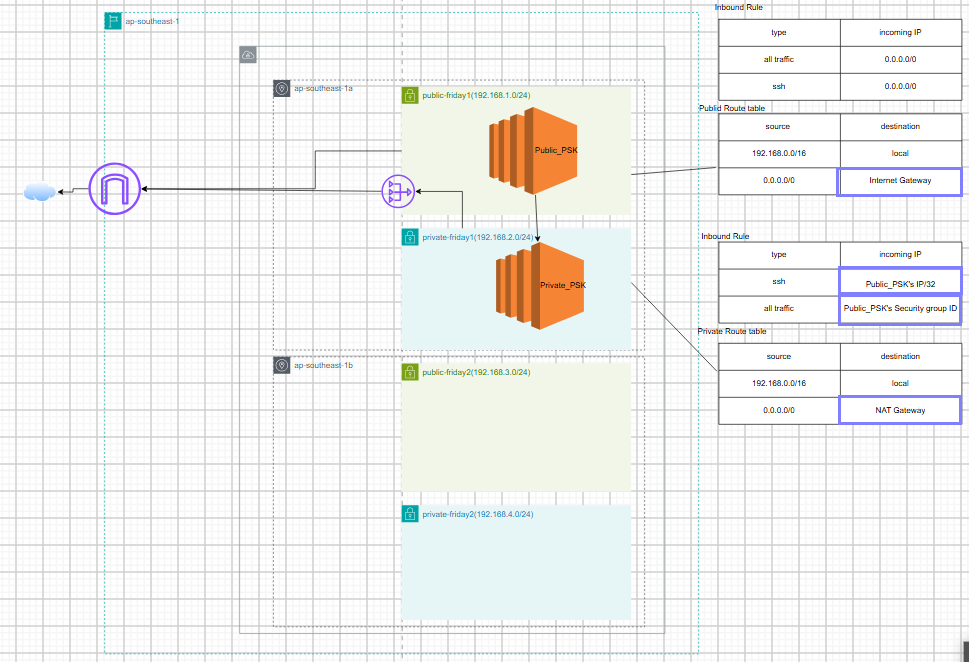
We are going to explore AWS resources with Practical Lab. Some prerequisites are

1. AWS free tier account
2. And Laptop. Yes, that is.

The reason behind this lab is to practice some beginner lab and make some memory about my achievement. This following diagram is what we’re going to practice.



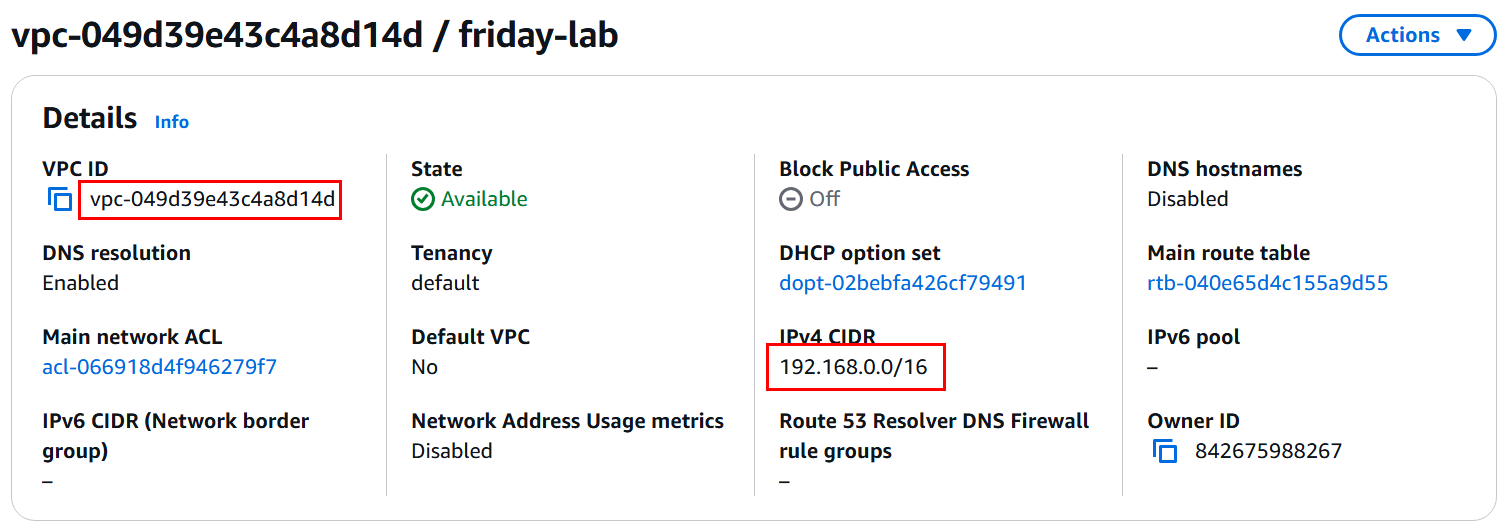
I will explain the overview of the project. There is sensitive server with private subnet which we can only access from another server within public subnet. And we want the sensitive server to get data from Internet. You can use a public NAT gateway to enable instances in a private subnet to send outbound traffic to the internet, while preventing the internet from establishing connections to the instances.

Let get start,

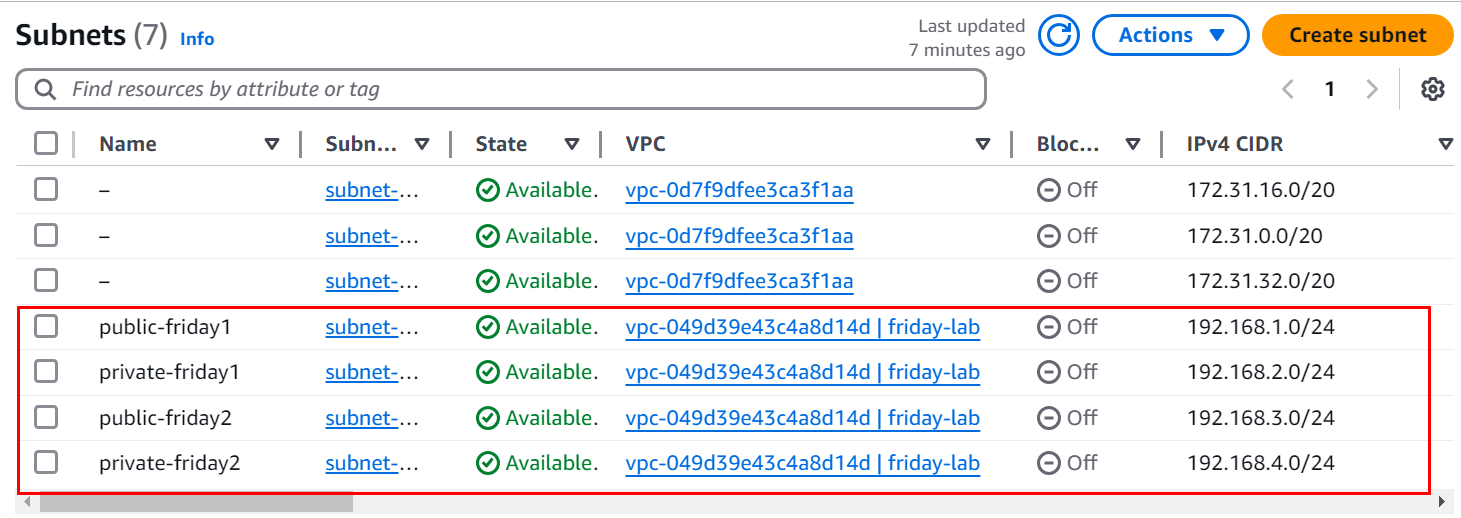
At first,

We already have default VPC in every region. But I want to make custom VPC so that we can build custom route table and subnets. The following image illustrates detail information of custom VPC named friday-lab because I make this on Friday. Anyway,

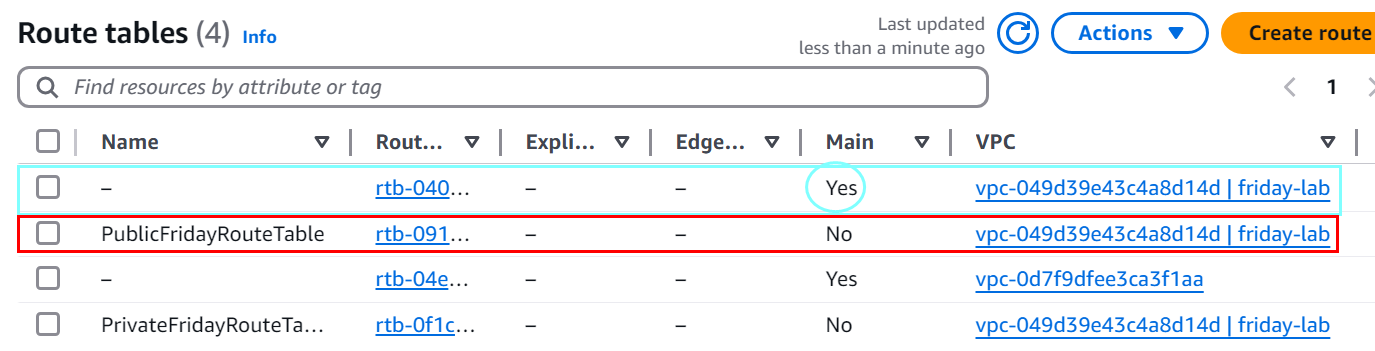
We focus on the IP address which is the whole ip address range we can use within this VPC.



