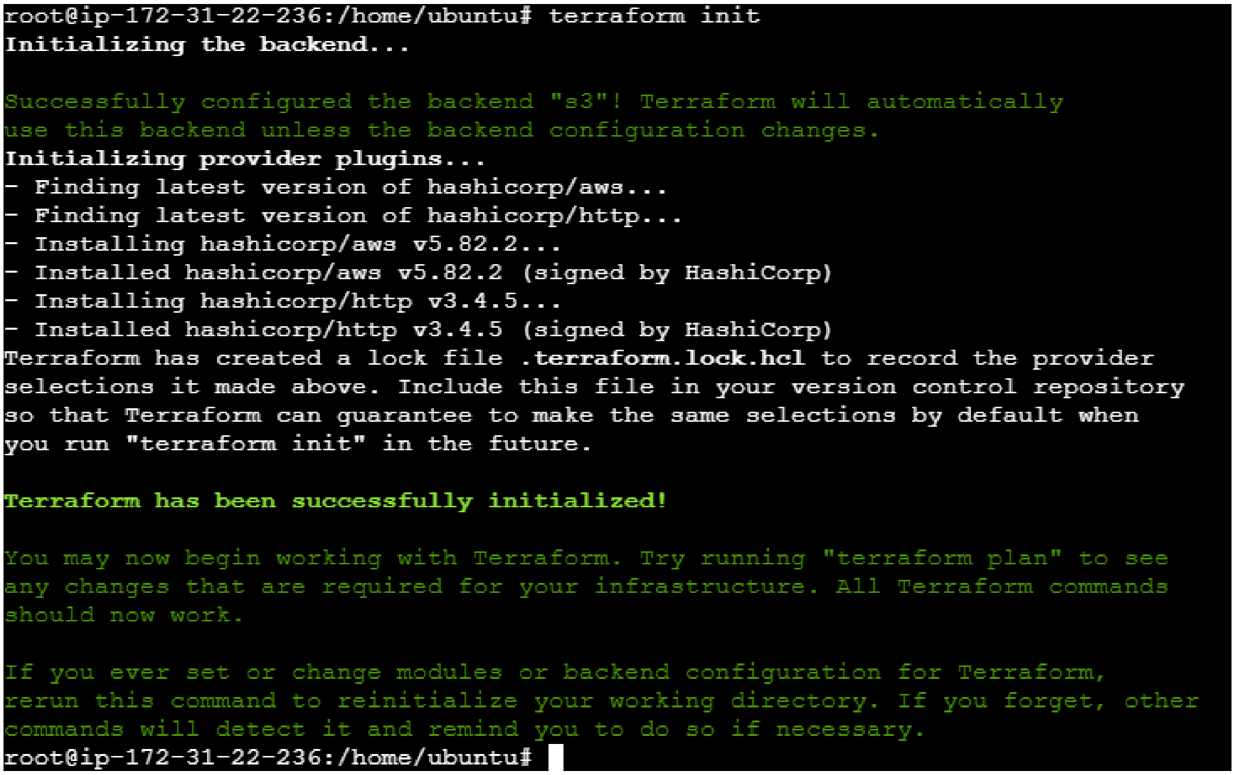
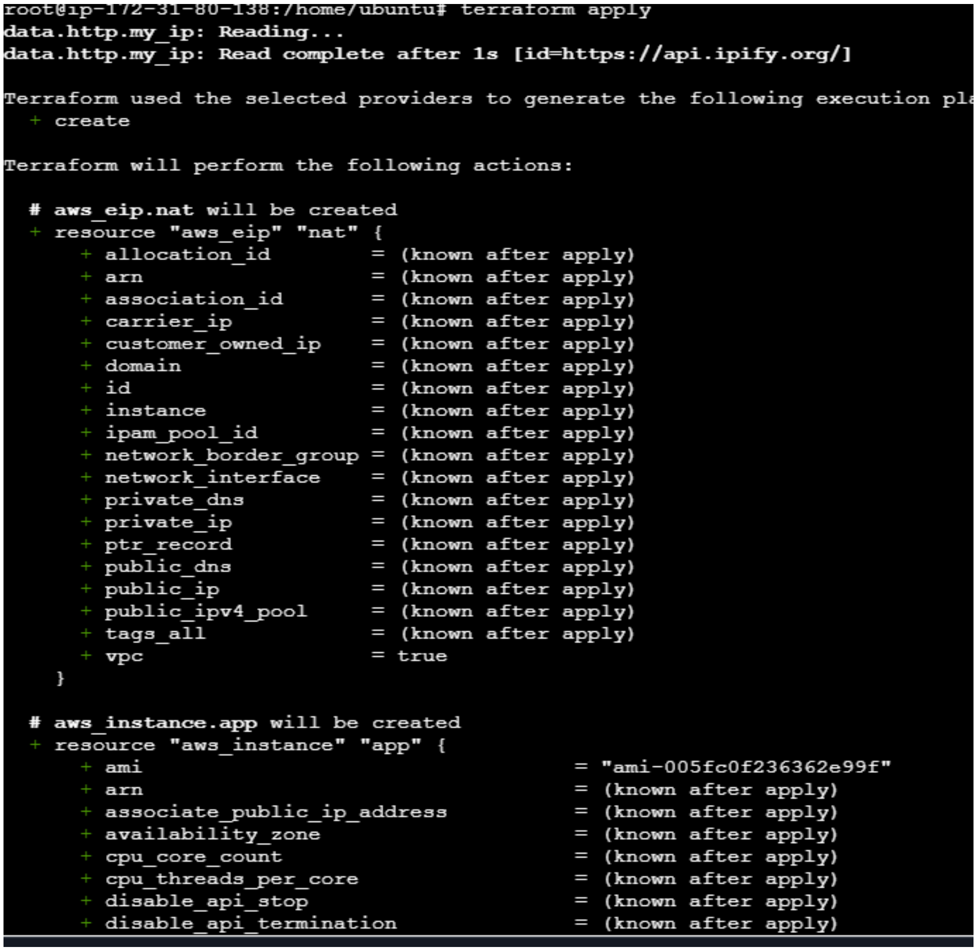
{Task1}

$ terraform init

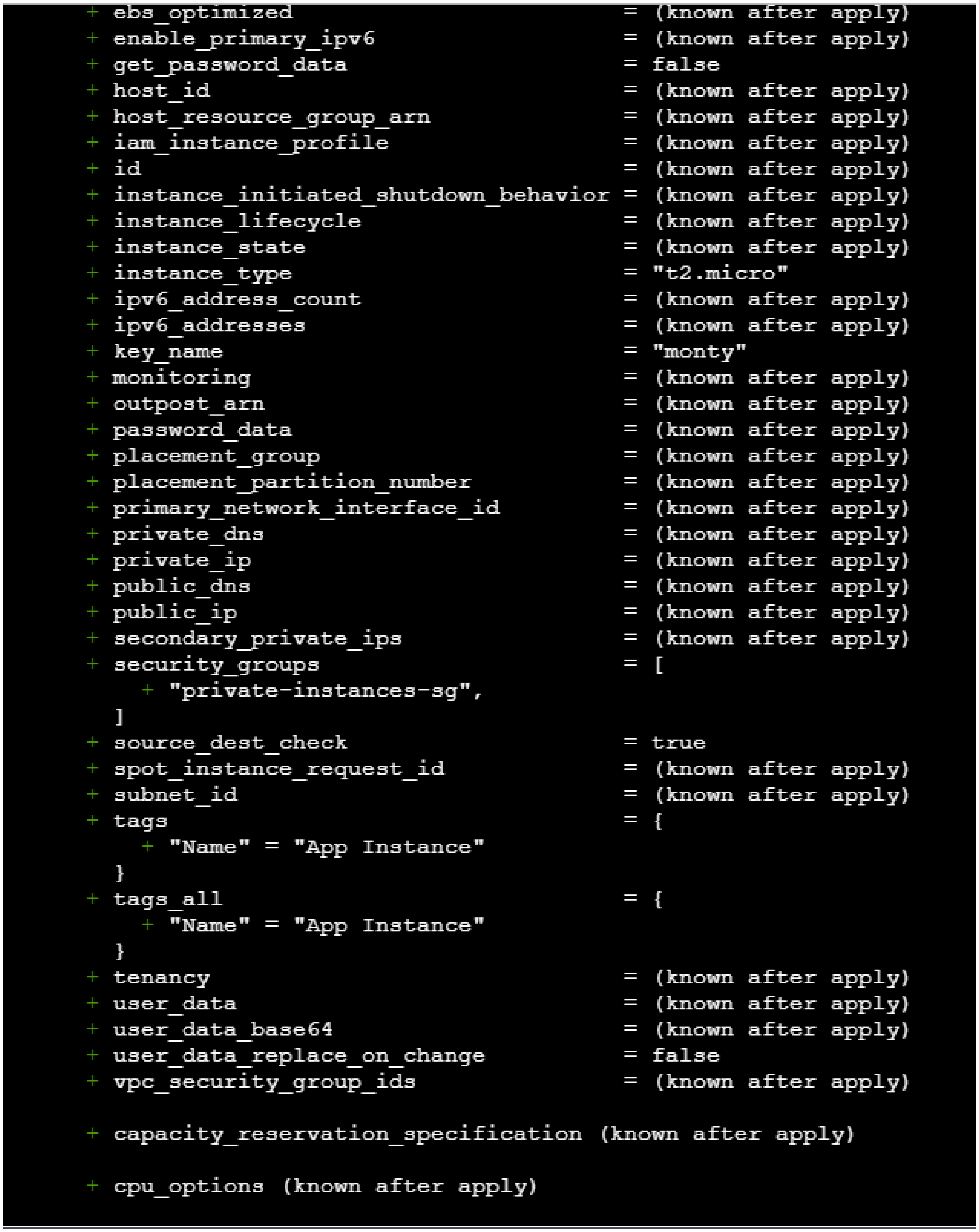


```
root@ip-172-31-22-236:/home/ubuntu# terraform init
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v5.82.2...
- Installed hashicorp/aws v5.82.2 (signed by HashiCorp)
- Installing hashicorp/http v3.4.5...
- Installed hashicorp/http v3.4.5 (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.
root@ip-172-31-22-236:/home/ubuntu#
```

