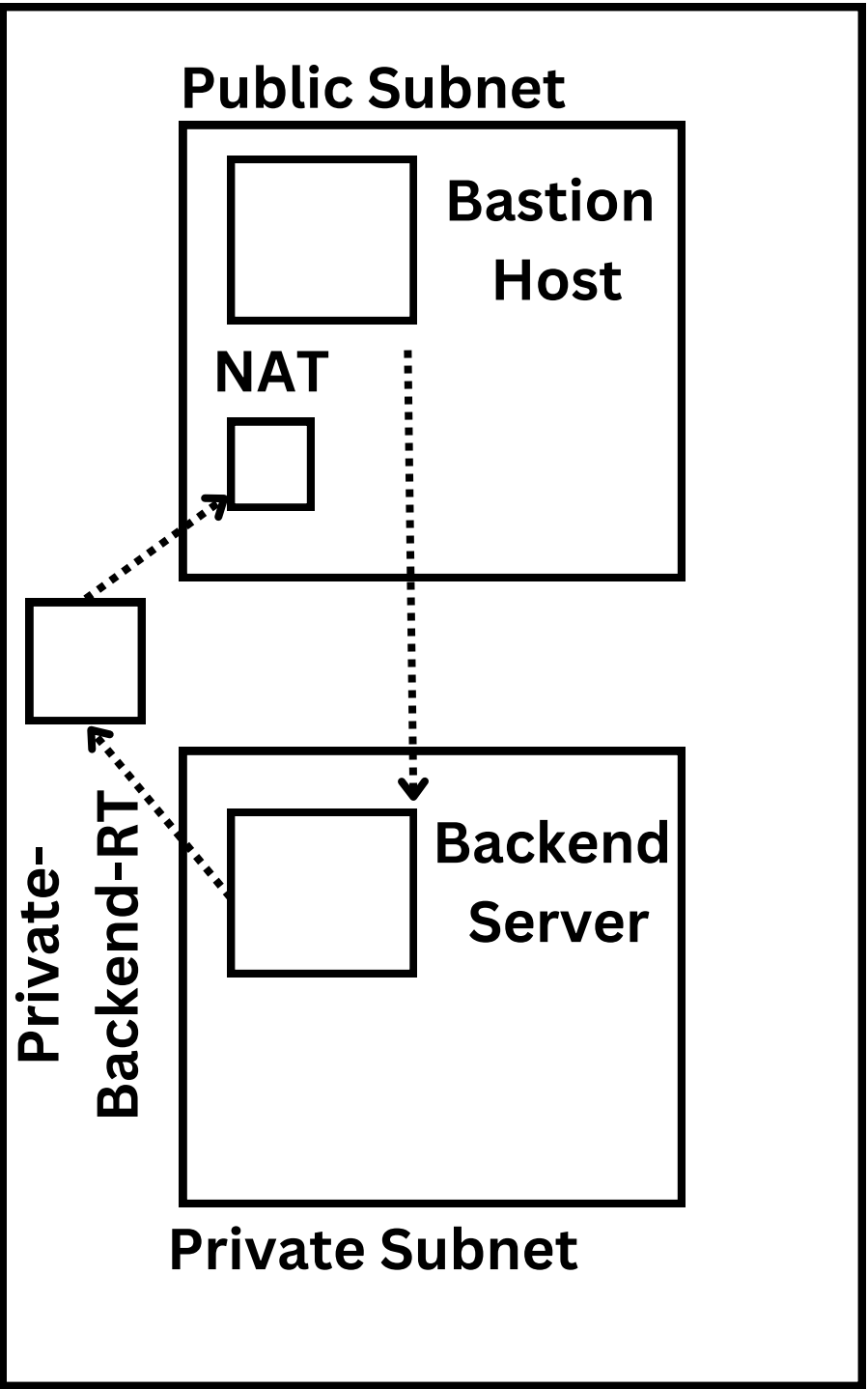
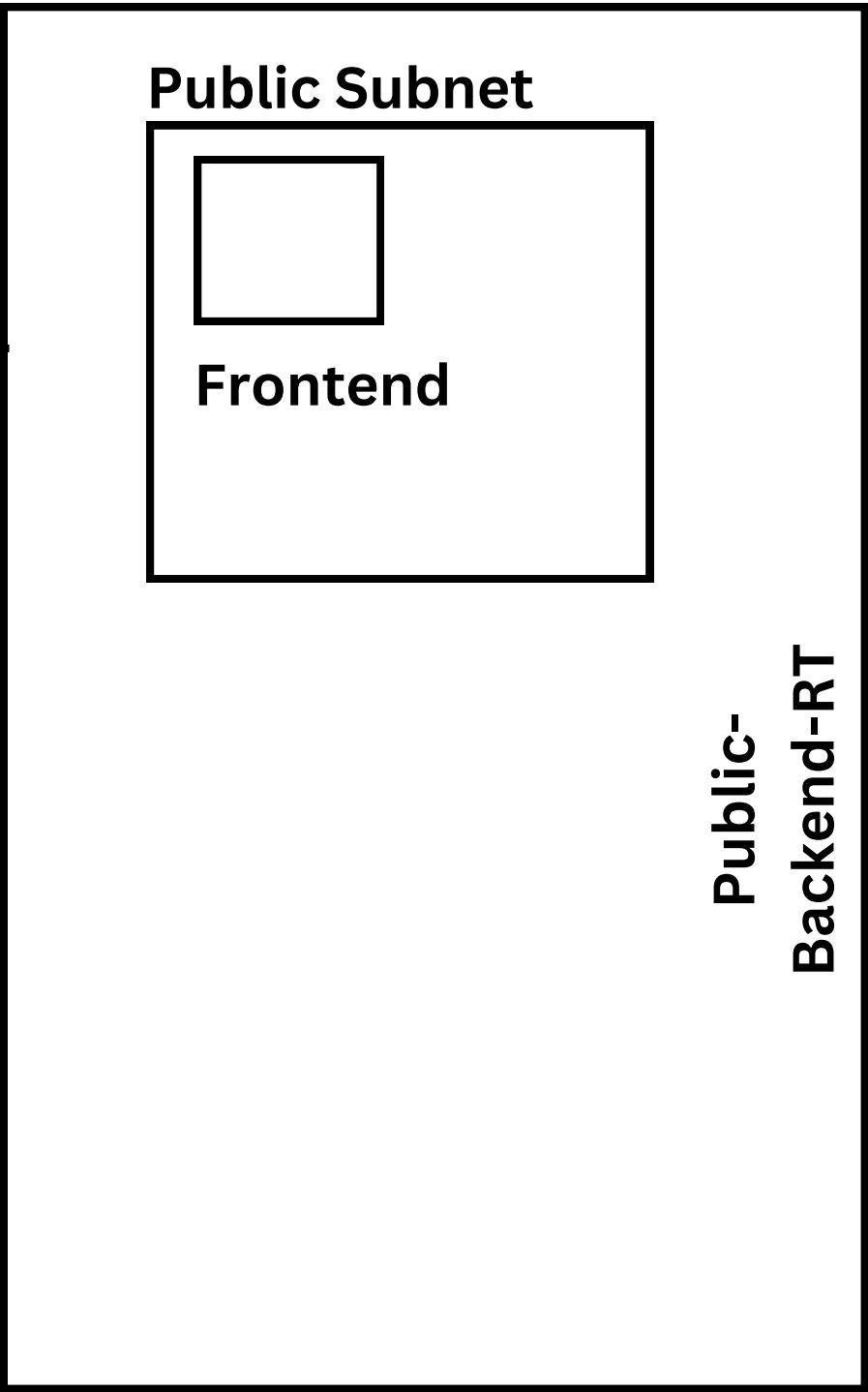


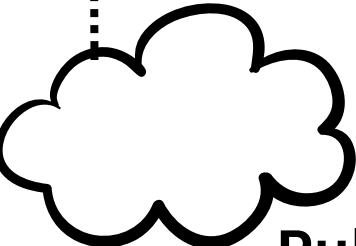
Frontend-VPC

Backend-VPC



VPC Peering

Public Access



Created a Backend VPC

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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 [Info](#)
Creates a tag with a key of 'Name' and a value that you specify.

