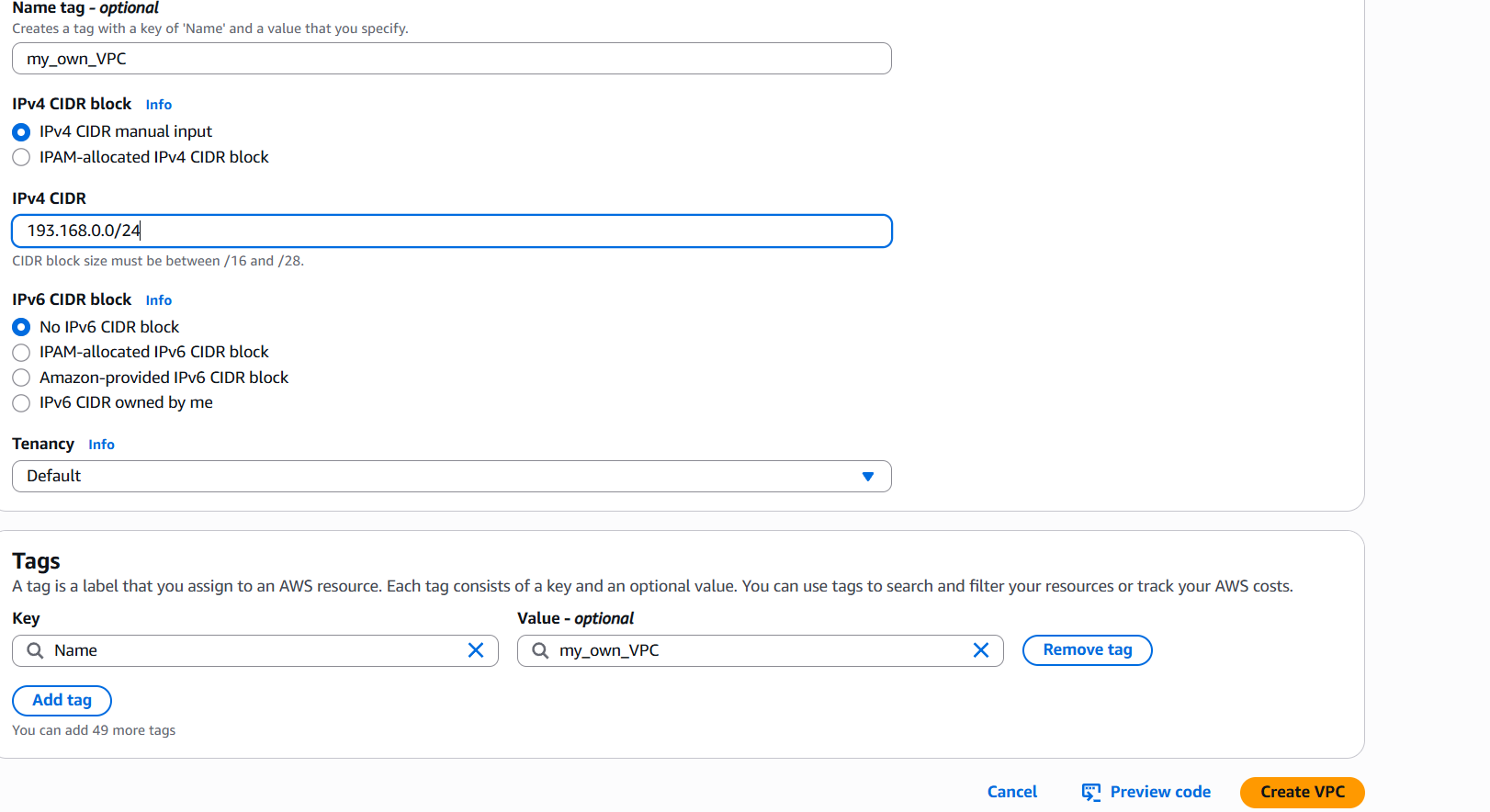
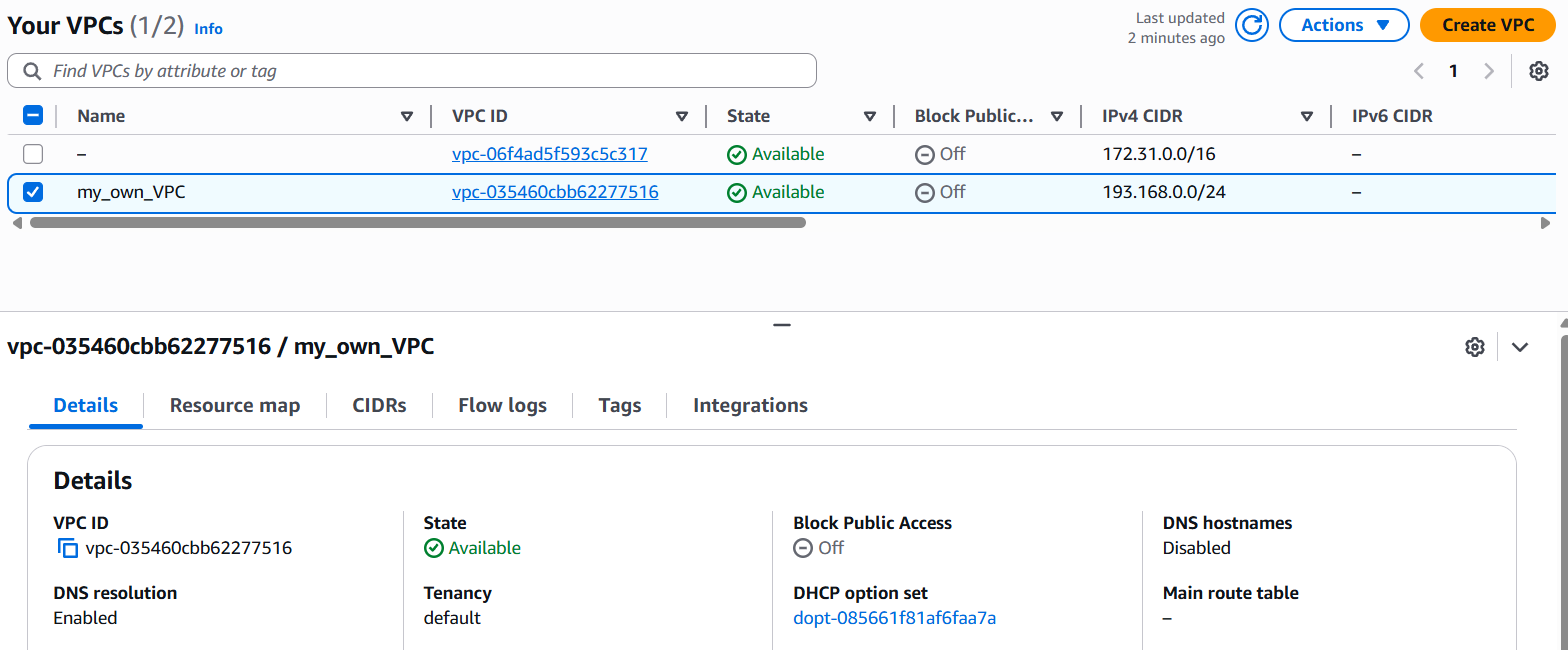
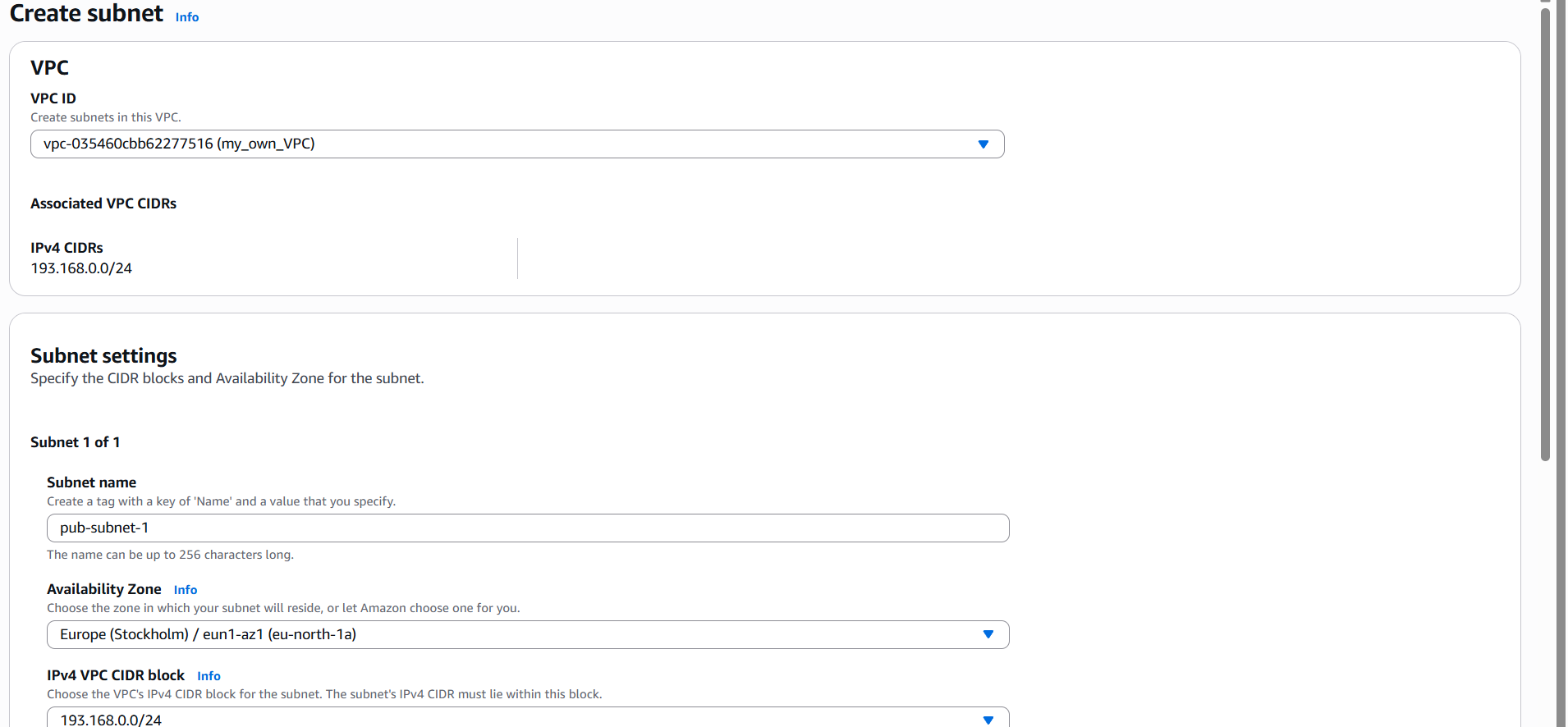
1. Create VPC with 2 private and 2 public subnets.

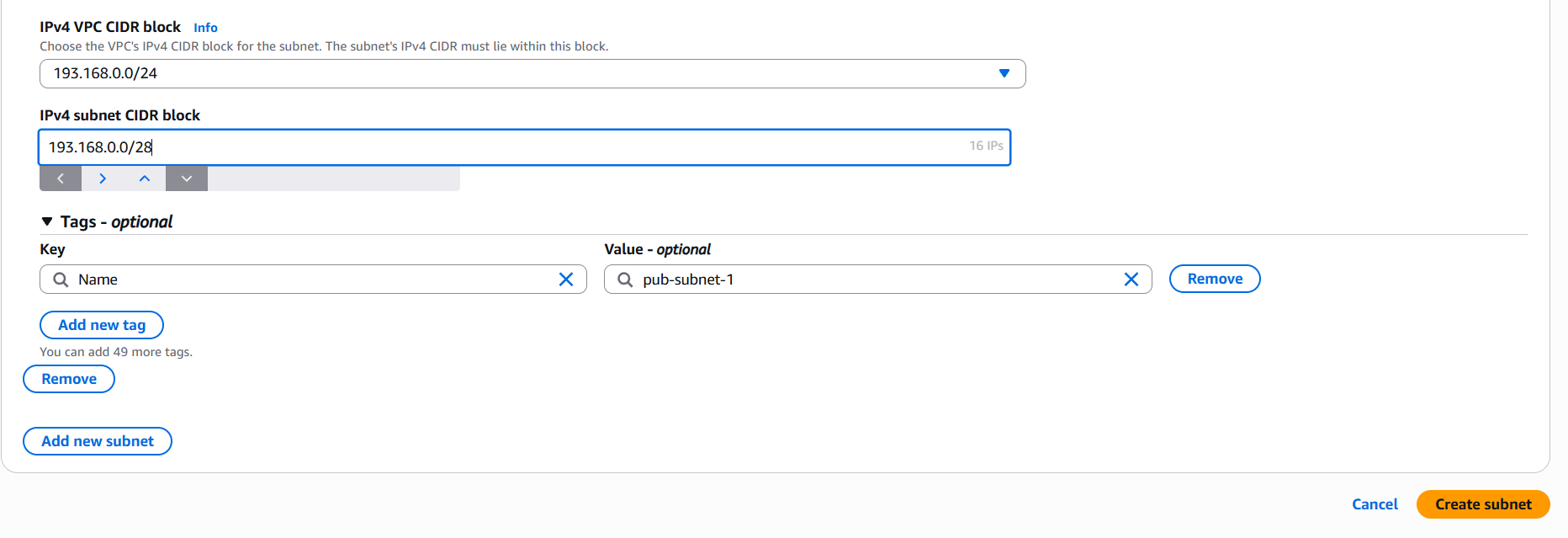
Step 1: Login to your AWS console, select VPC 🡪 Create VPC 🡪 Enter details 🡪 Create VPC.