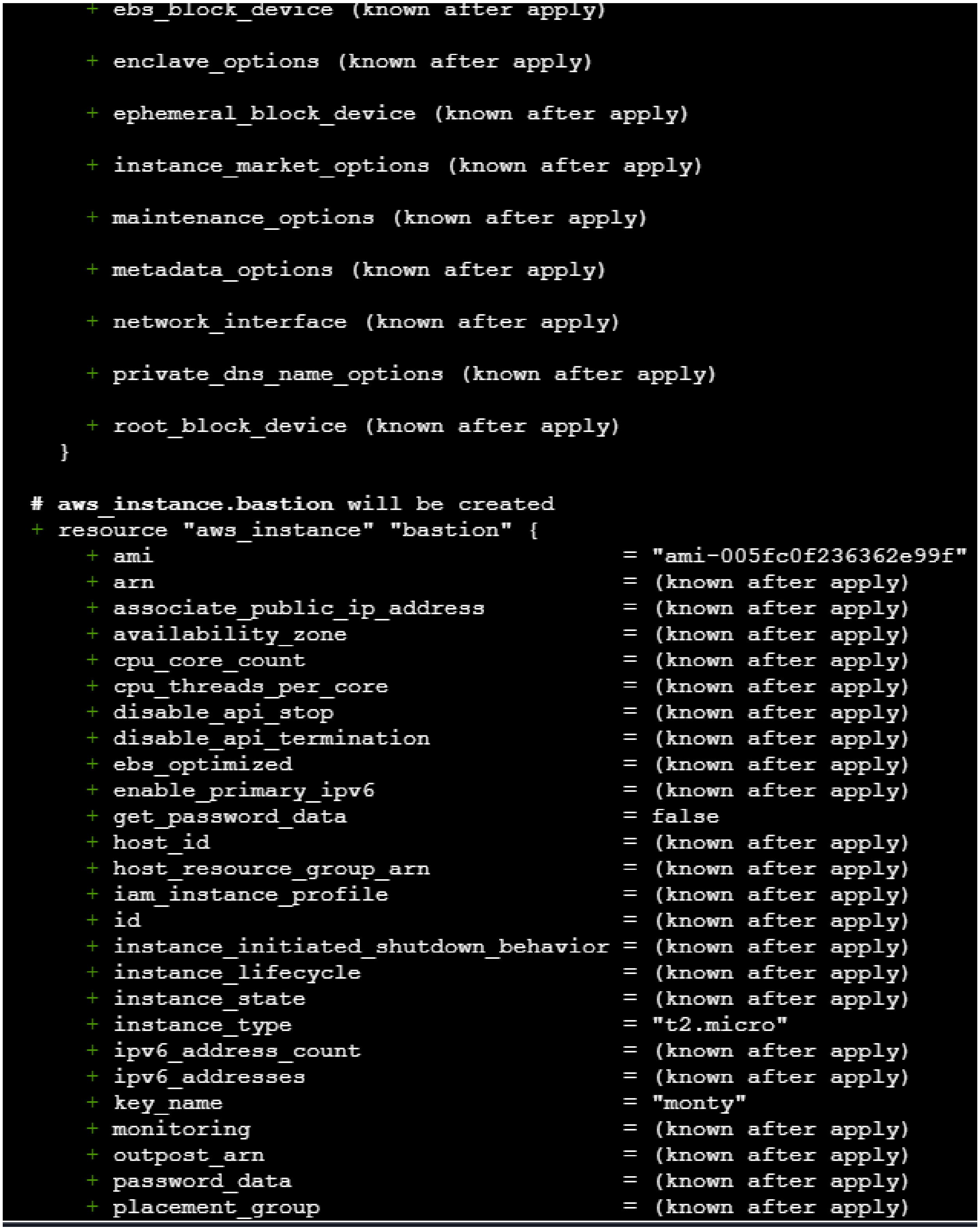
$ terraform apply



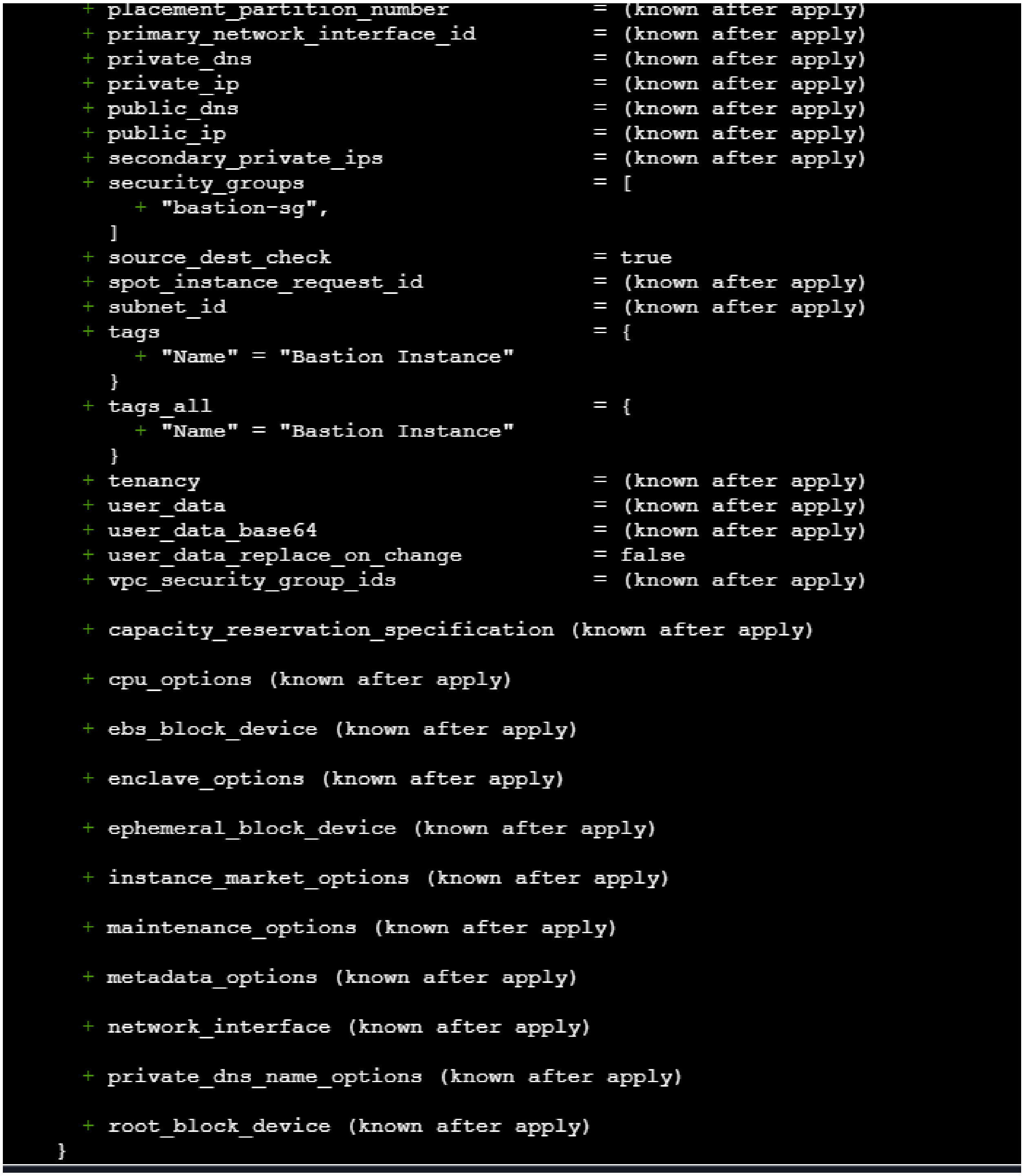
```
root@ip-172-31-80-138:/home/ubuntu# terraform apply
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 1s [id=https://api.ipify.org/]

Terraform used the selected providers to generate the following execution pla
  + create

Terraform will perform the following actions:

  # aws_eip.nat will be created
  + resource "aws_eip" "nat" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance             = (known after apply)
      + ipam_pool_id         = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + ptr_record           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags_all             = (known after apply)
      + vpc                  = true
    }

  # aws_instance.app will be created
  + resource "aws_instance" "app" {
      + ami                                  = "ami-005fc0f236362e99f"
      + 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)
```



