Name : Zuha Irfan

Roll No : 2023-BSE-073

Subject: Cloud Computing

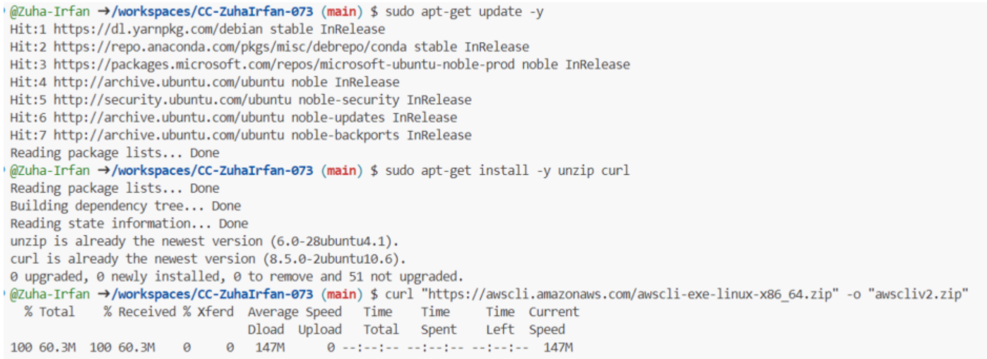
Semester & Section : V-B

Assignment 2 – Terraform & AWS Setup

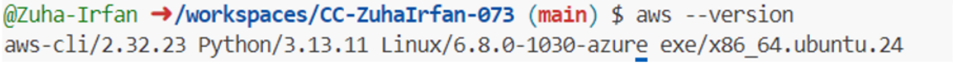
Phase 1: Environment Setup

1.1 Install AWS CLI

Screenshot: aws_cli_install.png



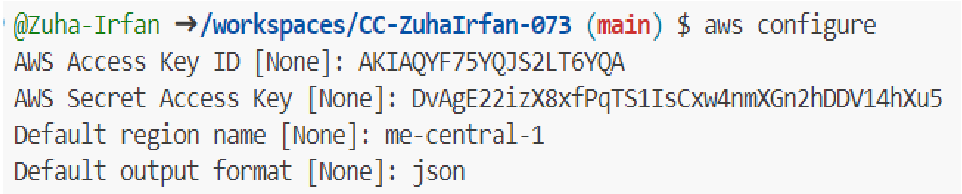
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ sudo apt-get update -y
Hit:1 https://dl.yarnpkg.com/debian stable InRelease
Hit:2 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease
Hit:3 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ sudo apt-get install -y unzip curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unzip is already the newest version (6.0-28ubuntu4.1).
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 60.3M  100 60.3M    0     0   147M      0 --:--:-- --:--:-- --:--:--  147M
```



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ aws --version
aws-cli/2.32.23 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
```

1.2 Configure AWS CLI

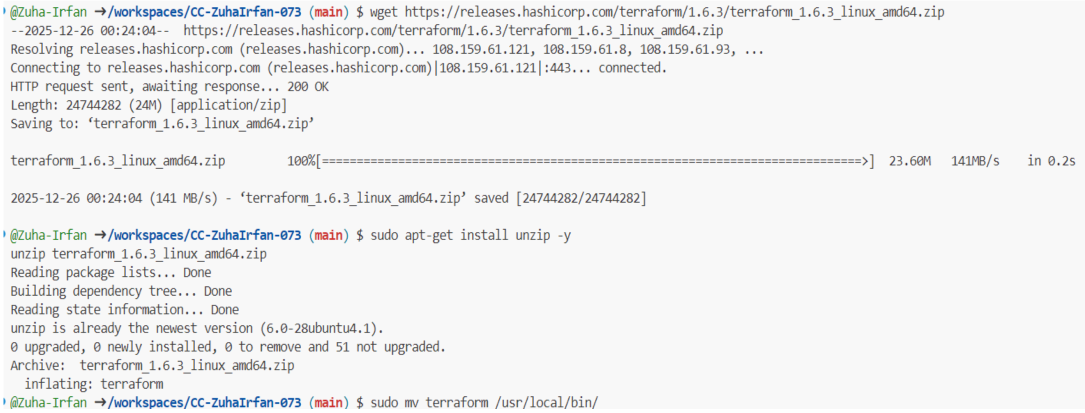
Screenshot: aws_configure.png



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ aws configure
AWS Access Key ID [None]: AKIAQYF75YQJS2LT6YQA
AWS Secret Access Key [None]: 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
Default region name [None]: me-central-1
Default output format [None]: json
```

1.3 Install Terraform

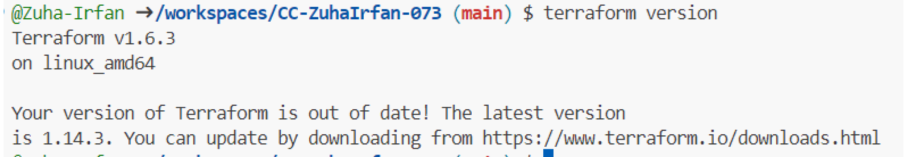
Screenshot: terraform_install.png



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ wget https://releases.hashicorp.com/terraform/1.6.3/terraform_1.6.3_linux_amd64.zip
--2025-12-26 00:24:04--  https://releases.hashicorp.com/terraform/1.6.3/terraform_1.6.3_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 108.159.61.121, 108.159.61.8, 108.159.61.93, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|108.159.61.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24744282 (24M) [application/zip]
Saving to: ‘terraform_1.6.3_linux_amd64.zip’

terraform_1.6.3_linux_amd64.zip        100%[=============================================================================>]  23.60M   141MB/s    in 0.2s

2025-12-26 00:24:04 (141 MB/s) - ‘terraform_1.6.3_linux_amd64.zip’ saved [24744282/24744282]

@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ sudo apt-get install unzip -y
unzip terraform_1.6.3_linux_amd64.zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unzip is already the newest version (6.0-28ubuntu4.1).
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
Archive:  terraform_1.6.3_linux_amd64.zip
  inflating: terraform
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ sudo mv terraform /usr/local/bin/
```