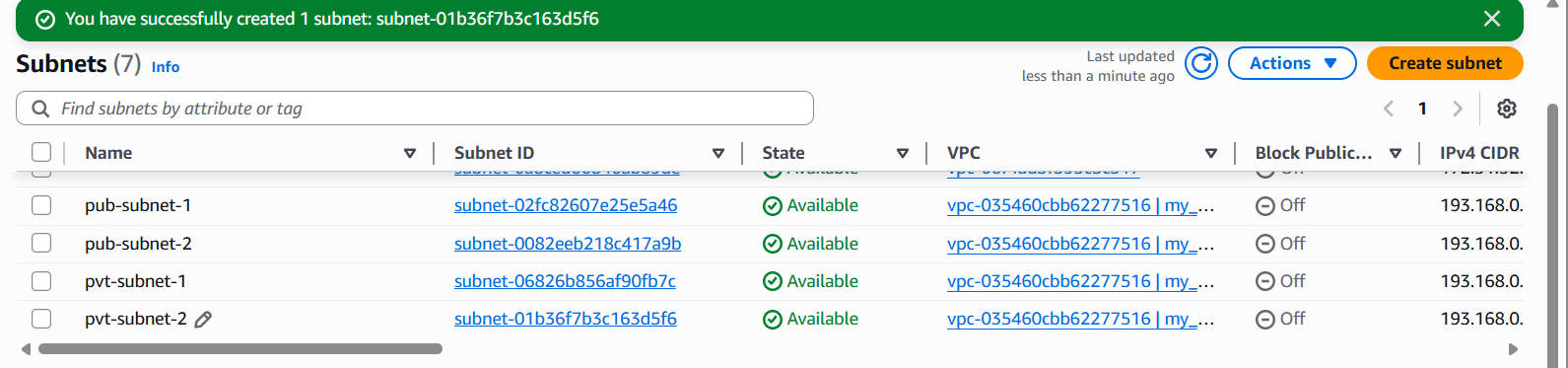




Step 2: Select subnet 🡪 create subnet 🡪Enter details, repeat the same to create 2 public and 2 private subnet.

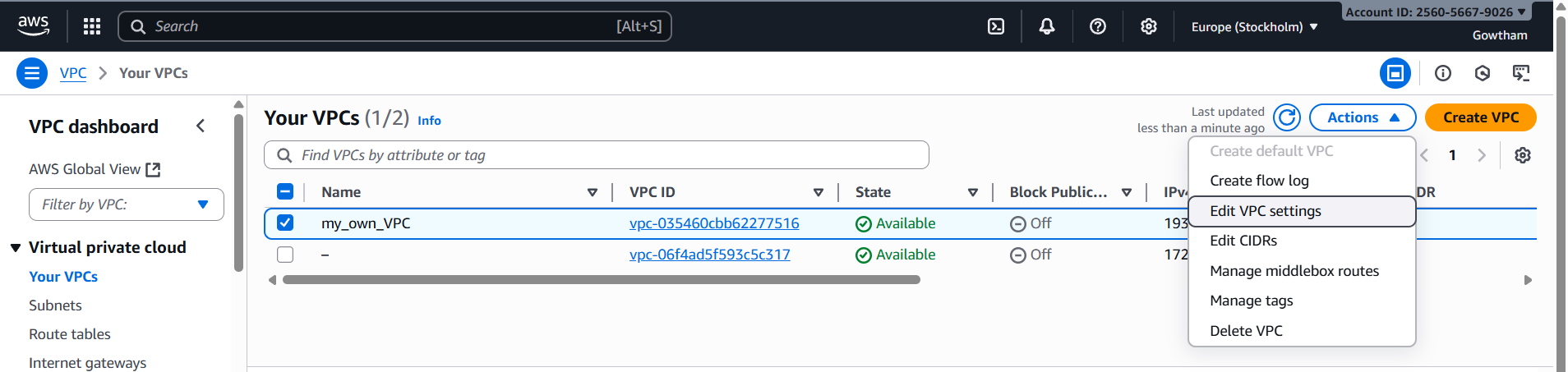




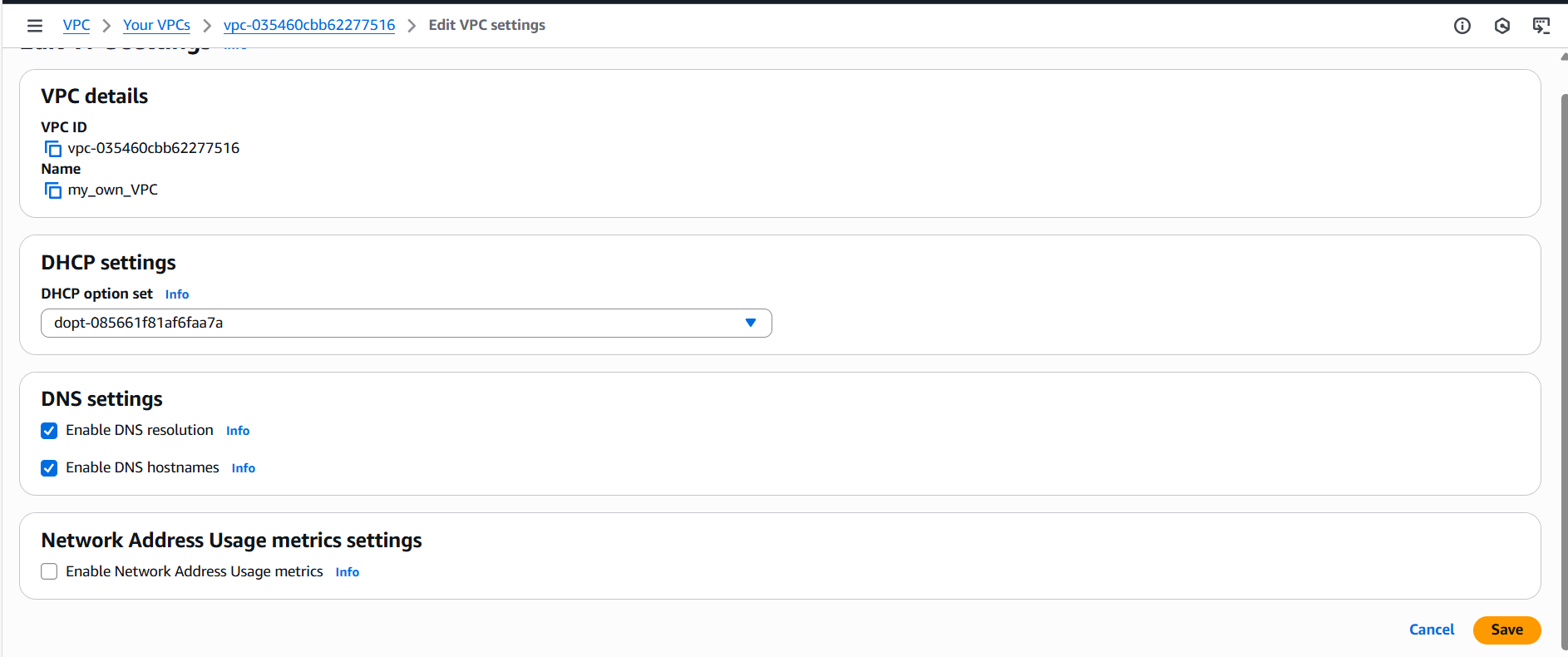


1. Enable DNS Hostname in VPC.

Step 1: Login to Aws console select VPC 🡪 your VPC 🡪 Actions 🡪 Edit VPC settings.

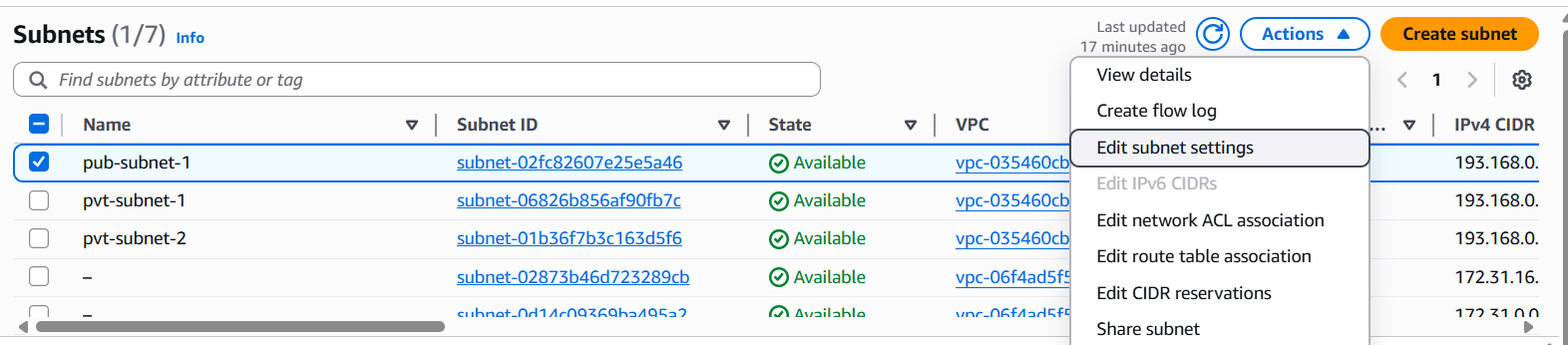


Step 2: Select Enable DNS hostnames under DNS settings and save.

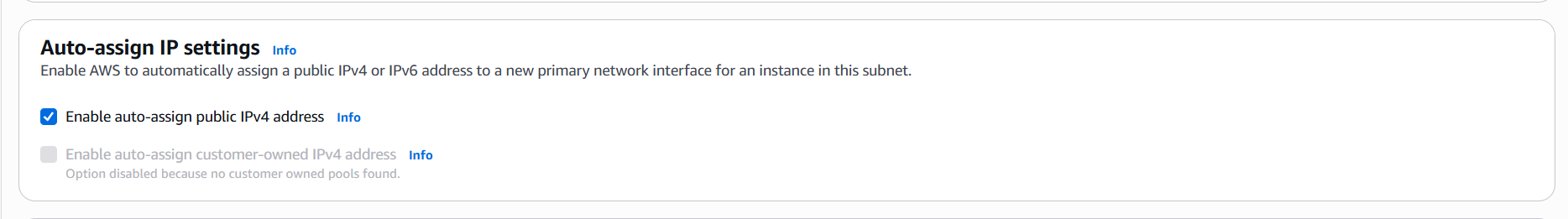


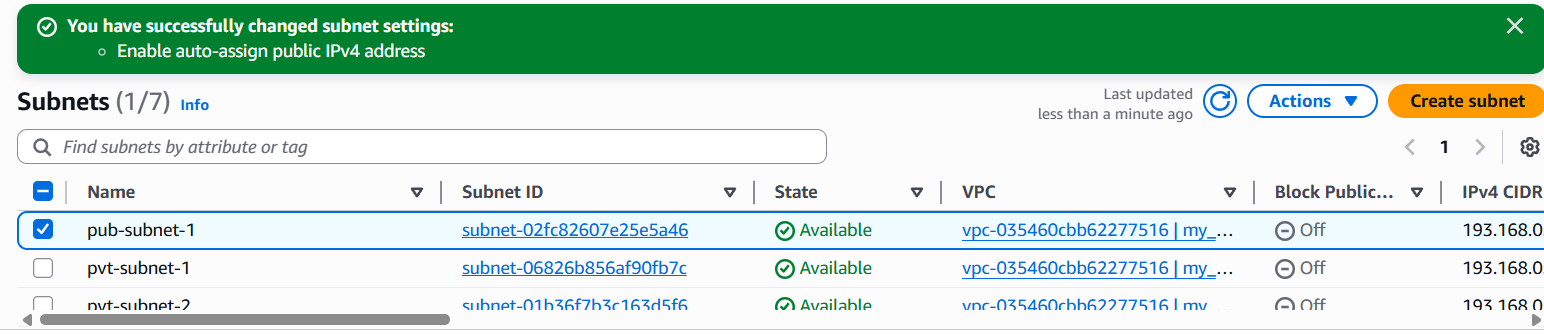
1. Enable Auto Assign Public IP in 2 public subnets.