Backend-VPC

IPv4 CIDR block [Info](#)
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /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

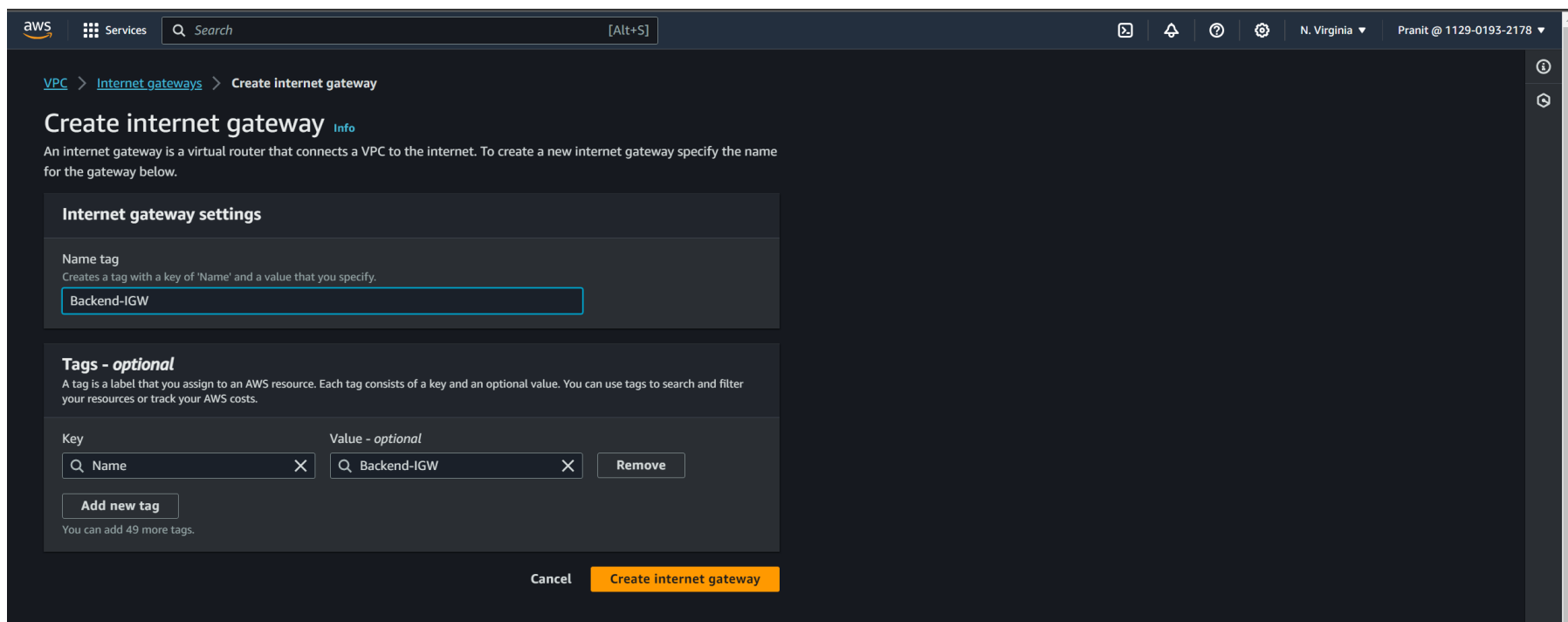
Created public-backend-subnet and private-backend-subnet

Subnets (2/2) [Info](#)

[Subnet ID : subnet-05eff582ae57fab0d](#) [Subnet ID : subnet-042a9160780b5fc57](#) [VPC : vpc-00a582db7cb25153e](#) [Clear filters](#)

<input checked="" type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/>	Public-Backend-Subnet	subnet-05eff582ae57fab0d	Available	vpc-00a582db7cb25153e Bac...	10.0.0.0/24	-
<input checked="" type="checkbox"/>	Private-Backend-Subnet	subnet-042a9160780b5fc57	Available	vpc-00a582db7cb25153e Bac...	10.0.1.0/24	-

Created backend Internet gateway



The screenshot shows the 'Create internet gateway' page in the AWS Management Console. The breadcrumb trail is 'VPC > Internet gateways > Create internet gateway'. The page title is 'Create internet gateway' with an 'Info' link. A description states: 'An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.' The 'Internet gateway settings' section contains a 'Name tag' field with the value 'Backend-IGW'. Below this is the 'Tags - optional' section, which includes a table with one tag: 'Name' as the key and 'Backend-IGW' as the value. There is an 'Add new tag' button and a note 'You can add 49 more tags.' At the bottom right are 'Cancel' and 'Create internet gateway' buttons.

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

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

Backend-IGW

Tags - optional
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.

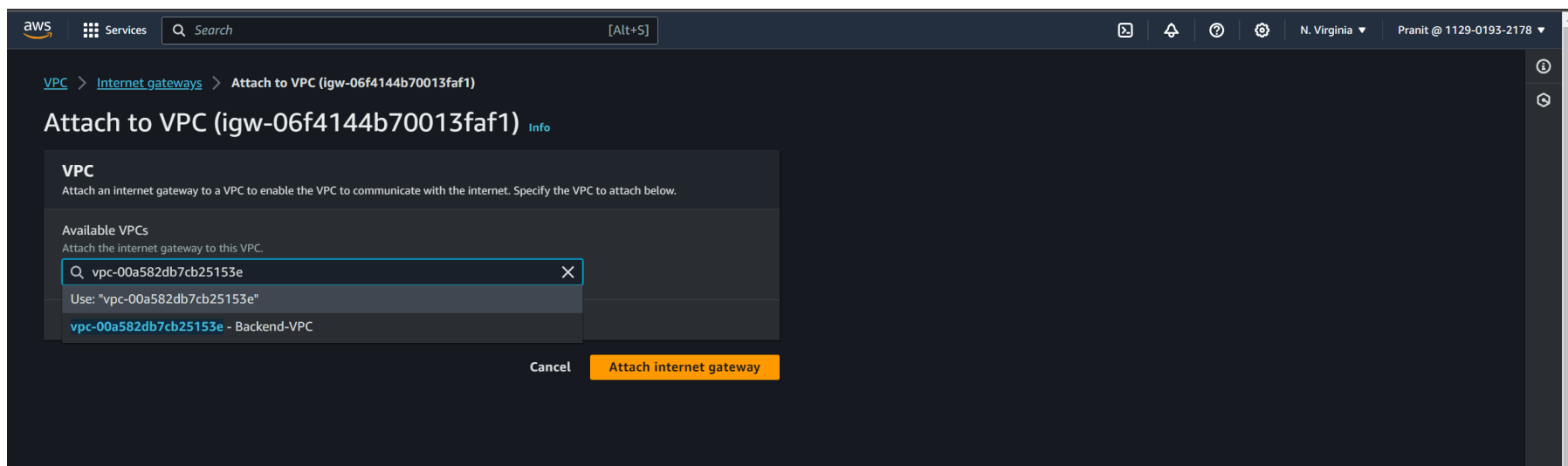
Key	Value - optional	
Name	Backend-IGW	Remove

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create internet gateway](#)

Attached backend internet gateway to backend VPC



The screenshot shows the 'Attach to VPC' page in the AWS Management Console. The breadcrumb trail is 'VPC > Internet gateways > Attach to VPC (igw-06f4144b70013faf1)'. The page title is 'Attach to VPC (igw-06f4144b70013faf1)' with an 'Info' link. A description states: 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' The 'Available VPCs' section contains a search bar with the value 'vpc-00a582db7cb25153e'. Below the search bar is a list of VPCs, with 'vpc-00a582db7cb25153e - Backend-VPC' selected. At the bottom right are 'Cancel' and 'Attach internet gateway' buttons.

Attach to VPC (igw-06f4144b70013faf1) [Info](#)

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

vpc-00a582db7cb25153e

Use: "vpc-00a582db7cb25153e"

vpc-00a582db7cb25153e - Backend-VPC

[Cancel](#) [Attach internet gateway](#)

Created Public route table and Private route table

aws

Services

Search

[Alt+S]

📄

🔔

🕒

⚙️

N. Virginia

Pranit @ 1129-0193-2178

VPC > Route tables > Create route table

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.

Public-Backend-RT

VPC

The VPC to use for this route table.

vpc-00a582db7cb25153e (Backend-VPC)

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

Q Name

Q Public-Backend-RT

Remove

Add new tag

You can add 49 more tags.

Cancel

Create route table

aws

Services

Search

[Alt+S]

📄

🔔

🕒

⚙️

N. Virginia

Pranit @ 1129-0193-2178

VPC > Route tables > Create route table

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.

