ASSIGNMENT: 1

TERRAFORM

NAME: OM VATS

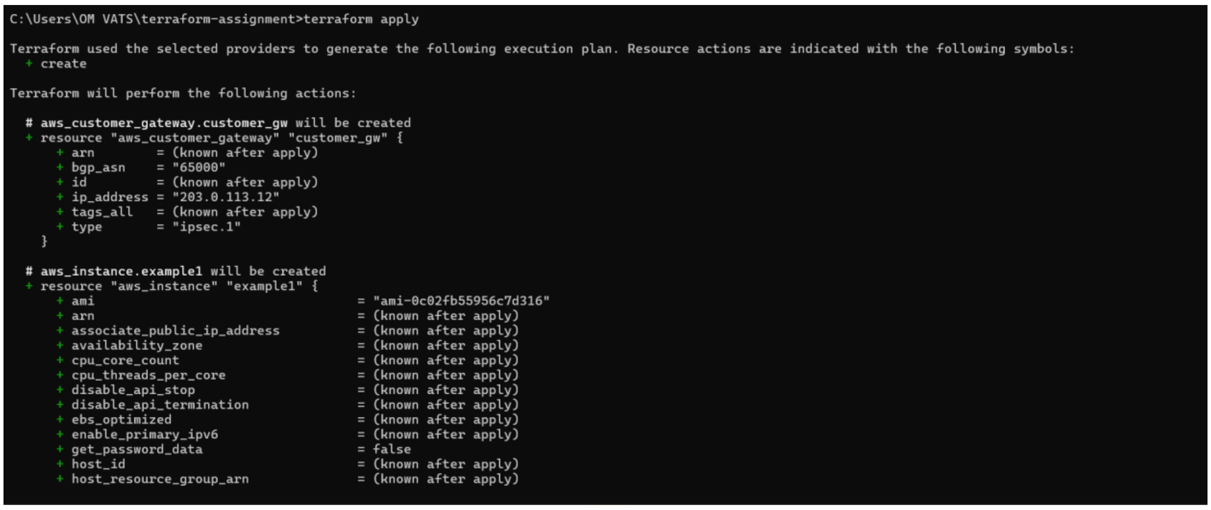
SPCM ASSIGNMENT

SAP'ID 500105231

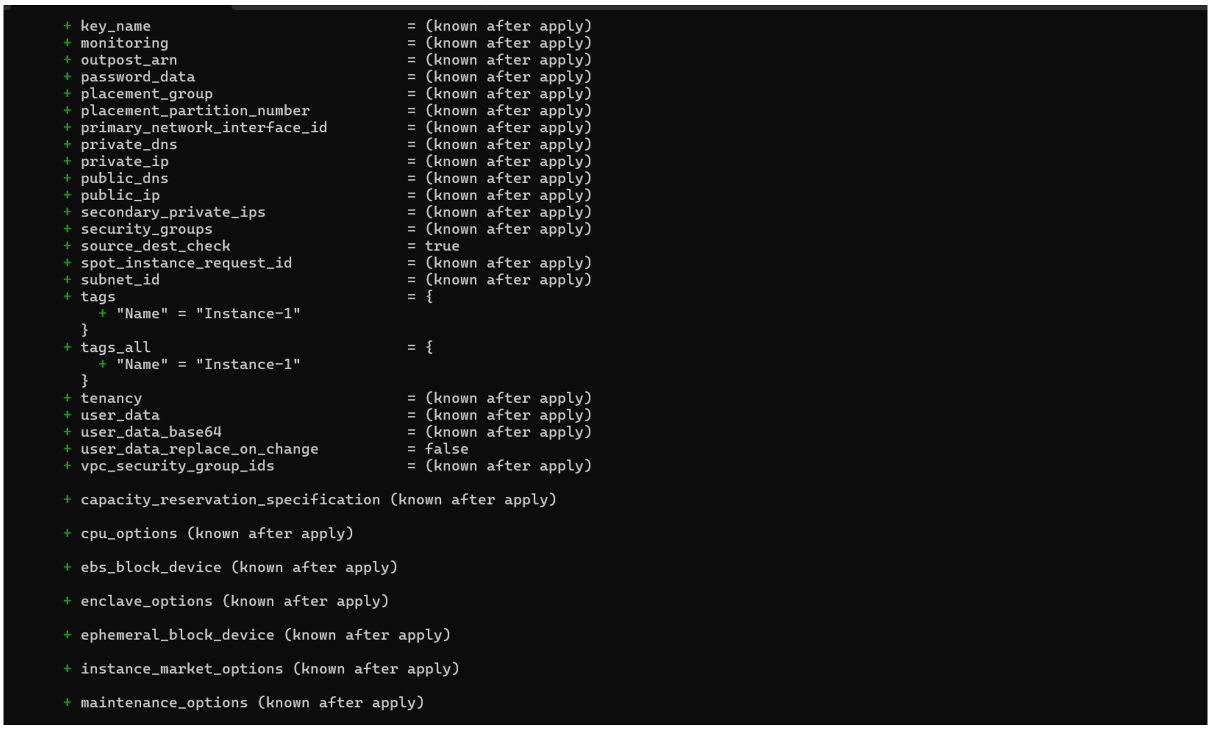
ROLL NO. R2142220269

Step 1: Creating Two T2 Micro EC2 Instances