Step 1: Login to AWS console select VPC 🡪 subnets 🡪 Actions 🡪 Edit subnet settings



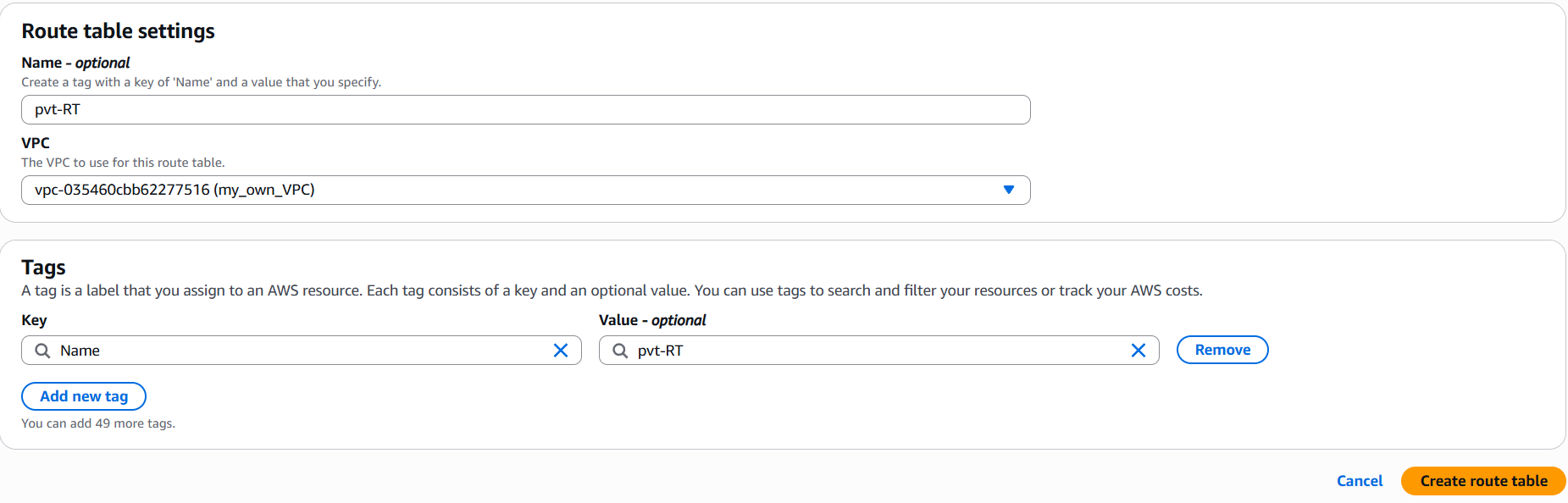
Step 2: Select Enable auto-assign and save. Repeat the same for 2nd subnet.

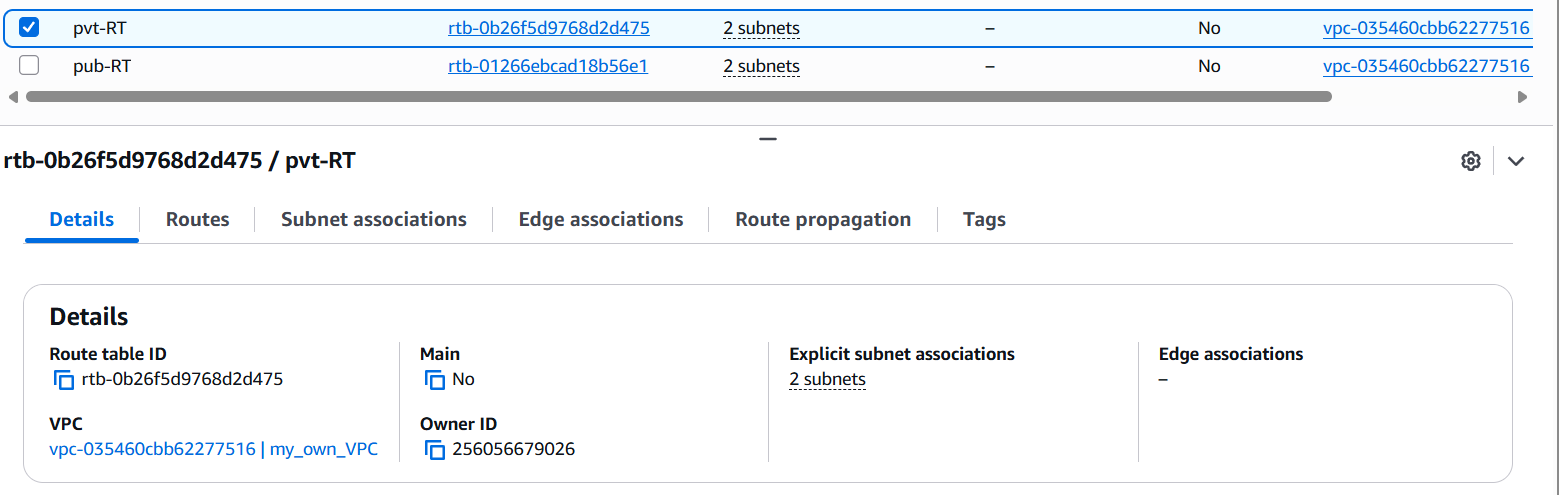




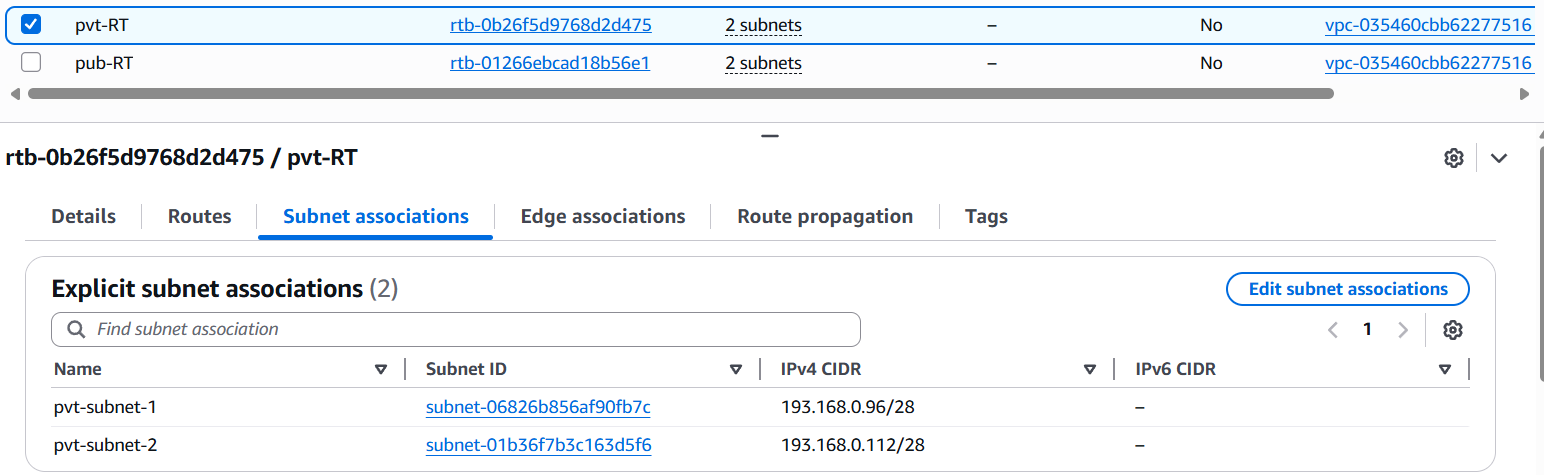
1. Add 2 private subnets in private route table.

Step 1: Login to AWS select VPC🡪Route table🡪Create RT🡪Enter details🡪Create Route table.



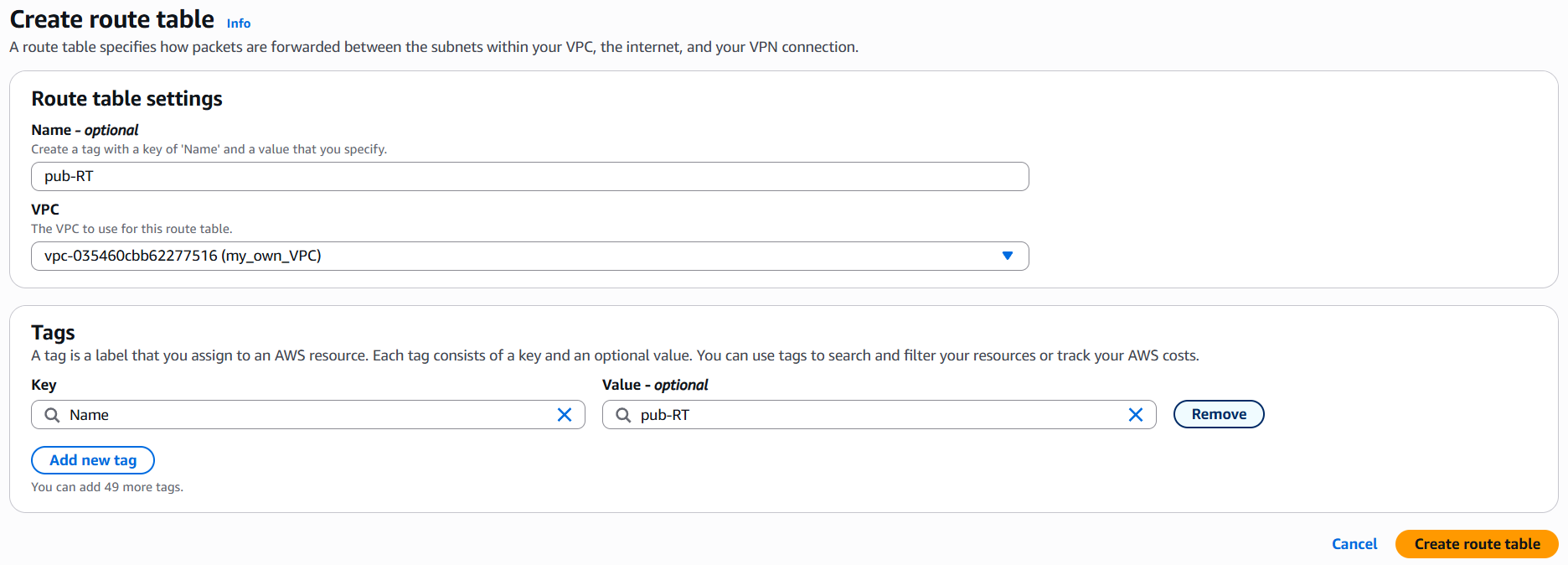


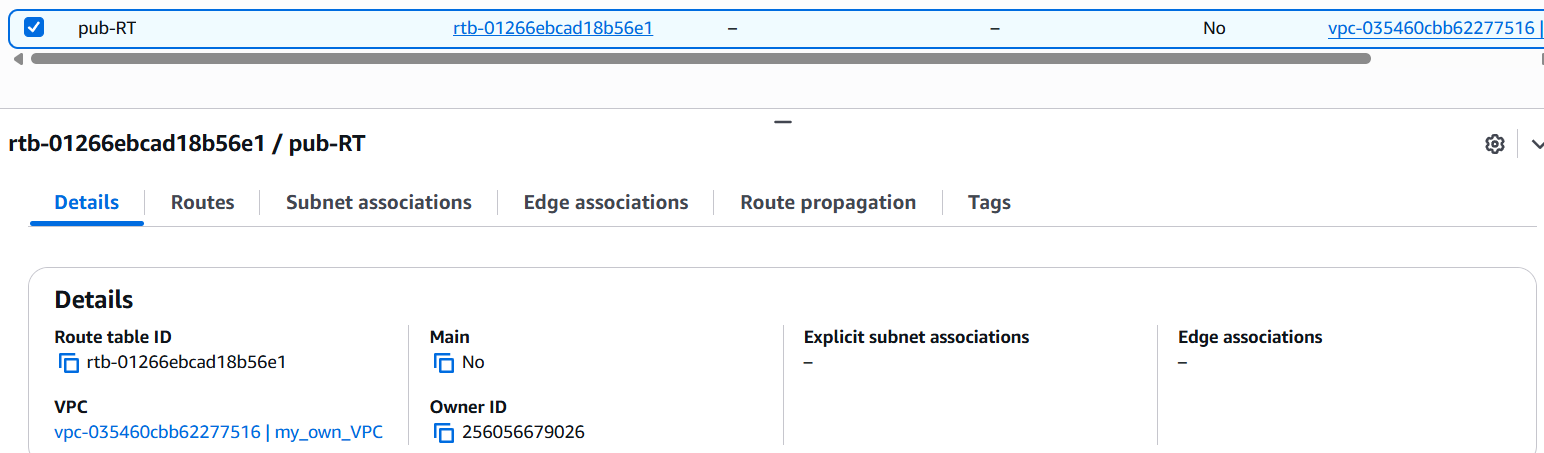
Step 2: select Subnet association 🡪 Edit SA 🡪 select private subnet 🡪 Save.