1.4 Verify Installations

Screenshot: verify_installation.png

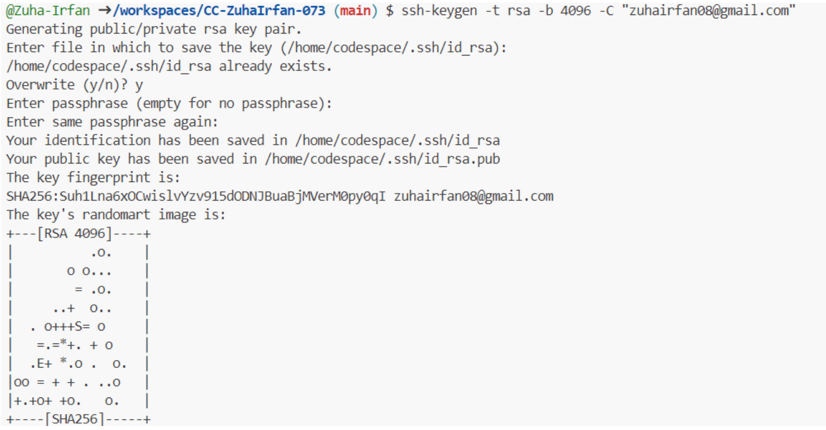


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ terraform version
Terraform v1.6.3
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.14.3. You can update by downloading from https://www.terraform.io/downloads.html
```

Phase 2: SSH Key Generation

21.Generate SSH Key Pair

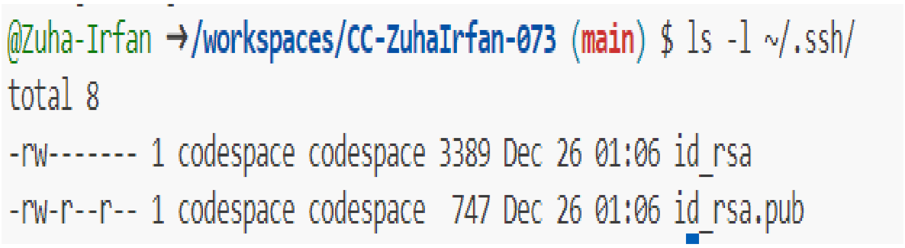
Screenshot: ssh_keygen.png



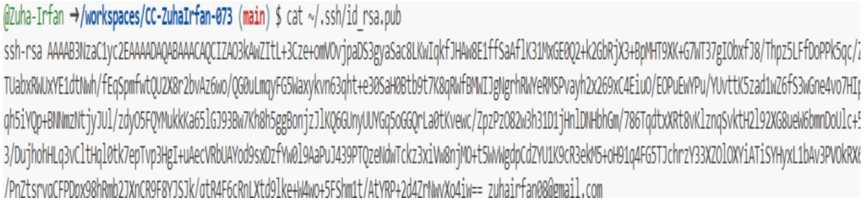
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ ssh-keygen -t rsa -b 4096 -C "zuhairfan08@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/codespace/.ssh/id_rsa):
/home/codespace/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/codespace/.ssh/id_rsa
Your public key has been saved in /home/codespace/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Suh1Lna6xOCwislvYzv915dODNJBuaBjMVerM0py0qI zuhairfan08@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|           .o.   |
|        o o...   |
|         = .o.   |
|      ..+  o..   |
|   . o+++S= o    |
|    =.=*+. + o   |
|   .E+ *.o .  o. |
|oo = + + . ..o   |
|+.+o+ +o.   o.   |
+----[SHA256]-----+
```

2.2 Verify SSH Keys

Screenshot: ssh_key_verify.png



```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ ls -l ~/.ssh/
total 8
-rw------- 1 codespace codespace 3389 Dec 26 01:06 id_rsa
-rw-r--r-- 1 codespace codespace  747 Dec 26 01:06 id_rsa.pub
```