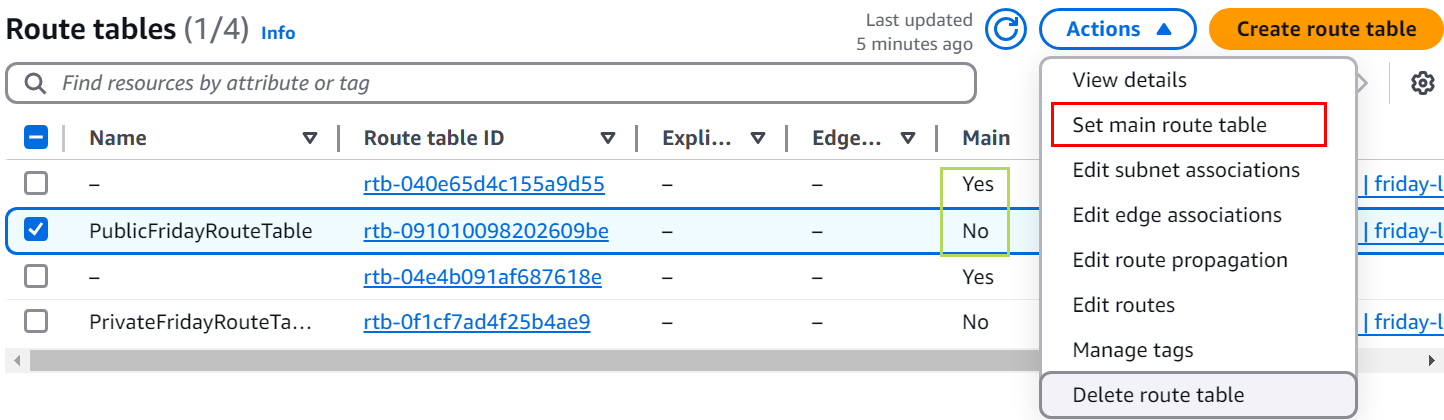
As you can see, I create two public and private subnets as shown in diagram. Each of them are associated with the custom VPC and have different private ip range within IPv4 CIDR block range.



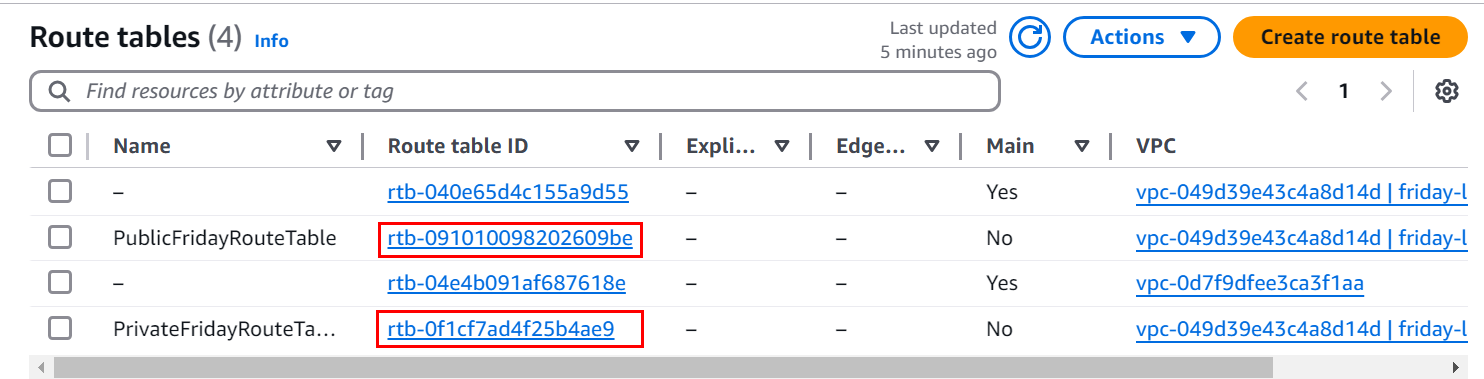
After we create custom VPC , amazon automatically create default main route for that VPC.  
I have to create public route for Internet access and private route for local access.



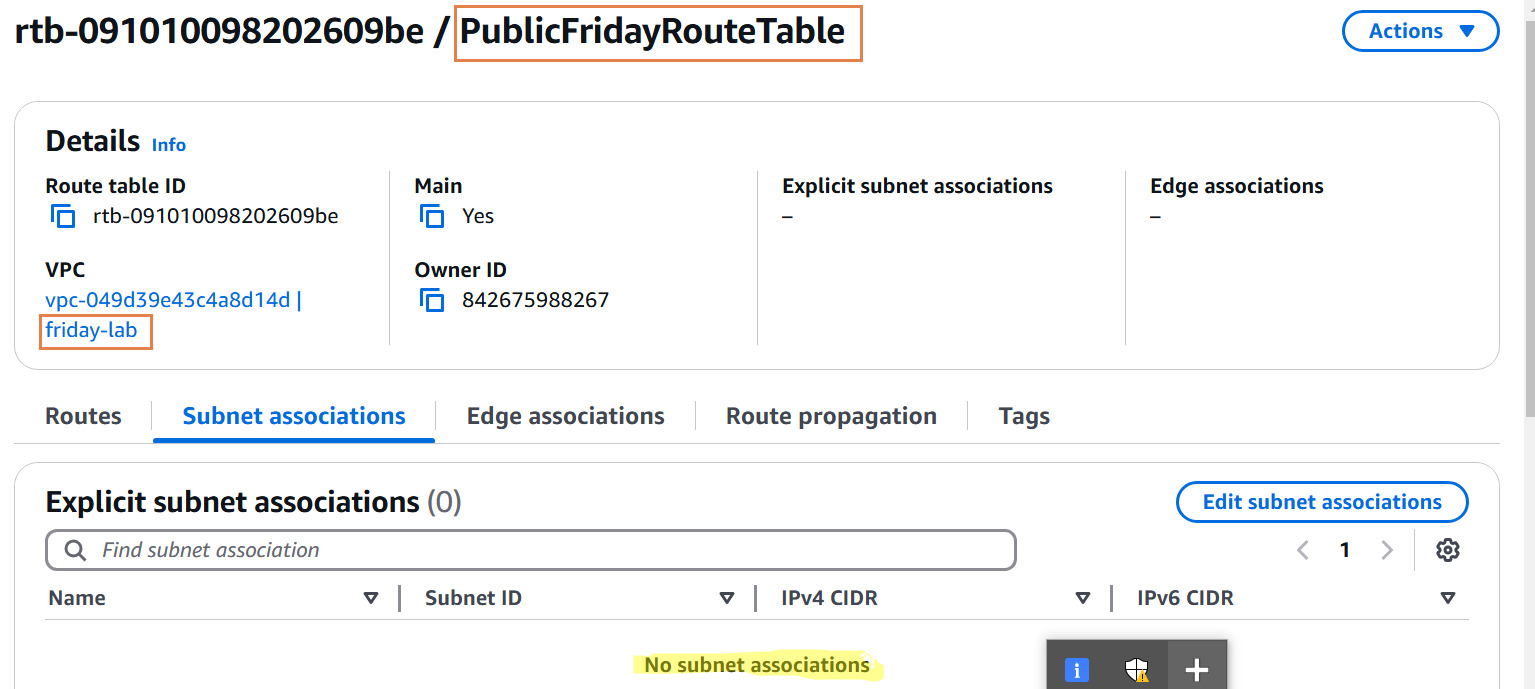
In order to set my custom route as a main route.



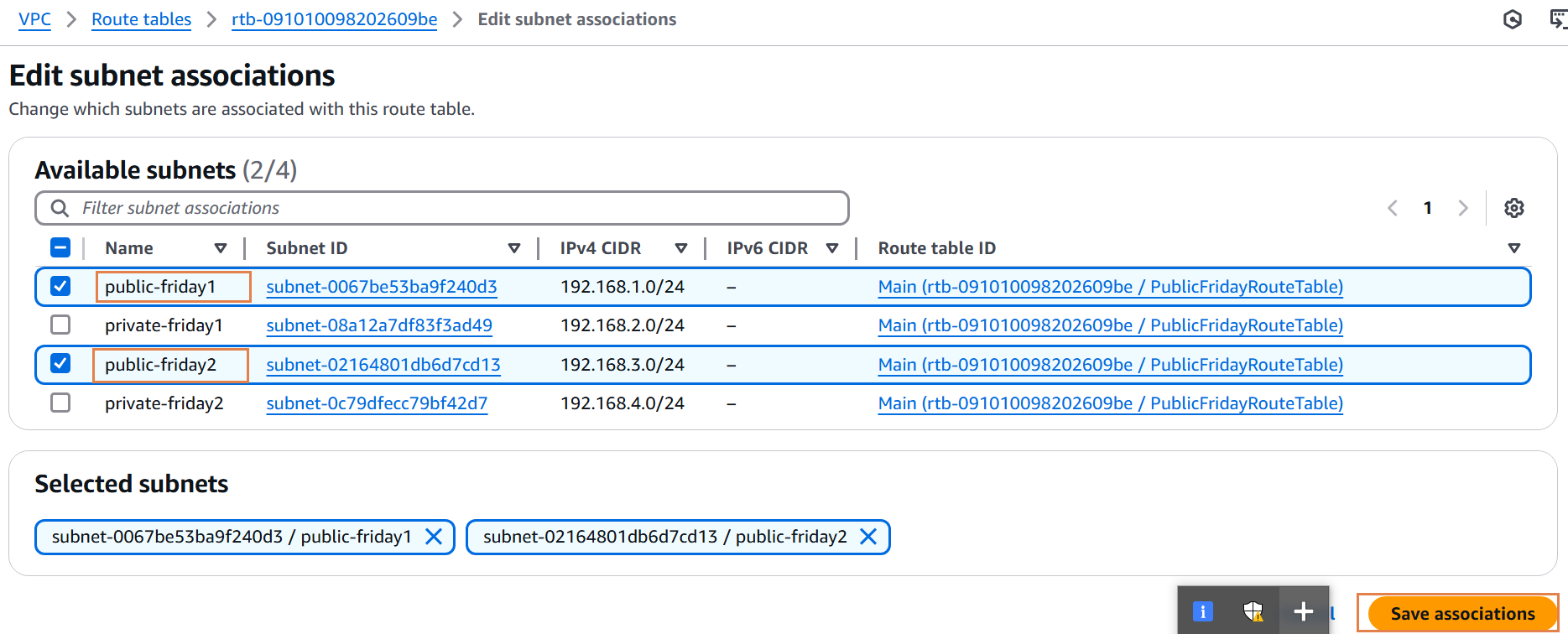
Select the custom public route and click actions, click Set main route table. After that, you can delete the default route if you want to. We need to associate subnet with appropriate route table. Click the Route table ID of public route.



