School of Computer Science

University of Petroleum and Energy Studies

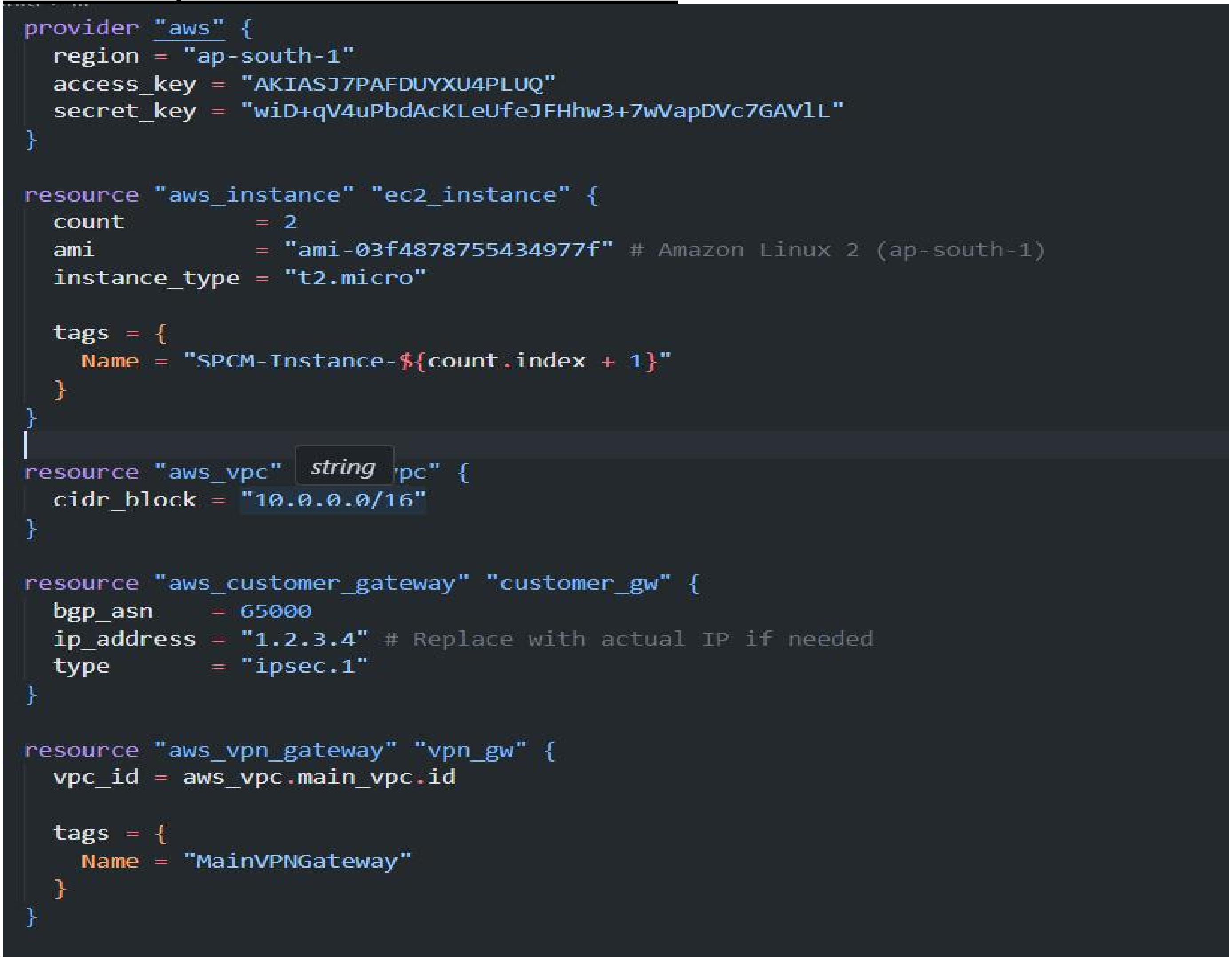
System Provisioning and Configuration Management Assignment 1

Submitted By: **Akshat Pandey** **500101788** **R2142220306** **DevOps B1**	Submitted To: **Dr. Hitesh Kumar Sharma**

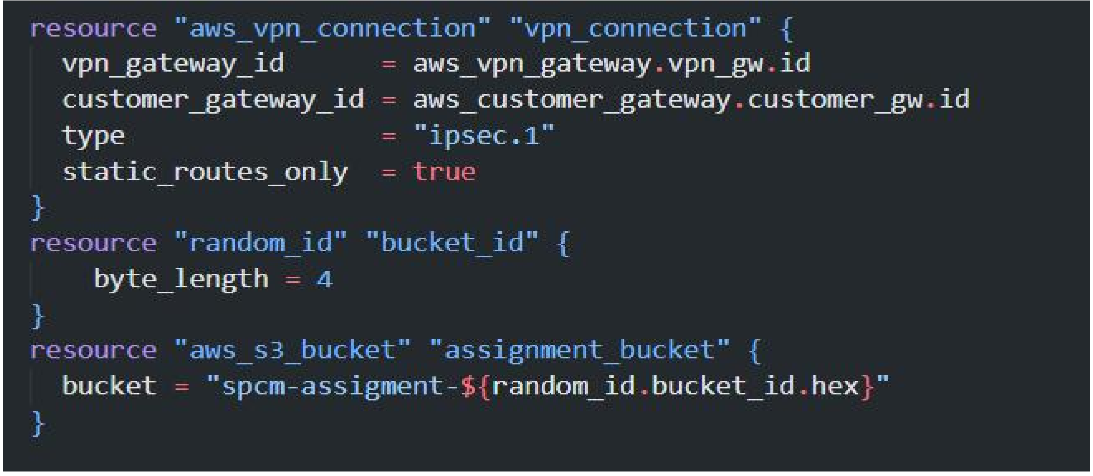
ASSIGNMENT 1

Write Terraform script to do perform following tasks on AWS cloud Platform:-
Step 1: Create two T2 Micro EC2 Instances.
Step 2: Create a VPN on AWS
Step 3: Create a S3 Bucket
Step 4: Write the code for step 1,2 and 3 in a IaC terraform file and run terraform commands to execute these steps.

Main.tf Script to Perform All the Above Task :-



```
provider "aws" {
  region = "ap-south-1"
  access_key = "AKIASJ7PAFDUYXU4PLUQ"
  secret_key = "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"
}

resource "aws_instance" "ec2_instance" {
  count         = 2
  ami           = "ami-03f4878755434977f" # Amazon Linux 2 (ap-south-1)
  instance_type = "t2.micro"

  tags = {
    Name = "SPCM-Instance-${count.index + 1}"
  }
}

resource "aws_vpc" string pc" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_customer_gateway" "customer_gw" {
  bgp_asn    = 65000
  ip_address = "1.2.3.4" # Replace with actual IP if needed
  type       = "ipsec.1"
}

resource "aws_vpn_gateway" "vpn_gw" {
  vpc_id = aws_vpc.main_vpc.id

  tags = {
    Name = "MainVPNGateway"
  }
}
```



```
resource "aws_vpn_connection" "vpn_connection" {
  vpn_gateway_id      = aws_vpn_gateway.vpn_gw.id
  customer_gateway_id = aws_customer_gateway.customer_gw.id
  type                = "ipsec.1"
  static_routes_only  = true
}
resource "random_id" "bucket_id" {
    byte_length = 4
}
resource "aws_s3_bucket" "assignment_bucket" {
  bucket = "spcm-assigment-${random_id.bucket_id.hex}"
}
```