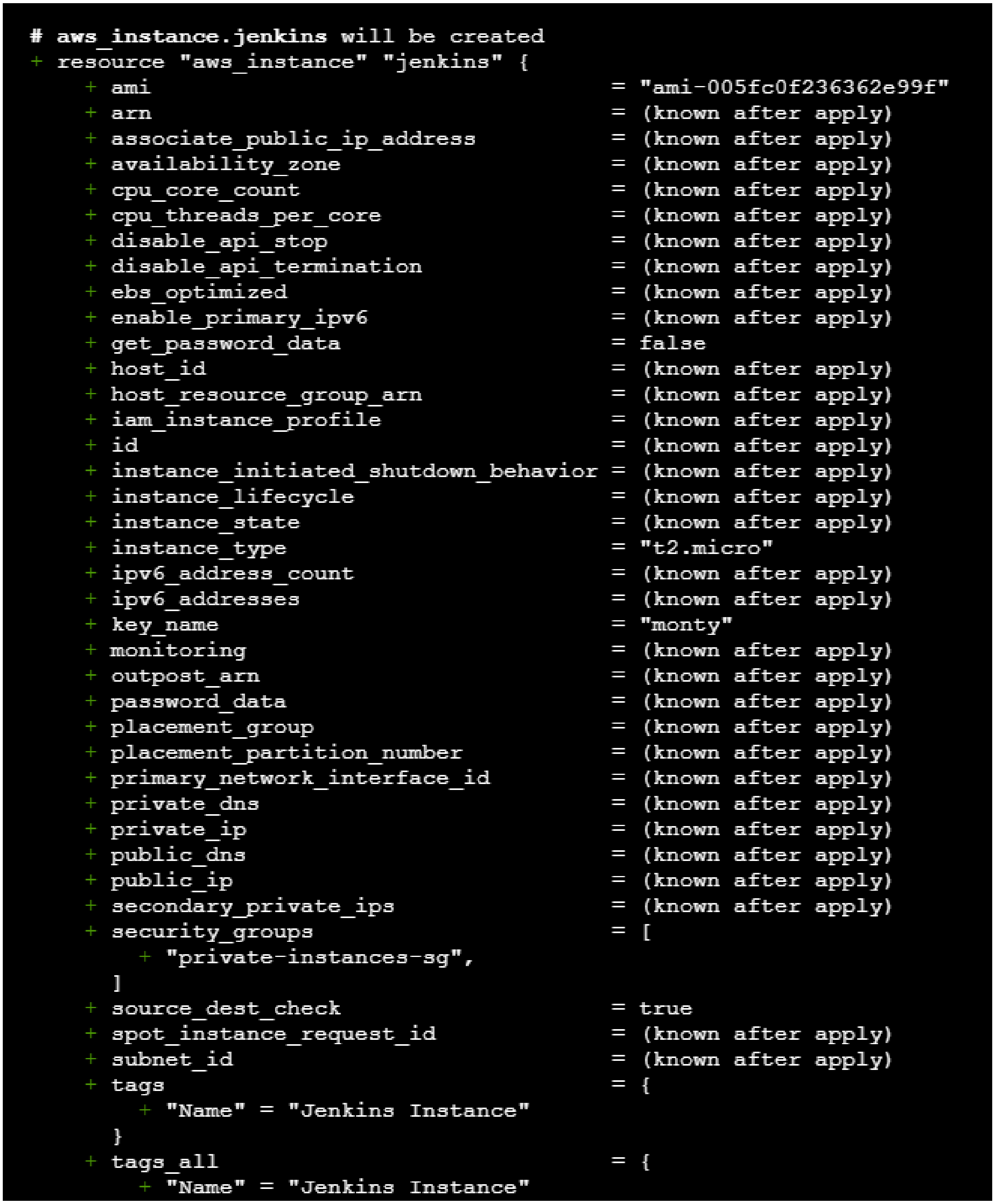
```
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (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                             = "monty"
      + 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                      = [
          + "private-instances-sg",
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "App Instance"
        }
      + tags_all                             = {
          + "Name" = "App Instance"
        }
      + 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)
```