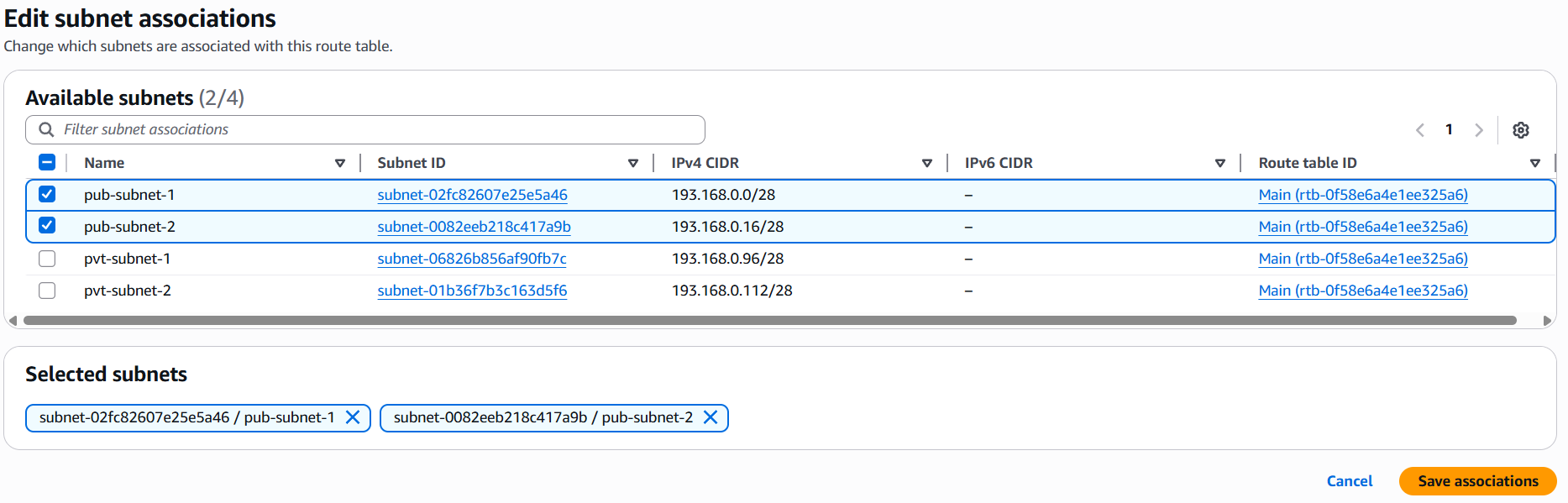
1. Add 2 public subnets in public route table.

Step 1:Login to AWS console select VPC🡪Route table🡪Create RT🡪Enter details🡪Create Route table.



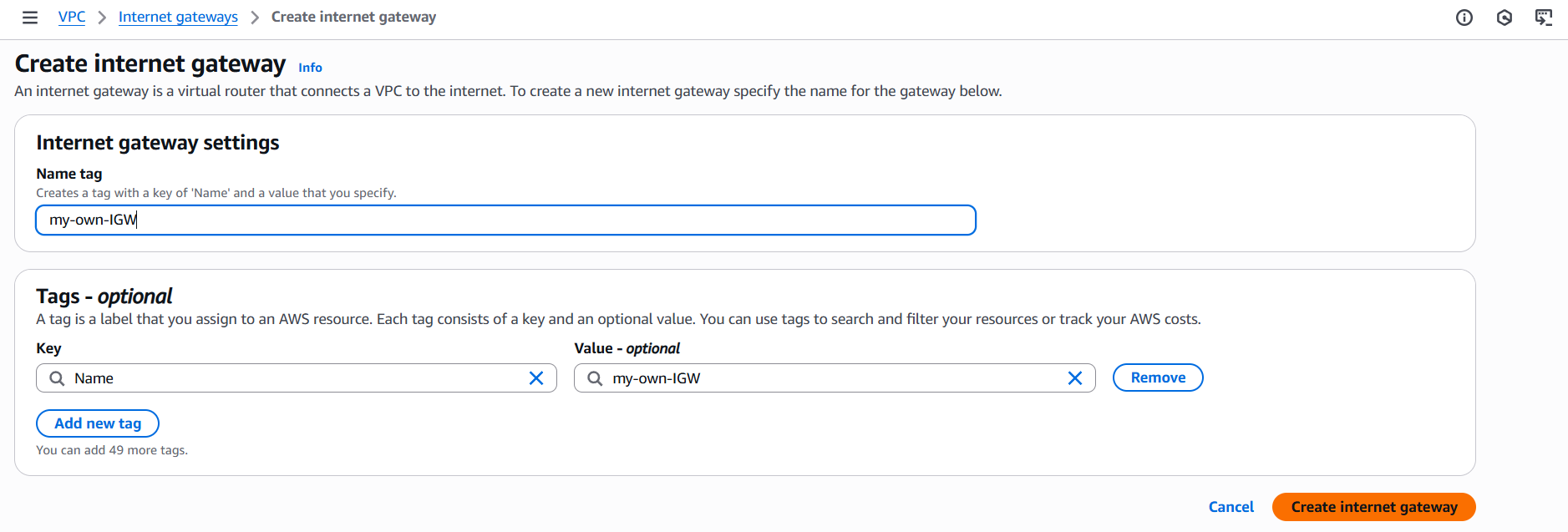


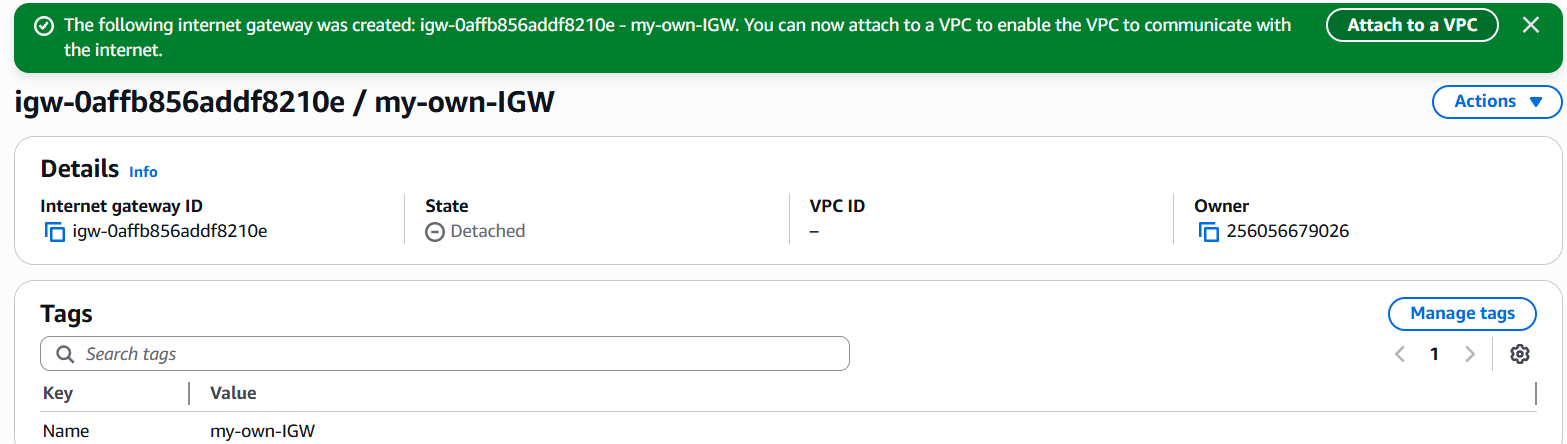
Step 2: select Subnet association 🡪 Edit SA 🡪 select public subnet 🡪 Save.



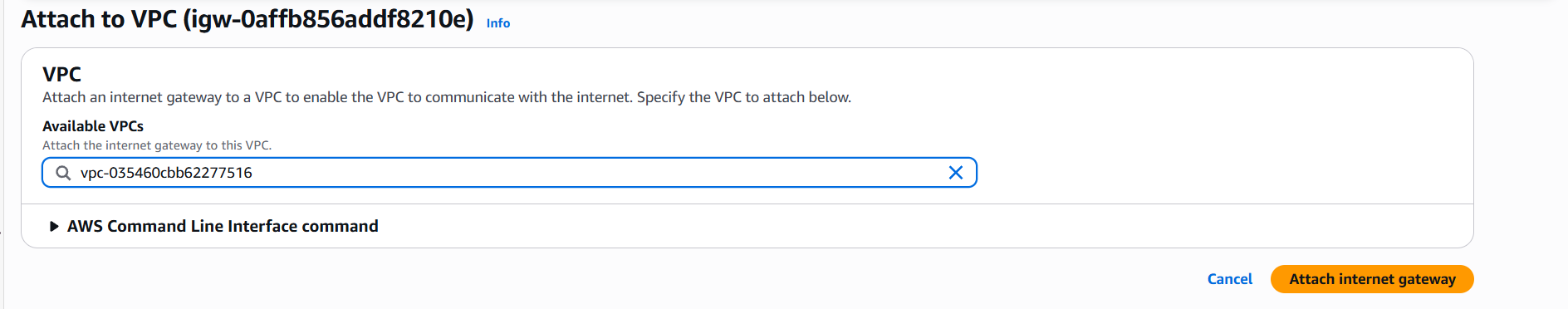
1. Public route table will have the routes to internet and local.

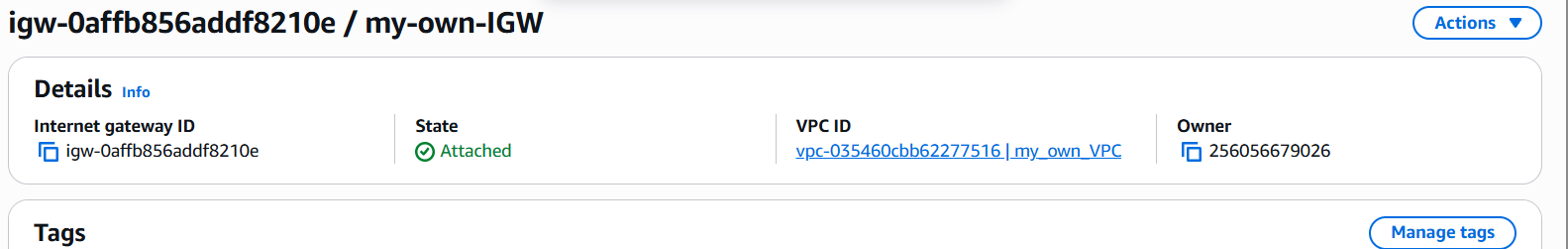
Step 1: Login to AWS console and select Internet gateways 🡪 create internet gateway🡪 enter details 🡪create IGW.



