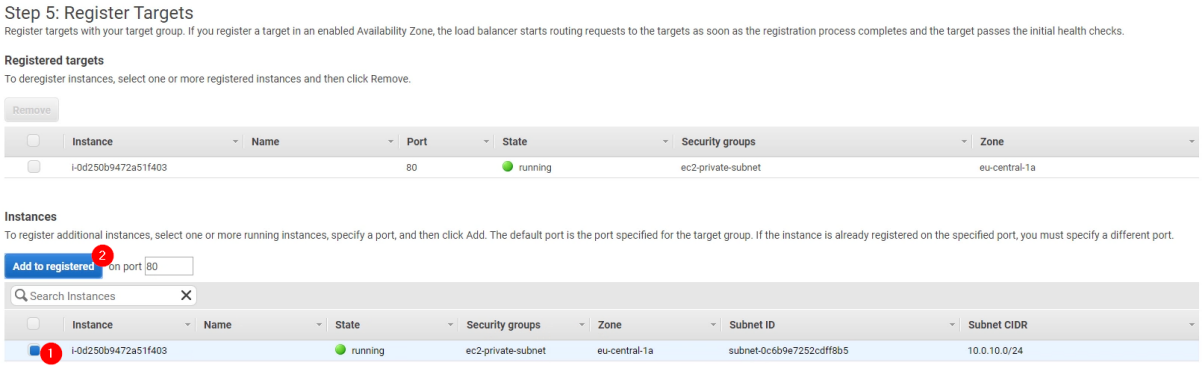
Attach a new Security Group to the Load Balancer:



Create a new Target Group for the Load Balancer:



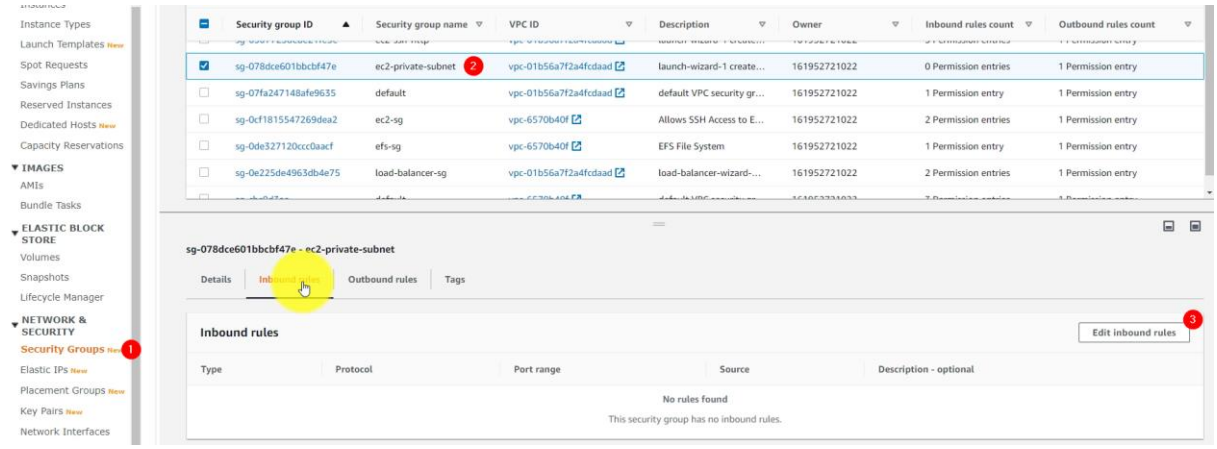
Register your Instance in your private subnet into the Target Group:

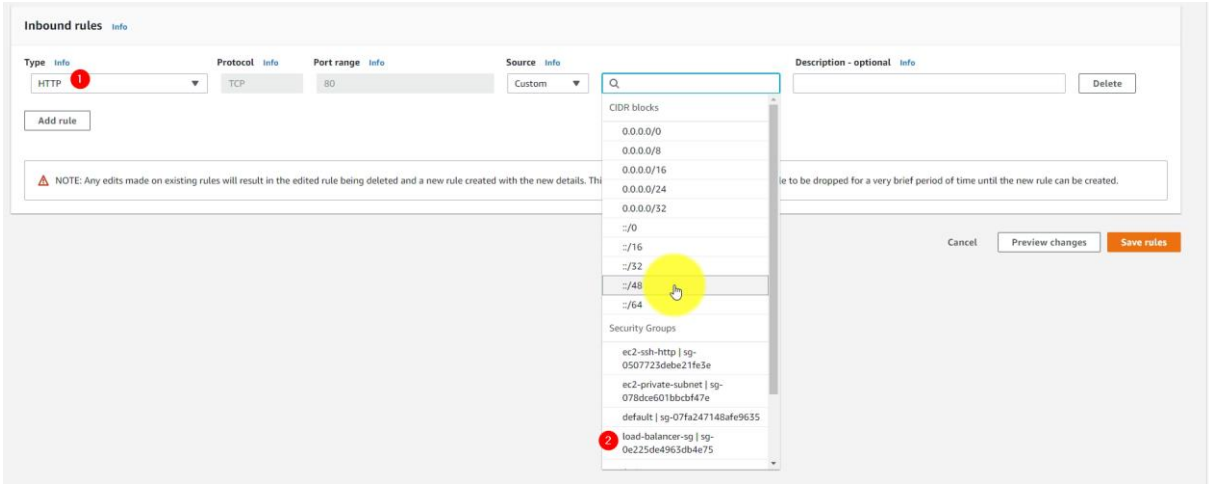


Then create the Load Balancer.

Allow Load-Balancer Traffic in the Security Group

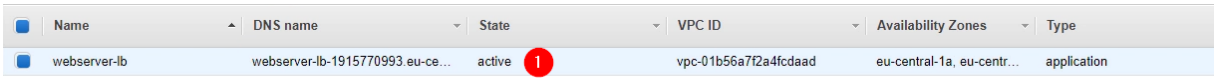
In the ec2-instance security group edit the inbound rules to allow Traffic from the Load Balancer to the EC2 Intance:

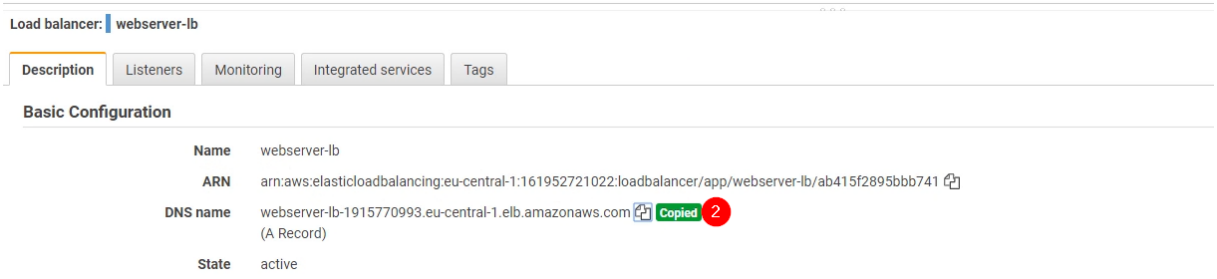




Test the Load Balancer

Wait until the load balancer is active, then copy the DNS Name and open the url in a new Tab:





You should see the hello apache string:

Clean Up

Tear down everything again:

1. Terminate the EC2 Instance
2. Delete the Load Balancer
3. Remove the Target Group
4. Delete the NAT Gateway
5. Disassociate the Elastic IP
6. Then Release the Elastic IP
7. Disassociate the Public Subnets from the Custom Route Table
8. Delete the Custom Route Table
9. Detach the Internet Gateway from the VPC
10. Delete the Internet Gateway
11. Delete the three Subnets from your VPC
12. Delete the VPC
13.