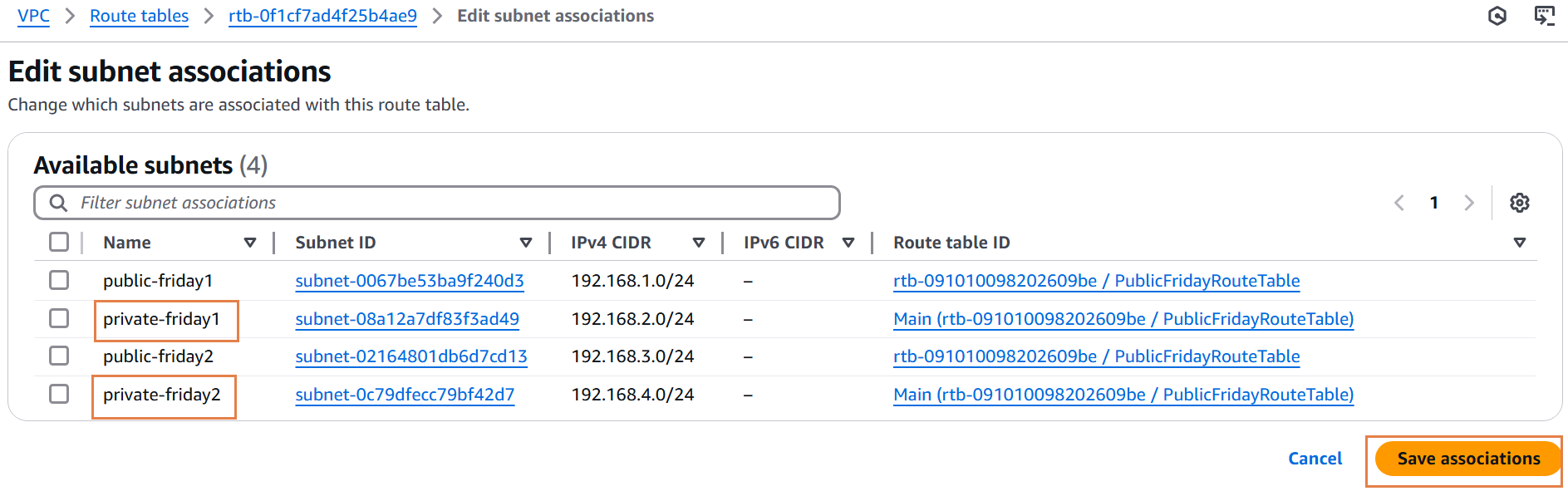
Click Subnet associations and under edit subnet associations, we can add public subnets with public route table. Noted that subnets which connected to public route and Internet gateway can only access the Internet.



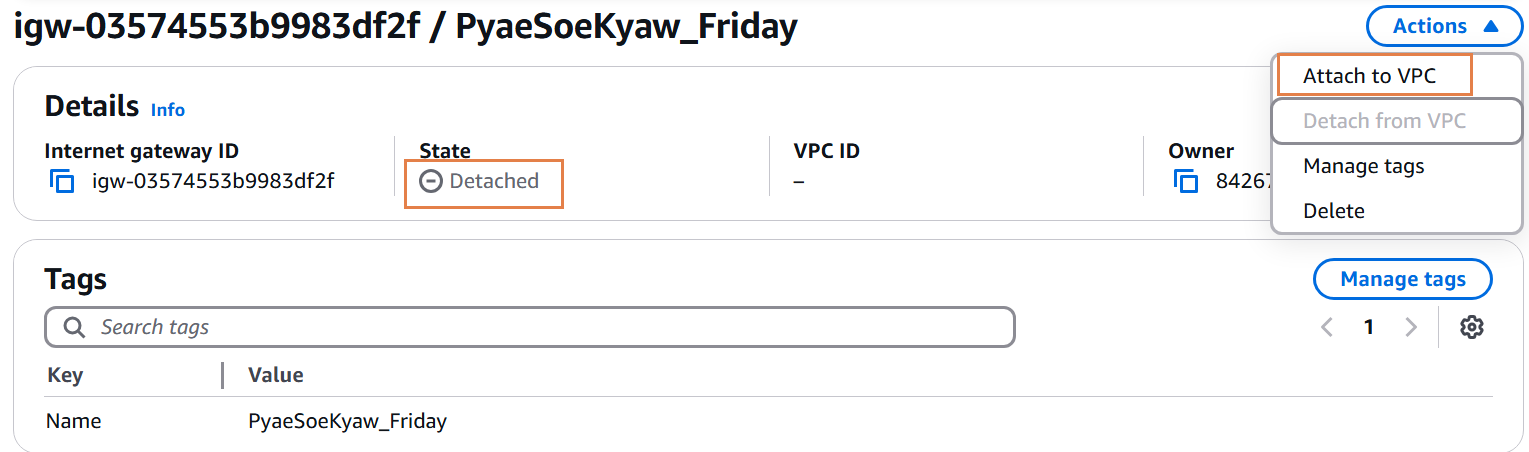
Check the public subnet and click save associations.



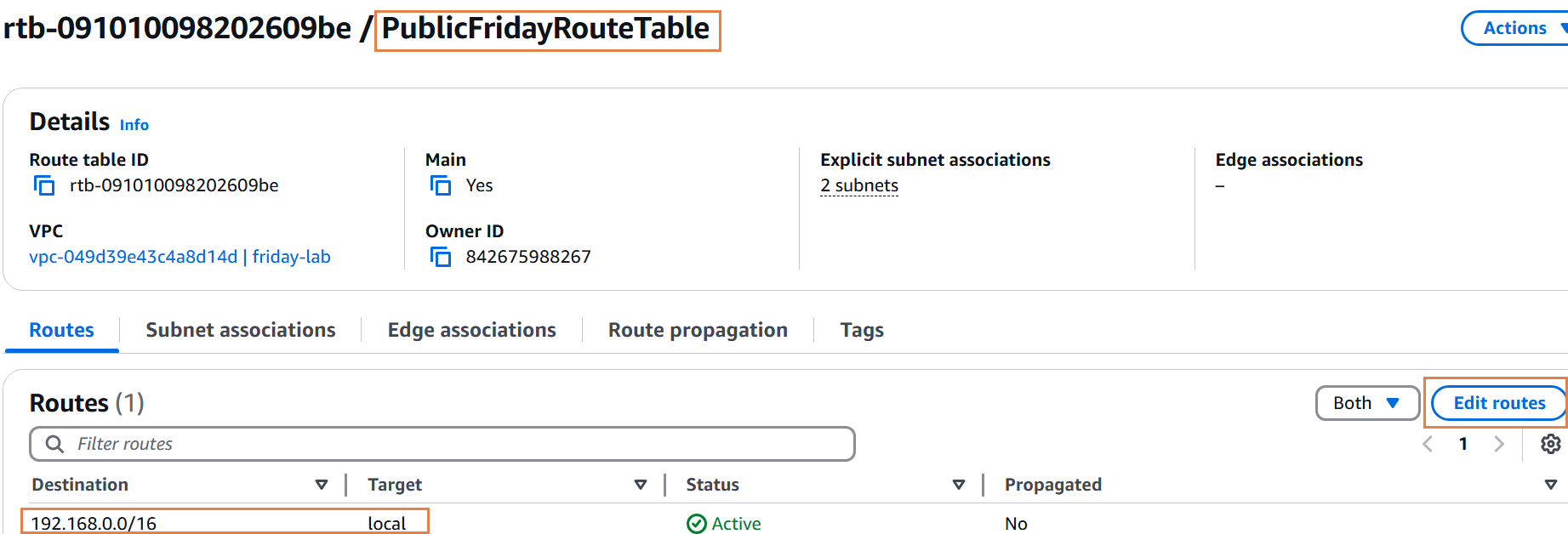
Just like the way we use preview image, we make another associations for private route with private subnets.