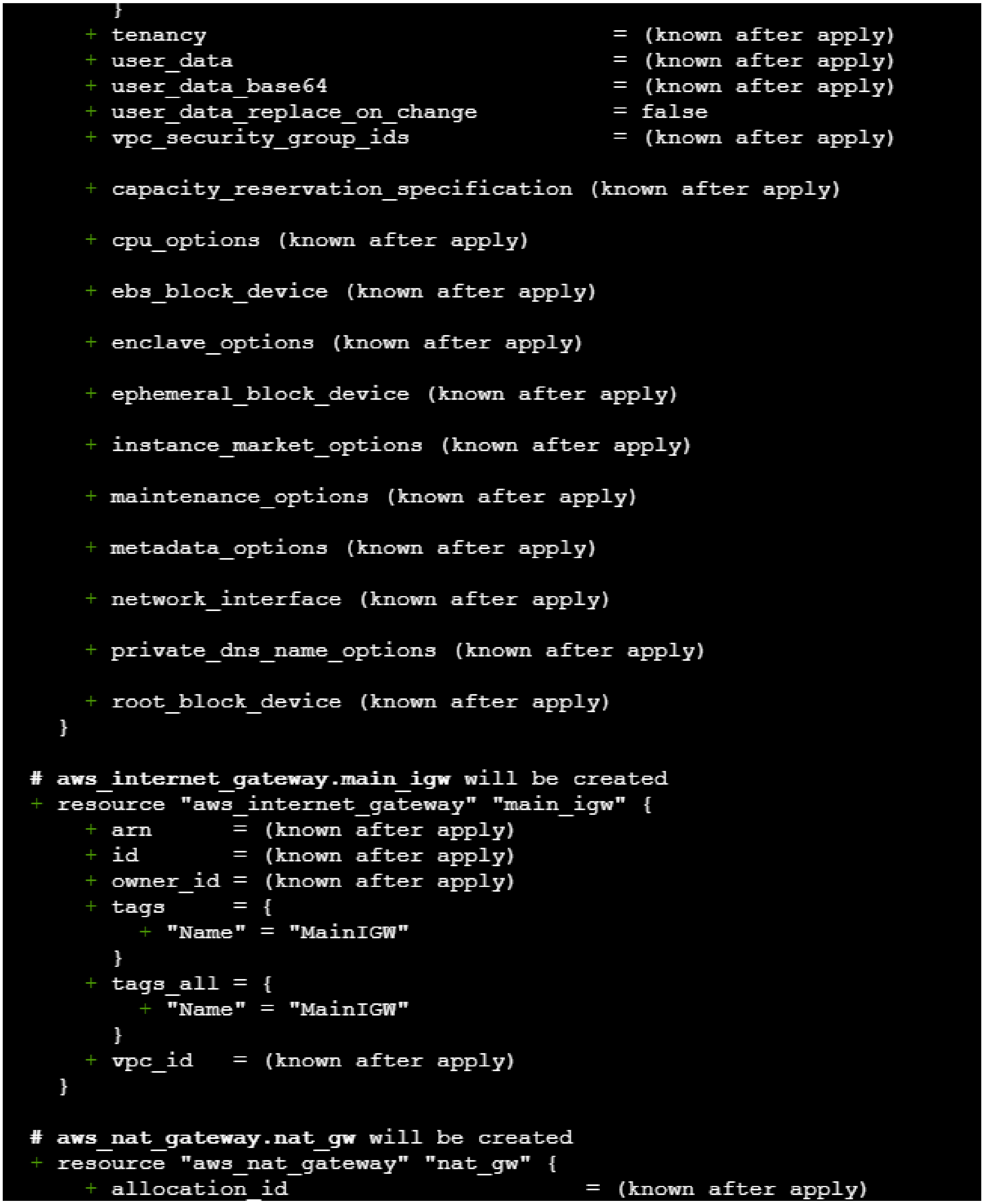
```
      + ebs_block_device (known after apply)

      + 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.bastion will be created
  + resource "aws_instance" "bastion" {
      + ami                                  = "ami-005fc0f236362e99f"
      + 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                             = "monty"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```



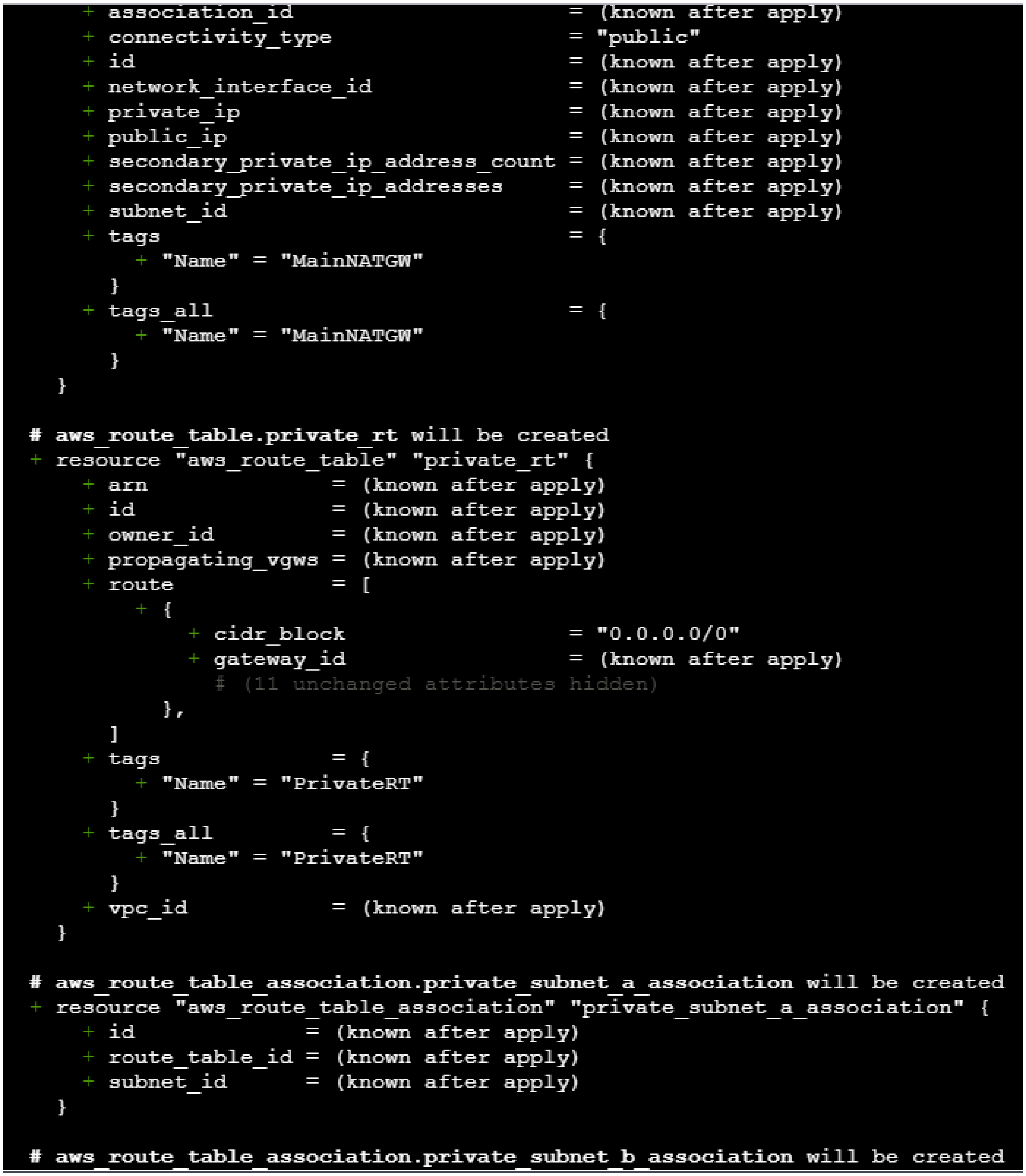
```
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = [
          + "bastion-sg",
        ]
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "Bastion Instance"
        }
      + tags_all                             = {
          + "Name" = "Bastion Instance"
        }
      + 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)
    }
```



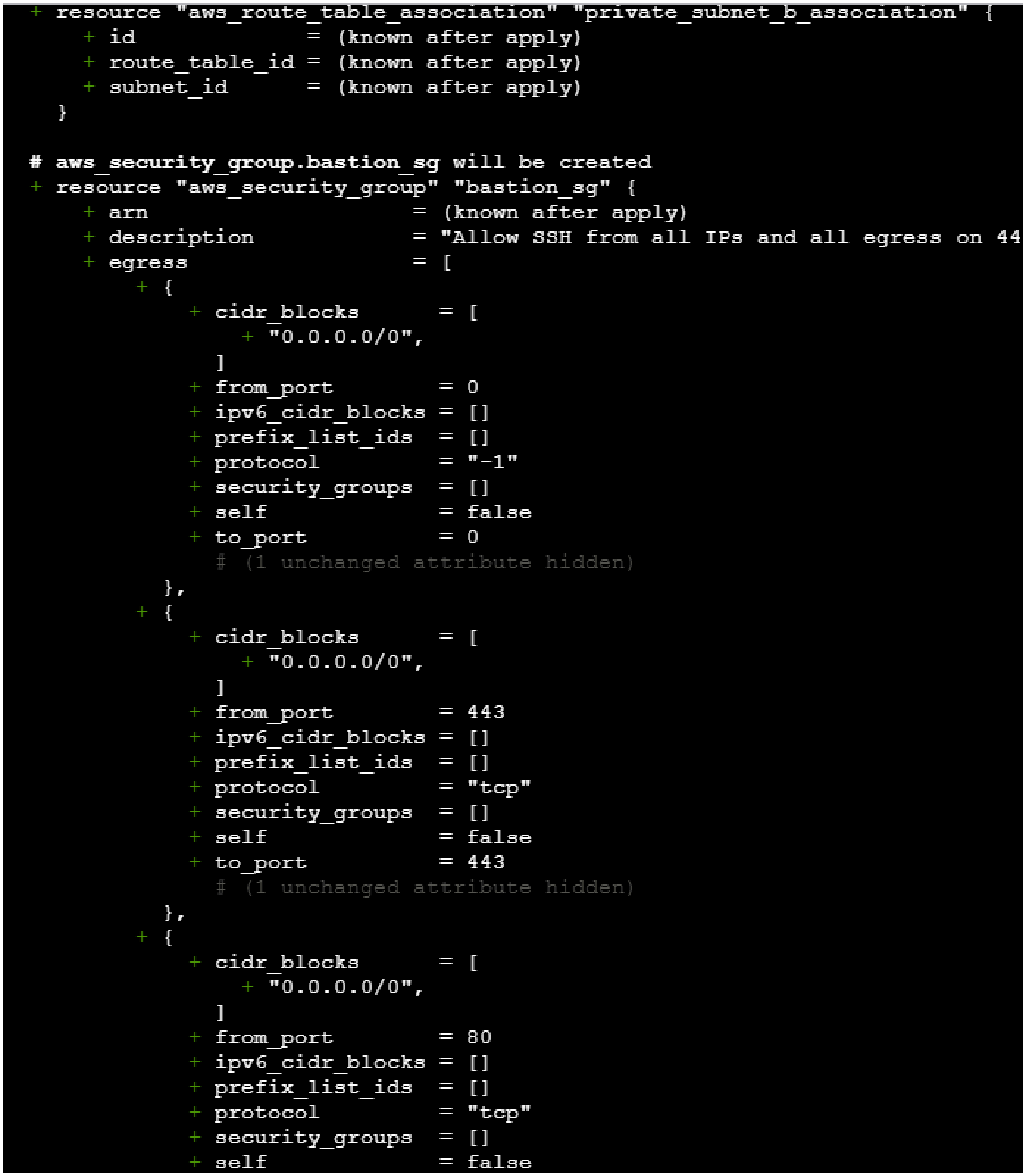
```
# aws_instance.jenkins will be created
+ resource "aws_instance" "jenkins" {
    + ami                                  = "ami-005fc0f236362e99f"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (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                             = "monty"
    + 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                      = [
        + "private-instances-sg",
      ]
    + source_dest_check                    = true
    + spot_instance_request_id             = (known after apply)
    + subnet_id                            = (known after apply)
    + tags                                 = {
        + "Name" = "Jenkins Instance"
      }
    + tags_all                             = {
        + "Name" = "Jenkins Instance"
```



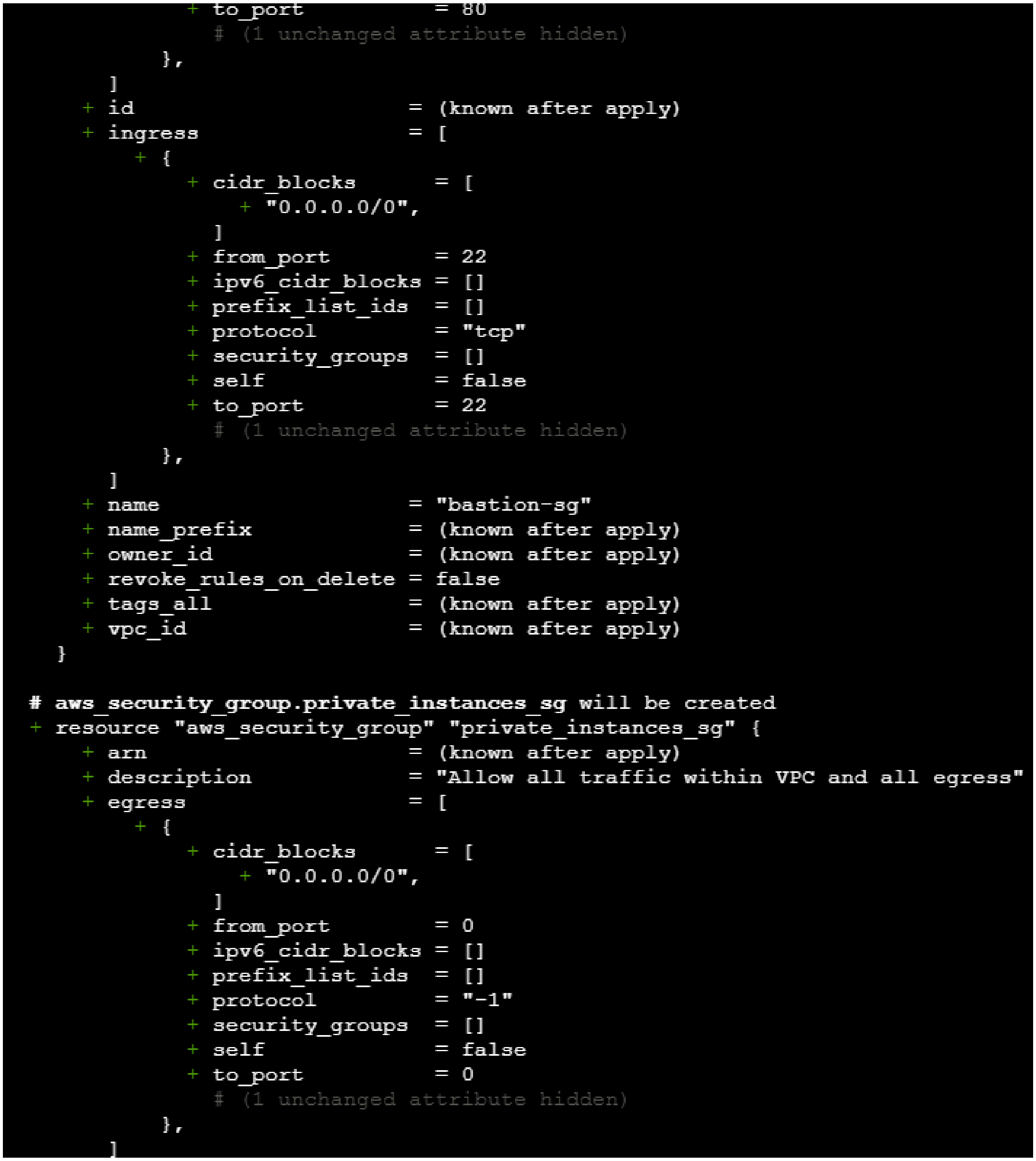
```
      }
    + tenancy                              = (known after apply)
    + user_data                            = (known after apply)
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (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_internet_gateway.main_igw will be created
+ resource "aws_internet_gateway" "main_igw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "MainIGW"
      }
    + tags_all = {
        + "Name" = "MainIGW"
      }
    + vpc_id   = (known after apply)
  }