Private-Backend-RT

VPC

The VPC to use for this route table.

vpc-00a582db7cb25153e (Backend-VPC)

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

Q Name

Q Private-Backend-RT

Remove

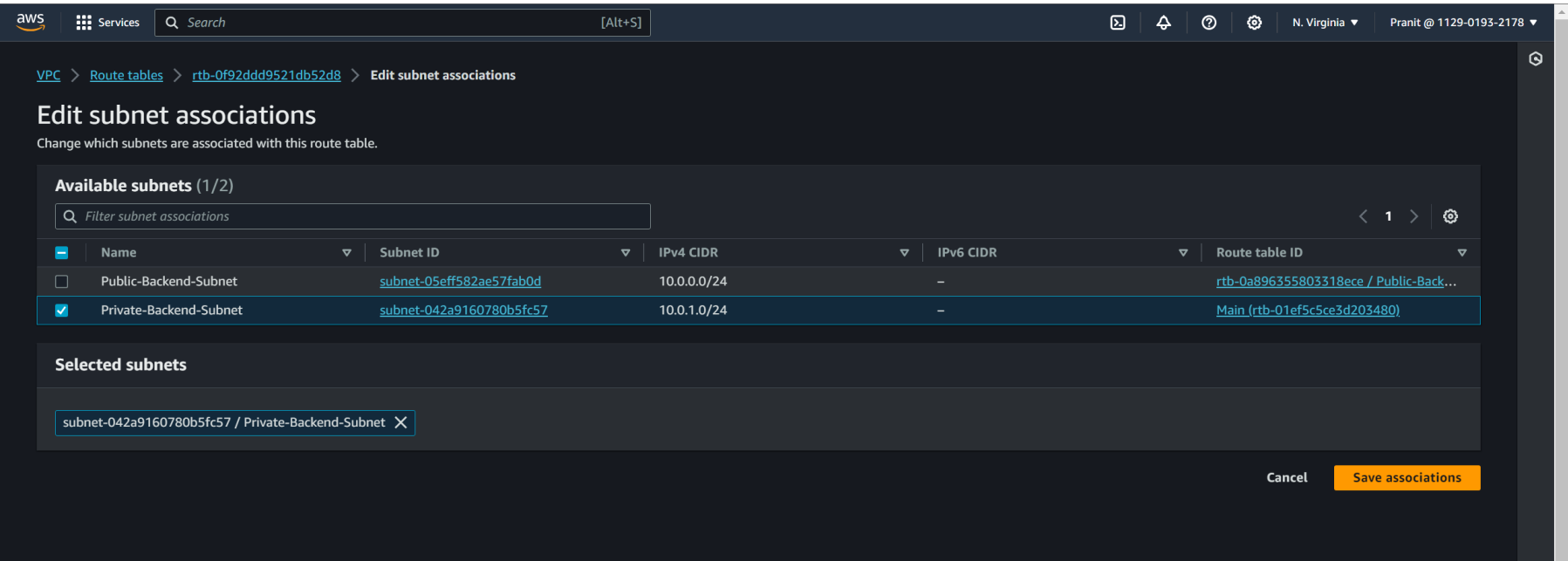
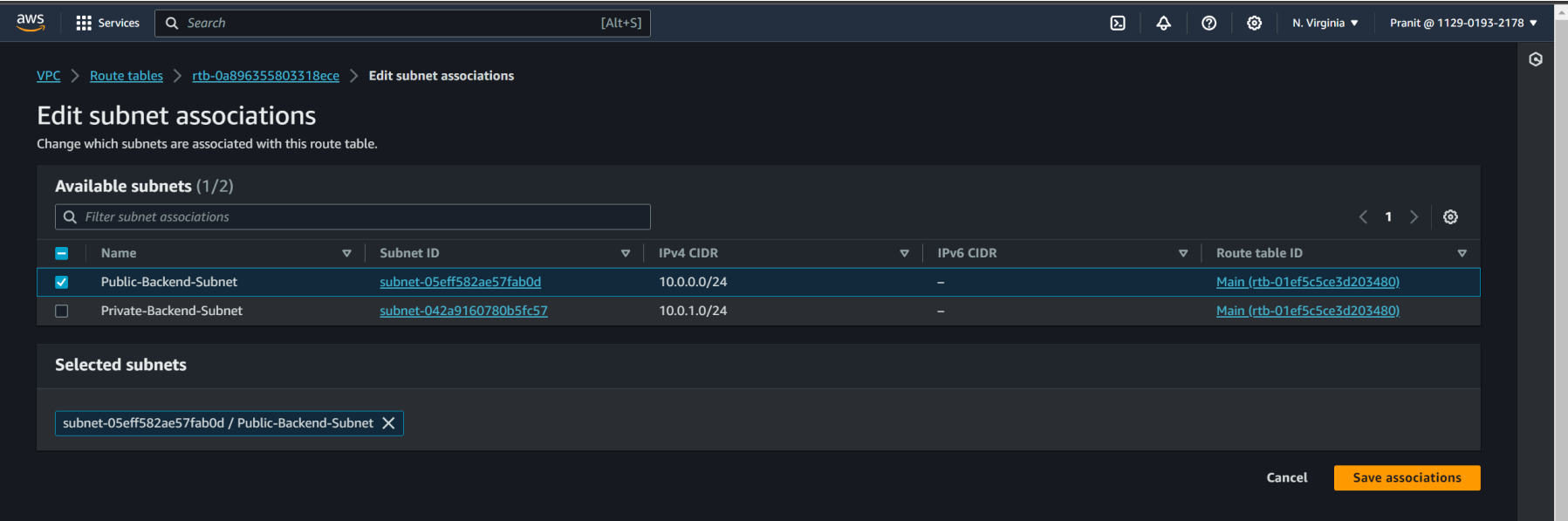
Add new tag

You can add 49 more tags.

Cancel

Create route table

Public subnet is associated with public route table and Private subnet is associated with private route table



Created NAT to make the private subnet to access internet(NAT is created on public subnet)

aws Services Search [Alt+S]

✓ Elastic IP address 54.145.147.59 (eipalloc-06fab8b979706ab47) allocated.

VPC > NAT gateways > Create NAT gateway

Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

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

NAT

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

subnet-05eff582ae57fab0d (Public-Backend-Subnet)

Connectivity type
Select a connectivity type for the NAT gateway.

☒ Public
☐ Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.

eipalloc-06fab8b979706ab47 [Allocate Elastic IP](#)

▶ **Additional settings** [Info](#)

Route traffic on private route table to NAT

aws Services Search [Alt+S]

VPC > Route tables > rtb-0f92ddd9521db52d8 > Edit routes

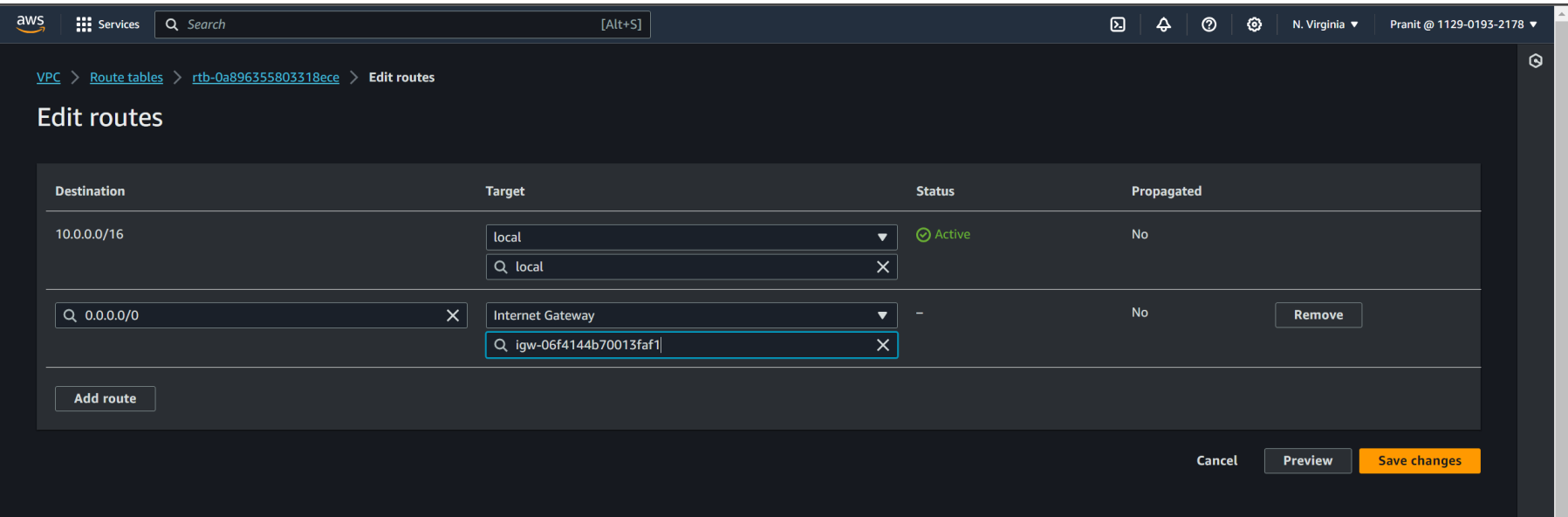
Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	✓ Active	No
0.0.0.0/0	NAT Gateway	-	No
	nat-00972255af7976e13		

