Task 1

```
Terraform will perform the following actions:

  # module.Portfolio.module.vpc.aws_default_network_acl.this[0] will be created
  + resource "aws_default_network_acl" "this" {
      + arn                    = (known after apply)
      + default_network_acl_id = (known after apply)
      + id                     = (known after apply)
      + owner_id               = (known after apply)
      + tags                   = {
          + "Name" = "my-vpc-default"
        }
      + tags_all               = {
          + "Name" = "my-vpc-default"
        }
      + vpc_id                 = (known after apply)

      + egress {
          + action          = "allow"
          + from_port       = 0
          + ipv6_cidr_block = "::/0"
```

Task 2

```
module.vpc.aws_eip.nat[0]: Creation complete after 1s [id=eipalloc-0e70659a354b207e9]
module.vpc.aws_nat_gateway.this[0]: Creating...
module.vpc.aws_nat_gateway.this[0]: Still creating... [10s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [20s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [30s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [40s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [50s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [1m0s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [1m10s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [1m20s elapsed]
module.vpc.aws_nat_gateway.this[0]: Still creating... [1m30s elapsed]
module.vpc.aws_nat_gateway.this[0]: Creation complete after 1m34s [id=nat-070493fbaa7bb64ee]
module.vpc.aws_route.private_nat_gateway[0]: Creating...
module.vpc.aws_route.private_nat_gateway[0]: Creation complete after 0s [id=r-rtb-0655b41ae4dec7a3d1080289494]

Apply complete! Resources: 15 added, 0 changed, 0 destroyed.

Outputs:

vpc-id = "vpc-0bc825e4fb6357c0d"
rohan@junie lab-vpc-juniemariam-main %
```

Task 3

```
Terminal   Local
aws_instance.bastion_host: Still destroying... [id=i-0a86a10dc075f95cd, 50s elapsed]
aws_instance.bastion_host: Still destroying... [id=i-0a86a10dc075f95cd, 1m0s elapsed]
aws_instance.bastion_host: Destruction complete after 1m1s
aws_instance.bastion_host: Creating...
aws_instance.bastion_host: Still creating... [10s elapsed]
aws_instance.bastion_host: Creation complete after 12s [id=i-0bb6eab5fc5fc57af]
aws_eip.bastion_eip: Creating...
aws_eip.bastion_eip: Creation complete after 2s [id=eipalloc-0ddfbf929ca62c1f4]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.

Outputs:

instance = "184.169.182.52"
vpc-id = "vpc-04f64e934a5585368"
rohan@junie lab-vpc-juniemariam-main %
```

```
ubuntu@ip-10-0-101-67:~$ TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
MAC_ID=$(curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/  -H "X-aws-ec2-metadata-token: $TOKEN")
VPC_ID=$(curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/${MAC_ID}/vpc-id -H "X-aws-ec2-metadata-token: $TOKEN")

echo "My vpc-id is: $VPC_ID"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    56  100    56    0     0  40000      0 --:--:-- --:--:-- --:--:-- 56000
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    18  100    18    0     0  15358      0 --:--:-- --:--:-- --:--:-- 18000
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    21  100    21    0     0  14767      0 --:--:-- --:--:-- --:--:-- 21000
My vpc-id is: vpc-04f64e934a5585368
ubuntu@ip-10-0-101-67:~$
```

Task 4

```
aws_eip.bastion_eip: Modifications complete after 2s [id=eipalloc-0ddfbf929ca62c1f4]

Apply complete! Resources: 3 added, 1 changed, 1 destroyed.

Outputs:

instance = "184.169.182.52"
private_ip = "10.0.1.179"
vpc-id = "vpc-04f64e934a5585368"
rohan@junie lab-vpc-juniemariam-main %
```

```
ubuntu@ip-10-0-101-9:~$ ping 10.0.1.179 -w 4
PING 10.0.1.179 (10.0.1.179) 56(84) bytes of data.
64 bytes from 10.0.1.179: icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from 10.0.1.179: icmp_seq=2 ttl=64 time=1.12 ms
64 bytes from 10.0.1.179: icmp_seq=3 ttl=64 time=1.03 ms
64 bytes from 10.0.1.179: icmp_seq=4 ttl=64 time=1.09 ms

--- 10.0.1.179 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.030/1.179/1.478/0.175 ms
```

Task 5

Created a private instance having the same key pair of bastion instance. And added a security group to the private instance to allow SSH access from the bastion host.

```
instance = "184.169.182.52"
private_ip = "10.0.1.14"
vpc-id = "vpc-04f64e934a5585368"
```

```
# Configuration for the Bastion Host
Host bastion
HostName 184.169.182.52
User ubuntu
IdentityFile /Users/rohan/CloudFiles/lab-vpc-juniemariam-main/key.pem
ForwardAgent yes

# Configuration for the Private Instance
Host private-instance
HostName 10.0.1.14
User ubuntu
ProxyJump bastion
```

```
rohan@junie lab-vpc-juniemariam-main % ssh -A ubuntu@184.169.182.52
The authenticity of host '184.169.182.52 (184.169.182.52)' can't be established.
ED25519 key fingerprint is SHA256:9koOOAuY1NEG15MORsnvbvRhBfZJNspmVfyjqykPW6o.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '184.169.182.52' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)
```

```
ubuntu@ip-10-0-101-199:~$ ssh ubuntu@10.0.1.14
The authenticity of host '10.0.1.14 (10.0.1.14)' can't be established.
ED25519 key fingerprint is SHA256:obnuvBu1zWlEJHNE+ax9fP4JP/8wDCEHLPcMrbwOb5k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.14' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Sep 24 06:30:27 UTC 2024
```