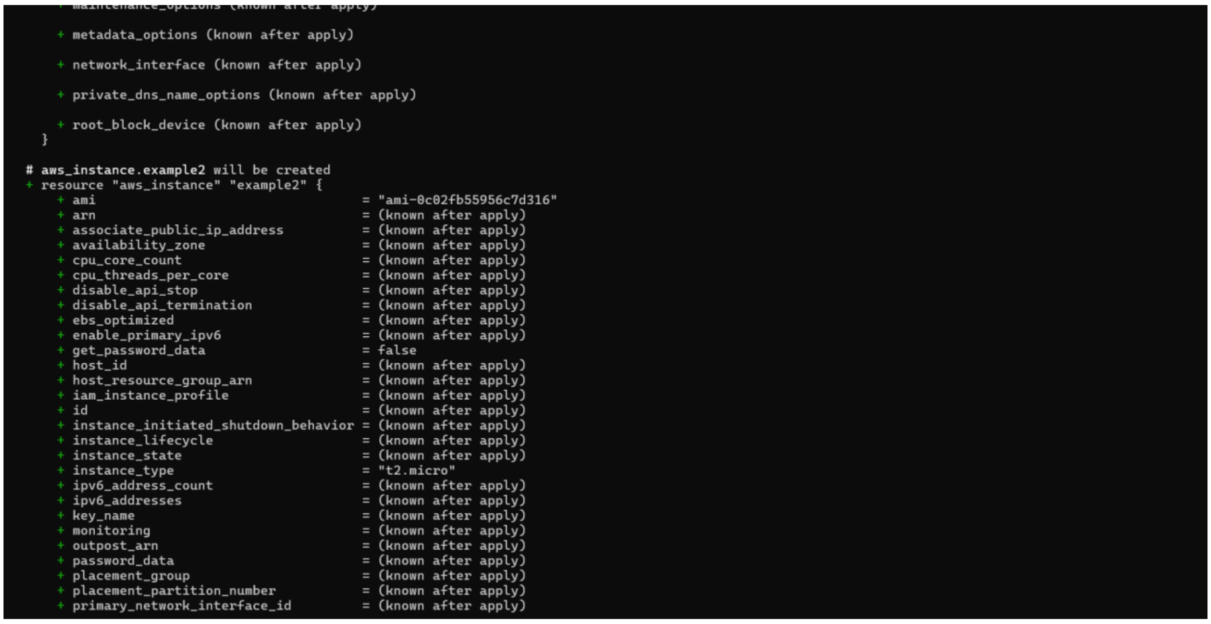
We created two EC2 instances using Terraform with the t2.micro type and Amazon Linux 2 AMI. Below is the Terraform configuration and the result after running terraform apply.