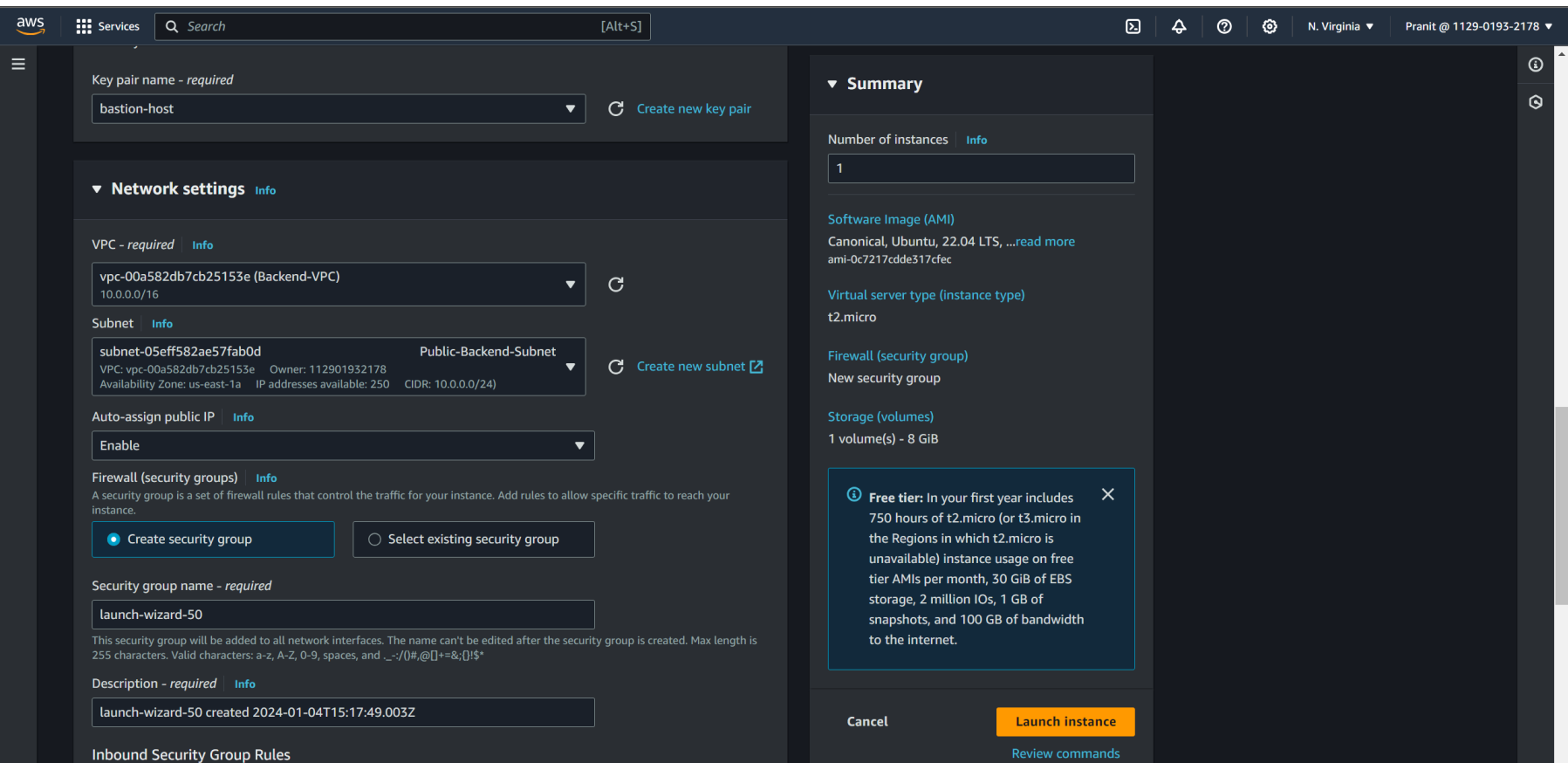
[Add route](#)

[Cancel](#) [Preview](#) [Save changes](#)

Route traffic on public route table to Internet Gateway



Created an EC2 instance(bastion host) on public backend subnet



Created an EC2 instance(backend server) on private backend subnet

The screenshot shows the AWS Management Console interface for creating an EC2 instance. The 'Network settings' section is expanded, showing the following configuration:

- Key pair name - required:** bastion-host
- Network settings:**
 - VPC - required:** vpc-00a582db7cb25153e (Backend-VPC)
 - Subnet:** subnet-042a9160780b5fc57 (Private-Backend-Subnet)
 - Auto-assign public IP:** Disable
 - Firewall (security groups):** Create security group (selected)
 - Security group name - required:** launch-wizard-51
 - Description - required:** launch-wizard-51 created 2024-01-04T15:19:38.098Z

The 'Summary' section on the right shows the following configuration:

- Number of instances:** 1
- Software Image (AMI):** Canonical, Ubuntu, 22.04 LTS, ...read more
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

A 'Free tier' notification is also visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.'

Login into bastion host

```
pranit@PREDATOR MINGW64 ~/Downloads
$ ssh -i "bastion-host.pem" ubuntu@3.83.205.27
The authenticity of host '3.83.205.27 (3.83.205.27)' can't be established.
ED25519 key fingerprint is SHA256:cGR1L+ovp101zLyvfga3G6ZW1w+RcvQ3wpoufidGmT4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.83.205.27' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)

* Documentation: https://help.ubuntu.com
```


Copied the ssh key to bastion host that is used to login the backend server

```
ubuntu@ip-10-0-0-9: ~  
ubuntu@ip-10-0-0-9:~$ vi bastion-host.pem  
ubuntu@ip-10-0-0-9:~$ cat bastion-host.pem  
-----BEGIN RSA PRIVATE KEY-----  
[REDACTED]  
-----END RSA PRIVATE KEY-----  
ubuntu@ip-10-0-0-9:~$ chmod 400 bastion-host.pem  
ubuntu@ip-10-0-0-9:~$ |
```

Login into backend server

```
ubuntu@ip-10-0-1-245: ~  
ubuntu@ip-10-0-0-9:~$ ssh -i "bastion-host.pem" ubuntu@10.0.1.245  
The authenticity of host '10.0.1.245 (10.0.1.245)' can't be established.  
ED25519 key fingerprint is SHA256:Vo0P6kjHL+mC+D39J5qNoGA4qzDURv1Fn1DkpPEKRdI.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '10.0.1.245' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1017-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Thu Jan  4 15:35:55 UTC 2024  
  
System load:  0.0           Processes:            95  
Usage of /:   20.5% of 7.57GB Users logged in:      0  
Memory usage: 20%          IPv4 address for eth0: 10.0.1.245  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-10-0-1-245:~$ |
```

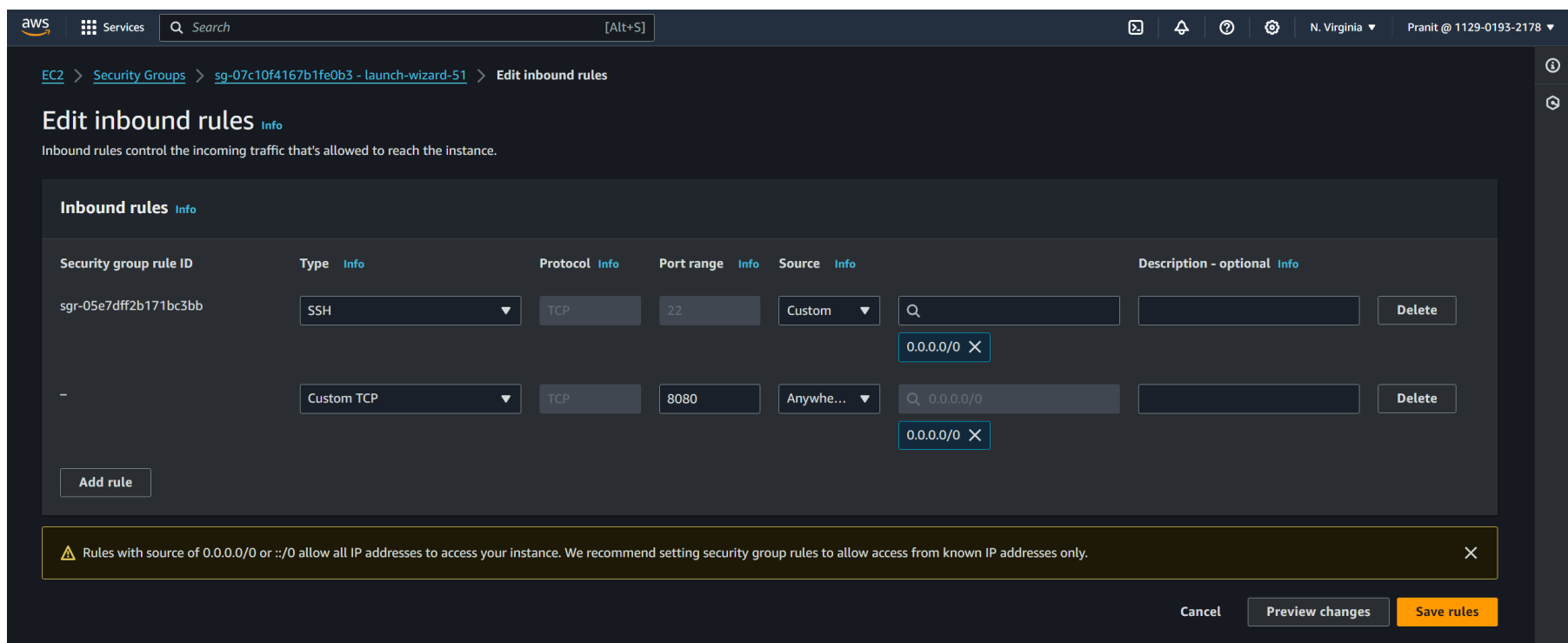
Checked whether the internet is accessible on backend server or not?

```
ubuntu@ip-10-0-1-245: ~  
ubuntu@ip-10-0-1-245:~$ ping google.com  
PING google.com (172.253.62.102) 56(84) bytes of data:  
64 bytes from bc-in-f102.1e100.net (172.253.62.102): icmp_seq=1 ttl=57 time=2.64 ms  
64 bytes from bc-in-f102.1e100.net (172.253.62.102): icmp_seq=2 ttl=57 time=2.01 ms  
64 bytes from bc-in-f102.1e100.net (172.253.62.102): icmp_seq=3 ttl=57 time=1.97 ms
```