# aws_nat_gateway.nat_gw will be created
+ resource "aws_nat_gateway" "nat_gw" {
    + allocation_id                      = (known after apply)
```



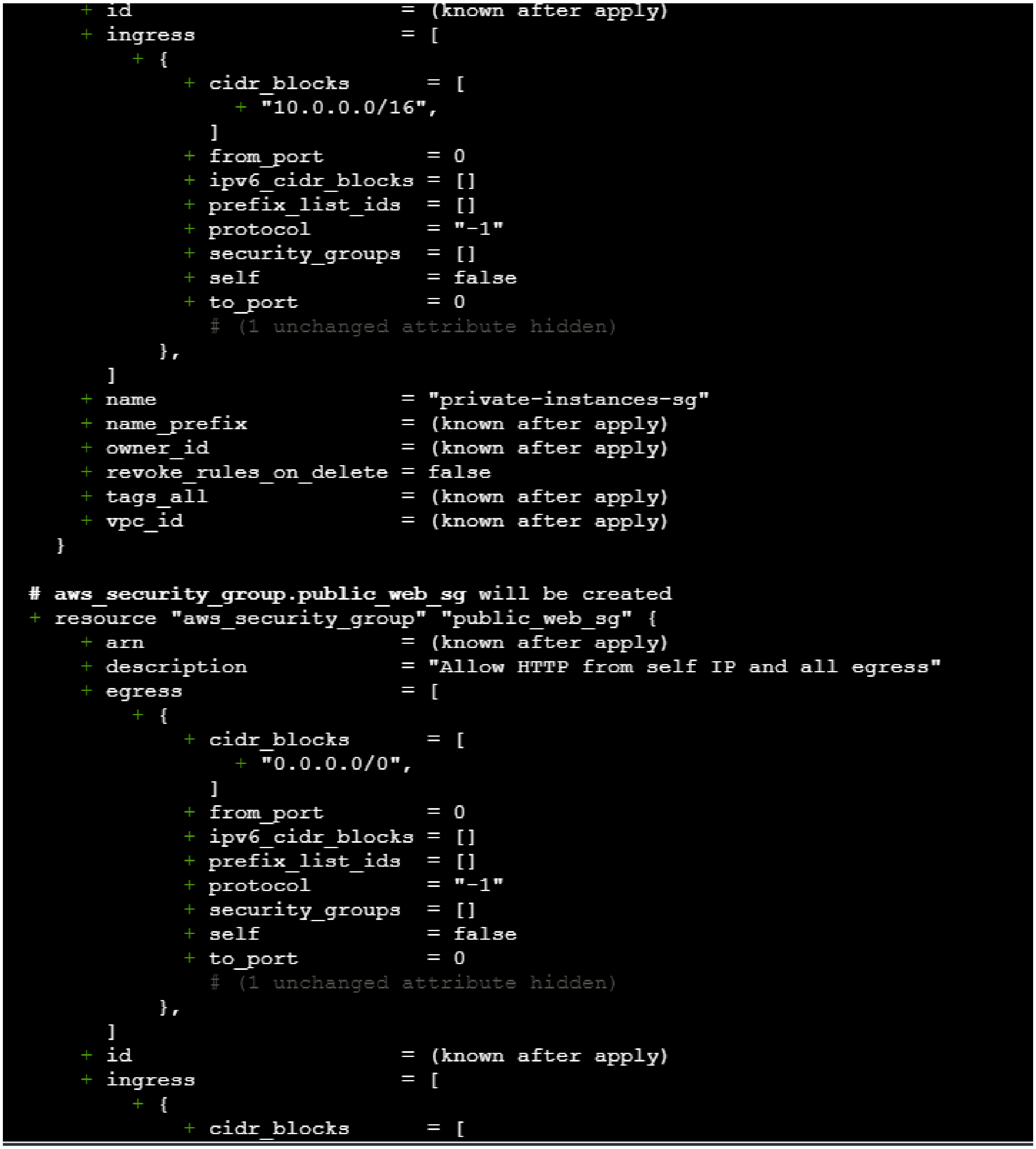
```
      + association_id                     = (known after apply)
      + connectivity_type                  = "public"
      + id                                 = (known after apply)
      + network_interface_id               = (known after apply)
      + private_ip                         = (known after apply)
      + public_ip                          = (known after apply)
      + secondary_private_ip_address_count = (known after apply)
      + secondary_private_ip_addresses     = (known after apply)
      + subnet_id                          = (known after apply)
      + tags                               = {
          + "Name" = "MainNATGW"
        }
      + tags_all                           = {
          + "Name" = "MainNATGW"
        }
    }

