Tasks To Be Performed:

1. Create an EC2 service in the default subnet in the Ohio region

```
  GNU nano 6.2
provider "aws"{
   region = "us-east-2"
   access_key = "AKIATVK4BHVRIRE5B5OU"
   secret_key = "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"
}
resource "aws_instance" "task-1"{
    ami = "ami-0cae6d6fe6048ca2c"
    instance_type = "t2.micro"
    key_name = "terra-kp"
    tags = {
      Name = "work-1"
    }
}
```

```
ubuntu@ip-172-31-28-242:~$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource act
  + create

Terraform will perform the following actions:

  # aws_instance.task-1 will be created
  + resource "aws_instance" "task-1" {
      + ami                                  = "ami-0cae6d6fe6048ca2c"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (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)
      + force_destroy                        = false
      + 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                             = "terra-kp"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (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)

      + primary_network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.
```

```
ubuntu@ip-172-31-28-242:~$ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource action
  + create

Terraform will perform the following actions:

  # aws_instance.task-1 will be created
  + resource "aws_instance" "task-1" {
      + ami                                  = "ami-0a627a85fdcfabbaa"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (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)
      + force_destroy                        = false
      + 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                             = "terra-kp"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```

```
      + primary_network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

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_instance.task-1: Creating...
aws_instance.task-1: Still creating... [00m10s elapsed]
aws_instance.task-1: Still creating... [00m20s elapsed]
aws_instance.task-1: Still creating... [00m30s elapsed]
aws_instance.task-1: Creation complete after 32s [id=i-0d2afd2f59bafa2c1]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-28-242:~$
```