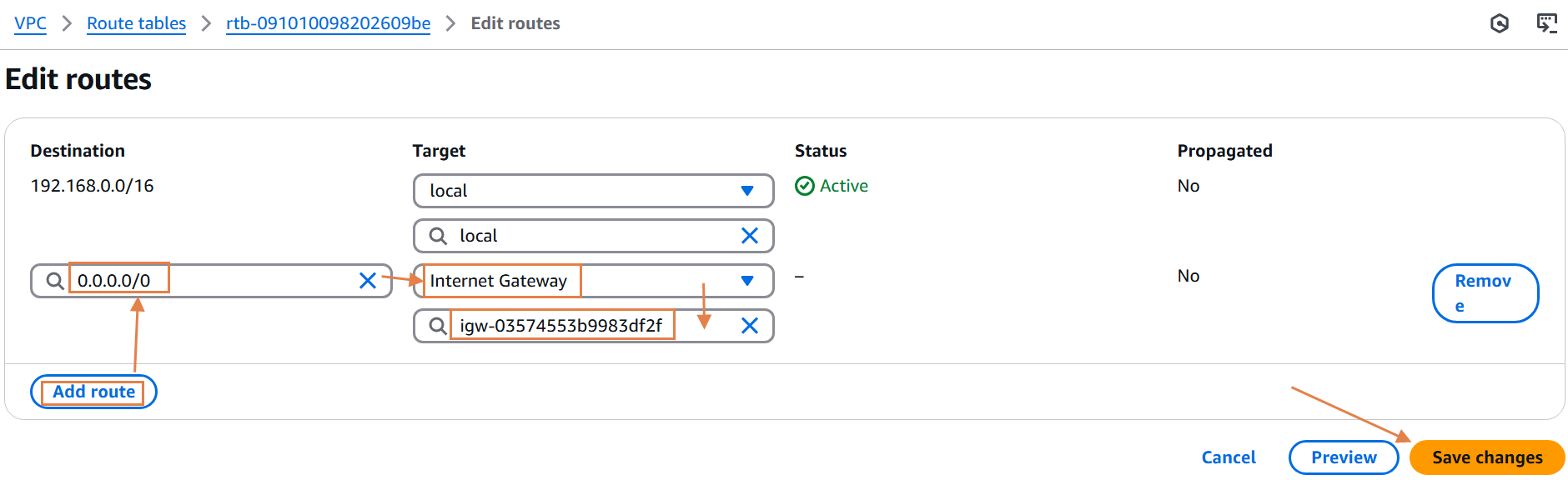


After that step, create custom Internet Gateway. Initially, It is not attached to any VPC so click Actions and Attach to VPC.

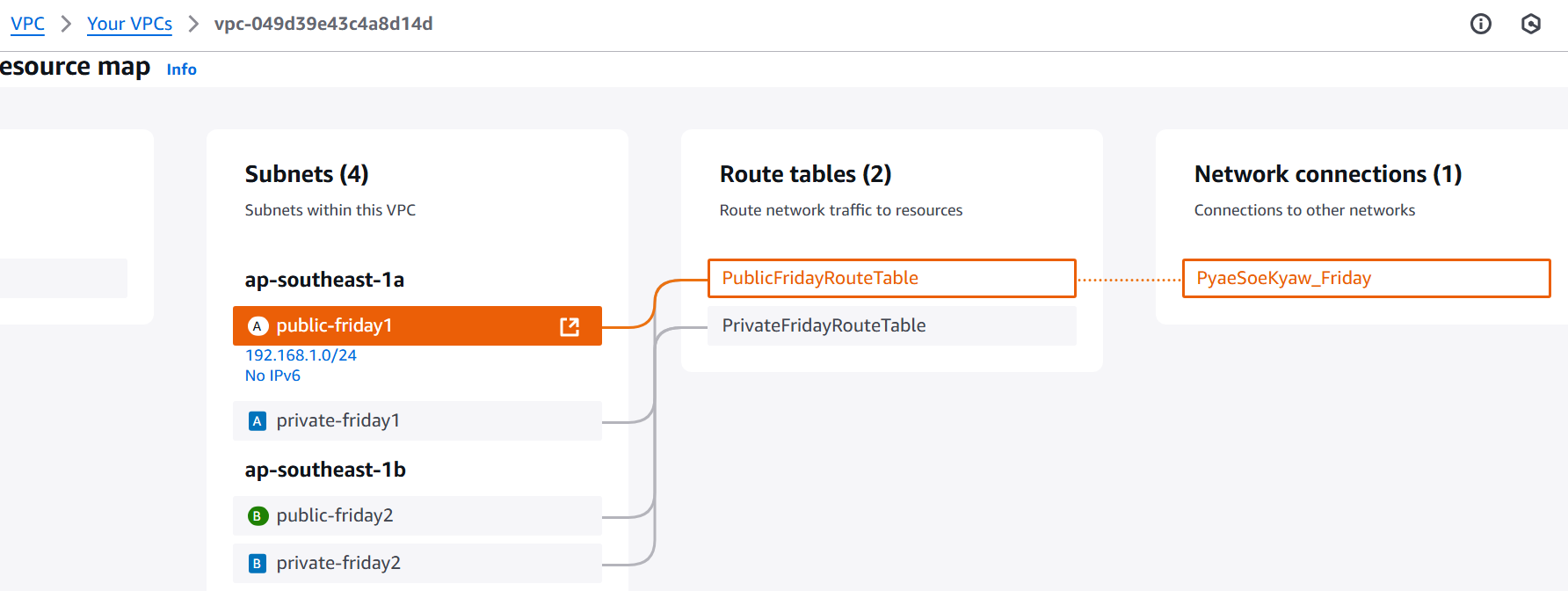


Make sure we attach our custom VPC with this Internet Gateway. Noted that our public route with public subnets does not have access to the Internet. For this reason, we update routing table which will lead to Internet gateway from any IP. Under route table, we can edit route to outcome our desired result.

You can see every step of adding new route and make sure Target should be our Internet Gateway.

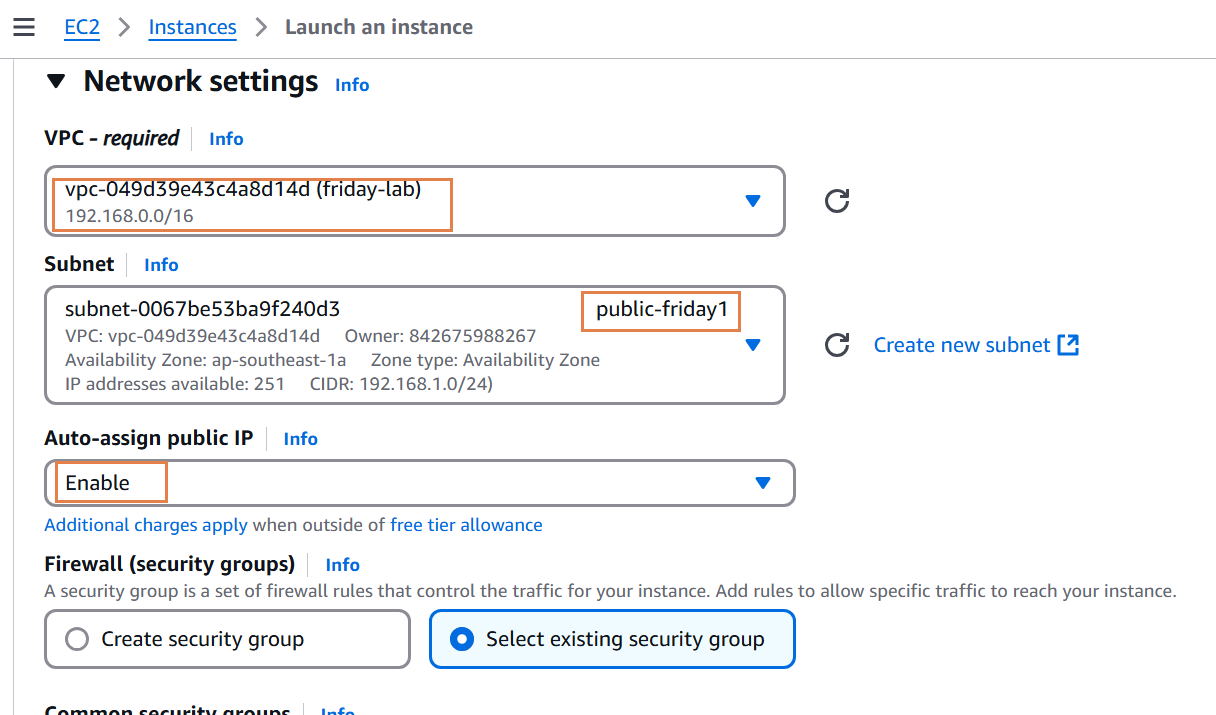


The background infra is done by now. We can see Resource map of the VPC how things are going through within the VPC. Every resource with public-friday subnet have access to the Internet.