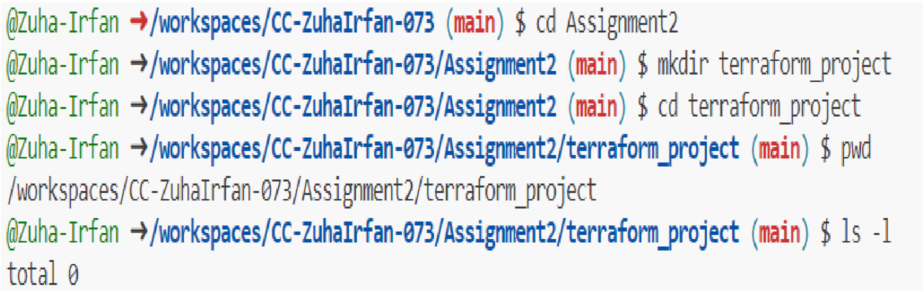


```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCIZAO3kAwZItL+3Cze+omVOvjpaDS3gyaSac8LKwIqkfJHAw8E1ffSaAflK31MxGE0Q2+k2GbRjX3+BpMHT9XK+G7WT37gIObxfJ8/Thpz5LFfDoPPk5qc/
TUabxRWUxYE1dtNwh/fEqSpmfwtQU2X8r2bvAz6wo/QG0uLmqyFG5Waxykvn63qht+e30SaH0Btb9t7K8qRwfBMWIJgNgrhRWYeRMSPvayh2x269xC4EiuO/EOPuEwYPu/YUvttK5zad1wZ6fS3wGne4vo7HI
qh5iYQp+BNNmzNtjyJUl/zdyO5FQYMukkKa65lGJ93Bw7Kh8h5ggBonjzJlKQ6GUnyUUYGq5oGGQrLa0tKvewc/ZpzPzO82w3h31D1jHnlDNHbhGm/786TqdtxXRt8vKlznqSvktH2l92XG8ueW6bmnDoUlc+
3/DujhohHLq3vCltHql0tk7epTvp3HgI+uAecVRbUAYod9sxDzfYw0l9AaPuJ439PTQzeNdwTckz3xiVw8njMO+t5WvWgdpCdZYU1K9cR3ekM5+oH91q4FG5TJchrzY33XZOlOXYiATiSYHyxL1bAv3PVOkRX
/PnZtsrvqCFPDpx98hRmb2JXnCR9F8YJSJk/qtR4F6cRnLXtd9lke+W4wo+5FShm1t/AtYRP+2d4ZrNwvXo4iw== zuhairfan08@gmail.com
```

Phase 3: Terraform Project Initialization

3.1 Create Project Directory

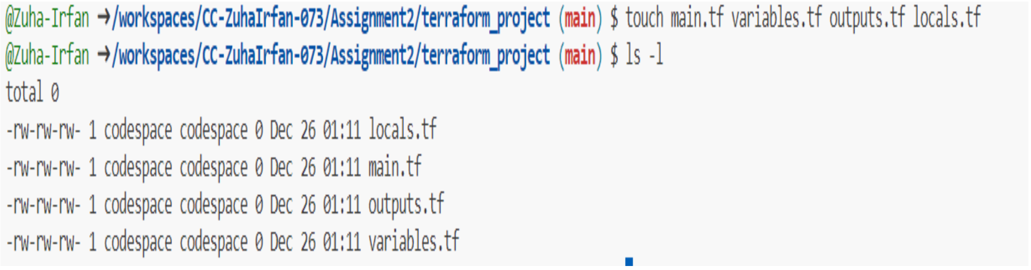
Screenshot: project_directory.png



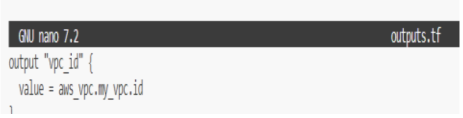
```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073 (main) $ cd Assignment2
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Assignment2 (main) $ mkdir terraform_project
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Assignment2 (main) $ cd terraform_project
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ pwd
/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ ls -l
total 0
```

3.2 Create Terraform Files

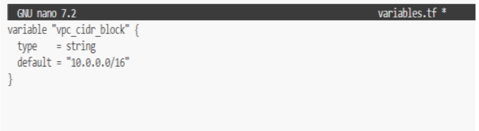
Screenshot: terraform_files.png



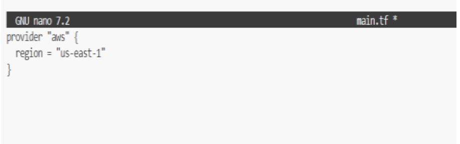
```
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ touch main.tf variables.tf outputs.tf locals.tf
@Zuha-Irfan ➜ /workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ ls -l
total 0
-rw-rw-rw- 1 codespace codespace 0 Dec 26 01:11 locals.tf
-rw-rw-rw- 1 codespace codespace 0 Dec 26 01:11 main.tf
-rw-rw-rw- 1 codespace codespace 0 Dec 26 01:11 outputs.tf
-rw-rw-rw- 1 codespace codespace 0 Dec 26 01:11 variables.tf
```



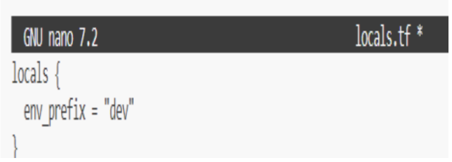
```
GNU nano 7.2                                   outputs.tf
output "vpc_id" {
  value = aws_vpc.my_vpc.id
}
```



```
GNU nano 7.2                                   variables.tf *
variable "vpc_cidr_block" {
  type    = string
  default = "10.0.0.0/16"
}
```