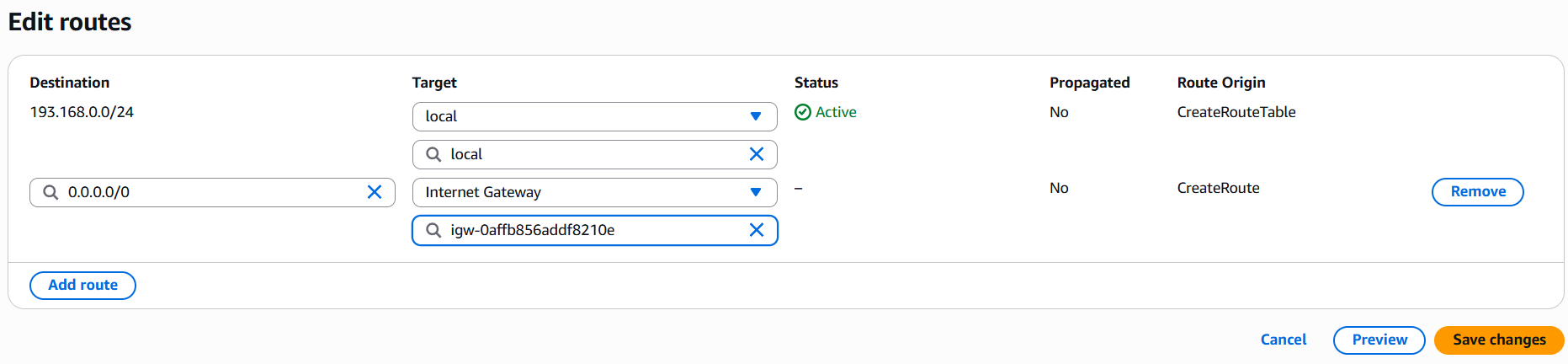


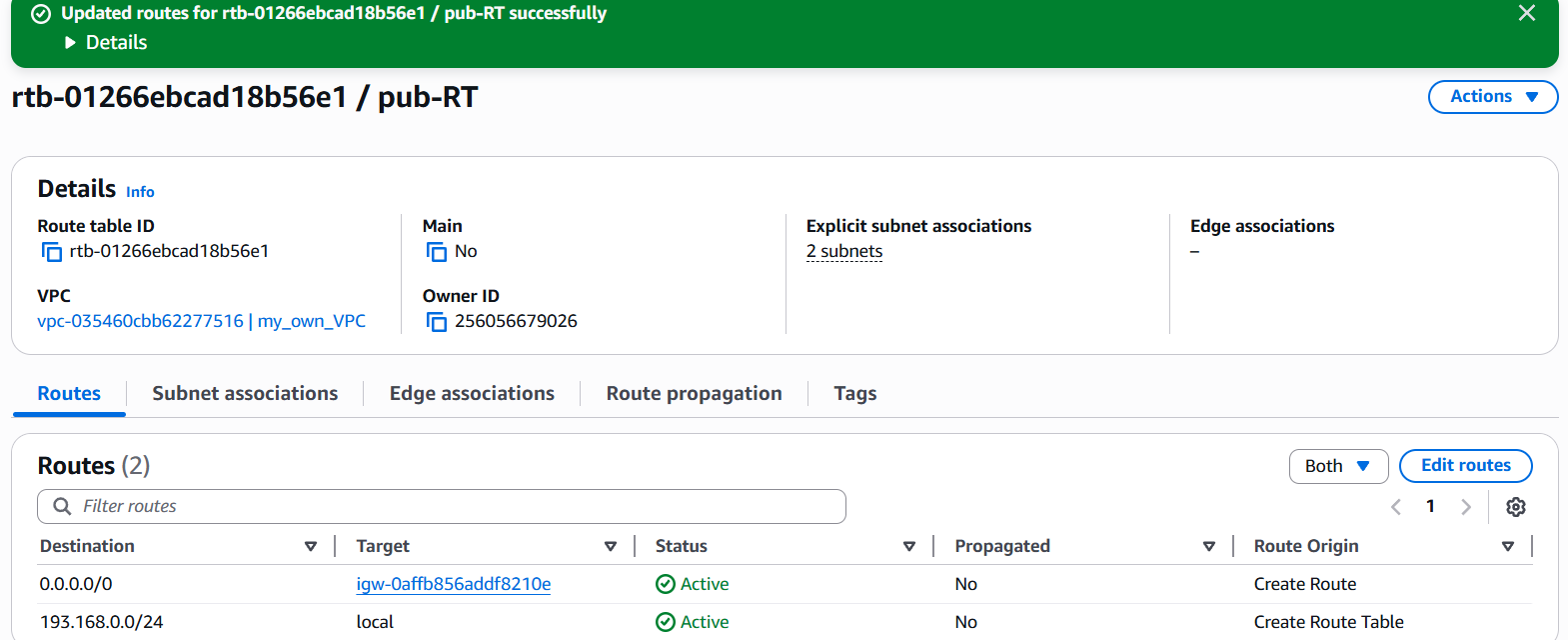
Step 2: Select Attach to VPC 🡪 Your VPC 🡪Attach IGW.





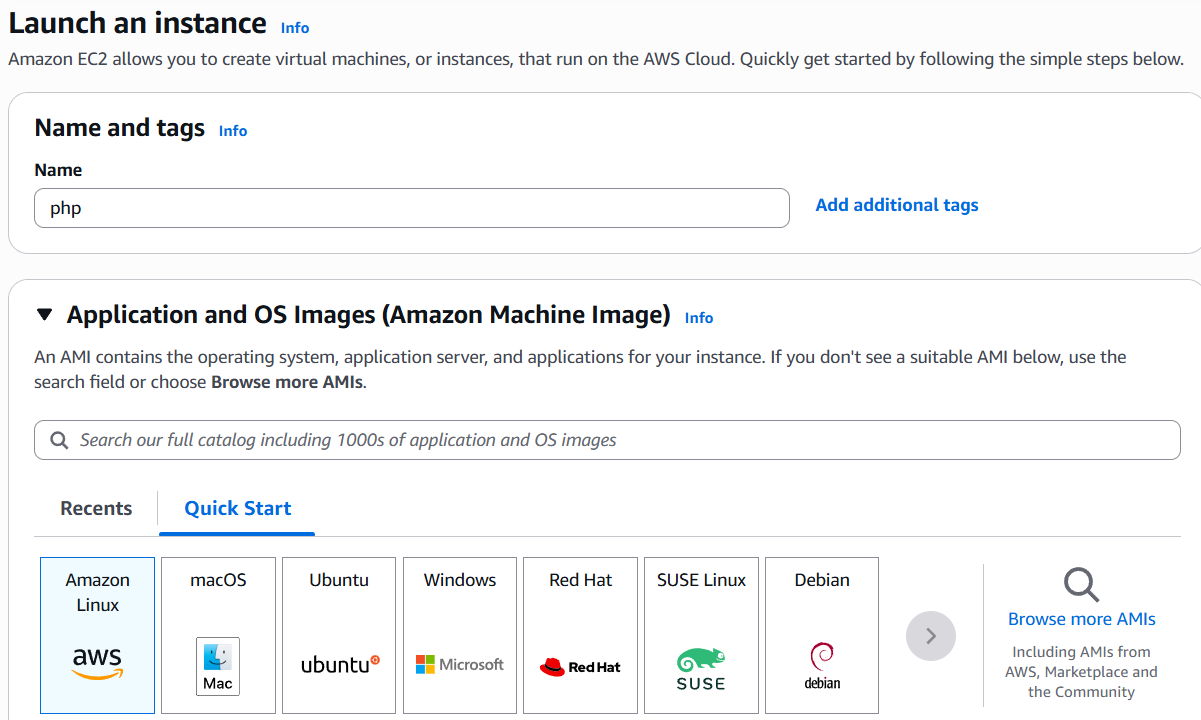
Step 3:Select Route🡪Edit route🡪Add route🡪select your IGW to grant internet access to public subnet.

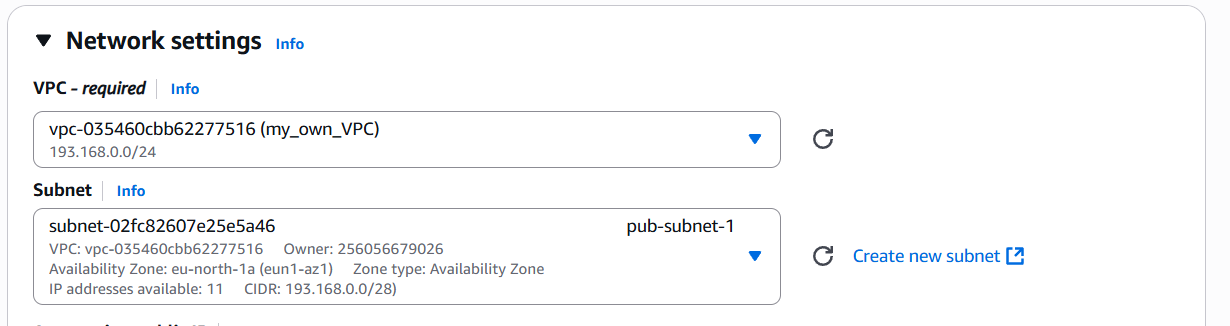


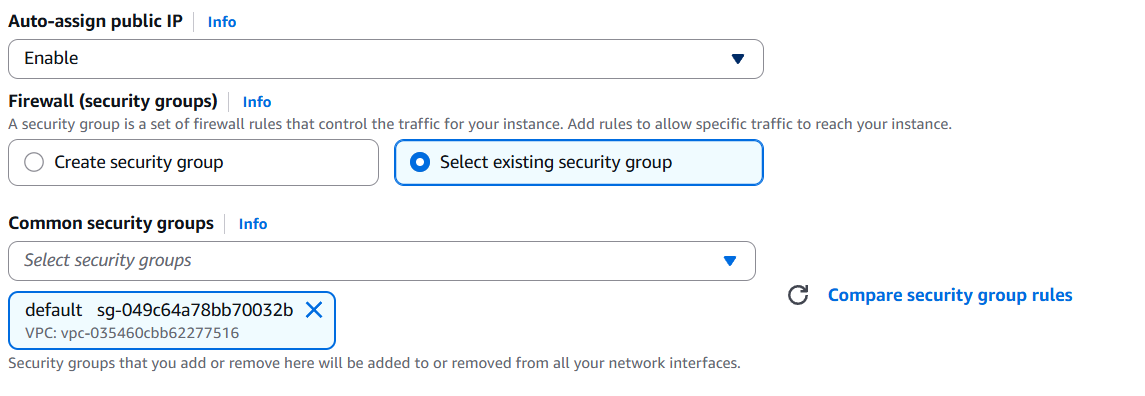


1. Create EC2 in public subnet with t2.micro and install PHP.

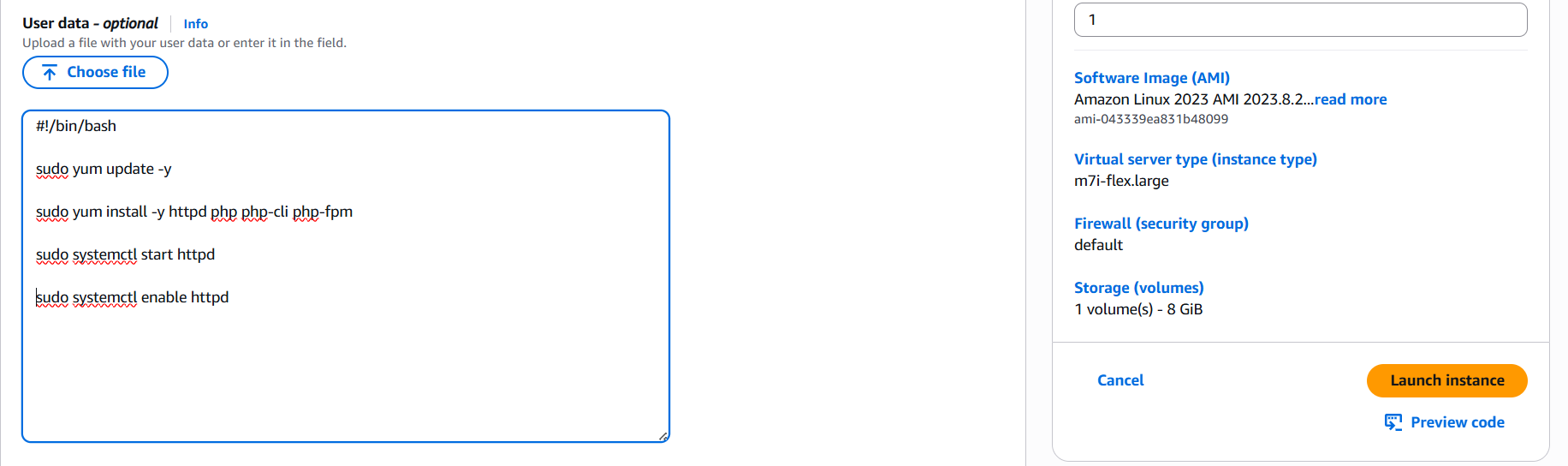
Step 1:Login to AWS console select EC2🡪Launch EC2🡪select required configurations.



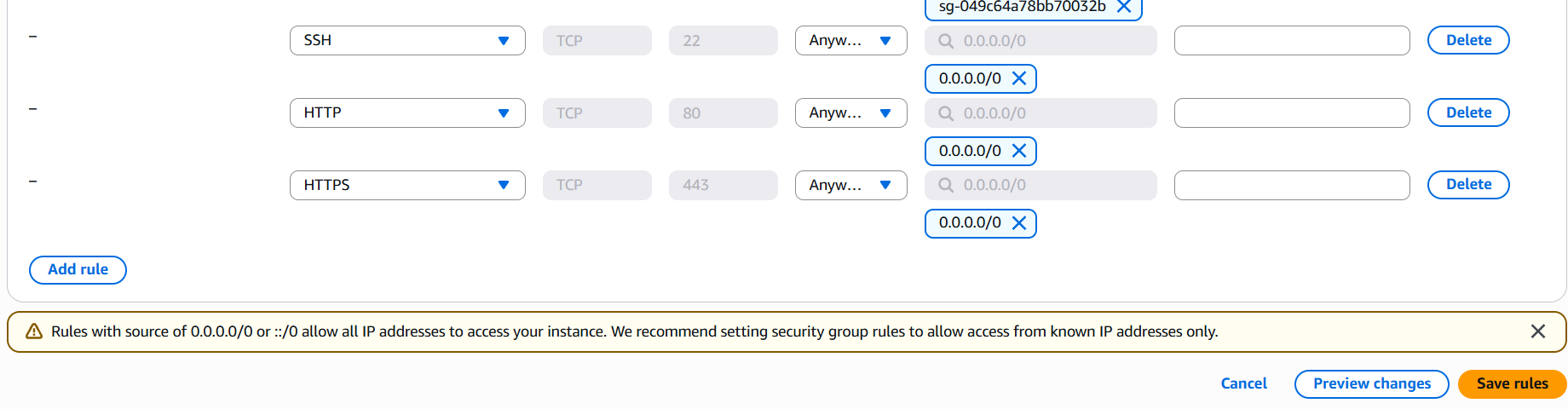




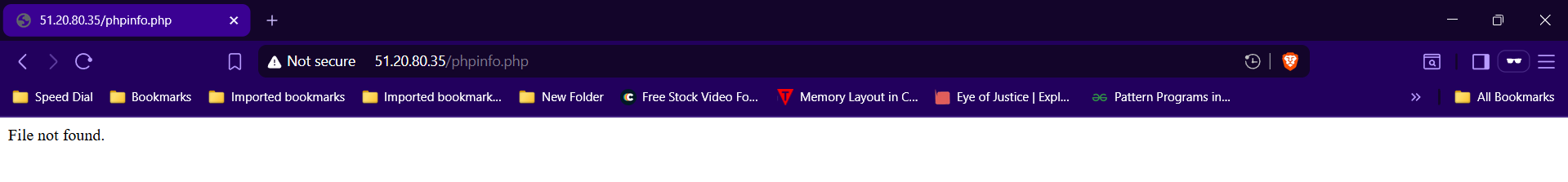
Step 2: In User data enter script to install httpd and php and click launch instance.