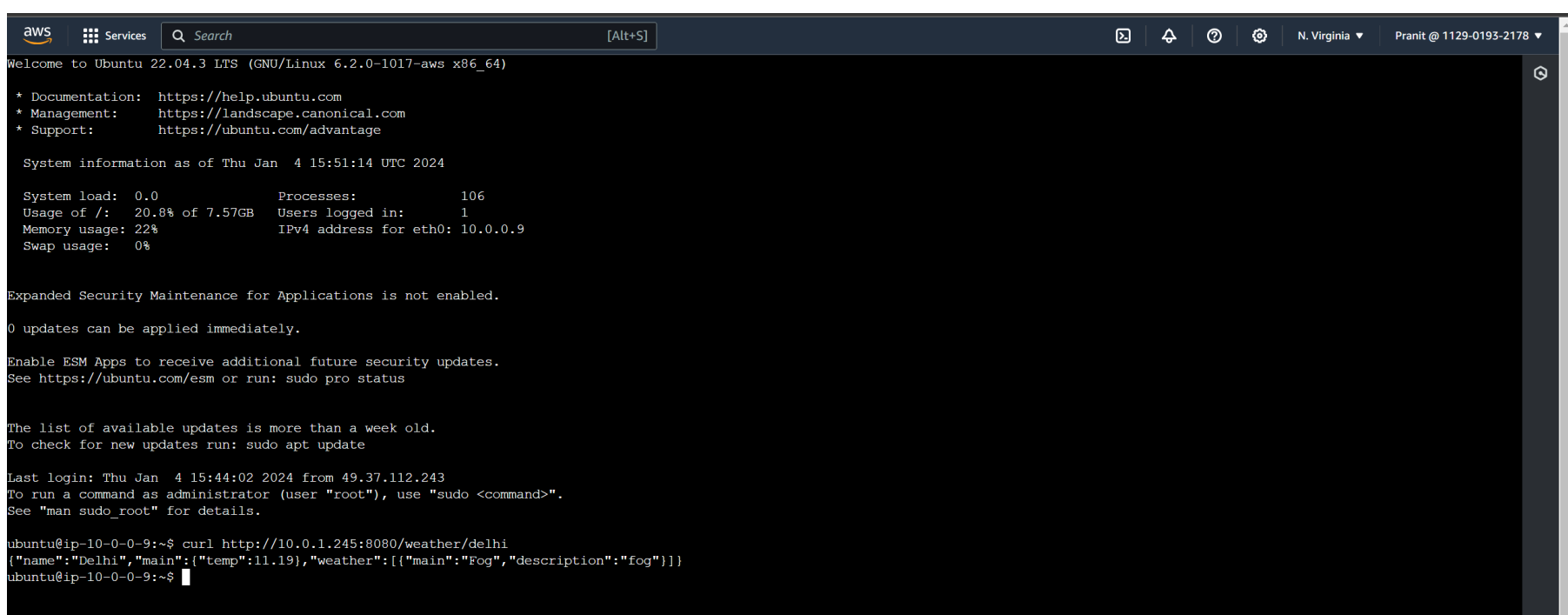
After cloned the git repository , run the backend server

```
ubuntu@ip-10-0-1-245: ~/weather-tracker  
ubuntu@ip-10-0-1-245:~/weather-tracker$ vi .apiConfig  
ubuntu@ip-10-0-1-245:~/weather-tracker$ go run main.go  
go: downloading github.com/rs/cors v1.10.1  
2024/01/04 15:50:20 Listening....
```

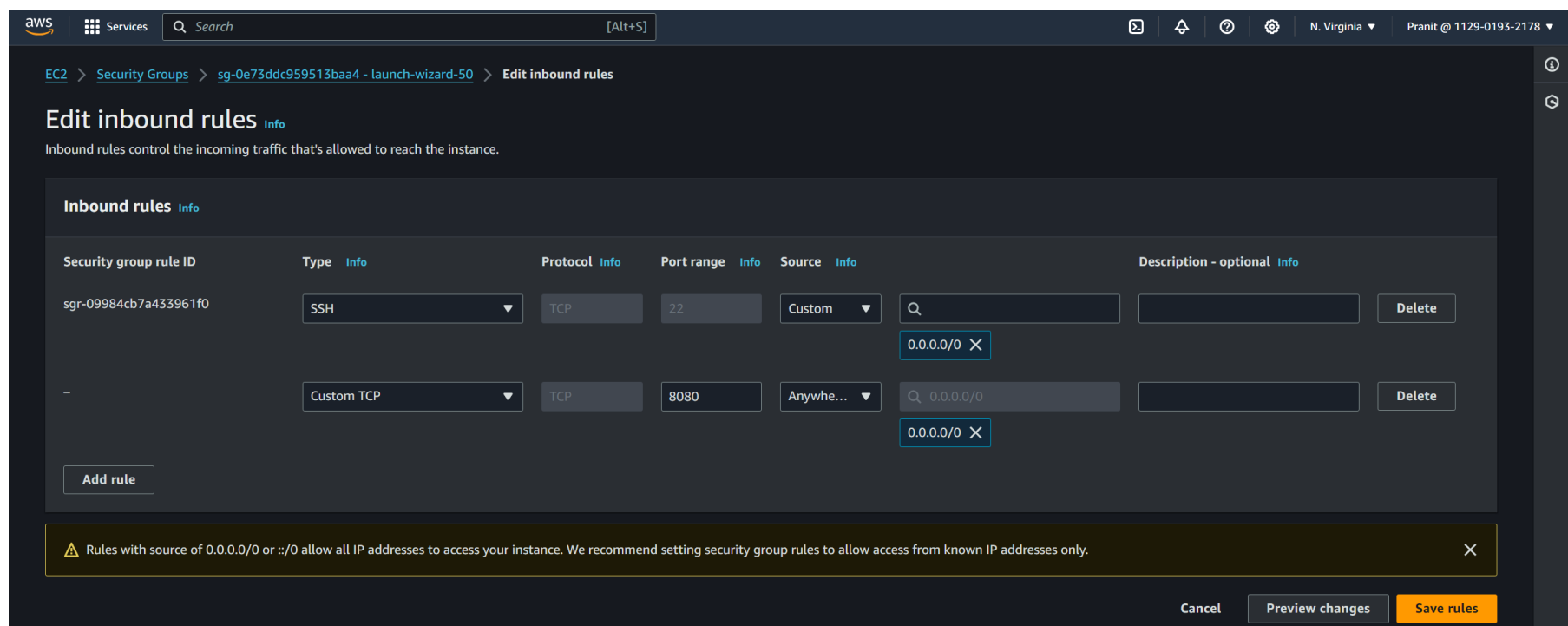
As the backend server runs on 8080 port , so added 8080 port to the security group used in the backend server



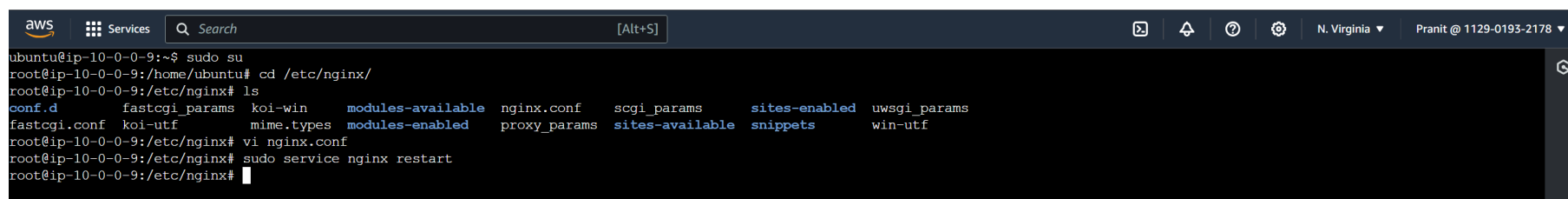
Get response on bastion host from the backend server