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Sep 24 06:17:24 2024 from 10.0.101.23
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-14:~$
```

```
ubuntu@ip-10-0-101-239:~$ ping google.com
PING google.com (142.251.46.206) 56(84) bytes of data.
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=1 ttl=118 time=1.79 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=2 ttl=118 time=1.86 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=3 ttl=118 time=1.96 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=4 ttl=118 time=1.79 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=5 ttl=118 time=1.95 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=6 ttl=118 time=1.80 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=7 ttl=118 time=1.81 ms
64 bytes from nuq04s45-in-f14.1e100.net (142.251.46.206): icmp_seq=8 ttl=118 time=1.96 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7011ms
rtt min/avg/max/mdev = 1.789/1.865/1.962/0.073 ms
```

Answers to the question:

1. One of the pain points of this lab was copying over old code from Lab: Compute and modifying some hard coded values to adjust it to our needs. Use what you learned from the self study articles and today's lab to propose a cleaner way to create the EC2 instances and security groups.

   Answer:
   This was quite a learning experience! Changing the hard coded values and replacing them with variables. So I changed the ec2.tf and sg.tf files introducing modules and variables replacing the hard coded values.
   To keep the variables I created a file variables.tf and also modified ec2.tf and sg.tf to fetch values dynamically.

   The cleaner modified code is as follows:
   ec2.tf

```
resource "tls_private_key" "key" {
 algorithm = "RSA"
 rsa_bits  = 4096
}

resource "aws_key_pair" "key_pair" {
 key_name   = var.key_name
 public_key = tls_private_key.key.public_key_openssh

 provisioner "local-exec" {
        command   =   "echo   '${tls_private_key.key.private_key_pem}'    >
${path.module}/key.pem && chmod 0700 ${path.module}/key.pem"
 }
```

```
}

resource "aws_instance" "bastion_host" {
 ami           = "ami-0d53d72369335a9d6"
 instance_type = var.instance_type
 key_name      = aws_key_pair.key_pair.key_name
 subnet_id     = module.vpc.public_subnets[0]


 security_groups = [aws_security_group.allow_ssh_http.id]

 #user_data = file("${path.module}/install_apache.sh")

}
resource "aws_eip" "bastion_eip" {
 instance = aws_instance.bastion_host.id
 domain   = var.domain
}

output "instance" {
 value = aws_eip.bastion_eip.public_ip
}

# Create an EC2 instance in the private subnet
resource "aws_instance" "private_instance" {
 ami                    = aws_instance.bastion_host.ami
 instance_type          = var.instance_type
 key_name               = aws_key_pair.key_pair.key_name
 subnet_id              = module.vpc.private_subnets[0]
     vpc_security_group_ids     =     [aws_security_group.icmp_access.id,
aws_security_group.ssh_access.id]

}

# Output the private IP address
output "private_ip" {
 value = aws_instance.private_instance.private_ip
}
```

sg.tf

```
resource "aws_security_group" "allow_ssh_http" {
 name        = var.sg_name_ssh_http
 description = "Allow SSH and HTTP traffic"
 vpc_id      = module.vpc.vpc_id

 tags = {
   Name = var.sg_name_ssh_http
 }
```

```
}

resource "aws_vpc_security_group_ingress_rule" "allow_ssh_ipv4" {
 security_group_id = aws_security_group.allow_ssh_http.id
 from_port         = var.ssh_ingress_from_port
 cidr_ipv4         = var.default_cidr_block
 ip_protocol       = "tcp"
 to_port           = var.ssh_ingress_to_port
}

resource "aws_vpc_security_group_ingress_rule" "allow_ssh_ipv6" {
 security_group_id = aws_security_group.allow_ssh_http.id
 from_port         = var.ssh_ingress_from_port
 cidr_ipv6         = var.ssh_ingress_cidr_ipv6
 ip_protocol       = "tcp"
 to_port           = var.ssh_ingress_to_port
}

resource "aws_vpc_security_group_egress_rule" "allow_all_traffic_ipv4" {
 security_group_id = aws_security_group.allow_ssh_http.id
 cidr_ipv4         = var.egress_cidr_ipv4
 ip_protocol       = "-1" # Allows all traffic
}

resource "aws_vpc_security_group_egress_rule" "allow_all_traffic_ipv6" {
 security_group_id = aws_security_group.allow_ssh_http.id
 cidr_ipv6         = var.egress_cidr_ipv6
 ip_protocol       = "-1" # Allows all traffic
}