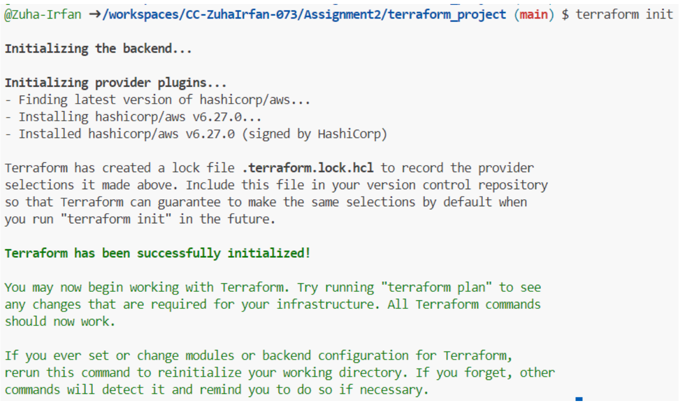
```
GNU nano 7.2                                   main.tf *
provider "aws" {
  region = "us-east-1"
}
```



```
GNU nano 7.2                                   locals.tf *
locals {
  env_prefix = "dev"
}
```

3.3 Initialize Terraform

Screenshot: terraform_init.png

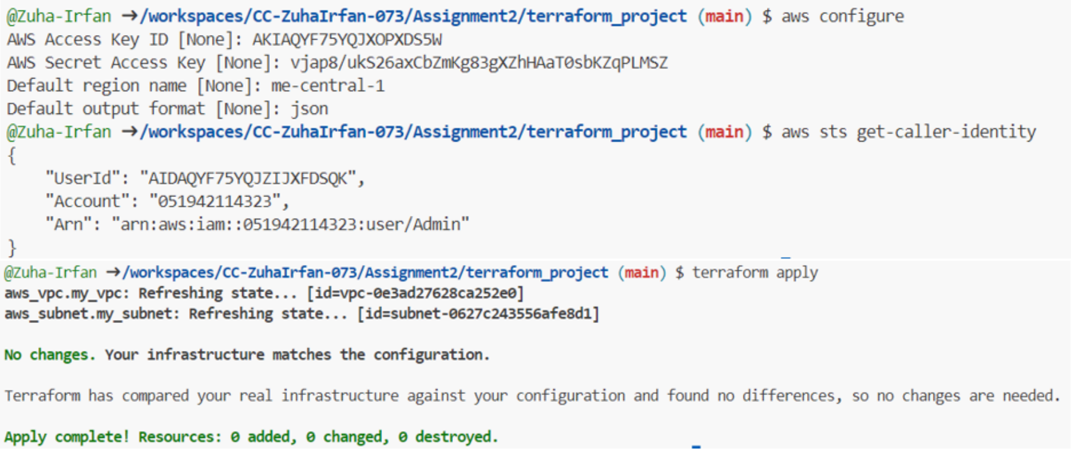


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (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.
```

Phase 4: Terraform Resource Creation

4.1 Configure AWS VPC

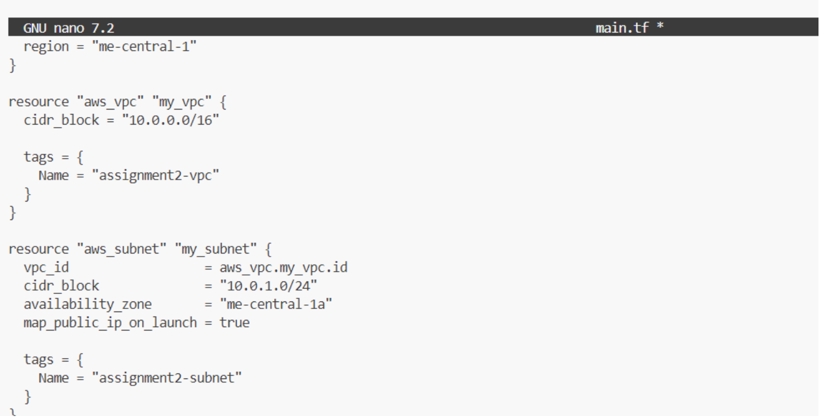
Screenshot: vpc_setup.png



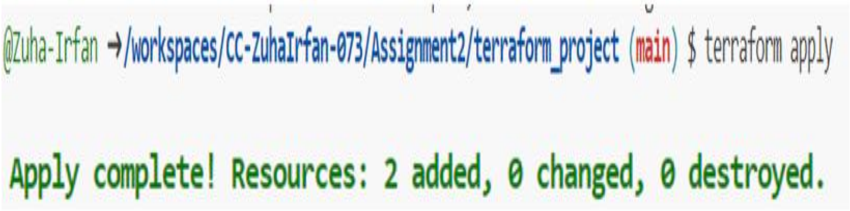
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ aws configure
AWS Access Key ID [None]: AKIAQYF75YQJXOPXDS5W
AWS Secret Access Key [None]: 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
Default region name [None]: me-central-1
Default output format [None]: json
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ aws sts get-caller-identity
{
    "UserId": "AIDAQYF75YQJZIJXFDSQK",
    "Account": "051942114323",
    "Arn": "arn:aws:iam::051942114323:user/Admin"
}
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-0e3ad27628ca252e0]
aws_subnet.my_subnet: Refreshing state... [id=subnet-0627c243556afe8d1]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

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

4.2 Configure Subnet

Screenshot: subnet_setup.png



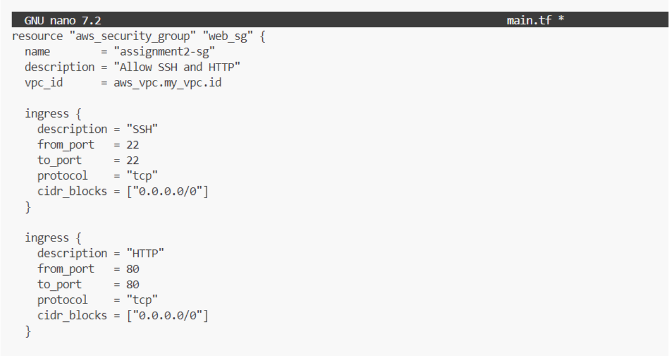
```
GNU nano 7.2                                                          main.tf *
  region = "me-central-1"
}

resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "assignment2-vpc"
  }
}

resource "aws_subnet" "my_subnet" {
  vpc_id                  = aws_vpc.my_vpc.id
  cidr_block              = "10.0.1.0/24"
  availability_zone       = "me-central-1a"
  map_public_ip_on_launch = true

  tags = {
    Name = "assignment2-subnet"
  }
```



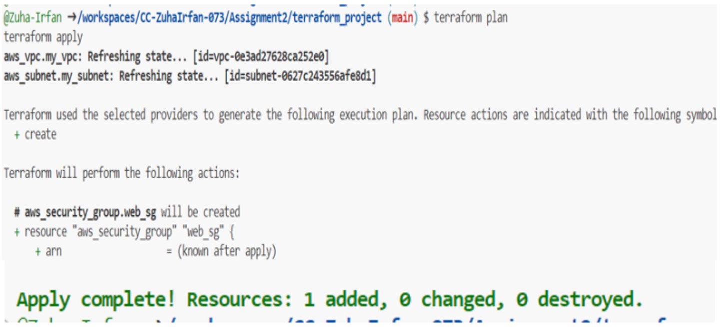
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform apply

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

4.3 Configure Security Groups

Screenshot: security_group.png



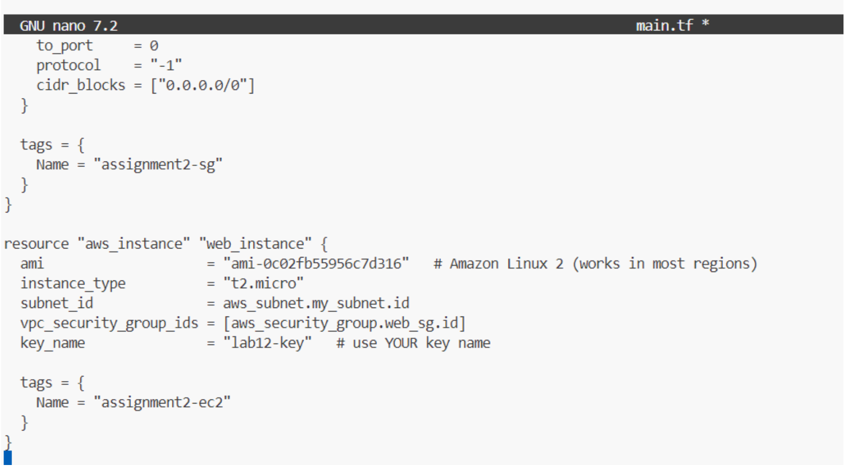
```
GNU nano 7.2                                                main.tf *
resource "aws_security_group" "web_sg" {
  name        = "assignment2-sg"
  description = "Allow SSH and HTTP"
  vpc_id      = aws_vpc.my_vpc.id

  ingress {
    description = "SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    description = "HTTP"
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
```



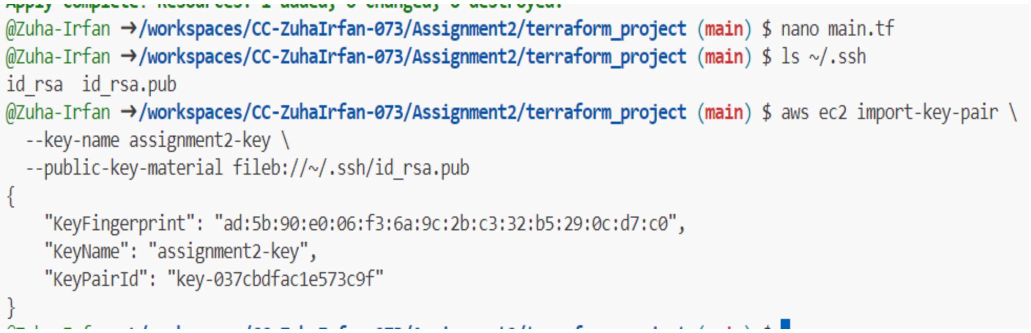
```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform plan
terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-0e3ad27628ca252e0]
aws_subnet.my_subnet: Refreshing state... [id=subnet-0627c243556afe8d1]

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

Terraform will perform the following actions:

  # aws_security_group.web_sg will be created
  + resource "aws_security_group" "web_sg" {
      + arn                     = (known after apply)