Port 8080 is added to security group of bastion host



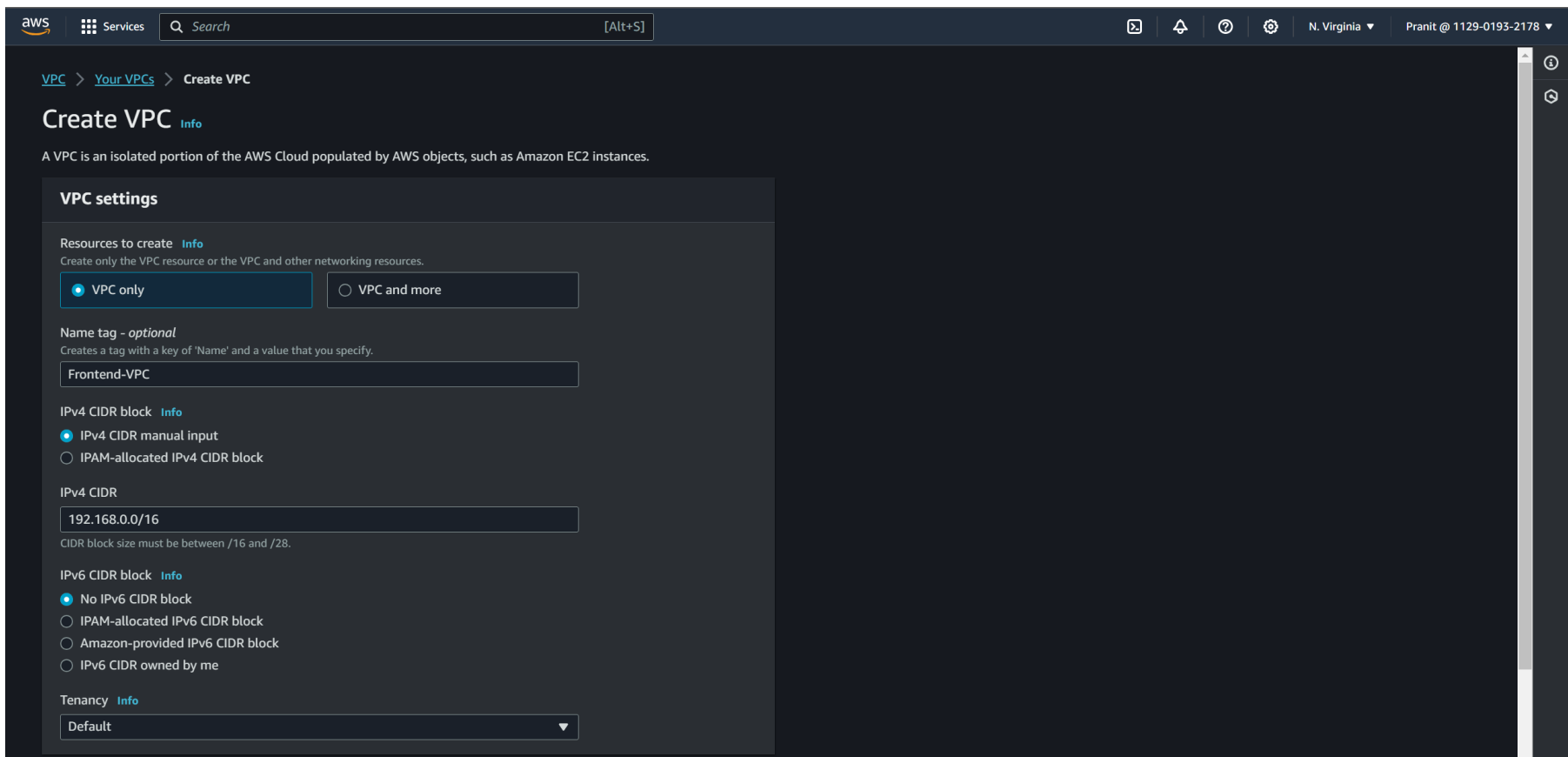
Configured a nginx proxy server



Accessed the api on backend server through the public IP of bastion host



Created a Frontend VPC



Created a public subnet on frontend VPC

The screenshot shows the 'Create subnet' page in the AWS Management Console. The page is titled 'Subnet settings' and includes instructions to specify CIDR blocks and Availability Zone. The 'Subnet 1 of 1' section contains the following fields:

- Subnet name:** A text input field containing 'Public Subnet'.
- Availability Zone:** A dropdown menu showing 'US East (N. Virginia) / us-east-1a'.
- IPv4 VPC CIDR block:** A dropdown menu showing '192.168.0.0/16'.
- IPv4 subnet CIDR block:** A text input field containing '192.168.0.0/24' with a '256 IPs' indicator.
- Tags - optional:** A section with a table for adding tags. The table has columns for 'Key' and 'Value - optional'. A tag with key 'Name' and value 'Public Subnet' is added. There are buttons for 'Add new tag', 'Remove', and 'Add new subnet'.

At the bottom of the page, there are 'Cancel' and 'Create subnet' buttons.

Created a Internet Gateway

The screenshot shows the 'Create internet gateway' page in the AWS Management Console. The page is titled 'Create internet gateway' and includes instructions to specify the name for the gateway. The 'Internet gateway settings' section contains the following fields:

- Name tag:** A text input field containing 'Frontend-IGW'.
- Tags - optional:** A section with a table for adding tags. The table has columns for 'Key' and 'Value - optional'. A tag with key 'Name' and value 'Frontend-IGW' is added. There are buttons for 'Add new tag', 'Remove', and 'Create internet gateway'.

At the bottom of the page, there are 'Cancel' and 'Create internet gateway' buttons.

Created a Route table

The screenshot shows the 'Create route table' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Route tables > Create route table'. The page title is 'Create route table' with an 'Info' link. A description states: '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.
Frontend-RT

VPC
The VPC to use for this route table.
vpc-0721dc77b37d4c5f3 (Frontend-VPC)

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
Name

Value - optional
Frontend-RT

Buttons: Add new tag, Cancel, Create route table

You can add 49 more tags.

Associate the public subnet with public route table(frontend-vpc)

The screenshot shows the 'Edit subnet associations' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Route tables > rtb-07c95dee0f95eded4 > Edit subnet associations'. The page title is 'Edit subnet associations'. A description states: 'Change which subnets are associated with this route table.'

Available subnets (1/1)

Filter subnet associations

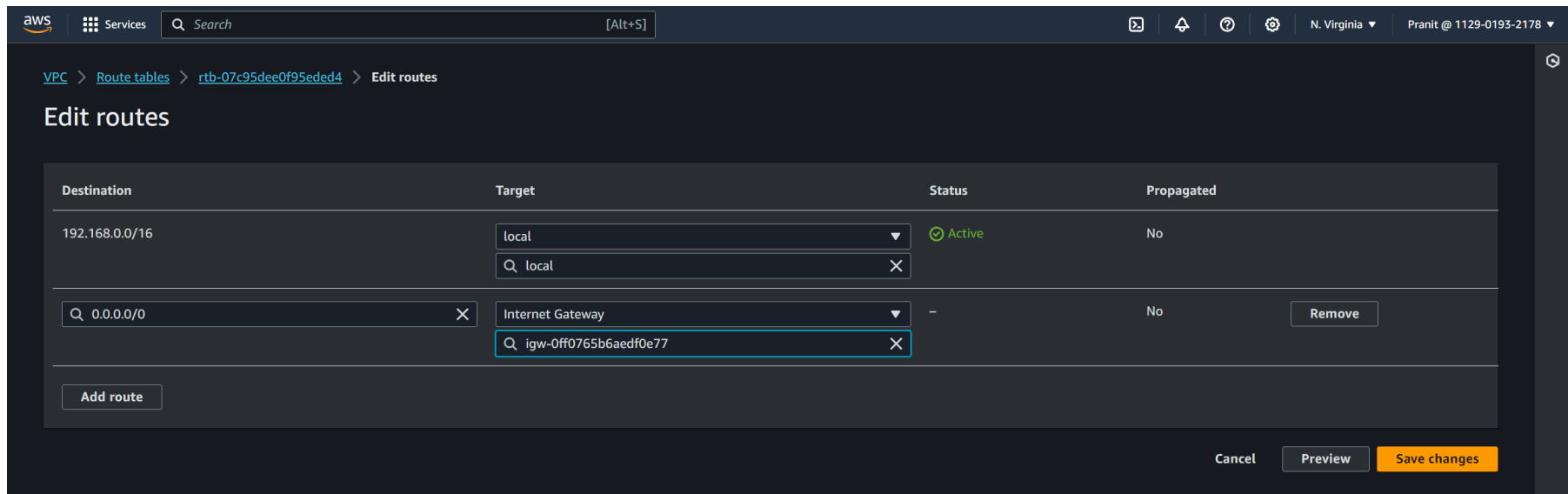
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	Public Subnet	subnet-03aed114adbe7f196	192.168.0.0/24	-	Main (rtb-05511a790a445be72)