# Security Group for ICMP traffic
resource "aws_security_group" "icmp_access" {
 name        = var.sg_name_icmp
 description = "Allow ICMP traffic"
 vpc_id      = module.vpc.vpc_id

 ingress {
   from_port   = -1 # Allows all ICMP traffic
   to_port     = -1 # Allows all ICMP traffic
   protocol    = var.icmp_protocol
   cidr_blocks = var.icmp_cidr_blocks
 }

 egress {
   from_port   = 0
   to_port     = 0
   protocol    = "-1" # Allows all outbound traffic
   cidr_blocks = [var.default_cidr_block]
 }
```

```
}

# Security Group for SSH access
resource "aws_security_group" "ssh_access" {
 name        = var.sg_name_ssh
 description = "Allow SSH access from Bastion Host"
 vpc_id      = module.vpc.vpc_id

 ingress {
   from_port   = var.ssh_ingress_from_port
   to_port     = var.ssh_ingress_to_port
   protocol    = "tcp"
   cidr_blocks = ["${aws_instance.bastion_host.private_ip}/32"]
 }

 egress {
   from_port   = 0
   to_port     = 0
   protocol    = "-1"
   cidr_blocks = [var.default_cidr_block]
 }
}
```

Finally variables.tf

```
variable "instance_type" {
 description = "EC2 instance type"
 type        = string
 default     = "t2.micro"
}

variable "key_name" {
 description = "Name of the key pair to use for SSH access"
 type        = string
 default     = "key"
}

variable "sg_name_ssh_http" {
 description = "Name for the SSH and HTTP security group"
 type        = string
 default     = "allow_ssh_http"
}

variable "sg_name_icmp" {
 description = "Name for the ICMP security group"
 type        = string
 default     = "icmp_access"
}
```

```
variable "icmp_protocol" {
 description = "Protocol for ICMP traffic"
 type        = string
 default     = "icmp"
}

variable "sg_name_ssh" {
 description = "Name for the SSH access security group from Bastion"
 type        = string
 default     = "ssh_access"
}

variable "default_cidr_block" {
 description = "Default CIDR block for inbound traffic"
 type        = string
 default     = "0.0.0.0/0"
}

variable "ssh_ingress_cidr_ipv6" {
 description = "CIDR block for SSH ingress (IPv6)"
 type        = string
 default     = "::/0"
}

variable "icmp_cidr_blocks" {
 description = "CIDR blocks for ICMP ingress"
 type        = list(string)
 default     = ["0.0.0.0/0"]
}

variable "egress_cidr_ipv4" {
 description = "CIDR block for outbound traffic (IPv4)"
 type        = string
 default     = "0.0.0.0/0"
}

variable "egress_cidr_ipv6" {
 description = "CIDR block for outbound traffic (IPv6)"
 type        = string
 default     = "::/0"
}

variable "ssh_ingress_from_port" {
 description = "Starting port for SSH ingress"
 type        = number
 default     = 22
}

variable "ssh_ingress_to_port" {
```

```
 description = "Ending port for SSH ingress"
 type        = number
 default     = 22
}
variable "domain" {
 description = "Domain type for the Elastic IP allocation (e.g., vpc or
standard)"
 type        = string
 default     = "vpc"  # Default value for instances in a VPC
}
```

After making the modifications I re- ran and checked if all the experimented functionalities worked as expected.

Results:

```
ubuntu@ip-10-0-101-90:~$ ssh ubuntu@10.0.1.96
The authenticity of host '10.0.1.96 (10.0.1.96)' can't be established.
ED25519 key fingerprint is SHA256:VLpxRYlrCOcZRTStAbncoRjdbin9hTjAFWSGoYkzueA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.96' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Sep 24 23:52:04 UTC 2024
```

```
ubuntu@ip-10-0-1-96:~$ ping google.com
PING google.com (142.250.191.46) 56(84) bytes of data.
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=1 ttl=117 time=2.21 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=2 ttl=117 time=1.95 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=3 ttl=117 time=2.17 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=4 ttl=117 time=1.84 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=5 ttl=117 time=2.07 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=6 ttl=117 time=1.79 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=7 ttl=117 time=1.81 ms
ç64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=8 ttl=117 time=1.90 ms
64 bytes from nuq04s42-in-f14.1e100.net (142.250.191.46): icmp_seq=9 ttl=117 time=1.97 ms
^C
--- google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8014ms
rtt min/avg/max/mdev = 1.785/1.967/2.206/0.143 ms
ubuntu@ip-10-0-1-96:~$
```

2. The VPC module automatically creates route tables for the subnets. What is the destination for the route to external traffic in the route table associated with the private subnet?

   ANSWER:
   In a VPC setup, the route table for a private subnet typically directs external traffic through a NAT (Network Address Translation) gateway or NAT instance. The destination for this traffic is set to 0.0.0.0/0, which allows all outbound internet access. The target for this route points to the NAT gateway or instance, which resides in a public subnet that has an Internet Gateway for internet connectivity. This setup enables instances in the private subnet to reach the internet for updates and downloads while keeping them secure from direct inbound traffic from the internet.