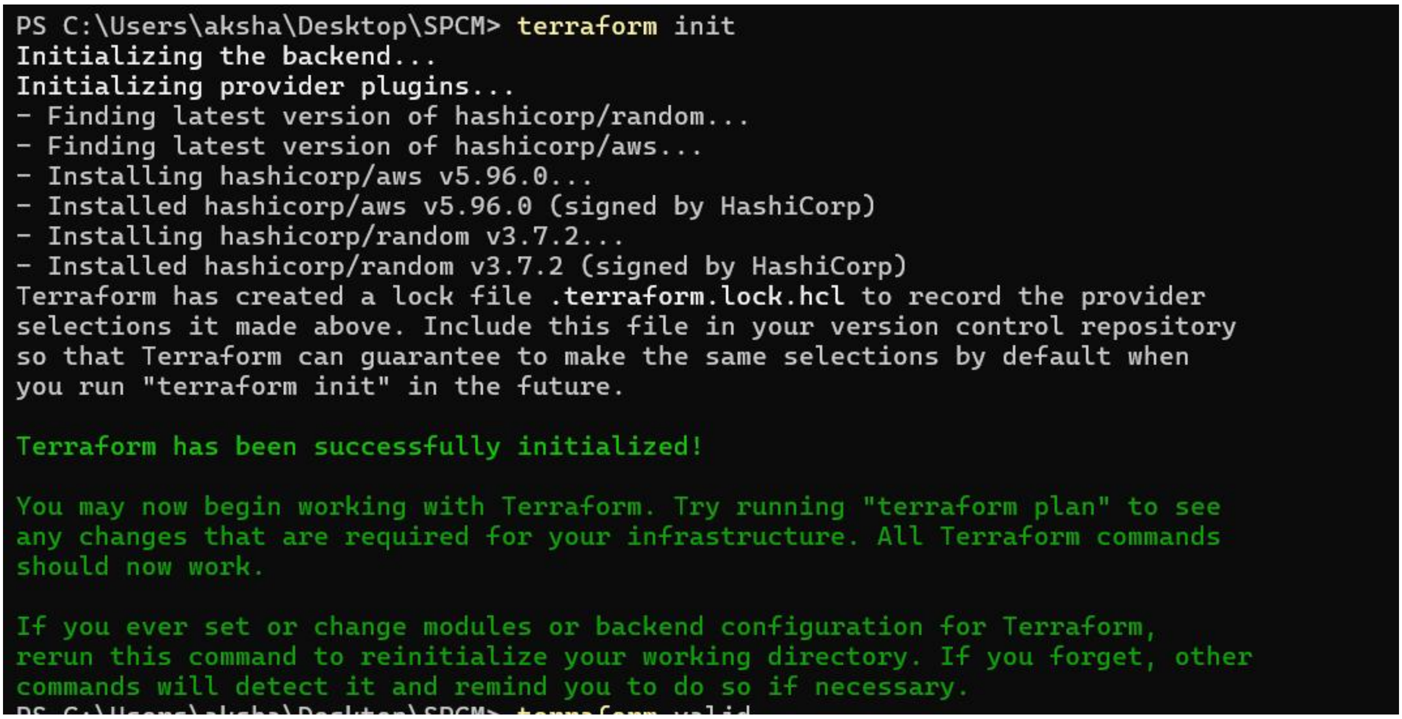
This Main.tf File Does :

- Connects to **AWS** in the **ap-south-1 (Mumbai)** region using provided credentials.
- Launches **2 EC2 instances** of type t2.micro using an Amazon Linux 2 AMI.
- Creates a **VPC** with a 10.0.0.0/16 CIDR block.
- Sets up a **Customer Gateway** and a **VPN Gateway**.
- Establishes a **VPN Connection** between the gateways using ipsec.1.
- Generates a random ID and creates a unique **S3 bucket** named spcm-assigment-<random-id>

Now Applying This File Using Terraform Command;-

1. Terraform Init

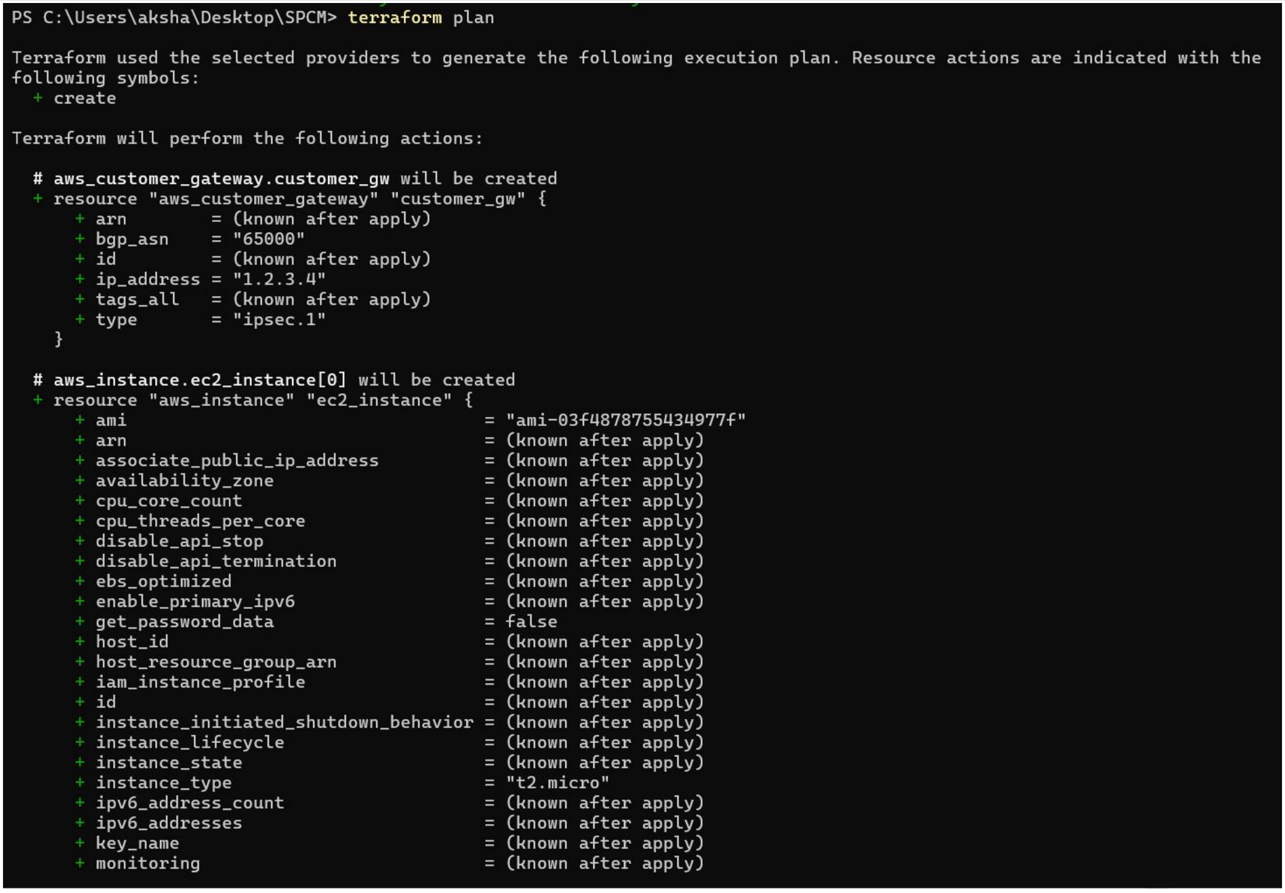
The terraform init command initializes the working directory containing Terraform configuration. It downloads the AWS provider plugin and sets up the backend for storing Terraform state. In the context of the above script, it prepares Terraform to create AWS resources like EC2, VPC, and VPN.