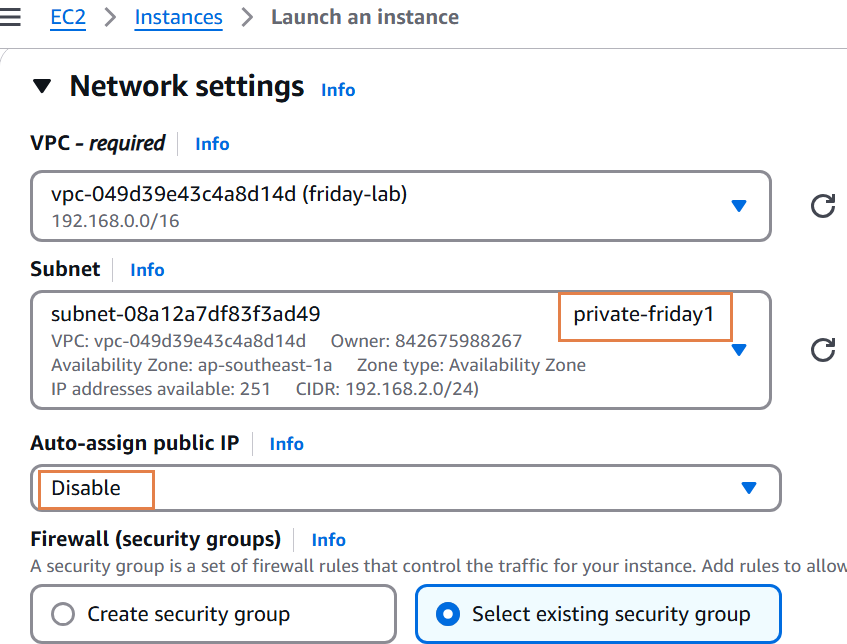


You might say “then how we access resource from Private subnet”. Even we make the resource isolated, we can still access from local resource within the VPC. “So, Can we make that resource to be more isolated from local,” answer is “Yes, we can configure with security group and make some rule to unconnectable the isolated resource.”

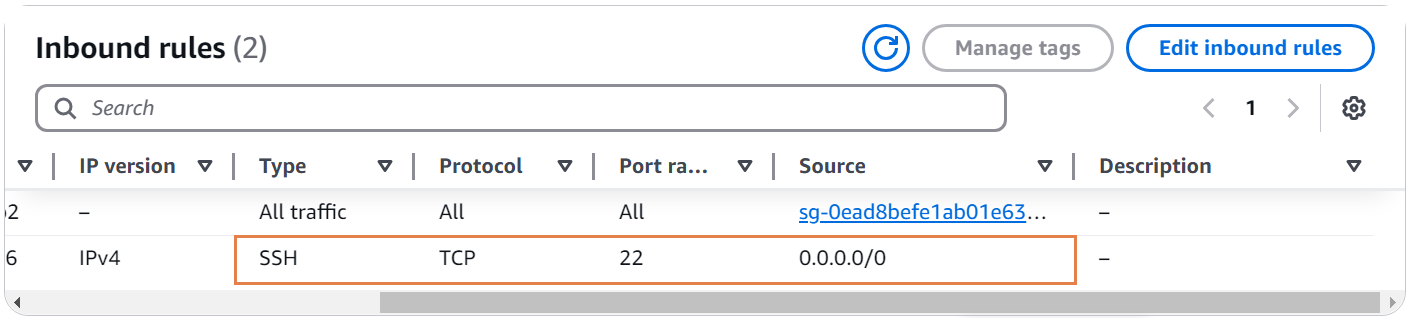
The second phrase is to create Instances. During the period of launching the instance, we need to edit Network settings so that we can implement the instance under desired VPC with specific subnet. Click our VPC (friday-lab) and choose public subnet which means this instance can access the Internet. And enable the Auto-assign public IP because we need to config from this instance to sensitive server.



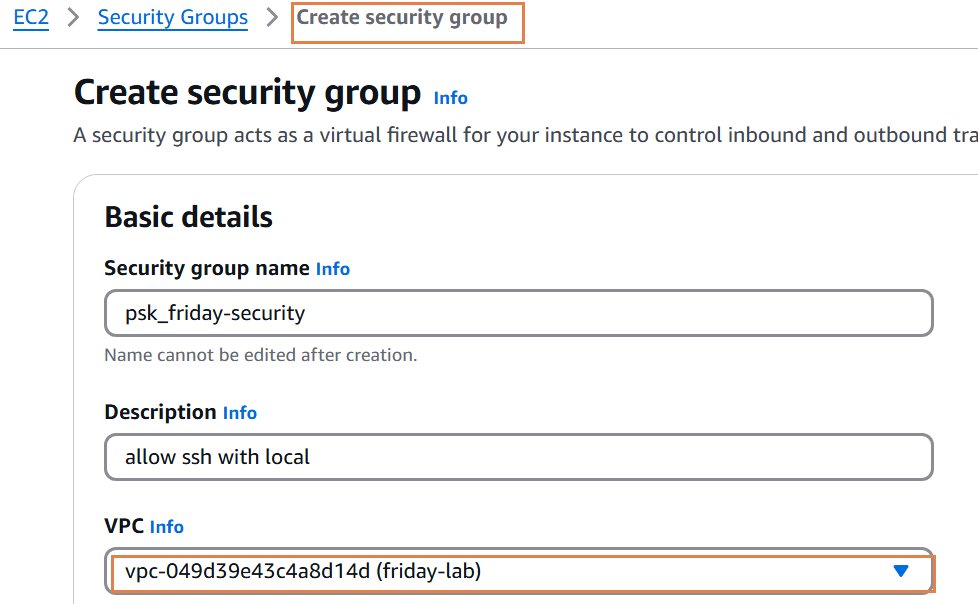
After that, we are about to create sensitive server which doesn’t require public IP and assign within private subnet. We will configure the following network setting for the sensitive server.



Since we create two servers. We will edit Inbound rules with SSH connection of Public server.



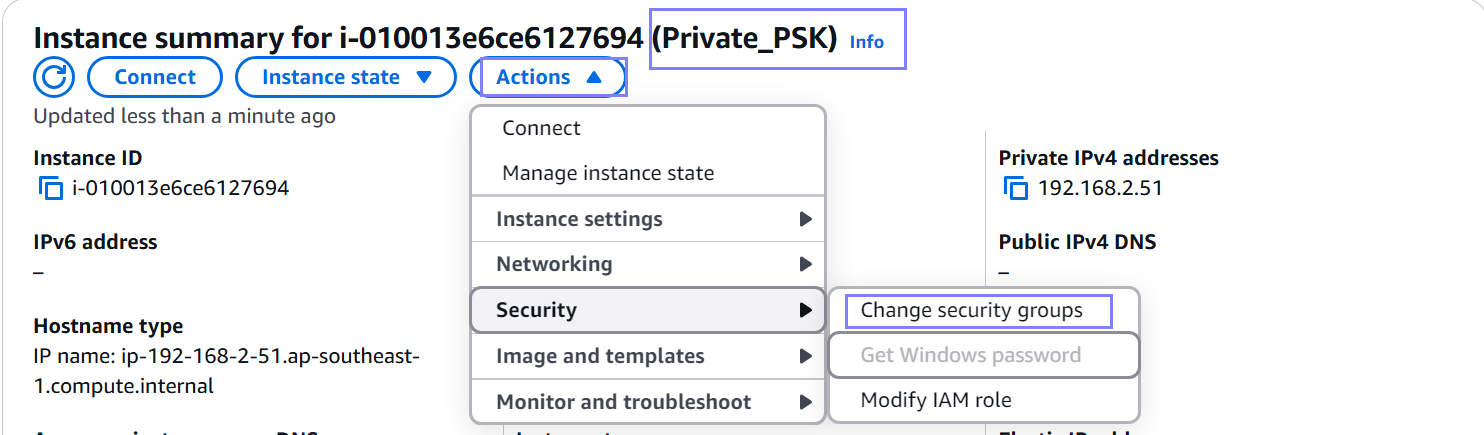
We need isolated security group which allow public server access private server within VPC.

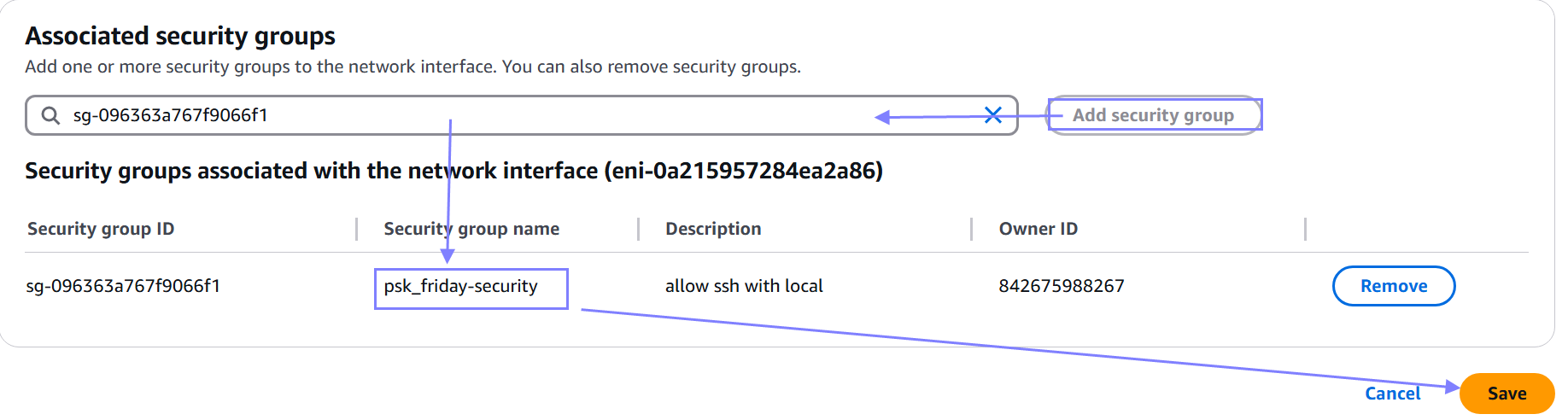


Within the new security group, we will edit the Inbound rules for Private server. As you can see here, add inbound communication from public server. In this scenario ,we need ssh connection with IP of public server and any connection with security group of public server.