  # aws_route_table.private_rt will be created
  + resource "aws_route_table" "private_rt" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = [
          + {
              + cidr_block                 = "0.0.0.0/0"
              + gateway_id                 = (known after apply)
                # (11 unchanged attributes hidden)
            },
        ]
      + tags             = {
          + "Name" = "PrivateRT"
        }
      + tags_all         = {
          + "Name" = "PrivateRT"
        }
      + vpc_id           = (known after apply)
    }

  # aws_route_table_association.private_subnet_a_association will be created
  + resource "aws_route_table_association" "private_subnet_a_association" {
      + id             = (known after apply)
      + route_table_id = (known after apply)
      + subnet_id      = (known after apply)
    }

  # aws_route_table_association.private_subnet_b_association will be created
```



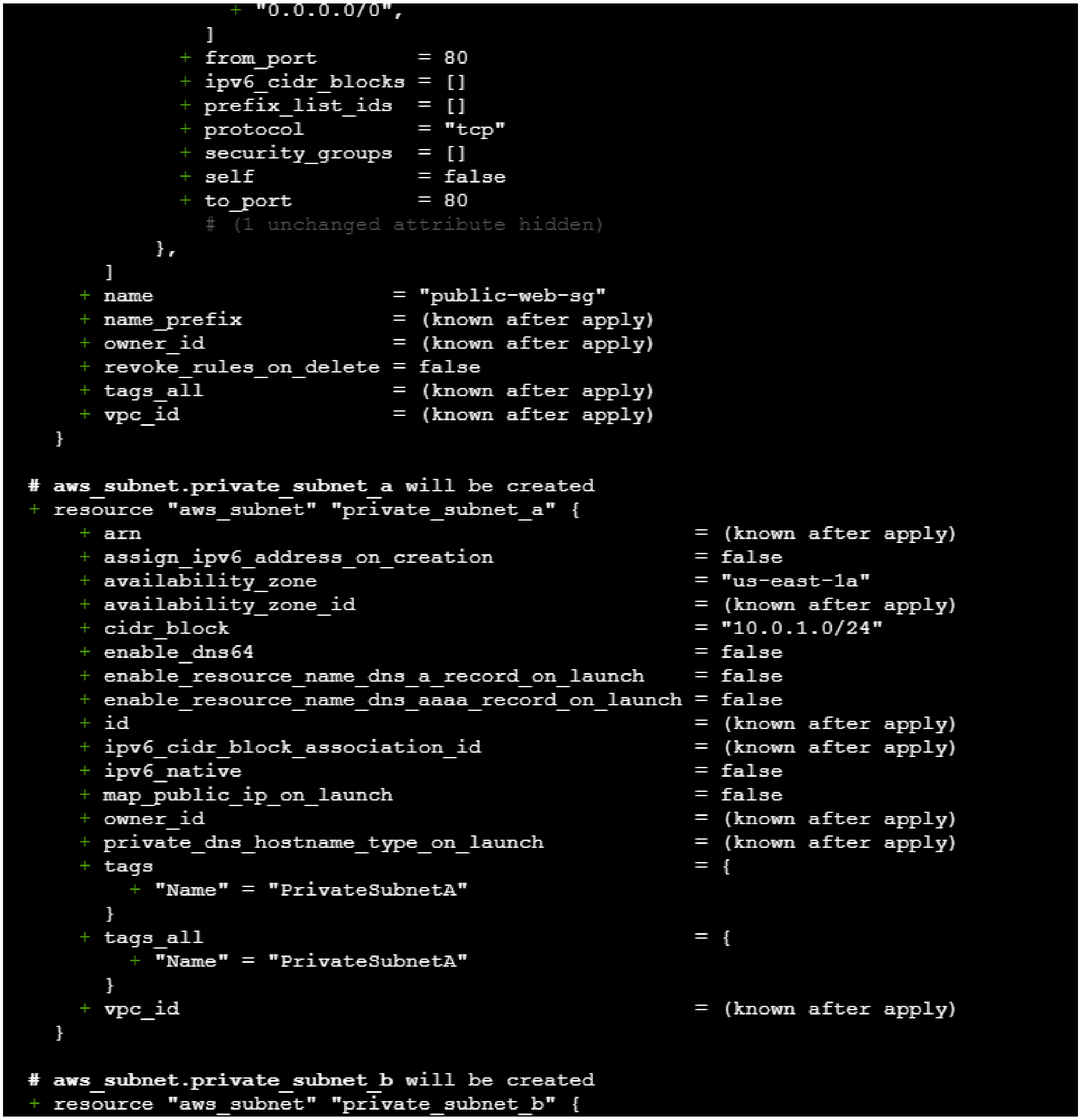
```
+ resource "aws_route_table_association" "private_subnet_b_association" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_security_group.bastion_sg will be created
+ resource "aws_security_group" "bastion_sg" {
    + arn                    = (known after apply)
    + description            = "Allow SSH from all IPs and all egress on 44
    + egress                 = [
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + from_port        = 0
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "-1"
            + security_groups  = []
            + self             = false
            + to_port          = 0
              # (1 unchanged attribute hidden)
          },
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + from_port        = 443
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 443
              # (1 unchanged attribute hidden)
          },
        + {
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + from_port        = 80
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
```



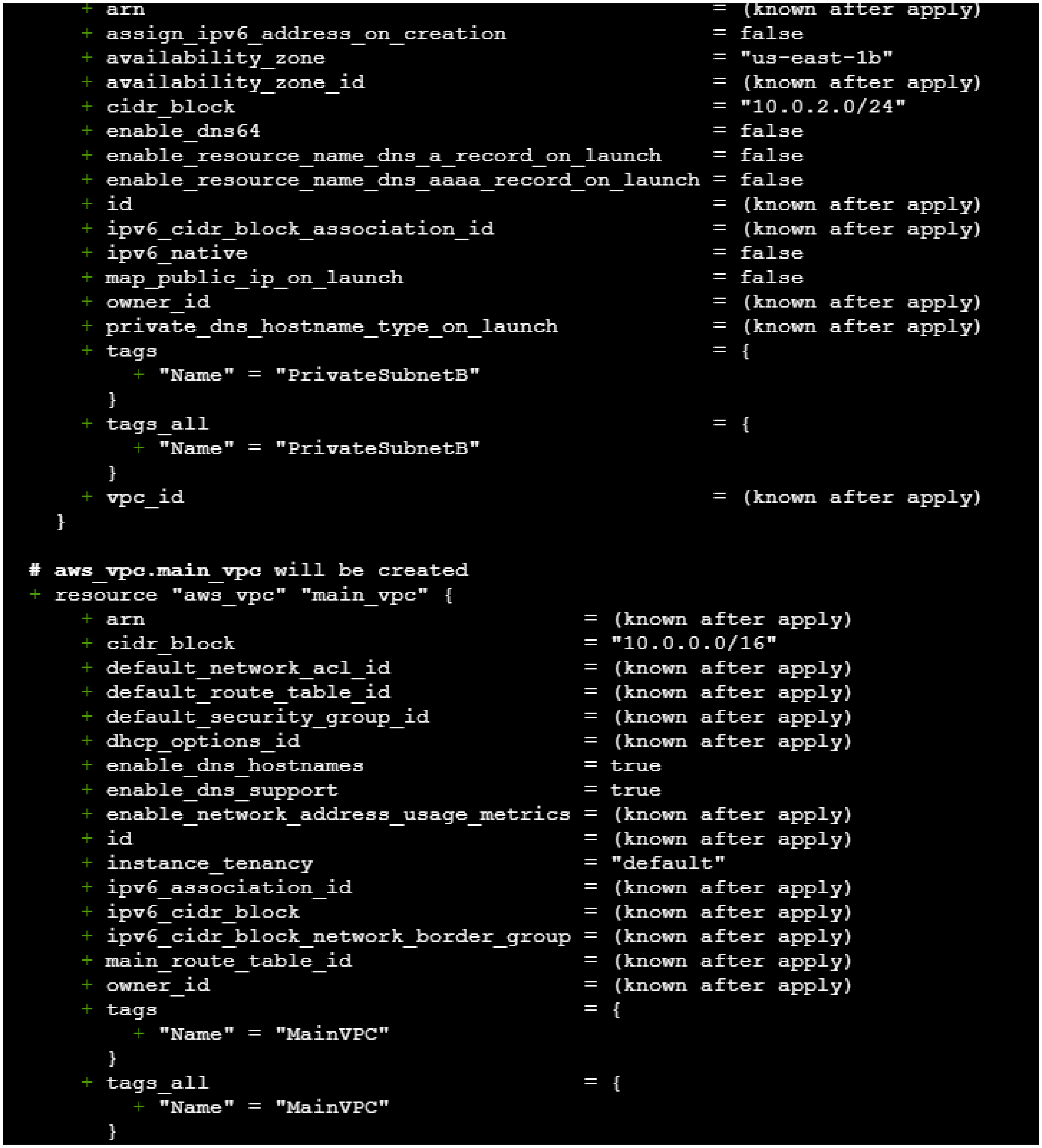
```
              + to_port          = 80
                # (1 unchanged attribute hidden)
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 22
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "bastion-sg"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