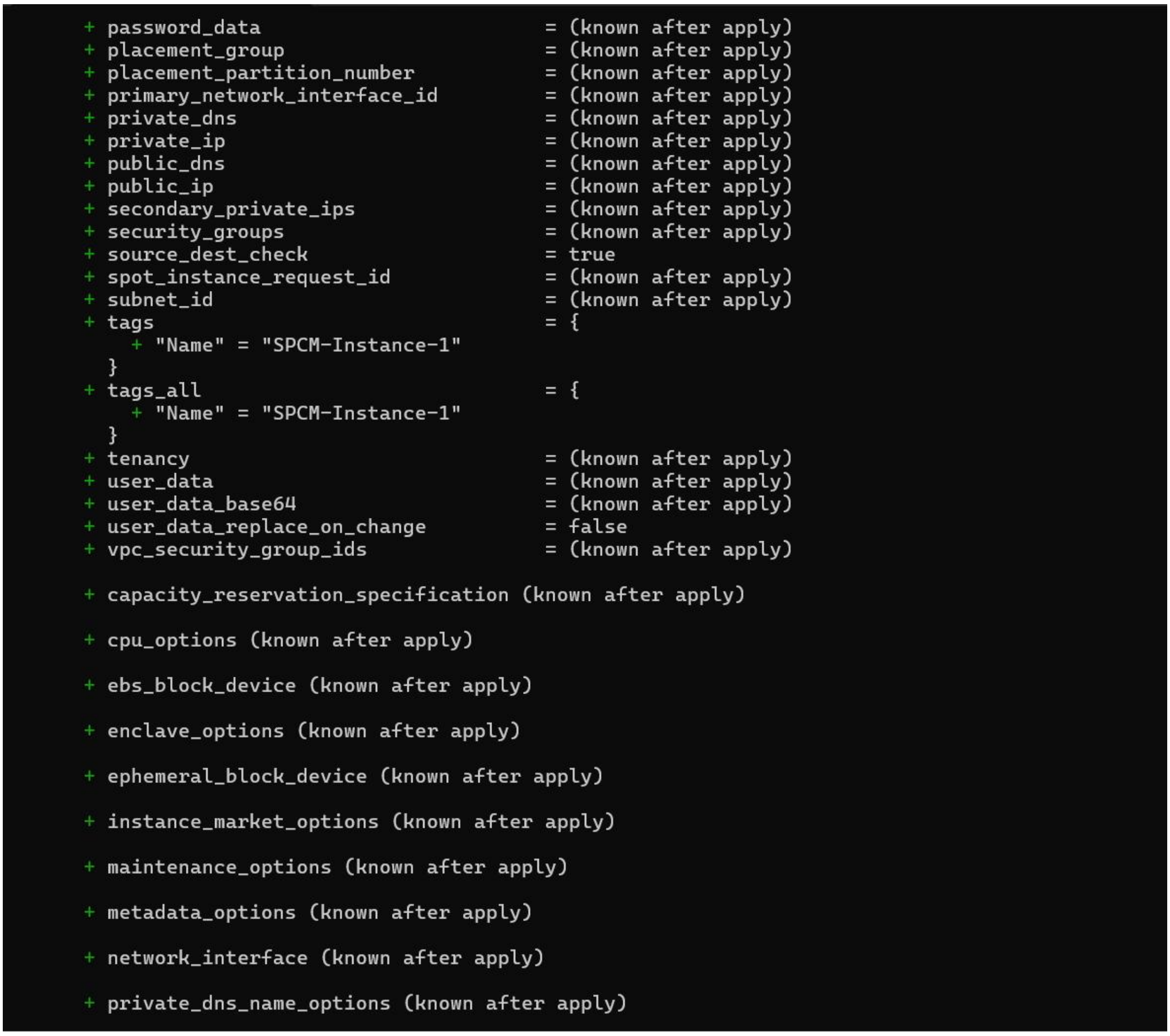
```
PS C:\Users\aksha\Desktop\SPCM> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/random...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.96.0...
- Installed hashicorp/aws v5.96.0 (signed by HashiCorp)
- Installing hashicorp/random v3.7.2...
- Installed hashicorp/random v3.7.2 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\aksha\Desktop\SPCM> terraform valid
```

2. Terraform Plan

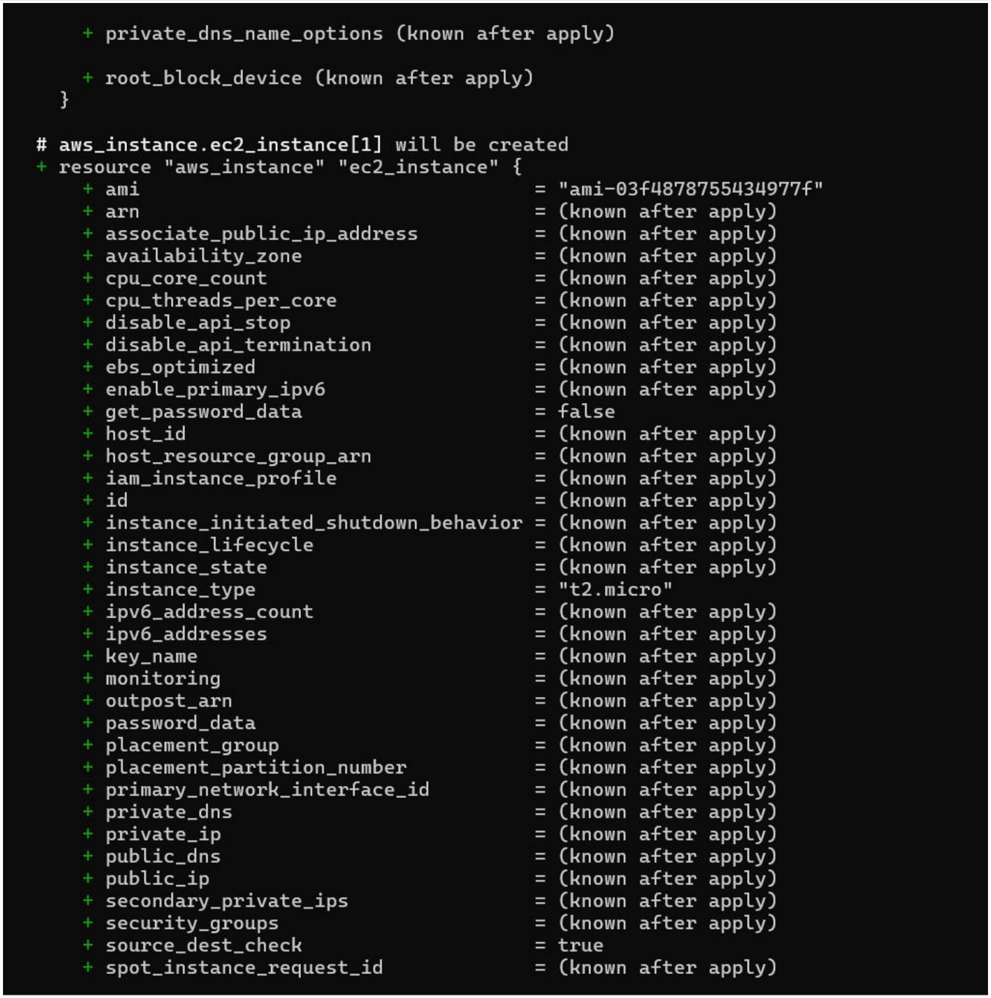
The terraform plan command shows the execution plan for the current configuration. It checks my script against the existing AWS infrastructure and outlines what will be created, changed, or destroyed. For our main.tf script, it previews the creation of EC2 instances, VPC, VPN setup, and the S3 bucket without making any changes yet.



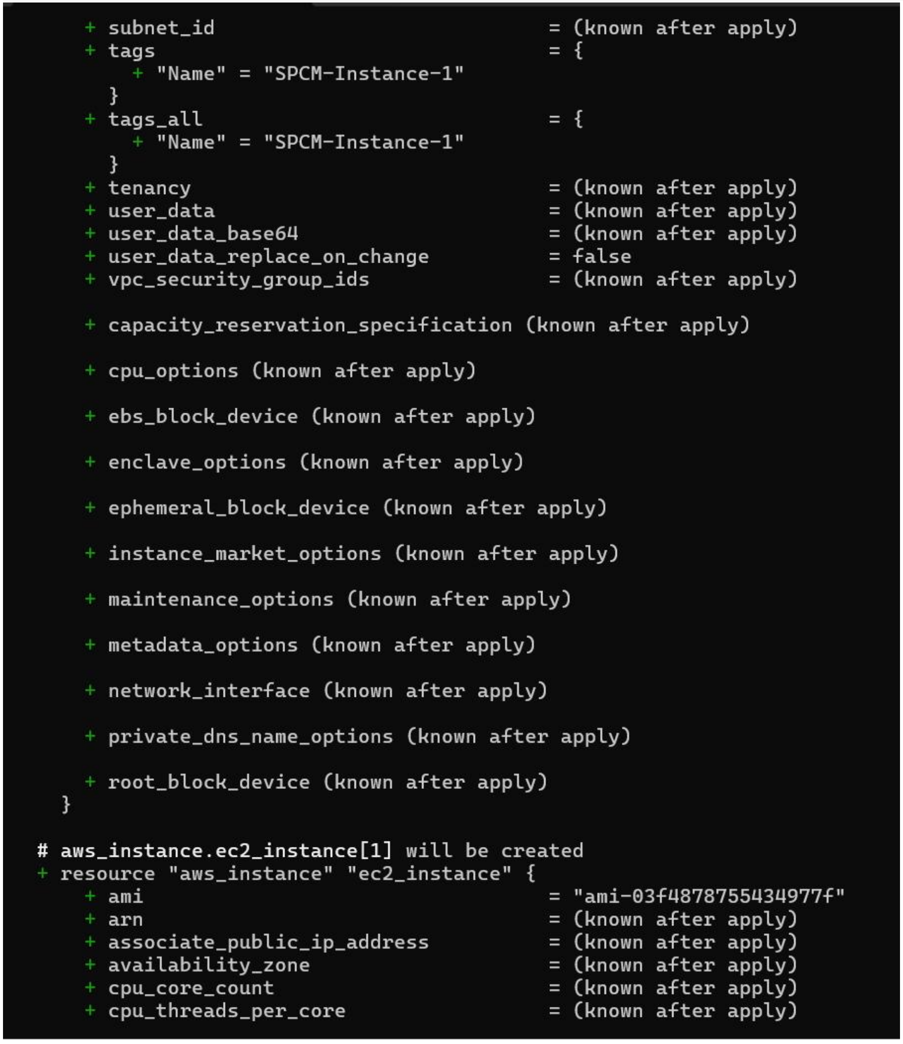
```
PS C:\Users\aksha\Desktop\SPCM> terraform plan

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 = "1.2.3.4"
      + tags_all   = (known after apply)
      + type       = "ipsec.1"
    }

  # aws_instance.ec2_instance[0] will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
```