Selected subnets

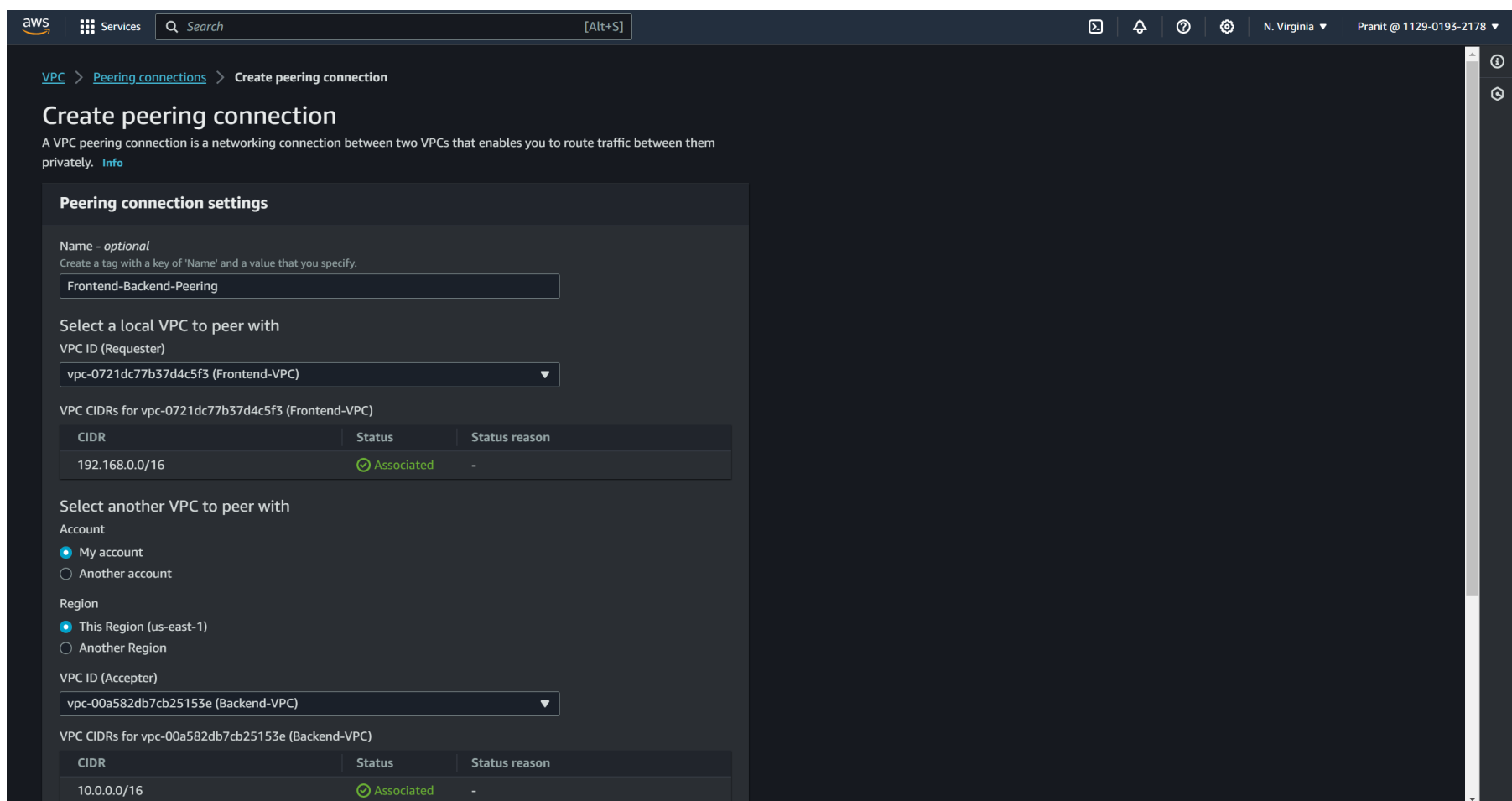
subnet-03aed114adbe7f196 / Public Subnet

Buttons: Cancel, Save associations

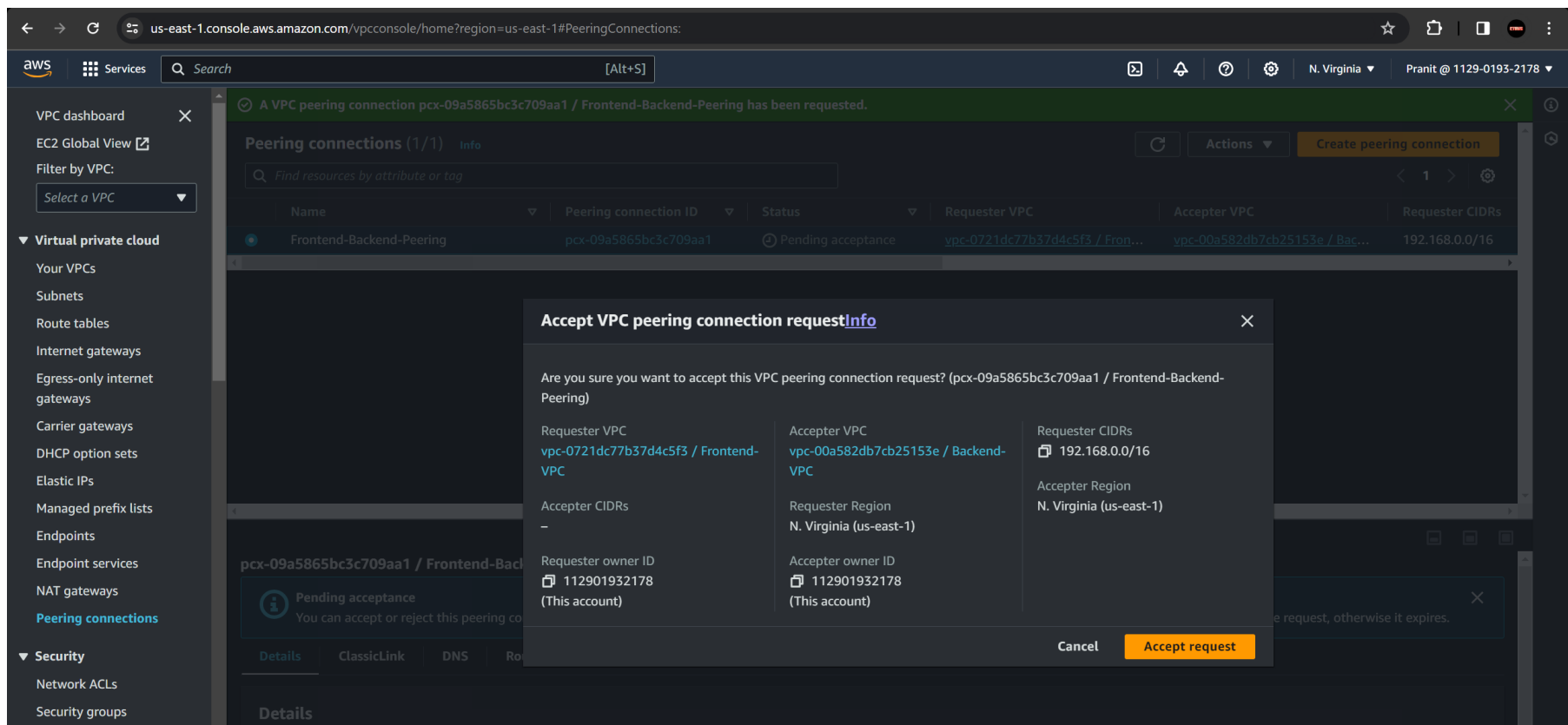
Added a route in route table to the target Internet Gateway