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

4.4 Configure EC2 Instance

Screenshot: ec2_instance.png

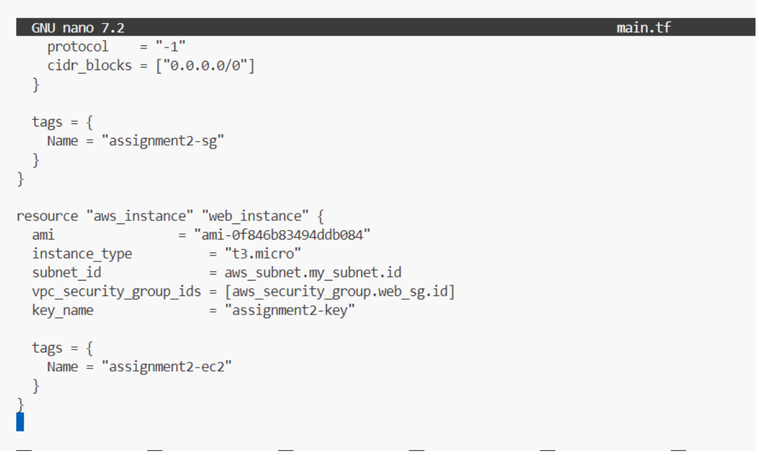


```
GNU nano 7.2                                                          main.tf *
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "assignment2-sg"
  }
}

resource "aws_instance" "web_instance" {
  ami                    = "ami-0c02fb55956c7d316"   # Amazon Linux 2 (works in most regions)
  instance_type          = "t2.micro"
  subnet_id              = aws_subnet.my_subnet.id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = "lab12-key"   # use YOUR key name

  tags = {
    Name = "assignment2-ec2"
  }
}
```

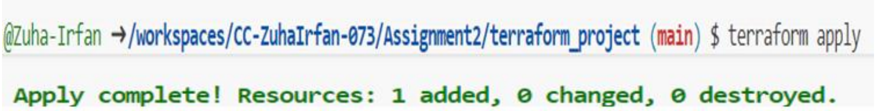


```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ nano main.tf
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ ls ~/.ssh
id_rsa  id_rsa.pub
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ aws ec2 import-key-pair \
  --key-name assignment2-key \
  --public-key-material fileb://~/.ssh/id_rsa.pub
{
    "KeyFingerprint": "ad:5b:90:e0:06:f3:6a:9c:2b:c3:32:b5:29:0c:d7:c0",
    "KeyName": "assignment2-key",
    "KeyPairId": "key-037cbdfac1e573c9f"
}
```

Screenshot: terraform_apply_ec2.png



```
GNU nano 7.2                                                  main.tf
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "assignment2-sg"
  }
}

resource "aws_instance" "web_instance" {
  ami                 = "ami-0f846b83494ddb084"
  instance_type          = "t3.micro"
  subnet_id              = aws_subnet.my_subnet.id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = "assignment2-key"

  tags = {
    Name = "assignment2-ec2"
  }
}
```

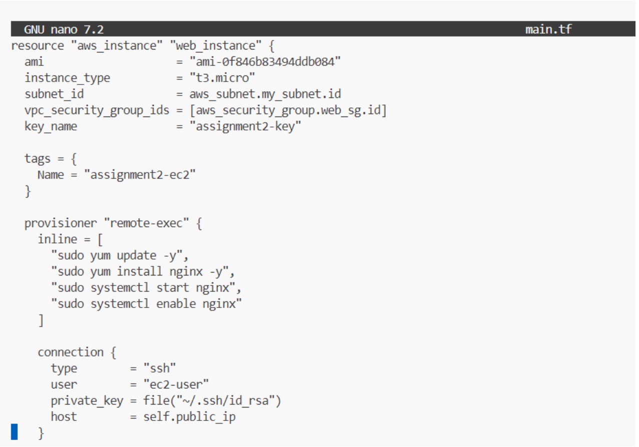


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform apply

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Phase 5: Remote Provisioning

5.1 Configure Remote-Exec Provisioner

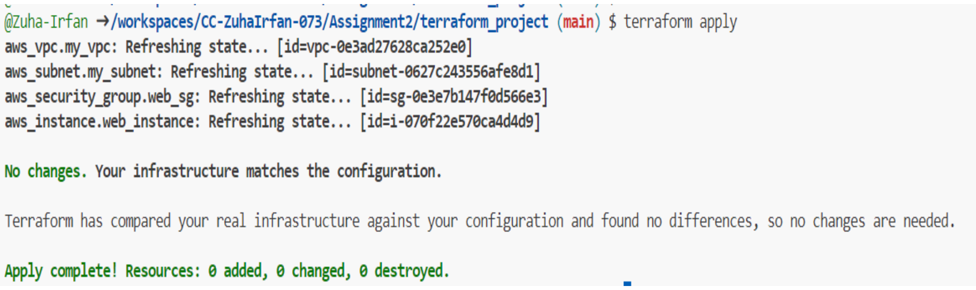
Screenshot: remote_exec.png



```
GNU nano 7.2                                                    main.tf
resource "aws_instance" "web_instance" {
  ami                    = "ami-0f846b83494ddb084"
  instance_type          = "t3.micro"
  subnet_id              = aws_subnet.my_subnet.id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = "assignment2-key"

  tags = {
    Name = "assignment2-ec2"
  }

  provisioner "remote-exec" {
    inline = [
      "sudo yum update -y",
      "sudo yum install nginx -y",
      "sudo systemctl start nginx",
      "sudo systemctl enable nginx"
    ]

    connection {
      type        = "ssh"
      user        = "ec2-user"
      private_key = file("~/.ssh/id_rsa")
      host        = self.public_ip
    }
```