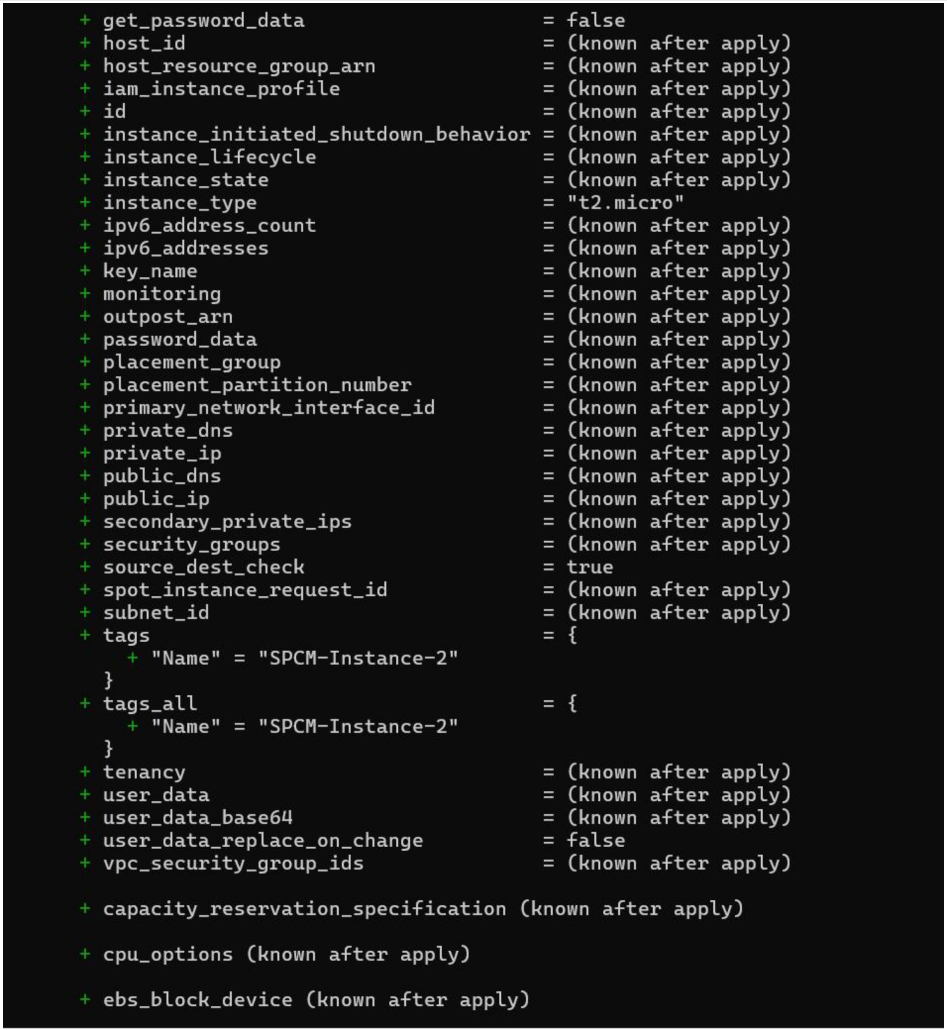
```
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (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" = "SPCM-Instance-1"
        }
      + tags_all                             = {
          + "Name" = "SPCM-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)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)
```



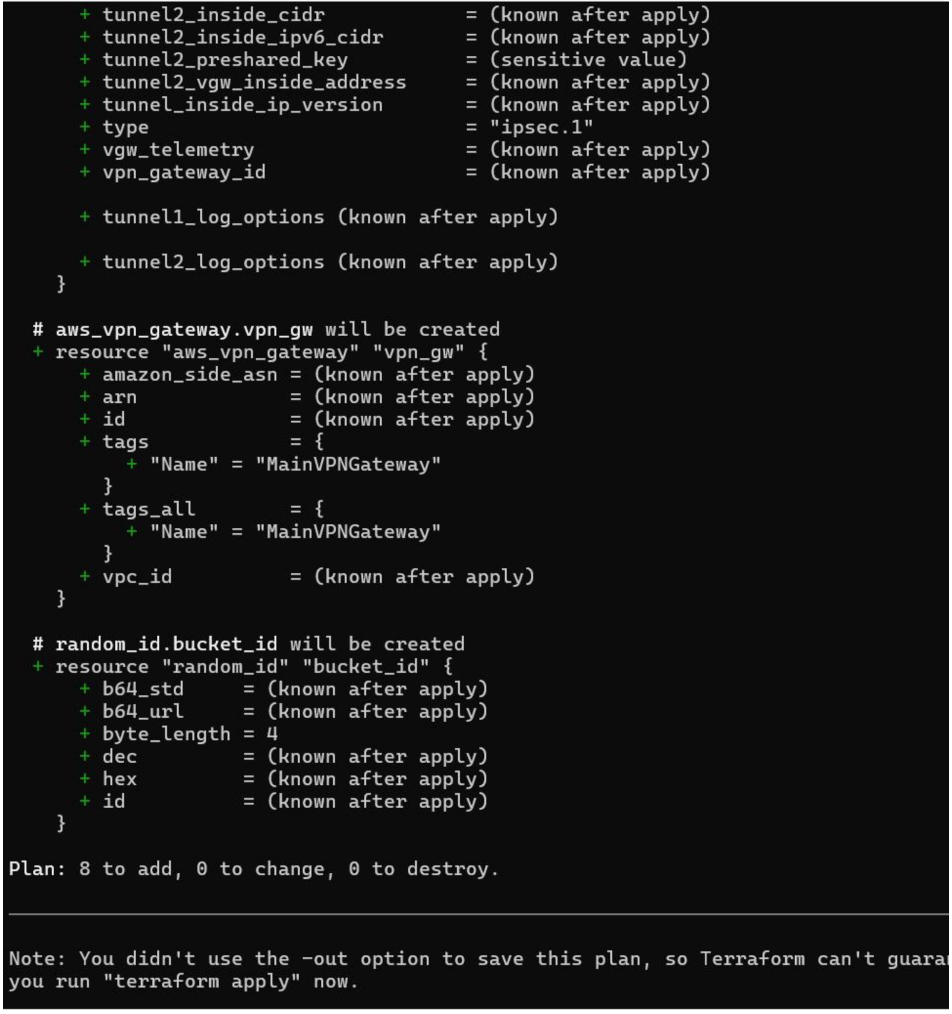
```
      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

  # aws_instance.ec2_instance[1] will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
      + 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" = "SPCM-Instance-1"
        }
      + tags_all                             = {
          + "Name" = "SPCM-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)

      + 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.ec2_instance[1] will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
```



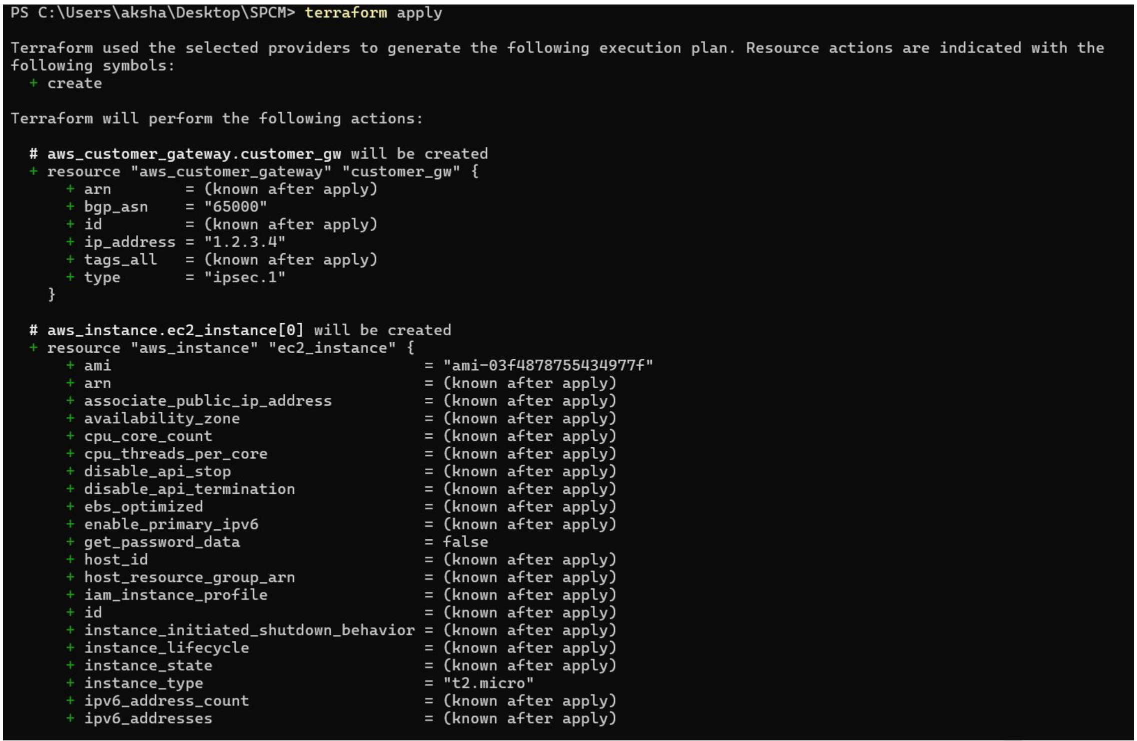
```
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (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)
      + 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" = "SPCM-Instance-2"
        }
      + tags_all                             = {
          + "Name" = "SPCM-Instance-2"
        }
      + 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)
```