Step 3: Change the inbound rule to allow SSH, HTTP, HTTPS ports.

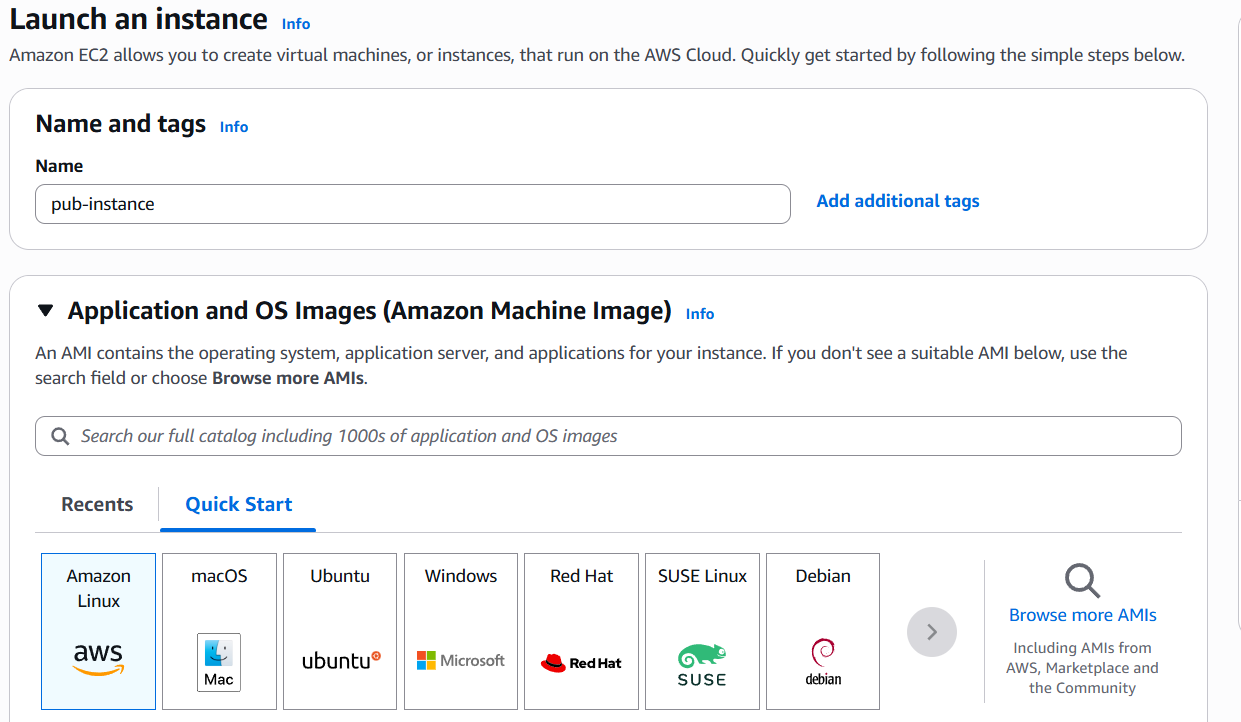


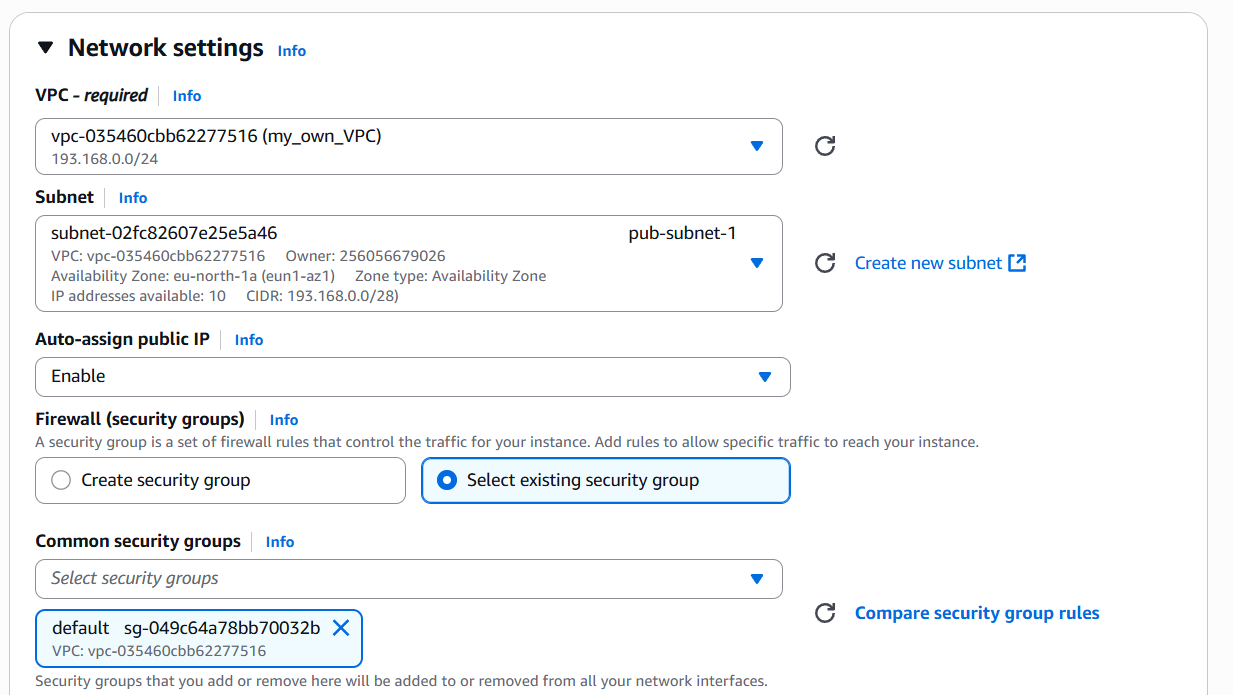
Step 4: copy the public of instance and run **public\_ip/phpinfo.php** in your browser.

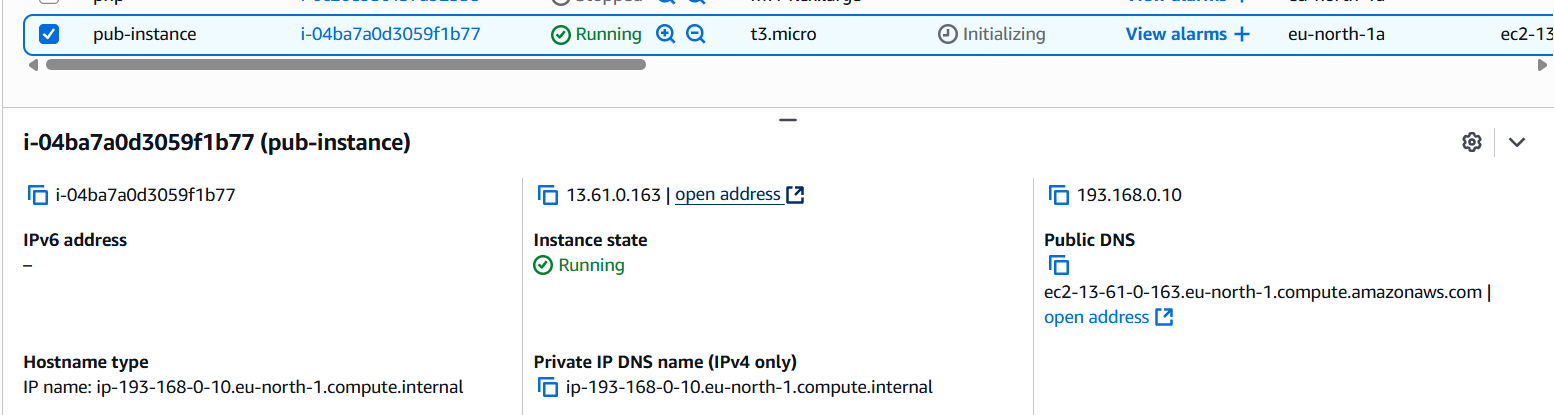


1. Configure NAT gateway in public subnet and connect to private instance.

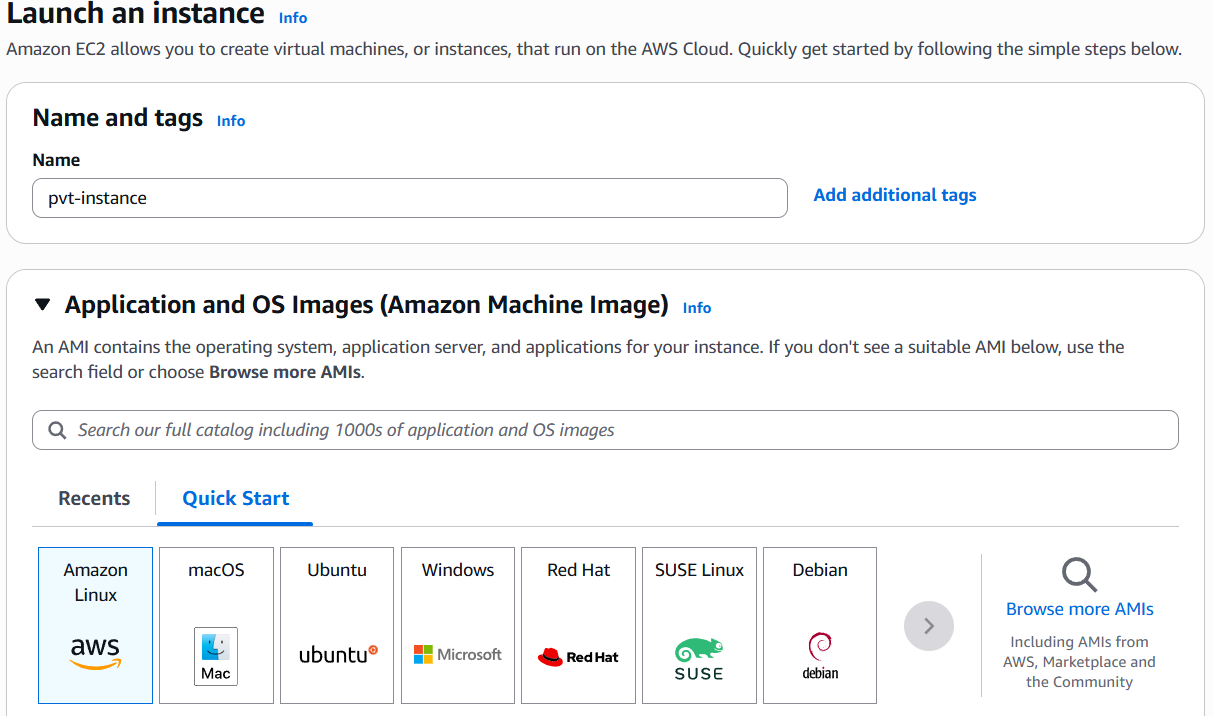
Step 1: Launch an instance in public subnet.