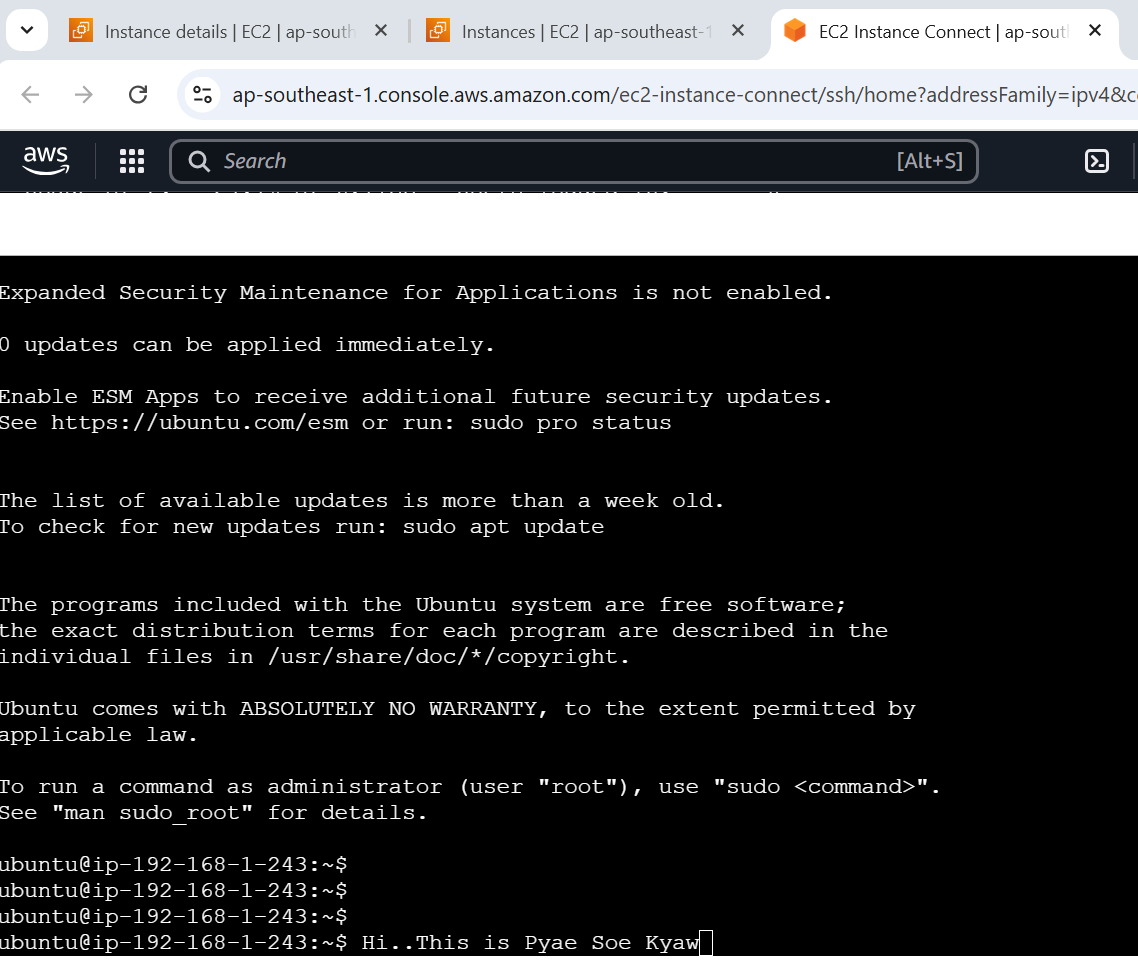


If you created the security group after the server, you need to change security group. Navigate to private server, take actions and changes security groups.





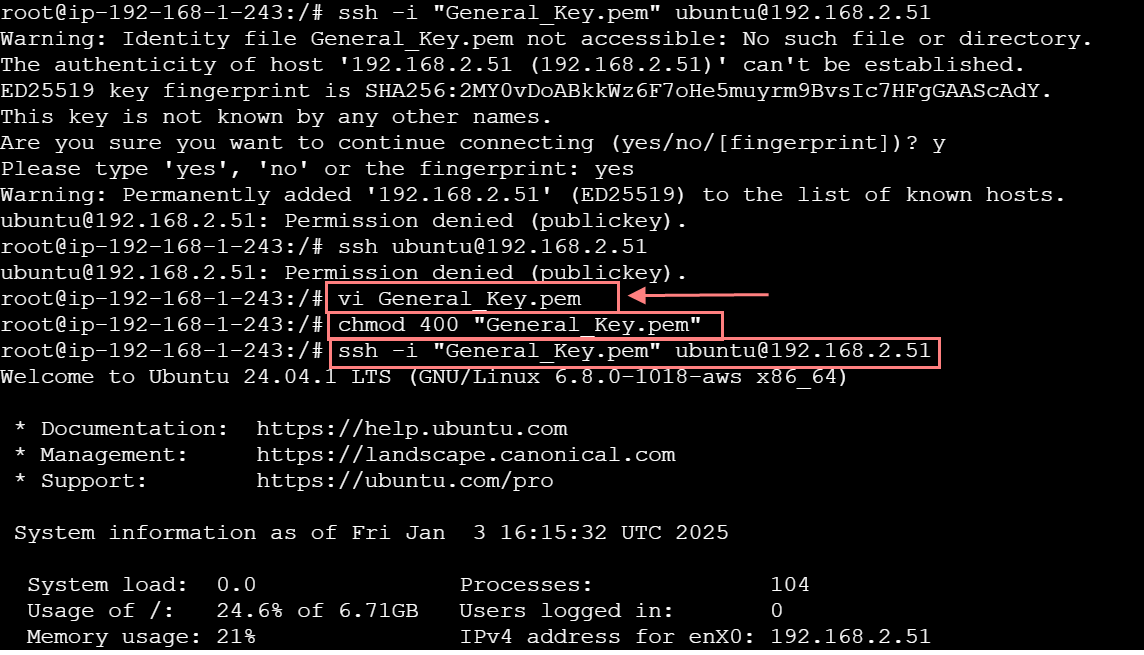
After saving the changes in private server. Establish the EC2 Instance Connect of public server

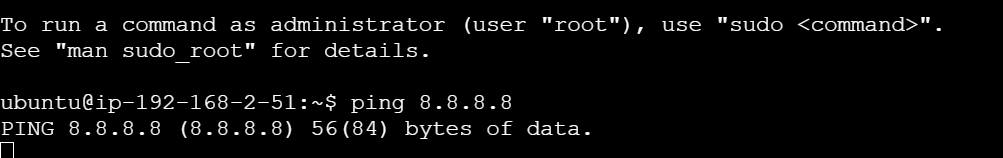


Even though we make things perfect, SSH connection need public key to access private Server. In the Terminal, create public key file(.pem). After typing “vi General\_Key.pem”, we must copy public key and paste it inside of the “General\_Key.pem”.

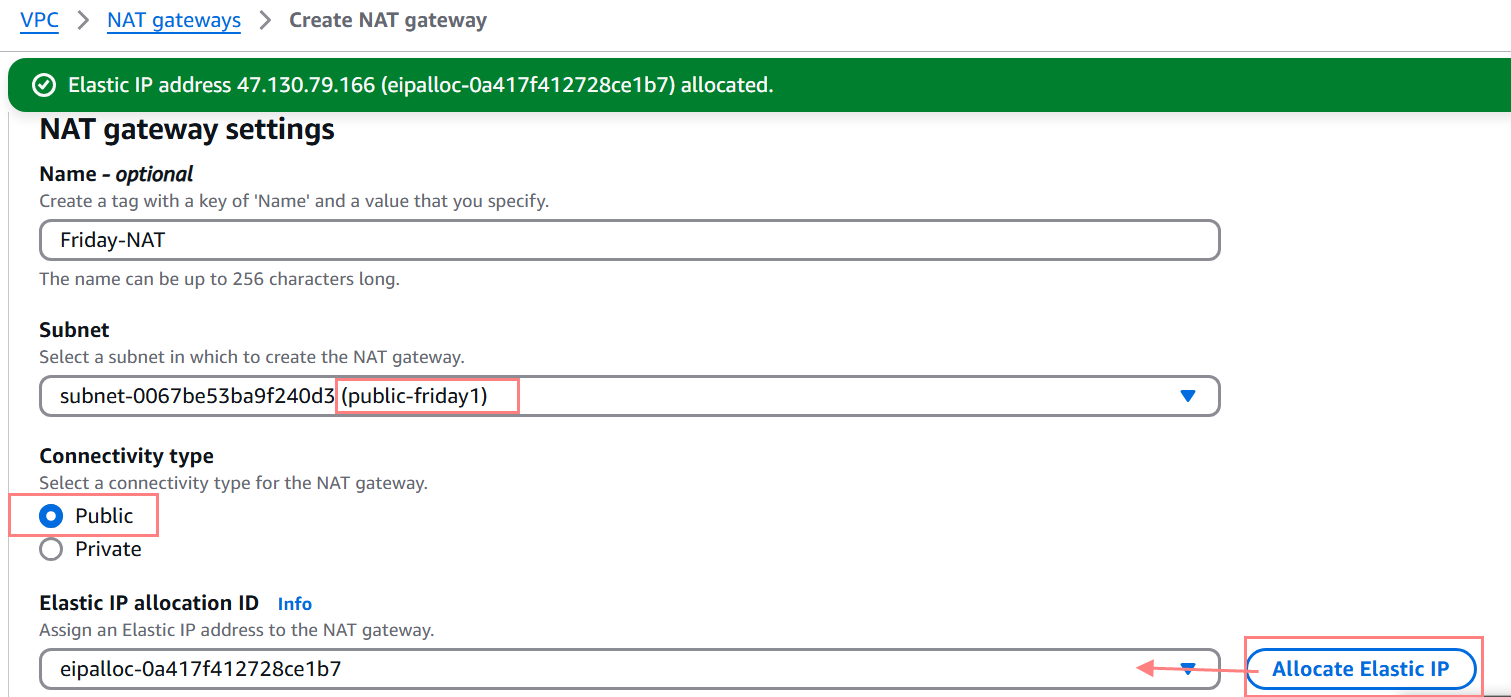


After edit the public key, change permission of the key file because we are using as a Ubuntu user which is not allowed to use this kind of feature. Finally, establish ssh command line and access the sensitive server within private subnet.