  # aws_security_group.private_instances_sg will be created
  + resource "aws_security_group" "private_instances_sg" {
      + arn                    = (known after apply)
      + description            = "Allow all traffic within VPC and all egress"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
                # (1 unchanged attribute hidden)
            },
        ]
```



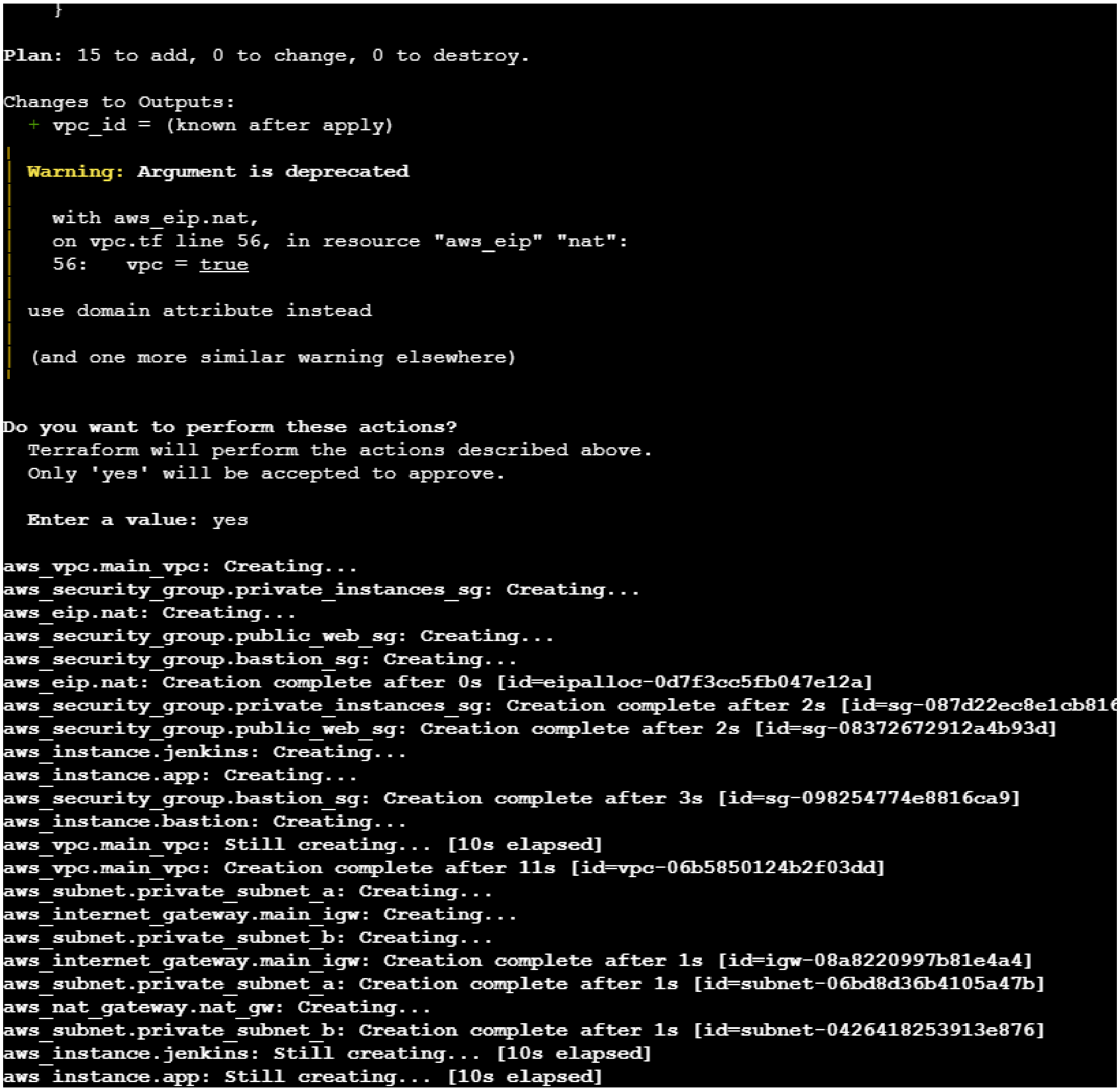
```
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
                  + "10.0.0.0/16",
                ]
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "private-instances-sg"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