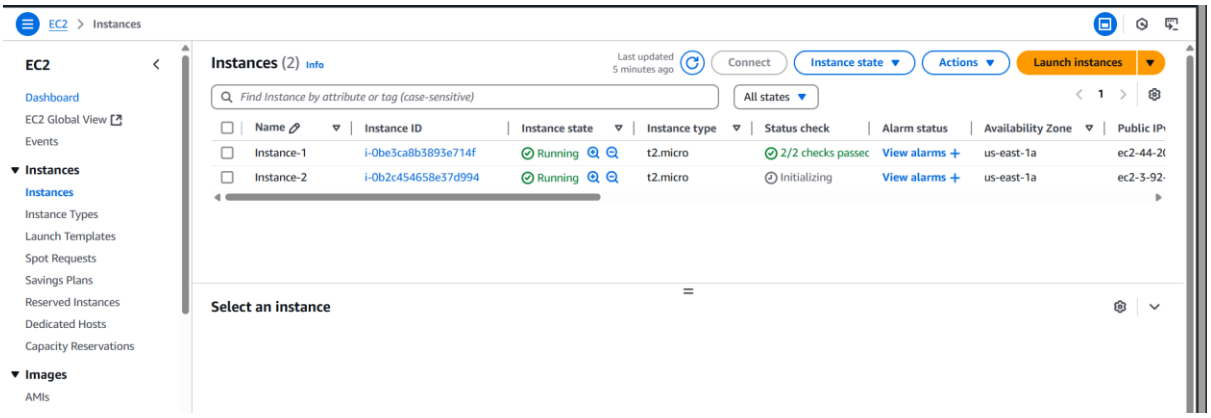
```
C:\Users\OM VATS\terraform-assignment>terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_customer_gateway.customer_gw will be created
  + resource "aws_customer_gateway" "customer_gw" {
      + arn        = (known after apply)
      + bgp_asn    = "65000"
      + id         = (known after apply)
      + ip_address = "203.0.113.12"
      + tags_all   = (known after apply)
      + type       = "ipsec.1"
    }

  # aws_instance.example1 will be created
  + resource "aws_instance" "example1" {
      + ami                                  = "ami-0c02fb55956c7d316"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
```



```
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "Instance-1"
        }
      + tags_all                             = {
          + "Name" = "Instance-1"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)
```



```
      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

  # aws_instance.example2 will be created
  + resource "aws_instance" "example2" {
      + ami                                  = "ami-0c02fb55956c7d316"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
```



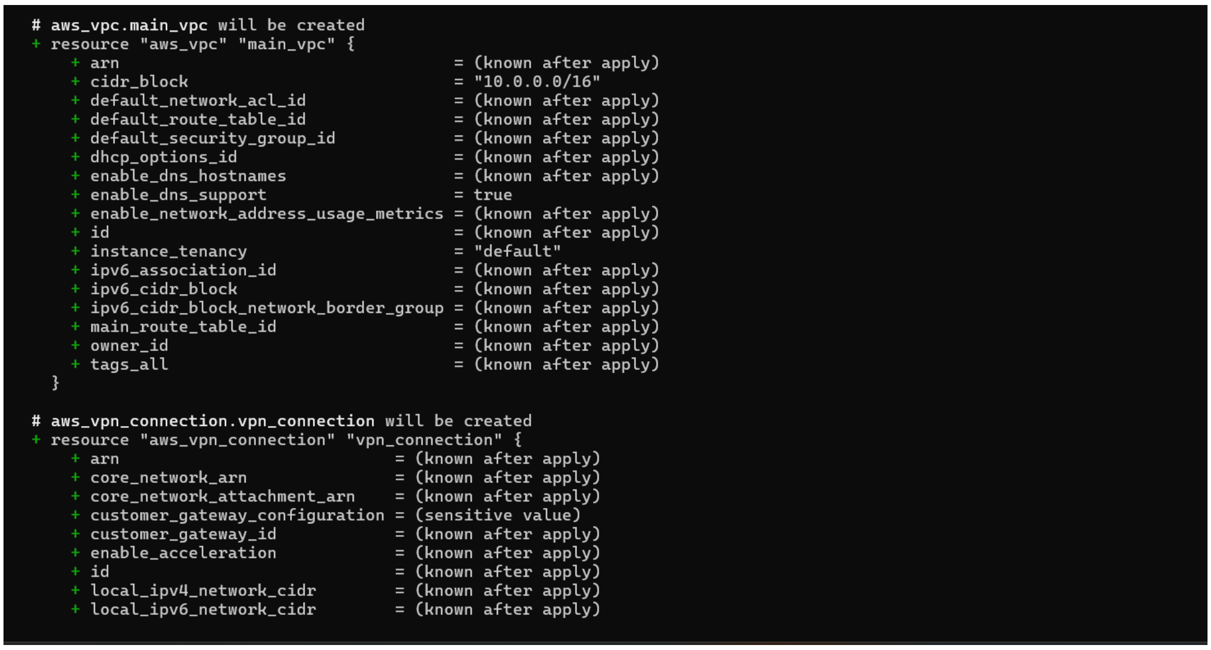
Instances (2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☐	Instance-1	i-0be3ca8b3893e714f	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-44-2(
☐	Instance-2	i-0b2c454658e37d994	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-3-92-