5.2 Apply Terraform Configuration

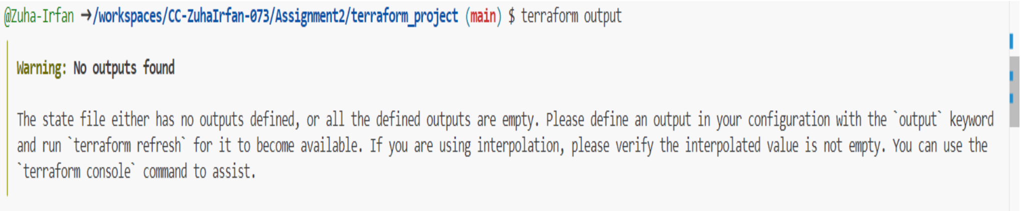
Screenshot: terraform_apply.png



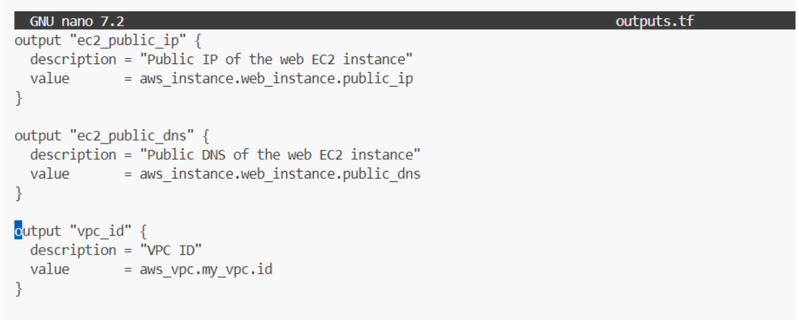
```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform apply
aws_vpc.my_vpc: Refreshing state... [id=vpc-0e3ad27628ca252e0]
aws_subnet.my_subnet: Refreshing state... [id=subnet-0627c243556afe8d1]
aws_security_group.web_sg: Refreshing state... [id=sg-0e3e7b147f0d566e3]
aws_instance.web_instance: Refreshing state... [id=i-070f22e570ca4d4d9]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

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

5.3 Verify Resource Creation

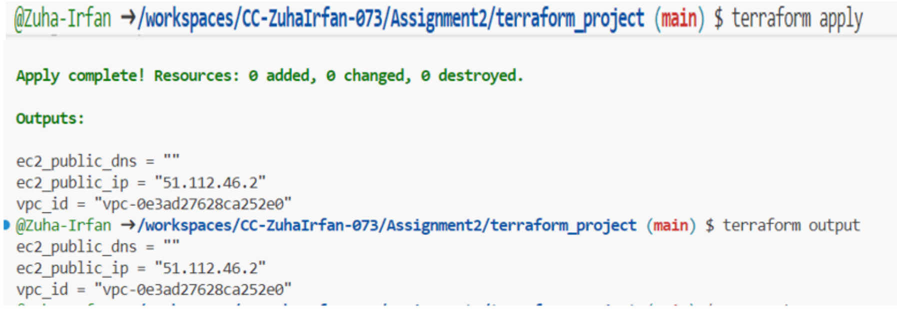
Screenshot: terraform_output.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform output

│ Warning: No outputs found
│
│ The state file either has no outputs defined, or all the defined outputs are empty. Please define an output in your configuration with the `output` keyword
│ and run `terraform refresh` for it to become available. If you are using interpolation, please verify the interpolated value is not empty. You can use the
│ `terraform console` command to assist.
```

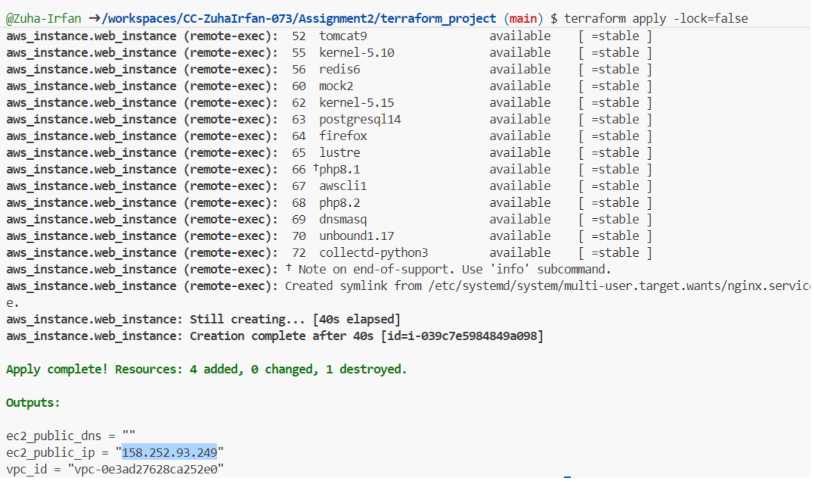


```
  GNU nano 7.2                                                         outputs.tf
output "ec2_public_ip" {
  description = "Public IP of the web EC2 instance"
  value       = aws_instance.web_instance.public_ip
}

output "ec2_public_dns" {
  description = "Public DNS of the web EC2 instance"
  value       = aws_instance.web_instance.public_dns
}

output "vpc_id" {
  description = "VPC ID"
  value       = aws_vpc.my_vpc.id
}
```