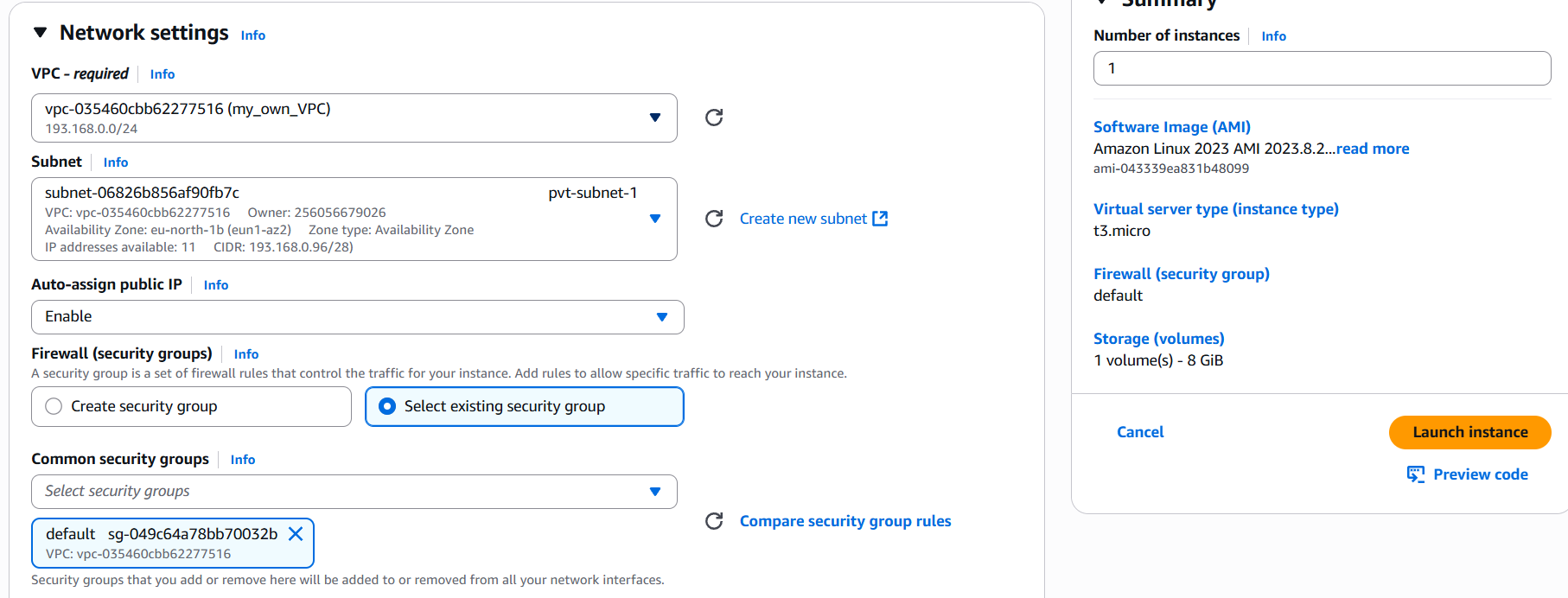


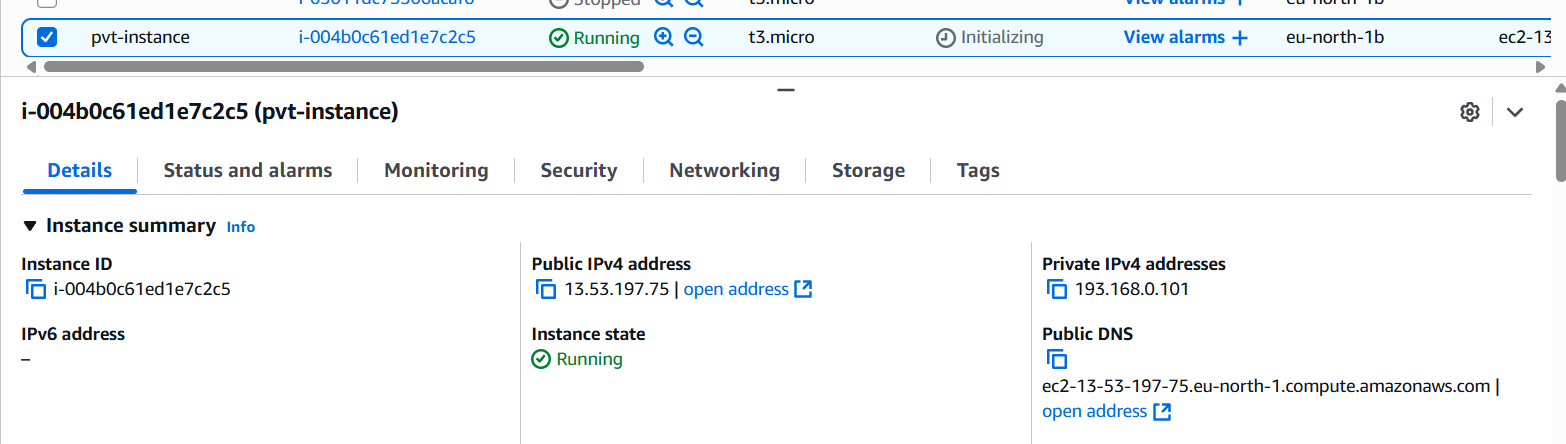




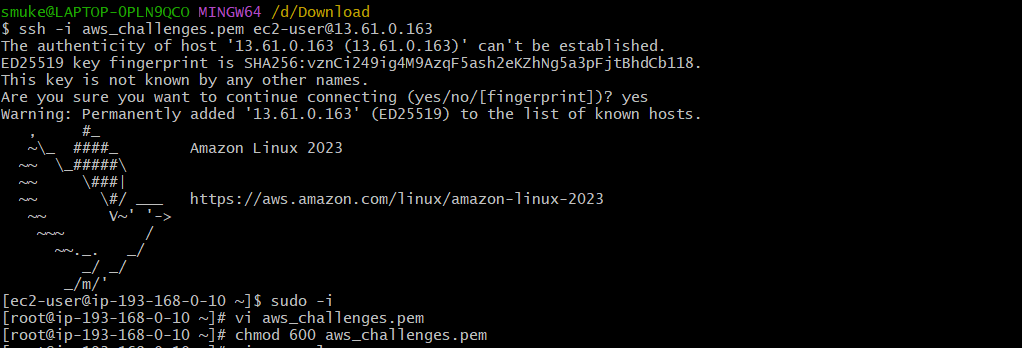
Step 2: Launch an instance in private subnet.



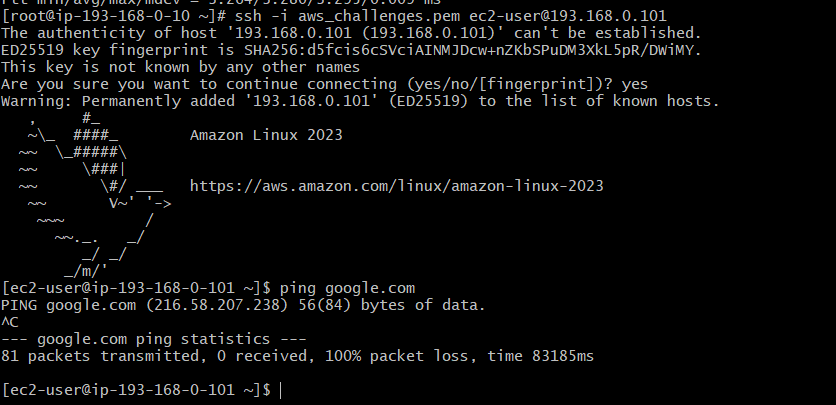




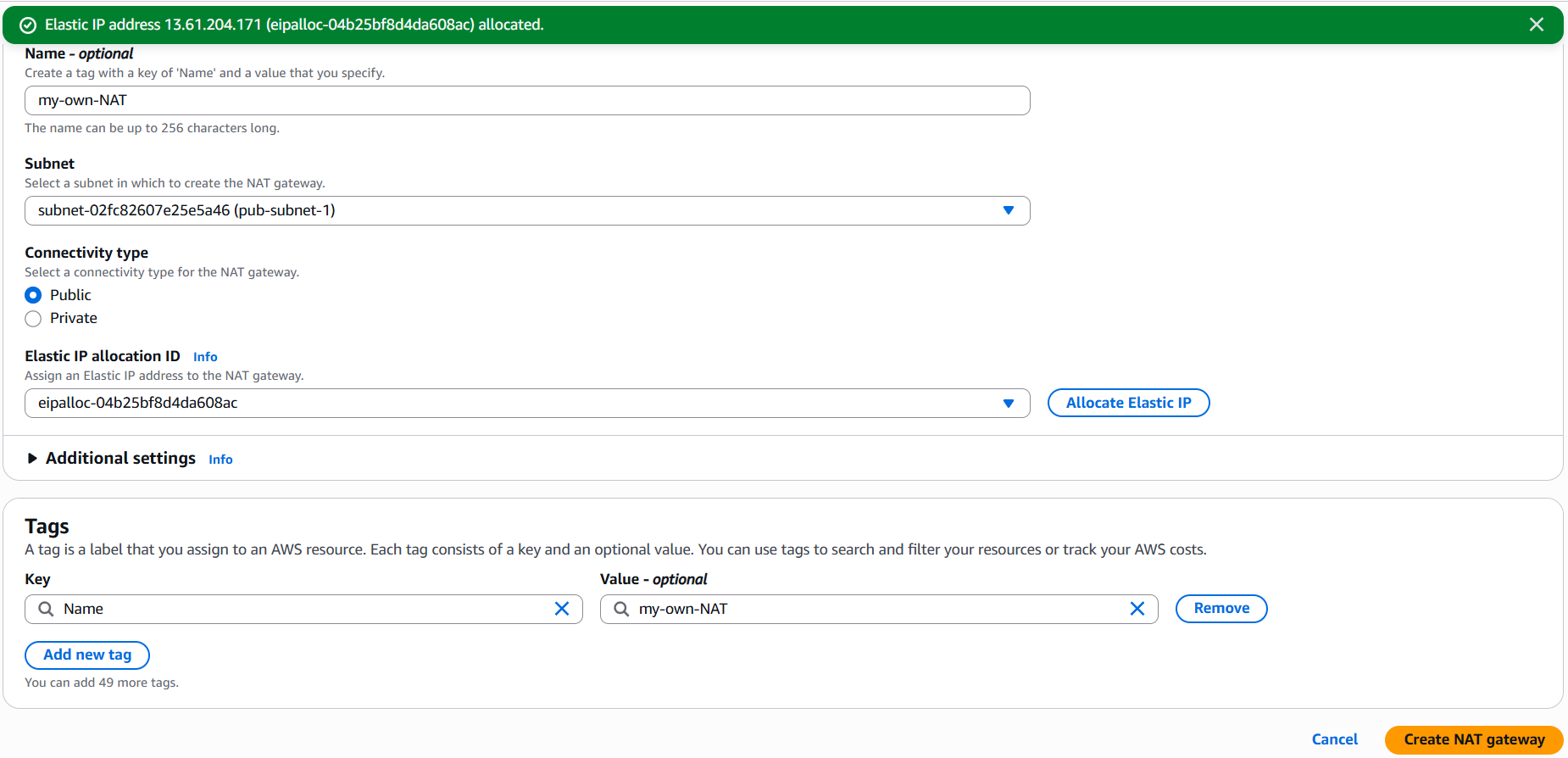
Step 3: Copy the **security\_key.pem** and paste it in new .pem file in your public instance.

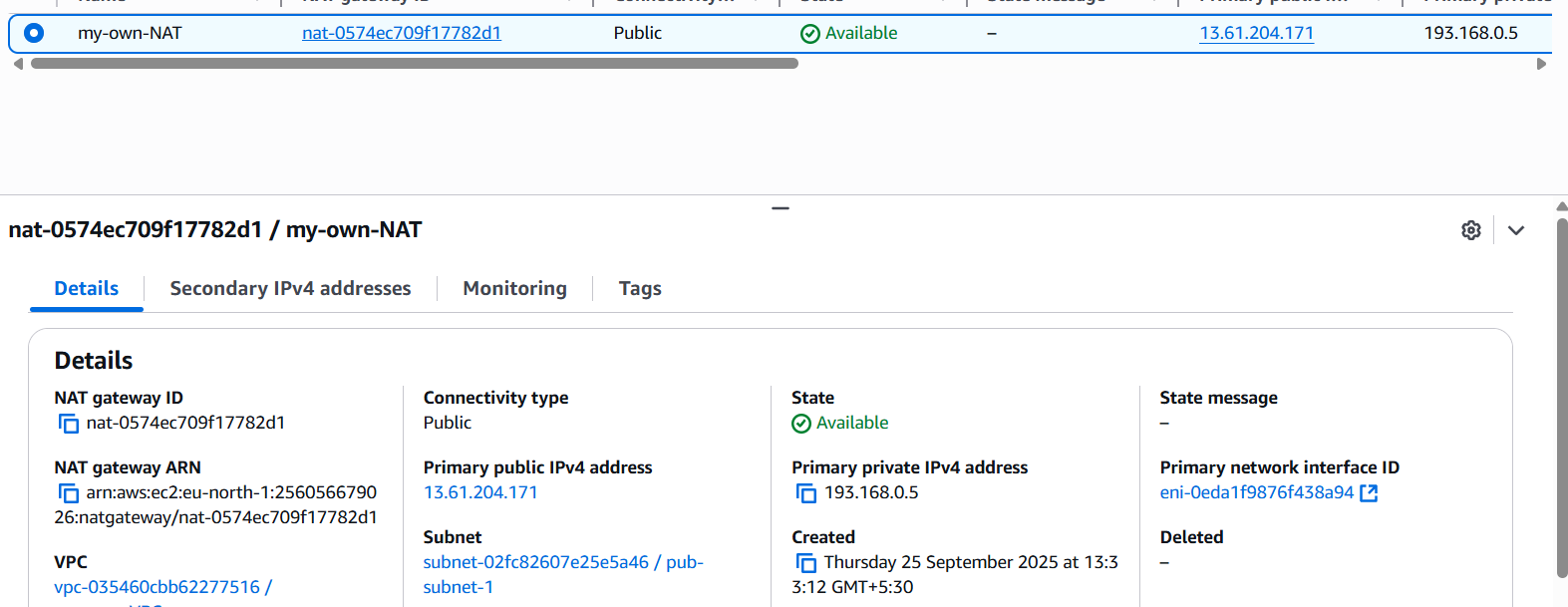


Step 4: Use **ssh -I security\_key.pem user\_name@private\_ip\_of\_pvt-instance** to connect to private instance and use **ping URL** to check for network connectivity.