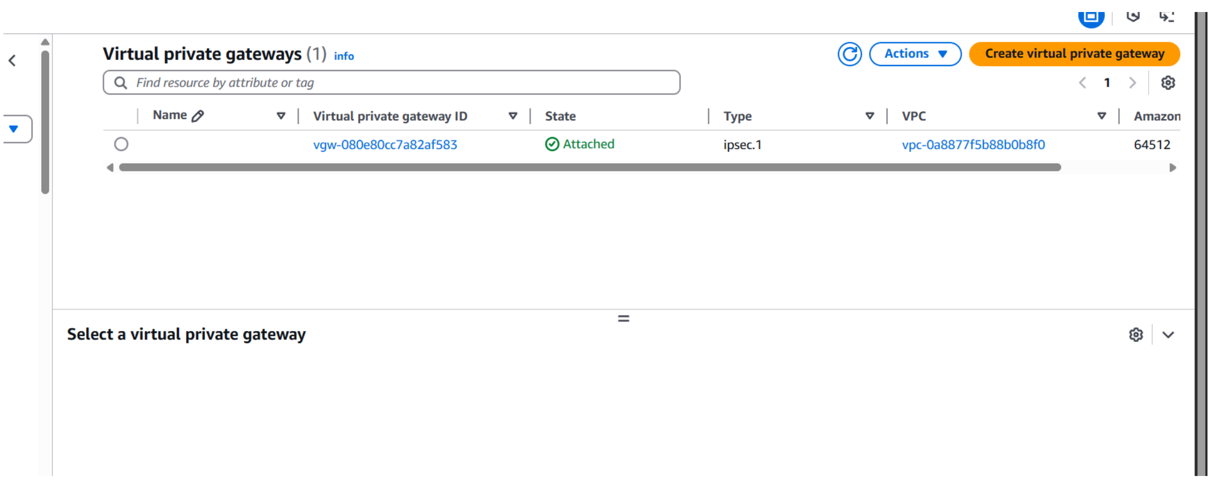
Select an instance

Step 2: Setting Up VPN on AWS

We set up a Virtual Private Gateway (VGW), a Customer Gateway (CGW with dummy IP), and a VPN Connection using Terraform. These simulate a secure VPN tunnel between AWS and an external network.



```
  # aws_vpc.main_vpc will be created
  + resource "aws_vpc" "main_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

  # aws_vpn_connection.vpn_connection will be created
  + resource "aws_vpn_connection" "vpn_connection" {
      + arn                            = (known after apply)
      + core_network_arn               = (known after apply)
      + core_network_attachment_arn    = (known after apply)
      + customer_gateway_configuration = (sensitive value)
      + customer_gateway_id            = (known after apply)
      + enable_acceleration            = (known after apply)
      + id                             = (known after apply)
      + local_ipv4_network_cidr        = (known after apply)
      + local_ipv6_network_cidr        = (known after apply)
```