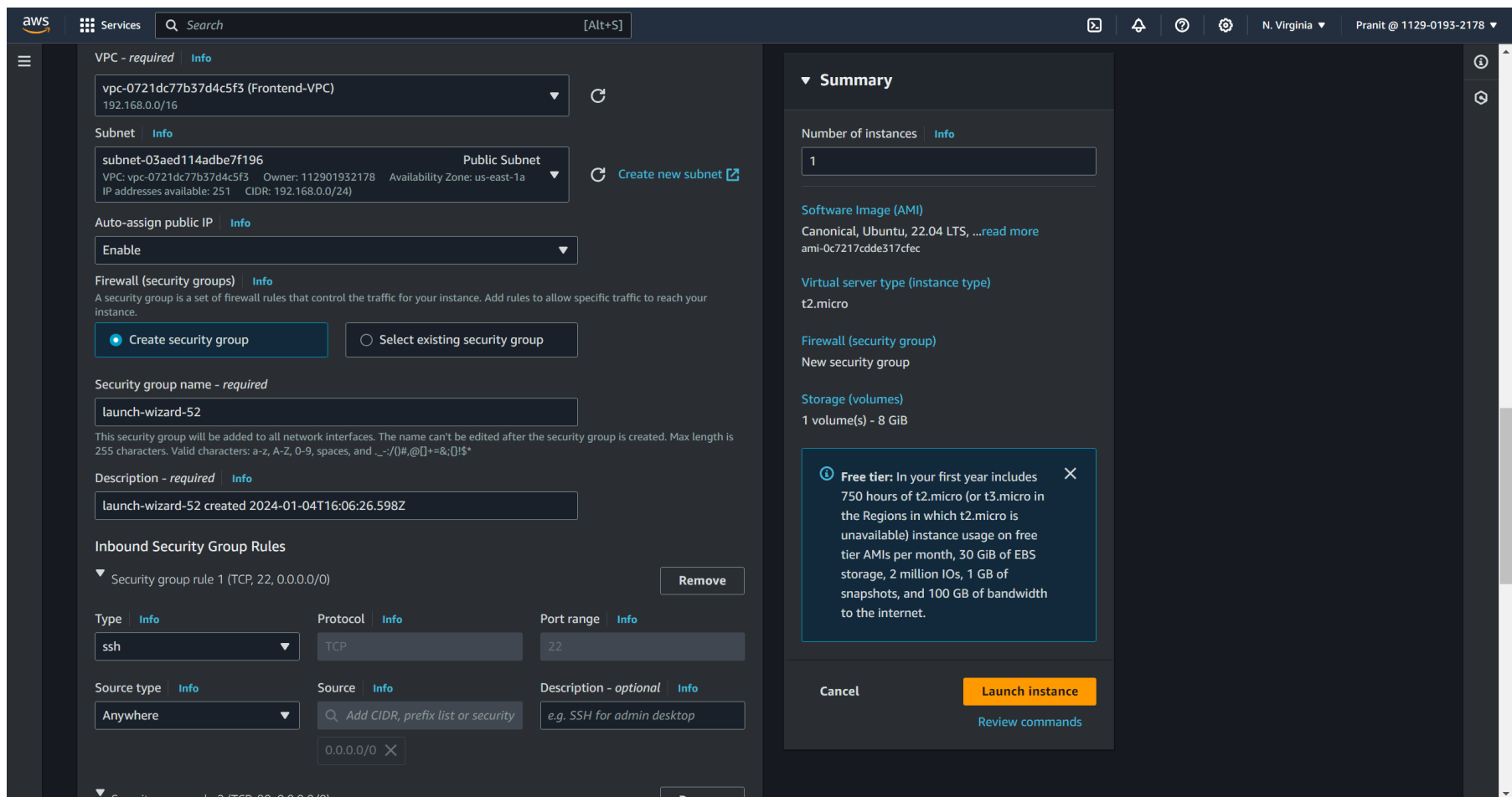
Create a VPC peering(to make connection between frontend and backend)



Accept the request



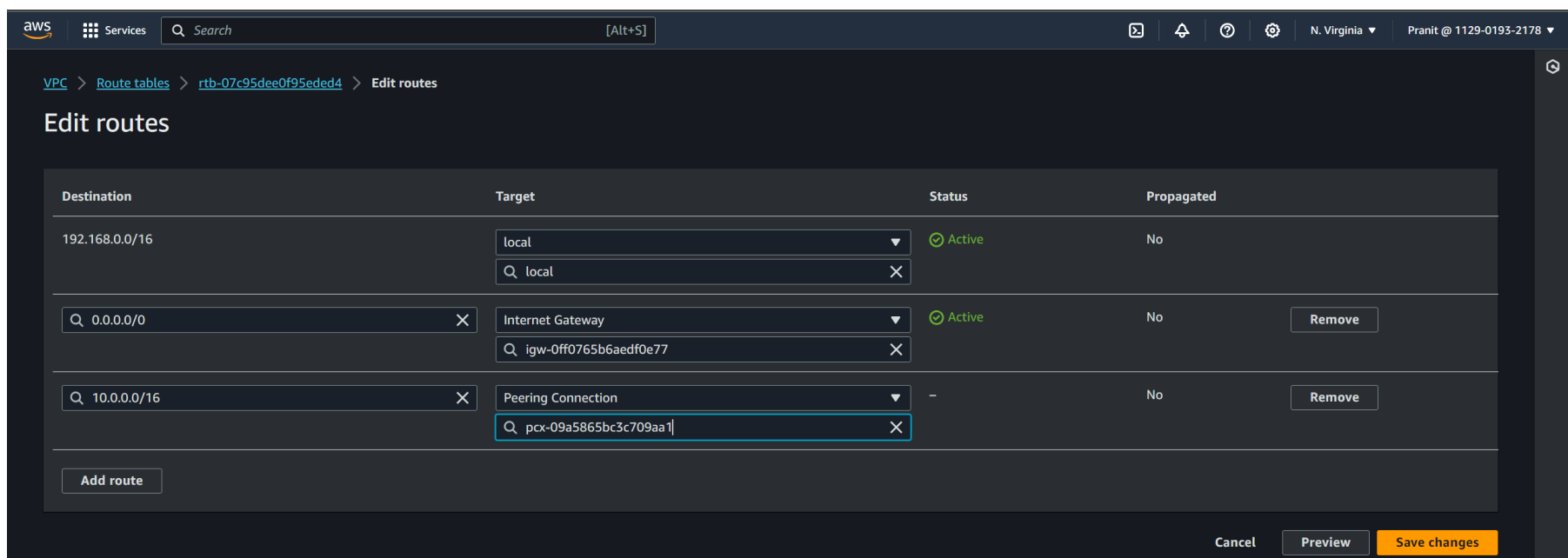
Launched an EC2 instance using Frontend VPC



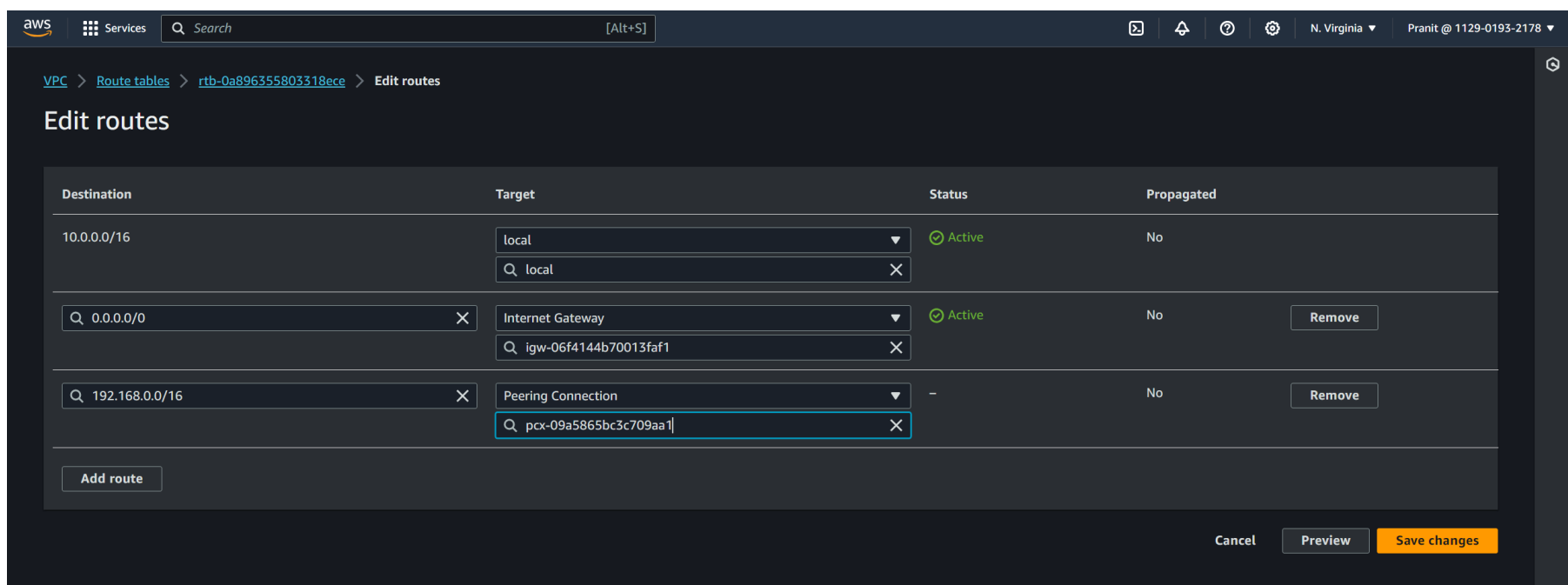
Created a proxy on frontend(EC2 instance)

```
aws
Services
Search
[Alt+S]
root@ip-192-168-0-221:/home/ubuntu/weather-tracker/static# sudo node proxy-server.js
[HPM] Proxy created: /weather -> http://10.0.0.9:8080
Proxy server is running on http://localhost:80
```

In the frontend route table added a route , where destination is the bastion host and target is peering connection



In the backend route table added a route , where destination is the frontend and target is peering connection



Finally accessed the website through Frontend EC2 instance's public IP and it is also successfully getting the responses from the backend server