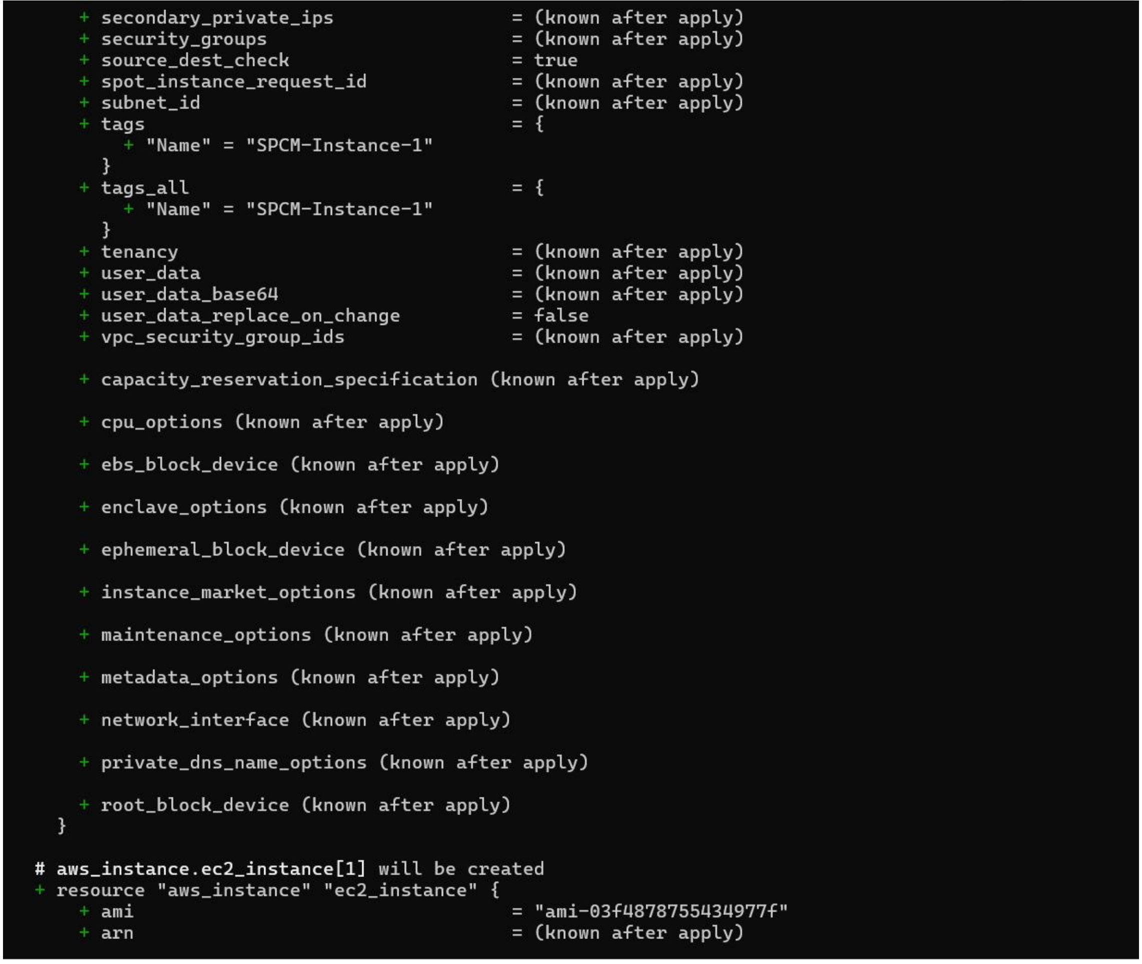
```
      + tunnel2_inside_cidr            = (known after apply)
      + tunnel2_inside_ipv6_cidr       = (known after apply)
      + tunnel2_preshared_key          = (sensitive value)
      + tunnel2_vgw_inside_address     = (known after apply)
      + tunnel_inside_ip_version       = (known after apply)
      + type                           = "ipsec.1"
      + vgw_telemetry                  = (known after apply)
      + vpn_gateway_id                 = (known after apply)

      + tunnel1_log_options (known after apply)

      + tunnel2_log_options (known after apply)
    }

  # aws_vpn_gateway.vpn_gw will be created
  + resource "aws_vpn_gateway" "vpn_gw" {
      + amazon_side_asn = (known after apply)
      + arn             = (known after apply)
      + id              = (known after apply)
      + tags            = {
          + "Name" = "MainVPNGateway"
        }
      + tags_all        = {
          + "Name" = "MainVPNGateway"
        }
      + vpc_id          = (known after apply)
    }

  # random_id.bucket_id will be created
  + resource "random_id" "bucket_id" {
      + b64_std     = (known after apply)
      + b64_url     = (known after apply)
      + byte_length = 4
      + dec         = (known after apply)
      + hex         = (known after apply)
      + id          = (known after apply)
    }

Plan: 8 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guara
you run "terraform apply" now.
```

3. Terraform Apply

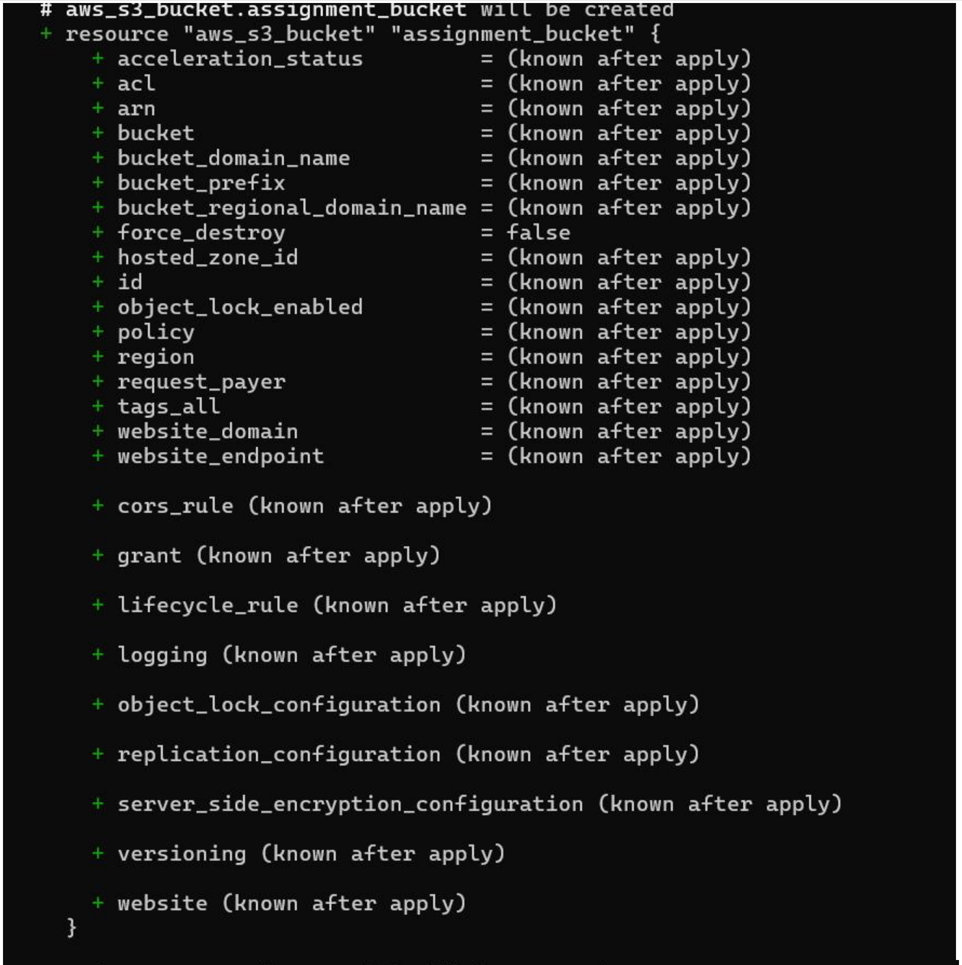
The terraform apply command executes the actions defined in the Terraform plan to provision infrastructure. It creates the specified AWS resources like EC2 instances, VPC, VPN connection, and S3 bucket as defined in the script. It will prompt to confirm before deployment, ensuring changes are intentional.