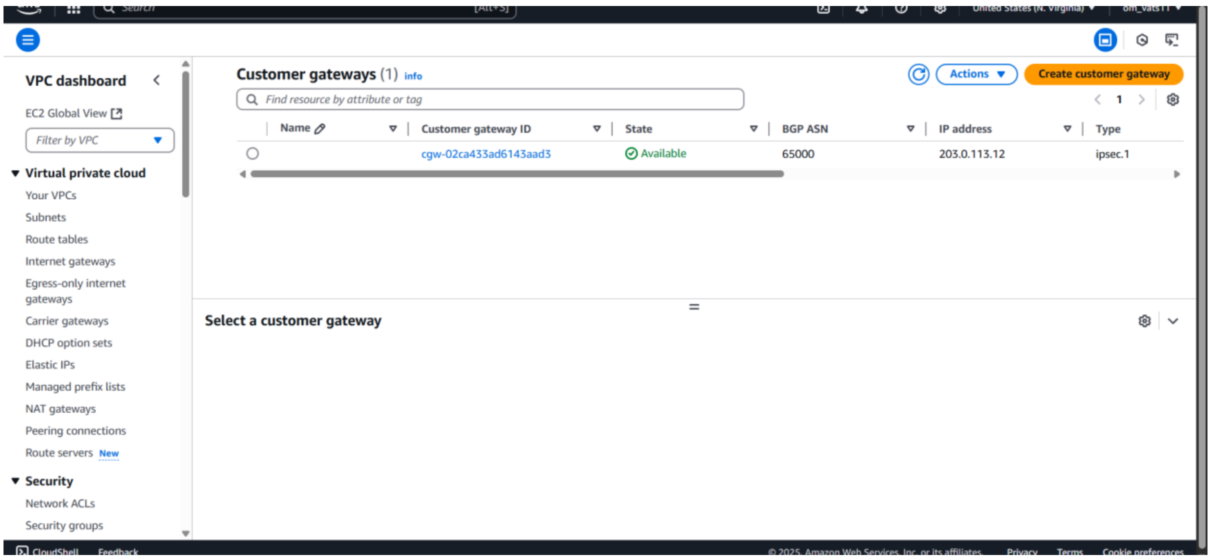


Actions
Create virtual private gateway
Find resource by attribute or tag

Virtual private gateways (1) Info

	Name	Virtual private gateway ID	State	Type	VPC	Amazon
○		vgw-080e80cc7a82af583	Attached	ipsec.1	vpc-0a8877f5b88b0b8f0	64512

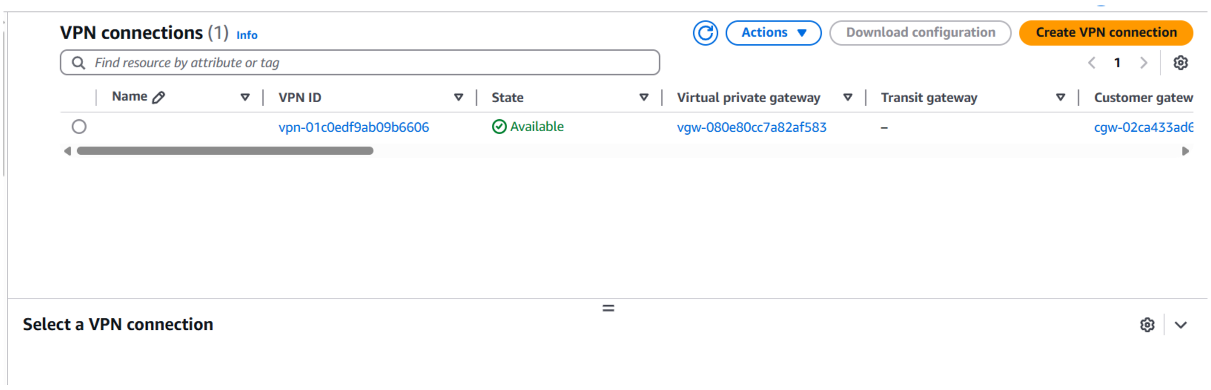
Select a virtual private gateway



Customer gateways (1) Info

Name	Customer gateway ID	State	BGP ASN	IP address	Type
	cgw-02ca433ad6143aad3	Available	65000	203.0.113.12	ipsec.1

