Assingment 1 Output

```
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_instance.test-instance will be created
 + resource "aws_instance" "test-instance" {
    + ami                            = "ami-0df7a207adb9748c7"
    + 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)
    + 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                       = "test-key-pair"
    + 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"    = "assingment-instance"
    + "purpose" = "Assignment"
   }
  + tags_all                             = {
    + "Name"    = "assingment-instance"
    + "purpose" = "Assignment"
   }
  + 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)

  + root_block_device {
    + delete_on_termination              = true
    + device_name                        = (known after apply)
    + encrypted                          = (known after apply)
    + iops                               = (known after apply)
    + kms_key_id                         = (known after apply)
    + throughput                         = (known after apply)
    + volume_id                          = (known after apply)
    + volume_size                        = 8
    + volume_type                        = "gp2"
   }
 }

# aws_internet_gateway.my_vpc_igw will be created
+ resource "aws_internet_gateway" "my_vpc_igw" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
  + tags     = {
    + "Name" = "My VPC IGW"
   }
  + tags_all = {
    + "Name" = "My VPC IGW"
   }
  + vpc_id   = (known after apply)
 }

# aws_route_table.public_subnet_rt will be created
+ resource "aws_route_table" "public_subnet_rt" {
```

```
+ arn              = (known after apply)
+ id               = (known after apply)
+ owner_id         = (known after apply)
+ propagating_vgws = (known after apply)
+ route            = [
    + {
        + carrier_gateway_id         = ""
        + cidr_block                 = ""
        + core_network_arn           = ""
        + destination_prefix_list_id = ""
        + egress_only_gateway_id     = ""
        + gateway_id                 = (known after apply)
        + ipv6_cidr_block            = "::/0"
        + local_gateway_id           = ""
        + nat_gateway_id             = ""
```