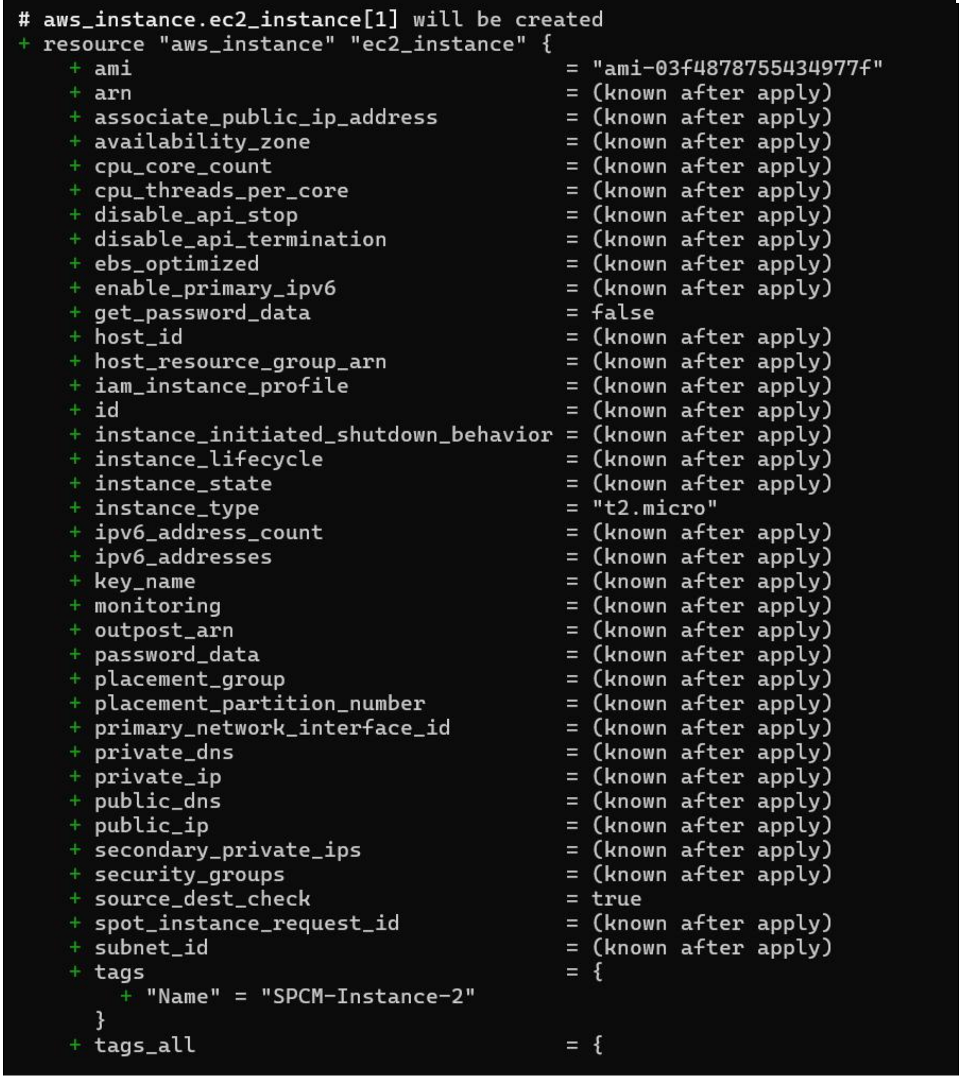
```
PS C:\Users\aksha\Desktop\SPCM> terraform apply

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 = "1.2.3.4"
      + tags_all   = (known after apply)
      + type       = "ipsec.1"
    }

  # aws_instance.ec2_instance[0] will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + 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)
```



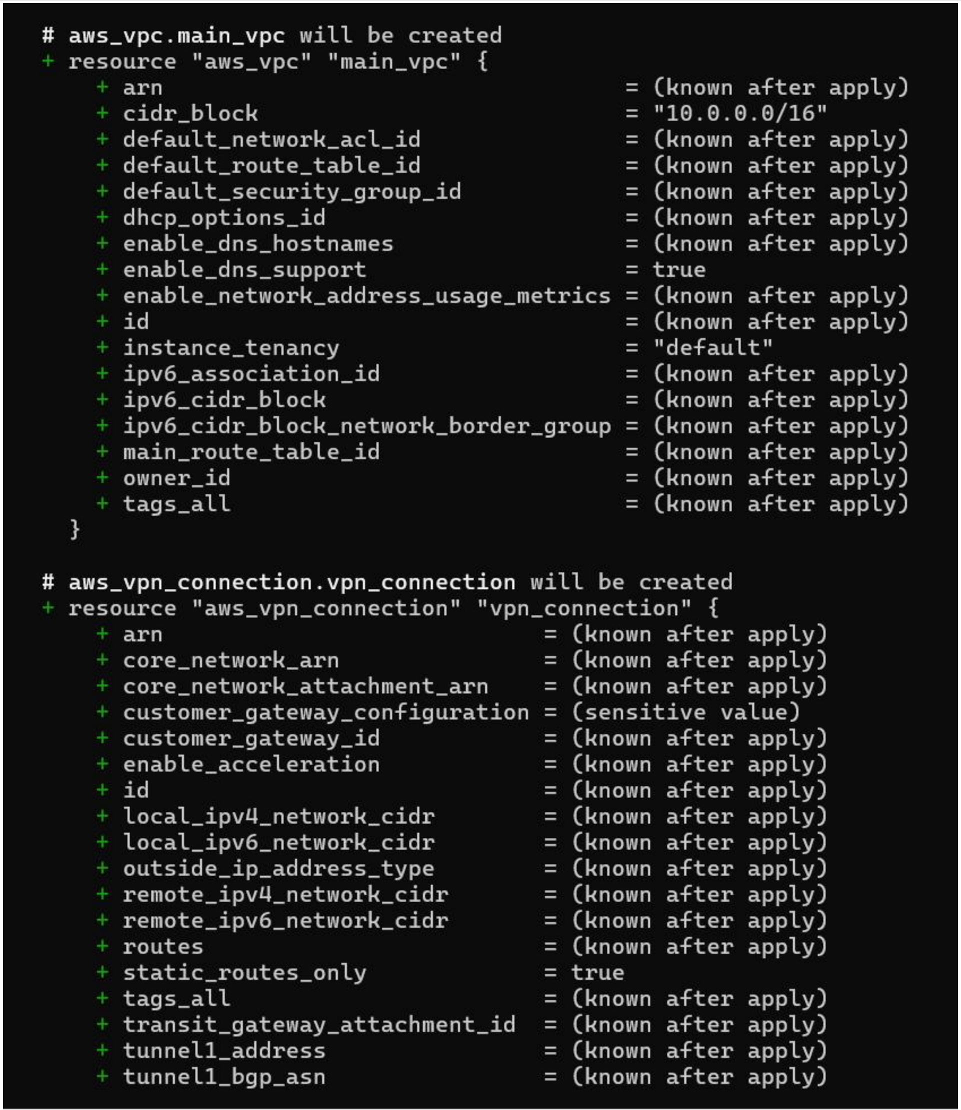
```
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "SPCM-Instance-1"
        }
      + tags_all                             = {
          + "Name" = "SPCM-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)

      + 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.ec2_instance[1] will be created
  + resource "aws_instance" "ec2_instance" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
```