Select a customer gateway



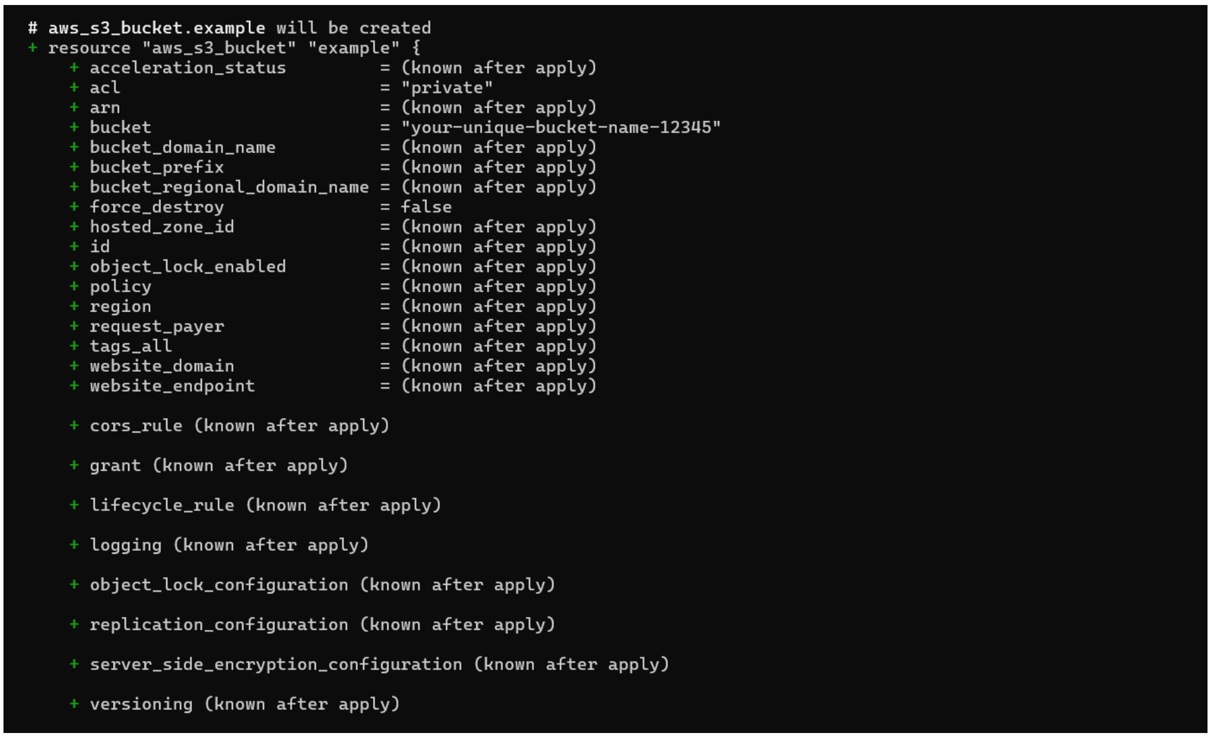
VPN connections (1) Info

Name	VPN ID	State	Virtual private gateway	Transit gateway	Customer gatew
	vpn-01c0edf9ab09b6606	Available	vgw-080e80cc7a82af583	–	cgw-02ca433ad6

Select a VPN connection

Step 3: Creating an S3 Bucket

An S3 bucket was created using Terraform. We specified a unique bucket name and set it as private using aws_s3_bucket resource.



```
# aws_s3_bucket.example will be created
+ resource "aws_s3_bucket" "example" {
    + acceleration_status         = (known after apply)
    + acl                         = "private"
    + arn                         = (known after apply)
    + bucket                      = "your-unique-bucket-name-12345"
    + bucket_domain_name          = (known after apply)
    + bucket_prefix               = (known after apply)
    + bucket_regional_domain_name = (known after apply)
    + force_destroy               = false
    + hosted_zone_id              = (known after apply)
    + id                          = (known after apply)
    + object_lock_enabled         = (known after apply)
    + policy                      = (known after apply)
    + region                      = (known after apply)
    + request_payer               = (known after apply)
    + tags_all                    = (known after apply)
    + website_domain              = (known after apply)
    + website_endpoint            = (known after apply)

    + cors_rule (known after apply)

    + grant (known after apply)

    + lifecycle_rule (known after apply)

    + logging (known after apply)

    + object_lock_configuration (known after apply)

    + replication_configuration (known after apply)

    + server_side_encryption_configuration (known after apply)

    + versioning (known after apply)
```