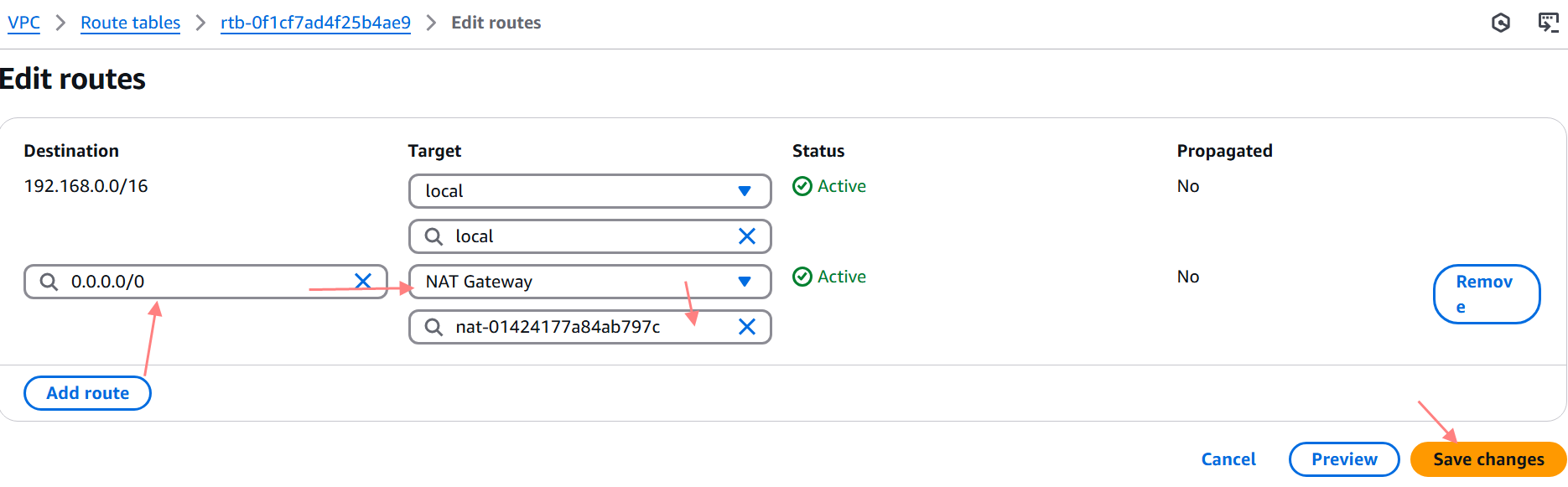


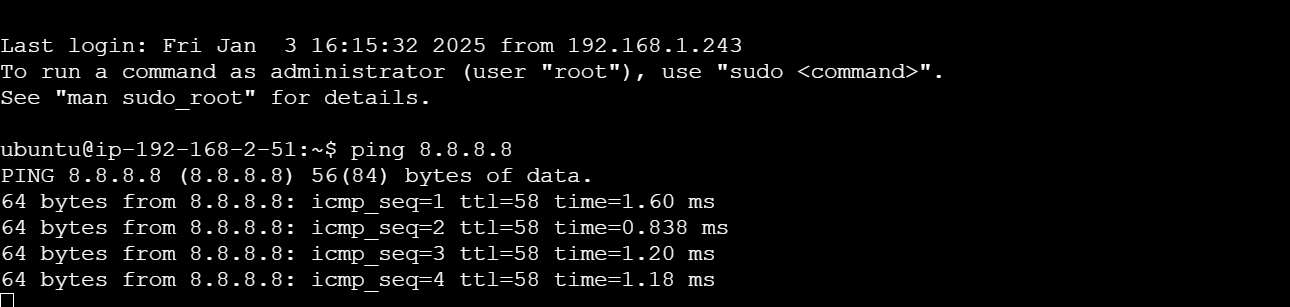


Even you can access the private server, it is not able to reach the Internet. This is where we need NAT gateway which allows the server to access the Internet by one side. Create the NAT gateway which need to place inside of public subnet because the NAT gateway will pass the Internet Gateway and exit the VPC. We use NAT gateway as a guard for private server. For many private , one NAT gateway is enough because it use Elastic IP for forwarding the traffic.



We need to add a route to NAT Gateway from private subnet. Make sure the destination would be any IP address.

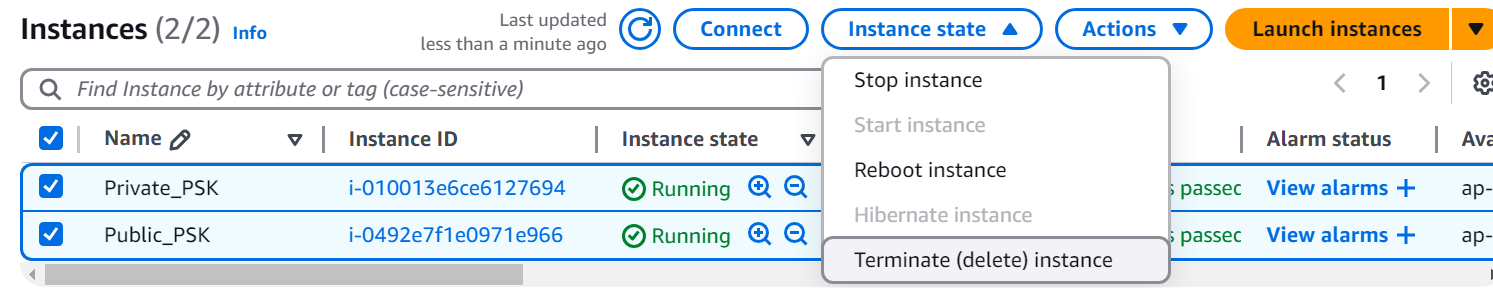




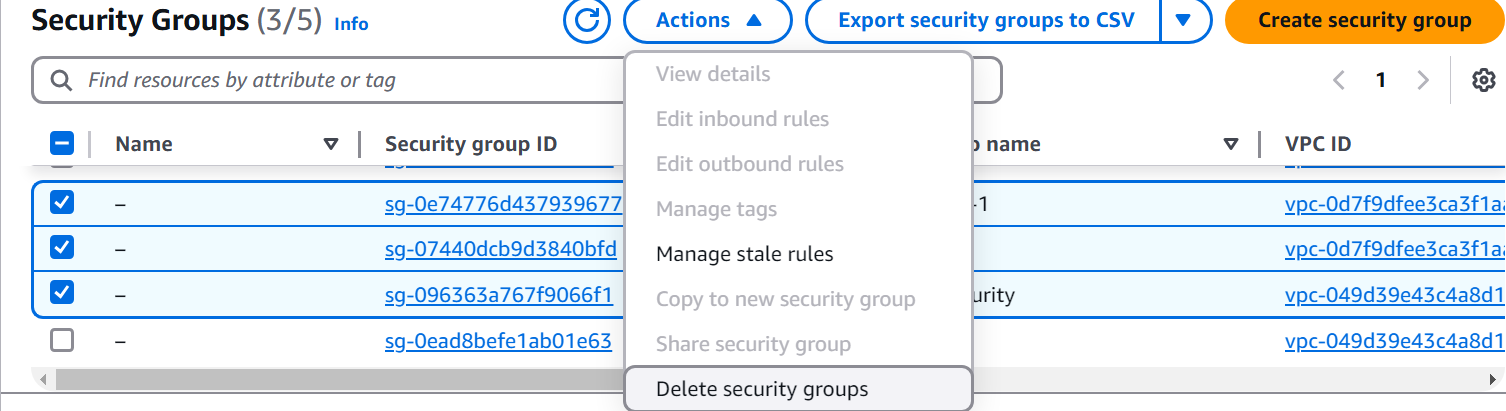
Congratulation . Well Done !

This is the final step.