  # aws_security_group.public_web_sg will be created
  + resource "aws_security_group" "public_web_sg" {
      + arn                    = (known after apply)
      + description            = "Allow HTTP from self IP and all egress"
      + egress                 = [
          + {
              + cidr_blocks      = [
                  + "0.0.0.0/0",
                ]
              + from_port        = 0
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "-1"
              + security_groups  = []
              + self             = false
              + to_port          = 0
                # (1 unchanged attribute hidden)
            },
        ]
      + id                     = (known after apply)
      + ingress                = [
          + {
              + cidr_blocks      = [
```



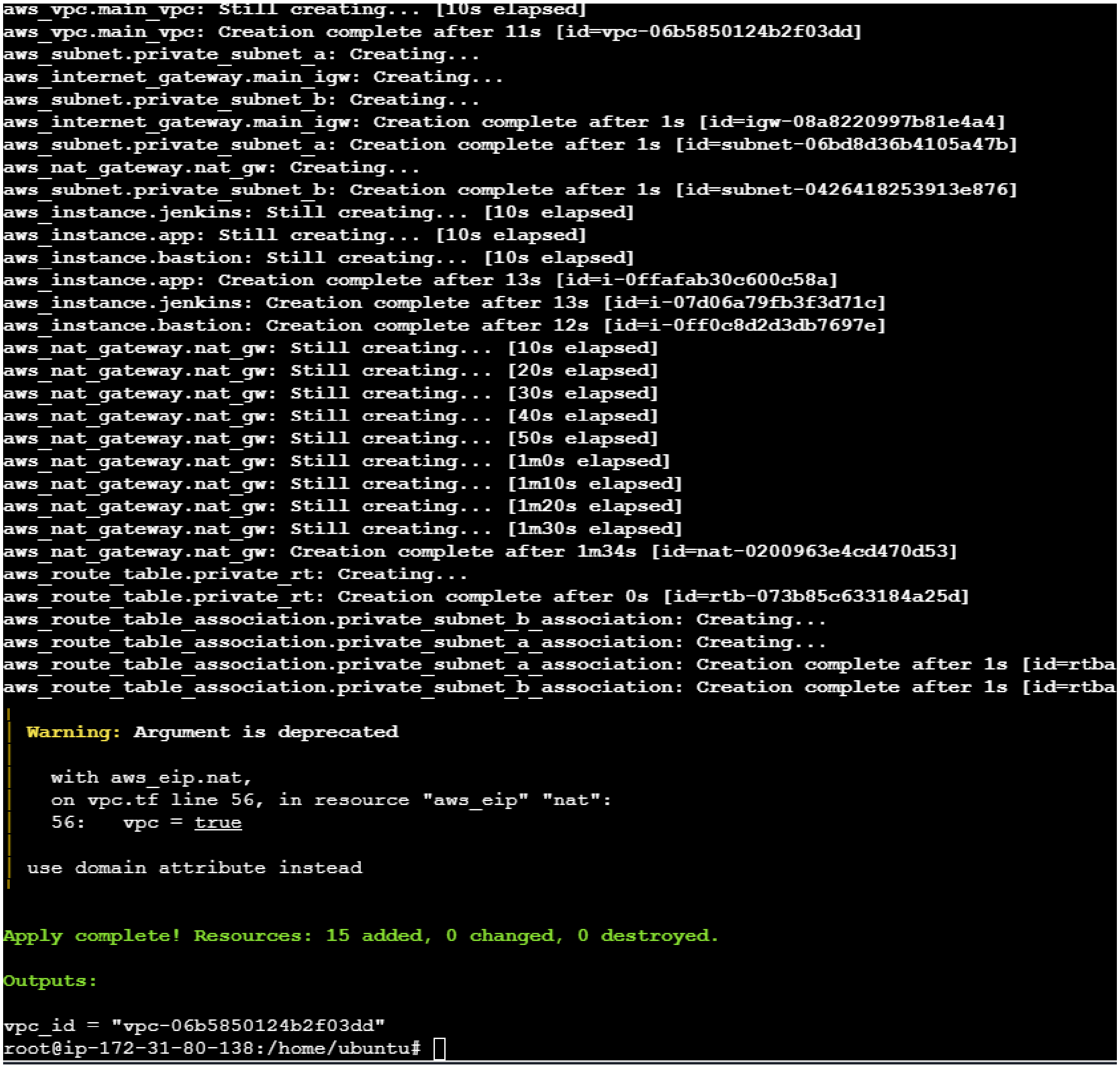
```
                  + "0.0.0.0/0",
                ]
              + from_port        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 80
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "public-web-sg"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + revoke_rules_on_delete = false
      + tags_all               = (known after apply)
      + vpc_id                 = (known after apply)
    }

  # aws_subnet.private_subnet_a will be created
  + resource "aws_subnet" "private_subnet_a" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "PrivateSubnetA"
        }
      + tags_all                                       = {
          + "Name" = "PrivateSubnetA"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_subnet.private_subnet_b will be created
  + resource "aws_subnet" "private_subnet_b" {
```



```
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1b"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.2.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "PrivateSubnetB"
        }
      + tags_all                                       = {
          + "Name" = "PrivateSubnetB"
        }
      + vpc_id                                         = (known after apply)
    }

  # 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                 = true
      + 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                                 = {
          + "Name" = "MainVPC"
        }
      + tags_all                             = {
          + "Name" = "MainVPC"
        }
```



```
    }

Plan: 15 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + vpc_id = (known after apply)
╷
│ Warning: Argument is deprecated
│
│   with aws_eip.nat,
│   on vpc.tf line 56, in resource "aws_eip" "nat":
│   56:   vpc = true
│
│ use domain attribute instead
│
│ (and one more similar warning elsewhere)
╵

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

aws_vpc.main_vpc: Creating...
aws_security_group.private_instances_sg: Creating...
aws_eip.nat: Creating...
aws_security_group.public_web_sg: Creating...
aws_security_group.bastion_sg: Creating...
aws_eip.nat: Creation complete after 0s [id=eipalloc-0d7f3cc5fb047e12a]
aws_security_group.private_instances_sg: Creation complete after 2s [id=sg-087d22ec8e1cb816
aws_security_group.public_web_sg: Creation complete after 2s [id=sg-08372672912a4b93d]
aws_instance.jenkins: Creating...
aws_instance.app: Creating...
aws_security_group.bastion_sg: Creation complete after 3s [id=sg-098254774e8816ca9]
aws_instance.bastion: Creating...
aws_vpc.main_vpc: Still creating... [10s elapsed]
aws_vpc.main_vpc: Creation complete after 11s [id=vpc-06b5850124b2f03dd]
aws_subnet.private_subnet_a: Creating...
aws_internet_gateway.main_igw: Creating...
aws_subnet.private_subnet_b: Creating...
aws_internet_gateway.main_igw: Creation complete after 1s [id=igw-08a8220997b81e4a4]
aws_subnet.private_subnet_a: Creation complete after 1s [id=subnet-06bd8d36b4105a47b]
aws_nat_gateway.nat_gw: Creating...
aws_subnet.private_subnet_b: Creation complete after 1s [id=subnet-0426418253913e876]
aws_instance.jenkins: Still creating... [10s elapsed]
aws_instance.app: Still creating... [10s elapsed]
```



```
aws_vpc.main_vpc: Still creating... [10s elapsed]
aws_vpc.main_vpc: Creation complete after 11s [id=vpc-06b5850124b2f03dd]
aws_subnet.private_subnet_a: Creating...
aws_internet_gateway.main_igw: Creating...
aws_subnet.private_subnet_b: Creating...
aws_internet_gateway.main_igw: Creation complete after 1s [id=igw-08a8220997b81e4a4]
aws_subnet.private_subnet_a: Creation complete after 1s [id=subnet-06bd8d36b4105a47b]
aws_nat_gateway.nat_gw: Creating...
aws_subnet.private_subnet_b: Creation complete after 1s [id=subnet-0426418253913e876]
aws_instance.jenkins: Still creating... [10s elapsed]
aws_instance.app: Still creating... [10s elapsed]
aws_instance.bastion: Still creating... [10s elapsed]
aws_instance.app: Creation complete after 13s [id=i-0ffafab30c600c58a]
aws_instance.jenkins: Creation complete after 13s [id=i-07d06a79fb3f3d71c]
aws_instance.bastion: Creation complete after 12s [id=i-0ff0c8d2d3db7697e]
aws_nat_gateway.nat_gw: Still creating... [10s elapsed]
aws_nat_gateway.nat_gw: Still creating... [20s elapsed]
aws_nat_gateway.nat_gw: Still creating... [30s elapsed]
aws_nat_gateway.nat_gw: Still creating... [40s elapsed]
aws_nat_gateway.nat_gw: Still creating... [50s elapsed]
aws_nat_gateway.nat_gw: Still creating... [1m0s elapsed]
aws_nat_gateway.nat_gw: Still creating... [1m10s elapsed]
aws_nat_gateway.nat_gw: Still creating... [1m20s elapsed]
aws_nat_gateway.nat_gw: Still creating... [1m30s elapsed]
aws_nat_gateway.nat_gw: Creation complete after 1m34s [id=nat-0200963e4cd470d53]
aws_route_table.private_rt: Creating...
aws_route_table.private_rt: Creation complete after 0s [id=rtb-073b85c633184a25d]
aws_route_table_association.private_subnet_b_association: Creating...
aws_route_table_association.private_subnet_a_association: Creating...
aws_route_table_association.private_subnet_a_association: Creation complete after 1s [id=rtba
aws_route_table_association.private_subnet_b_association: Creation complete after 1s [id=rtba
╷
│ Warning: Argument is deprecated
│
│   with aws_eip.nat,
│   on vpc.tf line 56, in resource "aws_eip" "nat":
│   56:   vpc = true
│
│ use domain attribute instead
╵

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

Outputs:

vpc_id = "vpc-06b5850124b2f03dd"
root@ip-172-31-80-138:/home/ubuntu# 
```