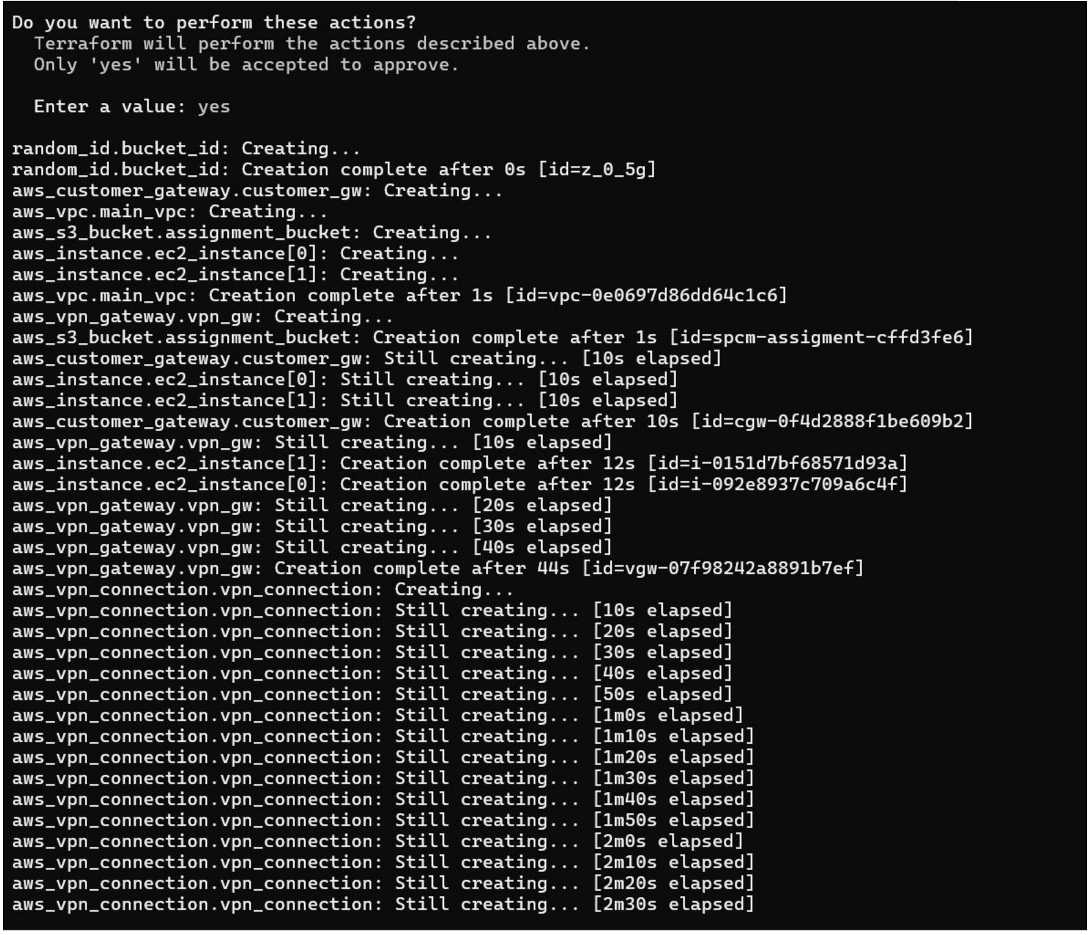
```
  # aws_s3_bucket.assignment_bucket will be created
  + resource "aws_s3_bucket" "assignment_bucket" {
      + acceleration_status         = (known after apply)
      + acl                         = (known after apply)
      + arn                         = (known after apply)
      + bucket                      = (known after apply)
      + 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)

      + website (known after apply)
    }
```



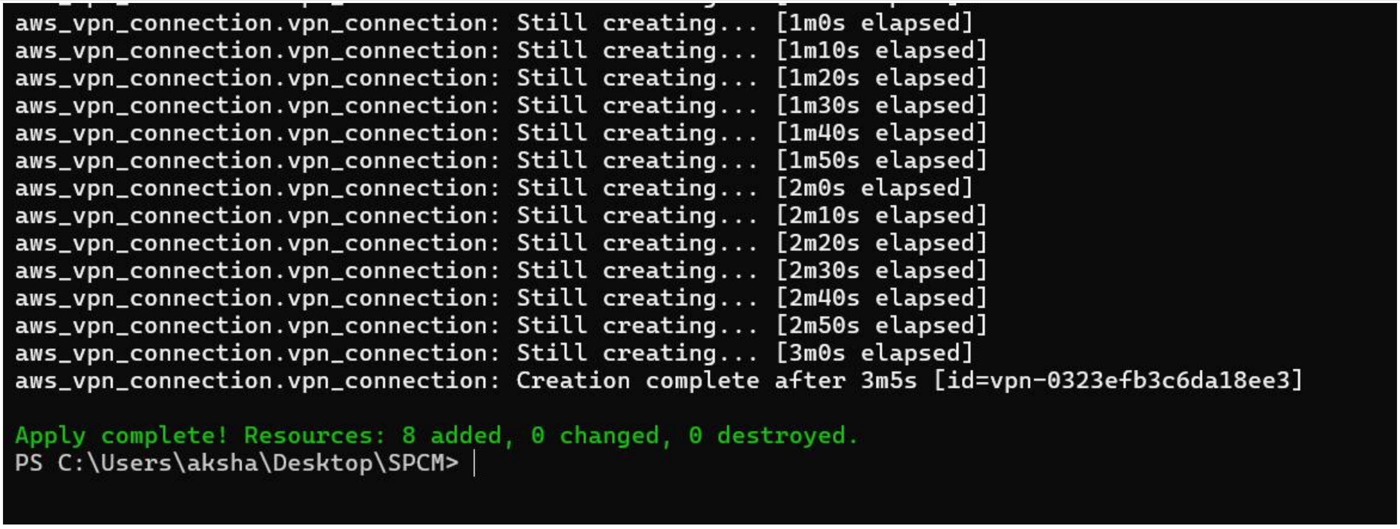
```
# aws_instance.ec2_instance[1] will be created
+ resource "aws_instance" "ec2_instance" {
    + ami                                  = "ami-03f4878755434977f"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (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)
    + 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" = "SPCM-Instance-2"
      }
    + tags_all                             = {
```



```
  # 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)
      + outside_ip_address_type        = (known after apply)
      + remote_ipv4_network_cidr       = (known after apply)
      + remote_ipv6_network_cidr       = (known after apply)
      + routes                         = (known after apply)
      + static_routes_only             = true
      + tags_all                       = (known after apply)
      + transit_gateway_attachment_id  = (known after apply)
      + tunnel1_address                = (known after apply)
      + tunnel1_bgp_asn                = (known after apply)
```



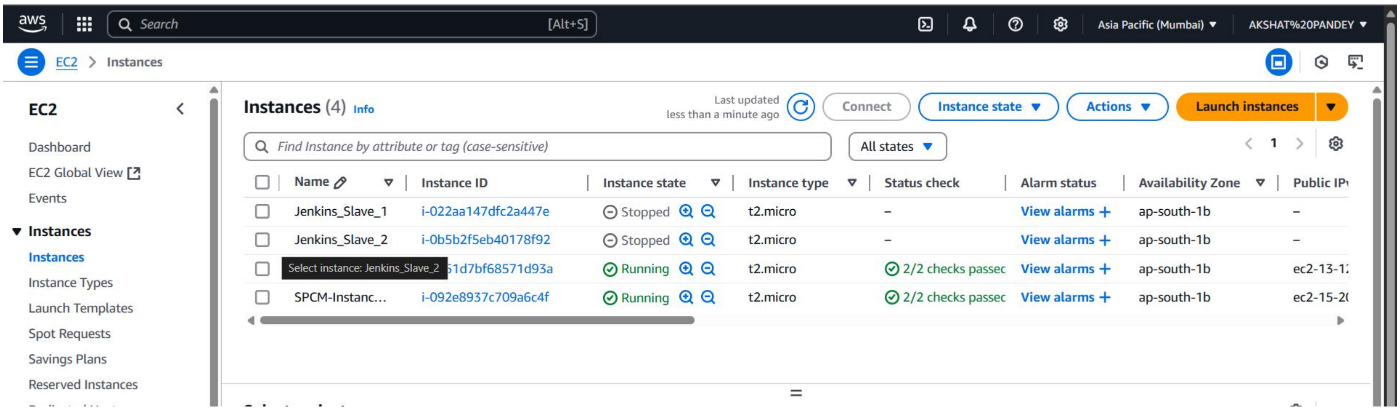
```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

random_id.bucket_id: Creating...
random_id.bucket_id: Creation complete after 0s [id=z_0_5g]
aws_customer_gateway.customer_gw: Creating...
aws_vpc.main_vpc: Creating...
aws_s3_bucket.assignment_bucket: Creating...
aws_instance.ec2_instance[0]: Creating...
aws_instance.ec2_instance[1]: Creating...
aws_vpc.main_vpc: Creation complete after 1s [id=vpc-0e0697d86dd64c1c6]
aws_vpn_gateway.vpn_gw: Creating...
aws_s3_bucket.assignment_bucket: Creation complete after 1s [id=spcm-assigment-cffd3fe6]
aws_customer_gateway.customer_gw: Still creating... [10s elapsed]
aws_instance.ec2_instance[0]: Still creating... [10s elapsed]
aws_instance.ec2_instance[1]: Still creating... [10s elapsed]
aws_customer_gateway.customer_gw: Creation complete after 10s [id=cgw-0f4d2888f1be609b2]
aws_vpn_gateway.vpn_gw: Still creating... [10s elapsed]
aws_instance.ec2_instance[1]: Creation complete after 12s [id=i-0151d7bf68571d93a]
aws_instance.ec2_instance[0]: Creation complete after 12s [id=i-092e8937c709a6c4f]
aws_vpn_gateway.vpn_gw: Still creating... [20s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [30s elapsed]
aws_vpn_gateway.vpn_gw: Still creating... [40s elapsed]
aws_vpn_gateway.vpn_gw: Creation complete after 44s [id=vgw-07f98242a8891b7ef]
aws_vpn_connection.vpn_connection: Creating...
aws_vpn_connection.vpn_connection: Still creating... [10s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [20s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [30s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [40s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [50s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m0s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m10s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m20s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m30s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m40s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m50s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m0s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m10s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m20s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m30s elapsed]
```



```
aws_vpn_connection.vpn_connection: Still creating... [1m0s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m10s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m20s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m30s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m40s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [1m50s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m0s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m10s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m20s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m30s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m40s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [2m50s elapsed]
aws_vpn_connection.vpn_connection: Still creating... [3m0s elapsed]
aws_vpn_connection.vpn_connection: Creation complete after 3m5s [id=vpn-0323efb3c6da18ee3]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
PS C:\Users\aksha\Desktop\SPCM> |
```