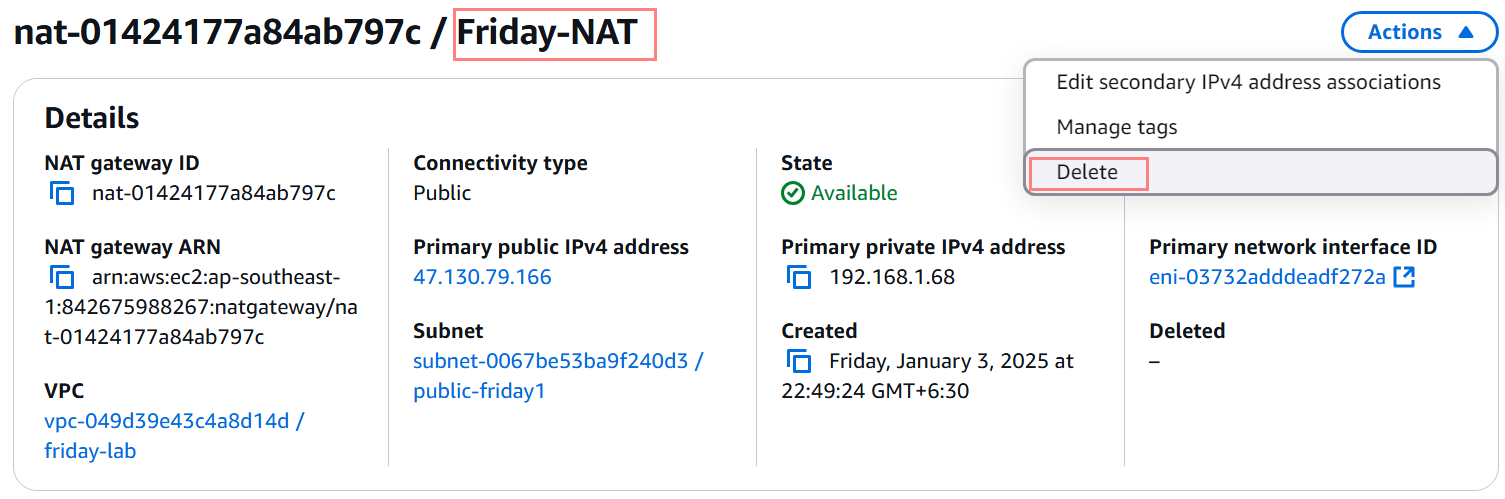
Step (1) : Terminate the Instances in their associate region.



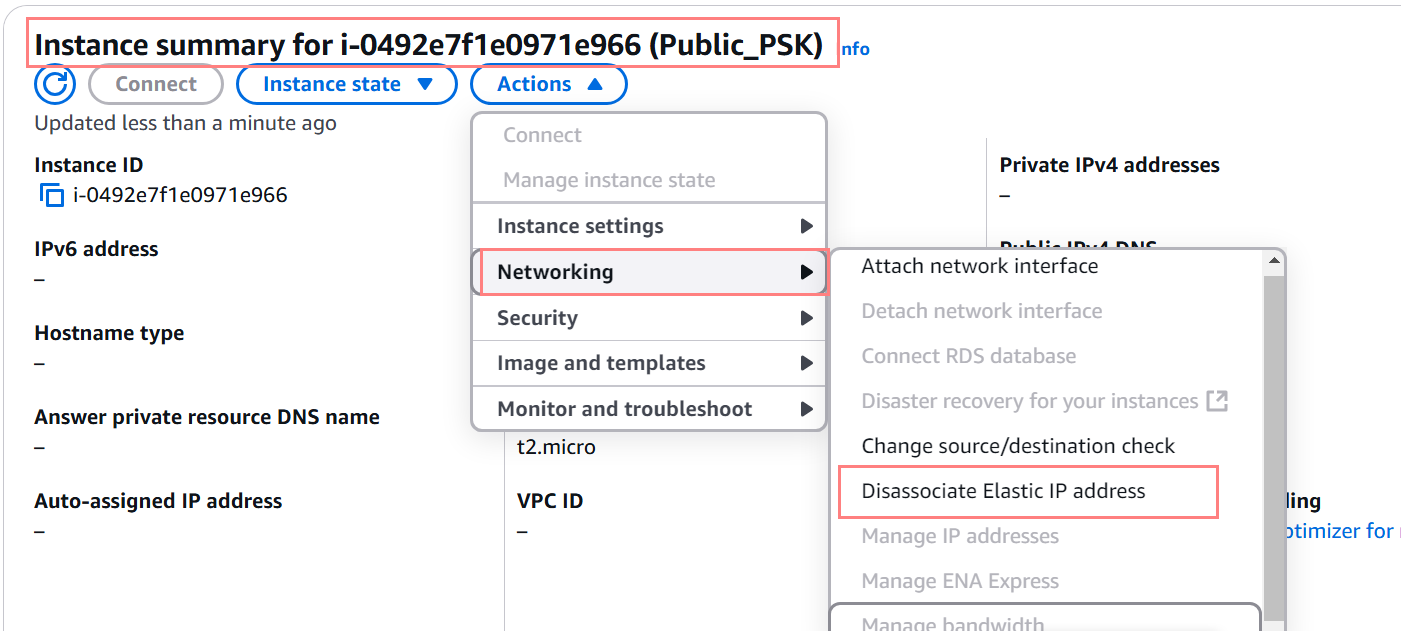
Step (2): Delete Security groups in EC2.



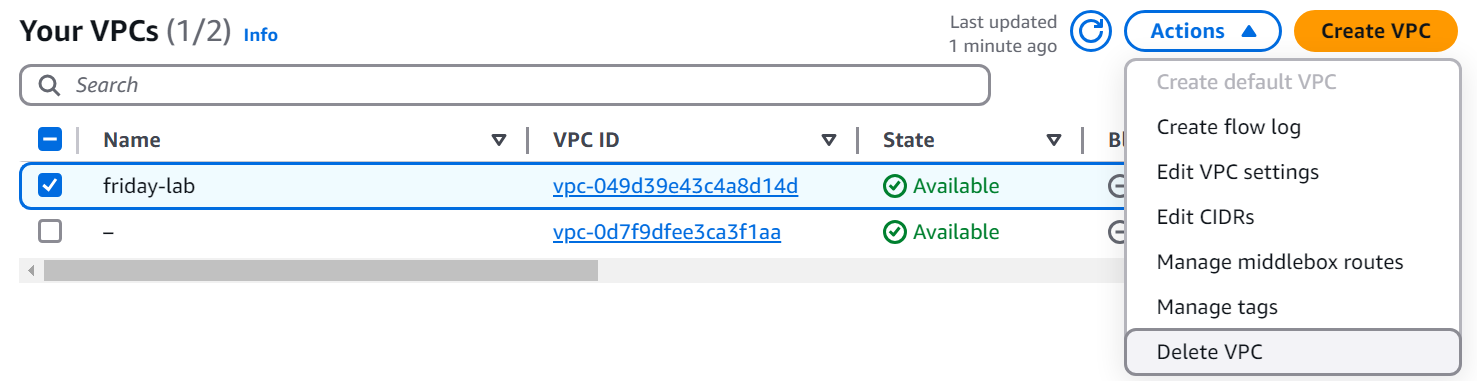
Step (3): Delete NAT gateway.



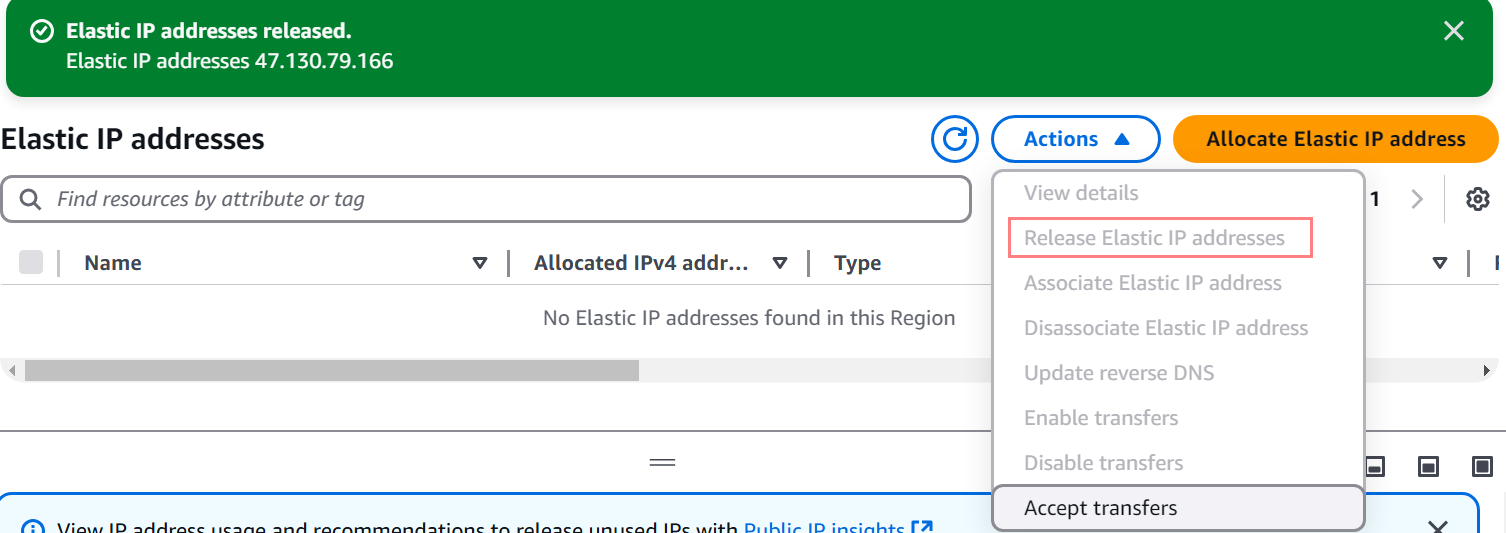
Step (4): Diassociate Elastic IP address in NAT Gateway.



Step (5): Delete custom VPC you have created.



Step (6): If you cannot delete Elastic Ip address, take Actions and Release Elastic IP addresses.



Don’t forget to check the Billing later !