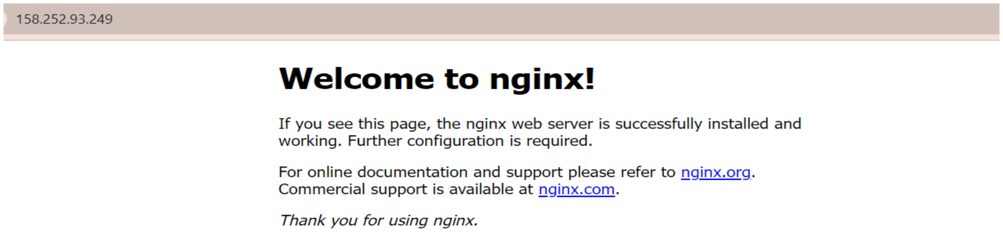


```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform apply

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

Outputs:

ec2_public_dns = ""
ec2_public_ip = "51.112.46.2"
vpc_id = "vpc-0e3ad27628ca252e0"
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform output
ec2_public_dns = ""
ec2_public_ip = "51.112.46.2"
vpc_id = "vpc-0e3ad27628ca252e0"
```

Screenshot: nginx_browser.png (if Nginx is installed)



```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ terraform apply -lock=false
aws_instance.web_instance (remote-exec):  52  tomcat9                  available    [ =stable ]
aws_instance.web_instance (remote-exec):  55  kernel-5.10              available    [ =stable ]
aws_instance.web_instance (remote-exec):  56  redis6                   available    [ =stable ]
aws_instance.web_instance (remote-exec):  60  mock2                    available    [ =stable ]
aws_instance.web_instance (remote-exec):  62  kernel-5.15              available    [ =stable ]
aws_instance.web_instance (remote-exec):  63  postgresql14             available    [ =stable ]
aws_instance.web_instance (remote-exec):  64  firefox                  available    [ =stable ]
aws_instance.web_instance (remote-exec):  65  lustre                   available    [ =stable ]
aws_instance.web_instance (remote-exec):  66 †php8.1                   available    [ =stable ]
aws_instance.web_instance (remote-exec):  67  awscli1                  available    [ =stable ]
aws_instance.web_instance (remote-exec):  68  php8.2                   available    [ =stable ]
aws_instance.web_instance (remote-exec):  69  dnsmasq                  available    [ =stable ]
aws_instance.web_instance (remote-exec):  70  unbound1.17              available    [ =stable ]
aws_instance.web_instance (remote-exec):  72  collectd-python3         available    [ =stable ]
aws_instance.web_instance (remote-exec): † Note on end-of-support. Use 'info' subcommand.
aws_instance.web_instance (remote-exec): Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.servic
e.
aws_instance.web_instance: Still creating... [40s elapsed]
aws_instance.web_instance: Creation complete after 40s [id=i-039c7e5984849a098]

Apply complete! Resources: 4 added, 0 changed, 1 destroyed.

Outputs:

ec2_public_dns = ""
ec2_public_ip = "158.252.93.249"
vpc_id = "vpc-0e3ad27628ca252e0"
```



158.252.93.249

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

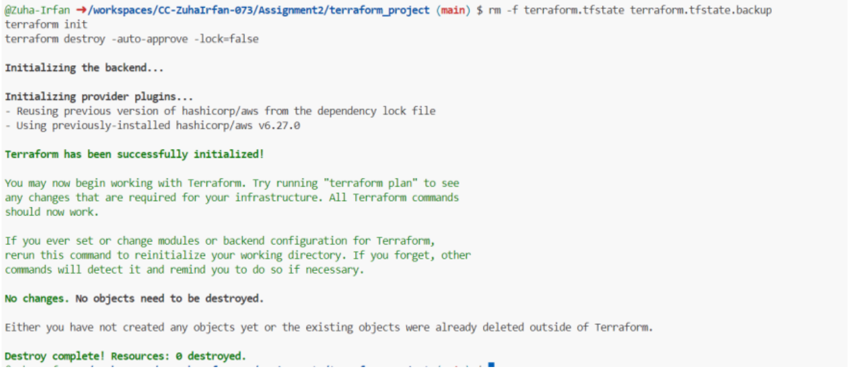
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Phase 6: Cleanup Resource

6.1 Destroy Terraform Resources

Screenshot: terraform_destroy.png



```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Assignment2/terraform_project (main) $ rm -f terraform.tfstate terraform.tfstate.backup
terraform init
terraform destroy -auto-approve -lock=false

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0

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.

No changes. No objects need to be destroyed.

Either you have not created any objects yet or the existing objects were already deleted outside of Terraform.

Destroy complete! Resources: 0 destroyed.
```