Verifying Resources Creating on AWS Console

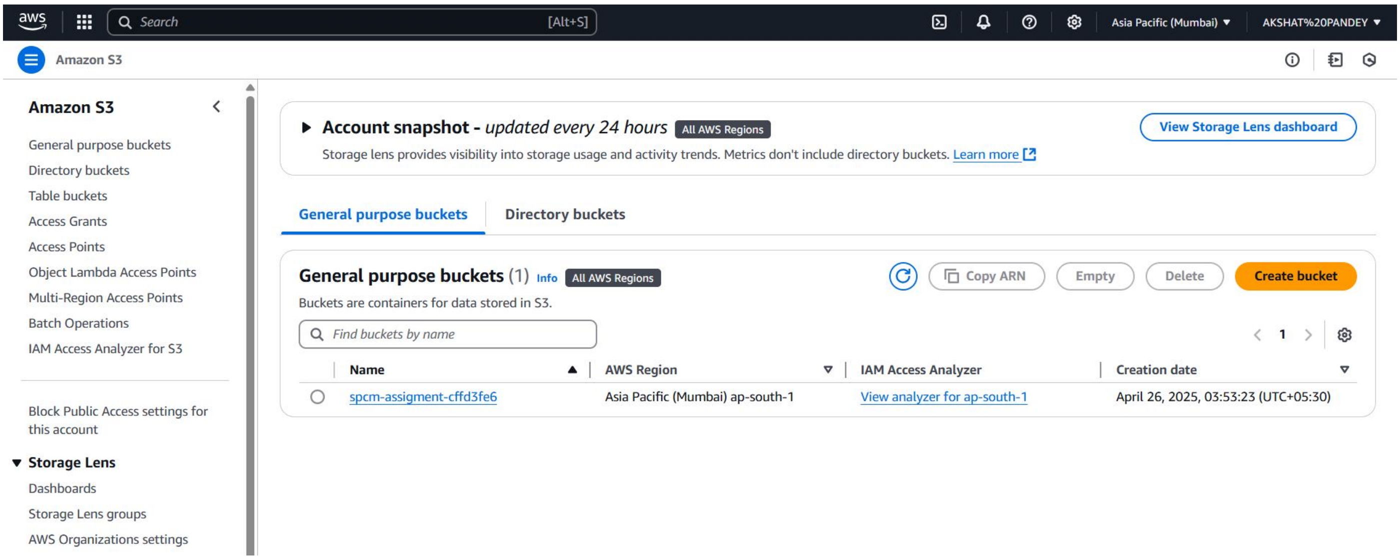
1. EC2 Instance



Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
□	Jenkins_Slave_1	i-022aa147dfc2a447e	Stopped	t2.micro	–	View alarms +	ap-south-1b	–
□	Jenkins_Slave_2	i-0b5b2f5eb40178f92	Stopped	t2.micro	–	View alarms +	ap-south-1b	–
□	Select instance: Jenkins_Slave_2	1d7bf68571d93a	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-1
□	SPCM-Instanc...	i-092e8937c709a6c4f	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-15-2(

2. S3 Bucket



▶ Account snapshot - *updated every 24 hours* All AWS Regions

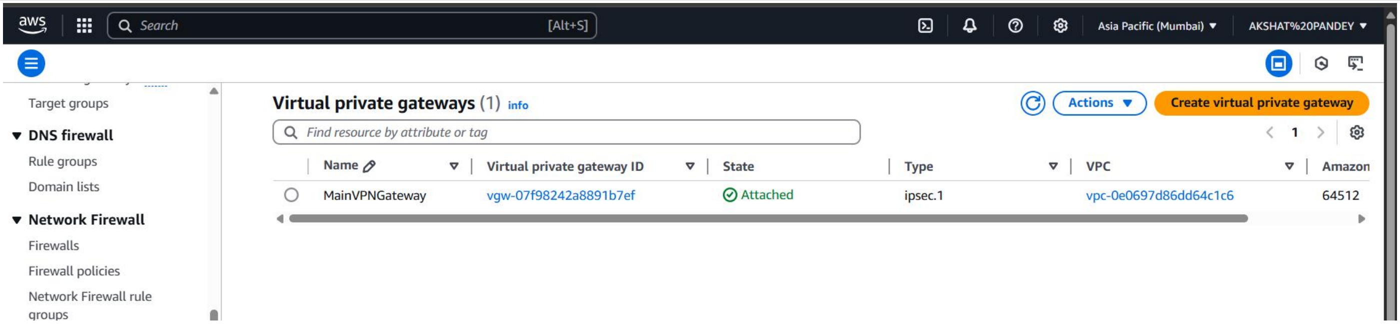
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	spcm-assigment-cffd3fe6	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 26, 2025, 03:53:23 (UTC+05:30)

3. VPN Gateway

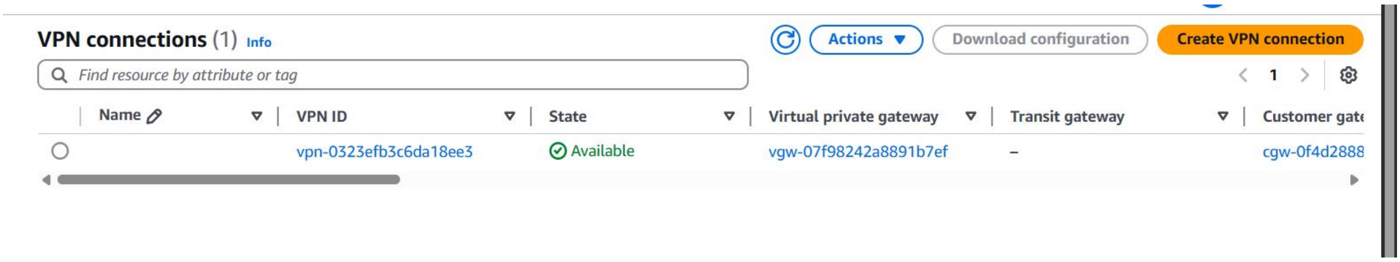


Virtual private gateways (1) info

Find resource by attribute or tag

	Name	Virtual private gateway ID	State	Type	VPC	Amazon
○	MainVPNGateway	vgw-07f98242a8891b7ef	✓ Attached	ipsec.1	vpc-0e0697d86dd64c1c6	64512

4. VPN Connection



VPN connections (1) Info

Find resource by attribute or tag

	Name	VPN ID	State	Virtual private gateway	Transit gateway	Customer gate
○		vpn-0323efb3c6da18ee3	✓ Available	vgw-07f98242a8891b7ef	–	cgw-0f4d2888