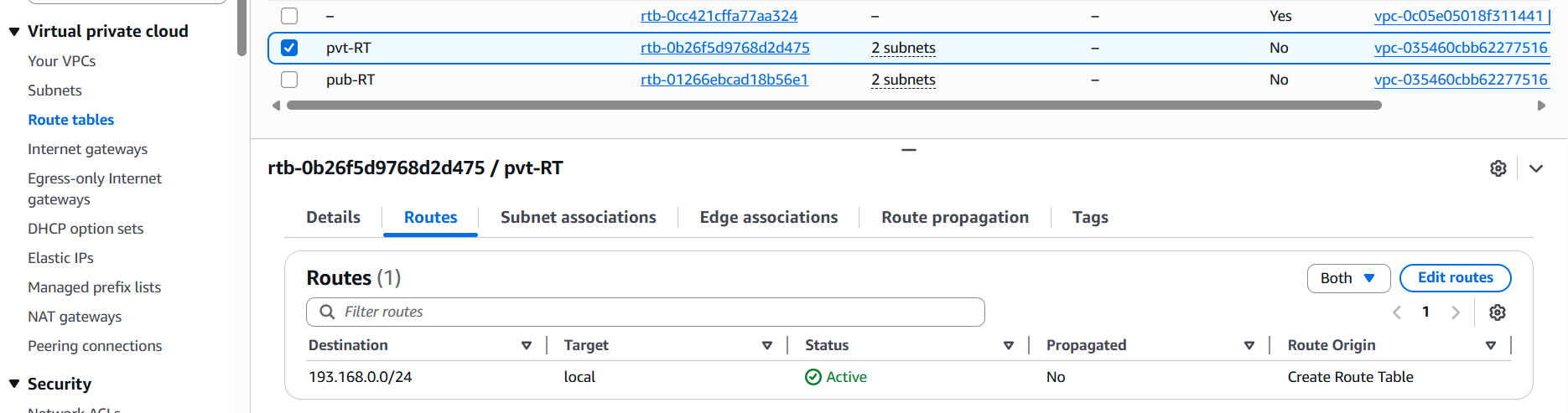


Step 5: In AWS console goto NAT gateways🡪create NAT gateway🡪Enter details.

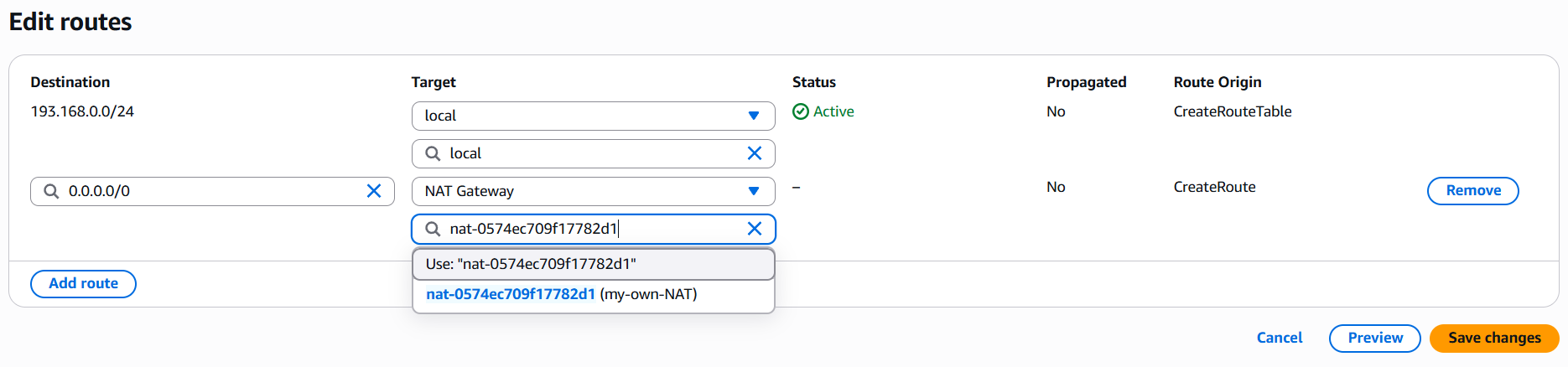




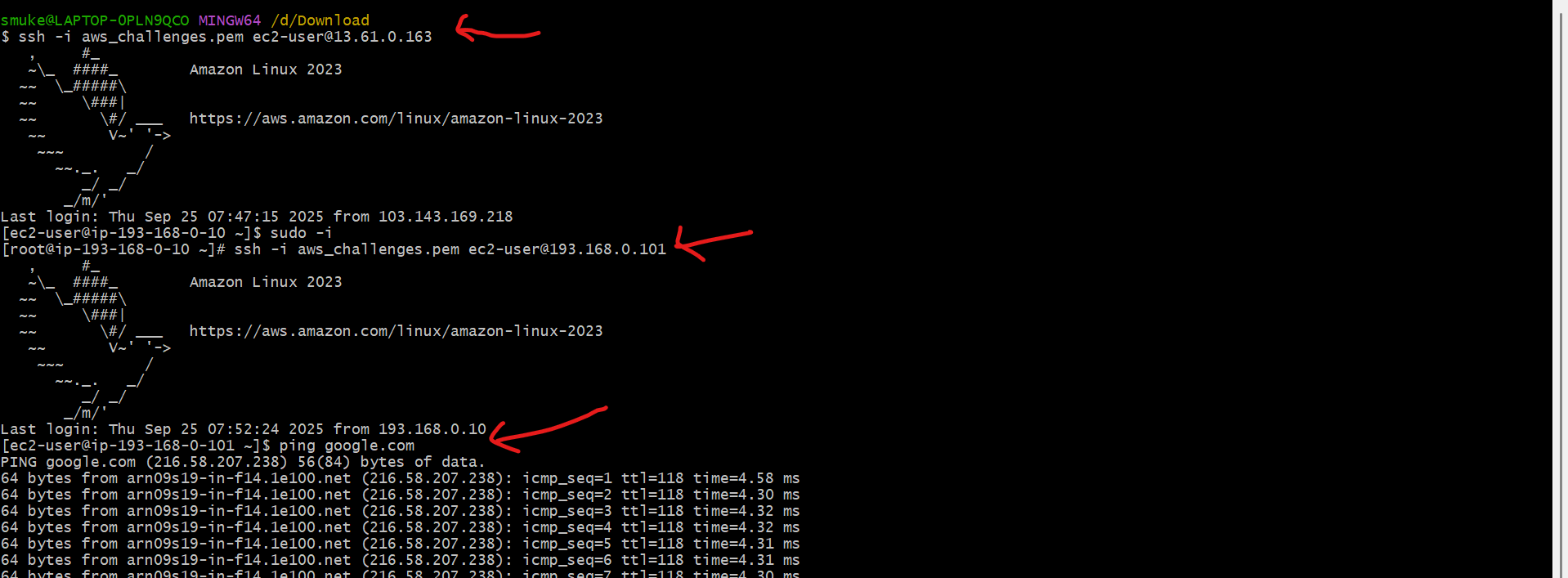
Step 6: Goto Route table🡪select private RT🡪Routes🡪Edit Routes.



Step 7: In Edit Routes select Destination, Target and save changes.

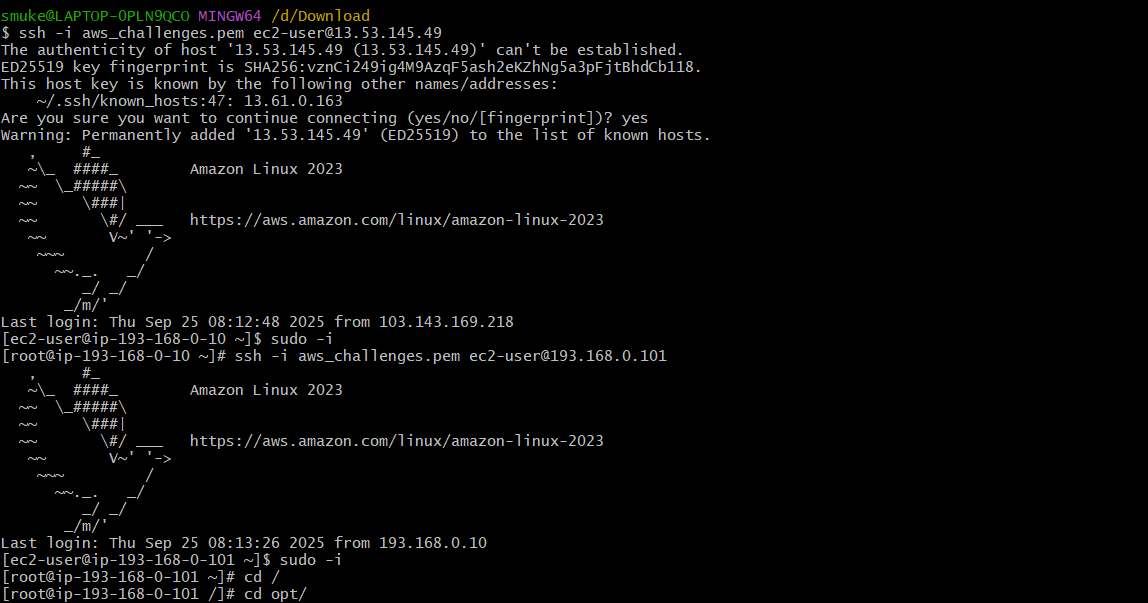


Step 8: Now try **ping URL** in private instance, it will connect to the host.

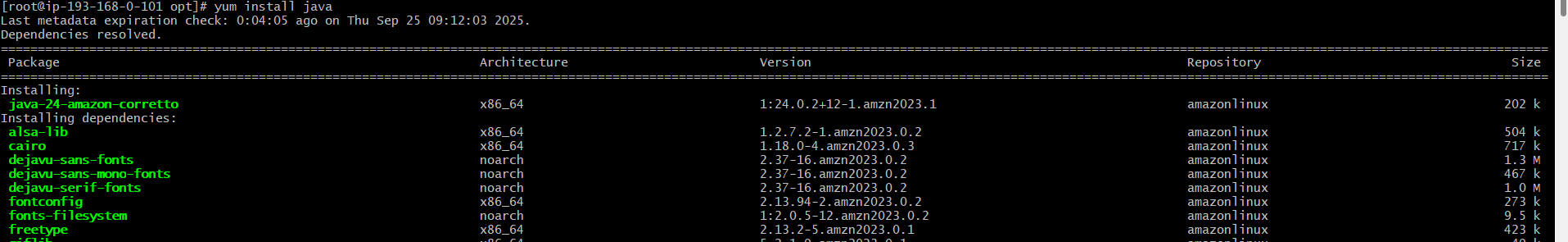


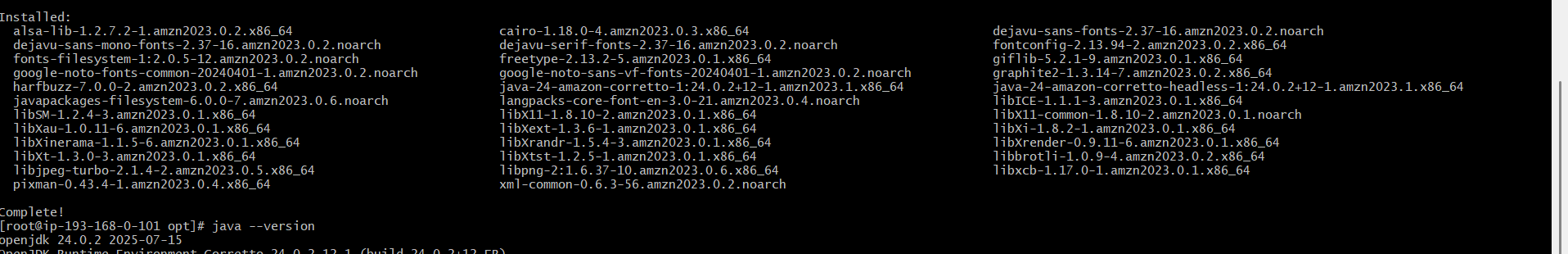
1. Install Apache Tomcat in private EC2 and deploy a sample app.

Step 1: Connect to private instance using Jump/Bastian server and goto **opt/** directory.



Step 2: Install java using **sudo yum install java** command.





Step 3: Download apache tomcat using **wget tomcat\_URL** and extract it.

