Name: Maira Malik

Reg.no: 2023-BSE-040

Subject: Cloud Computing

LAB # 12

Task 1:

1.

```
@Maira222 →/workspaces/lab12 (main) $ mkdir -p ~/Lab12
@Maira222 →/workspaces/lab12 (main) $ cd ~/Lab12
@Maira222 →~/Lab12 $
```

2.

```
@Maira222 →~/Lab12 $ touch main.tf variables.tf outputs.tf locals.tf terraform.t
fvars entry-script.sh
@Maira222 →~/Lab12 $ ls -la
total 12
drwxr-xr-x 2 codespace codespace 4096 Jan 10 07:09 .
drwxr-x--- 1 codespace codespace 4096 Jan 10 07:06 ..
-rw-r--r-- 1 codespace codespace    0 Jan 10 07:09 entry-script.sh
-rw-r--r-- 1 codespace codespace    0 Jan 10 07:09 locals.tf
-rw-r--r-- 1 codespace codespace    0 Jan 10 07:09 main.tf
-rw-r--r-- 1 codespace codespace    0 Jan 10 07:09 outputs.tf
-rw-r--r-- 1 codespace codespace    0 Jan 10 07:09 terraform.tfvars
-rw-r--r-- 1 codespace codespace    0 Jan 10 07:09 variables.tf
```

3.

```
 GNU nano 7.2                                variables.tf
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
variable "public_key" {}
variable "private_key" {}
```

4.

```
 GNU nano 7.2                                outputs.tf
output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}
```

5.

```
 GNU nano 7.2                                locals.tf
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}
```

6.

```
  GNU nano 7.2                          terraform.tfvars *
vpc_cidr_block = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix = "dev"
instance_type = "t3.micro"
public_key = "~/.ssh/id_ed25519.pub"
private_key = "~/.ssh/id_ed25519"
```

7.

```
  GNU nano 7.2                                  main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
     Name = "${var.env_prefix}-vpc"
  }
}
```

8.

```
  GNU nano 7.2                                entry-script.sh
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
```

9.

```
@Maira222 ➜~/Lab12 $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
The key fingerprint is:
SHA256:eLBSO5YR8G3a2Td65S/KYt6/cimUy43gEBbTHq13558 codespace@codespaces-e9d4dd
The key's randomart image is:
+--[ED25519 256]--+
|    ...  . .     |
|     . oo o .    |
|      = o+ o     |
|     . @ooo . . .|
|    . O.S...oo.o |
|     o o. .oo+  .|
|         o.+.+..o|
|          ++* =E.|
|         o..+*oo.|
+----[SHA256]-----+
@Maira222 ➜~/Lab12 $
```

10.

```
@Maira222 ➜~/Lab12 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (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.
@Maira222 ➜~/Lab12 $
```

11.

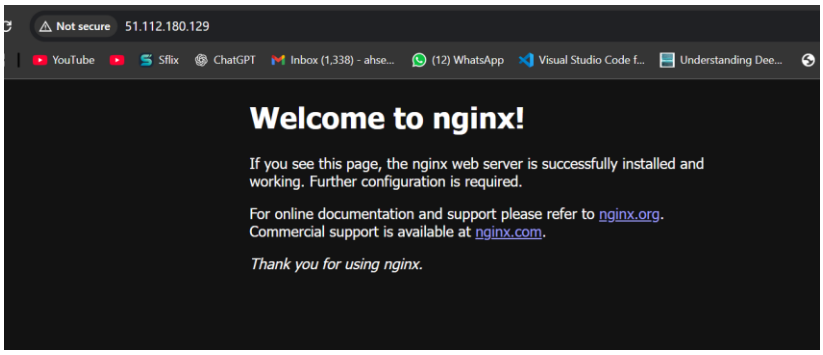
```
@Maira222 ➜~/Lab12 $ terraform apply -auto-approve
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]

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

Terraform will perform the following actions:

  # aws_default_route_table.main_rt will be created
  + resource "aws_default_route_table" "main_rt" {
      + arn                    = (known after apply)
      + default_route_table_id = (known after apply)
      + id                     = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "me-central-1"
      + route                  = [
          + {
              + cidr_block                = "0.0.0.0/0"
              + gateway_id                = (known after apply)
                # (10 unchanged attributes hidden)
```

```
@Maira222 ➜~/Lab12 $ terraform apply -auto-approve
          + "Name" = "dev-vpc"
        }
    }

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
aws_key_pair.ssh-key: Creation complete after 1s [id=serverkey]
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-0cc1c2f90912c6ddc]
aws_internet_gateway.myapp_igw: Creating...
aws_subnet.myapp_subnet_1: Creating...
aws_default_security_group.default_sg: Creating...
aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-073c23e396f367307]
aws_default_route_table.main_rt: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 0s [id=subnet-0c780bea9c0bce6f7]
aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-0f628655b7fef6e2f]
aws_default_security_group.default_sg: Creation complete after 2s [id=sg-0f1c3383dd024af47]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-0ac4767cfc5b899d9]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
```

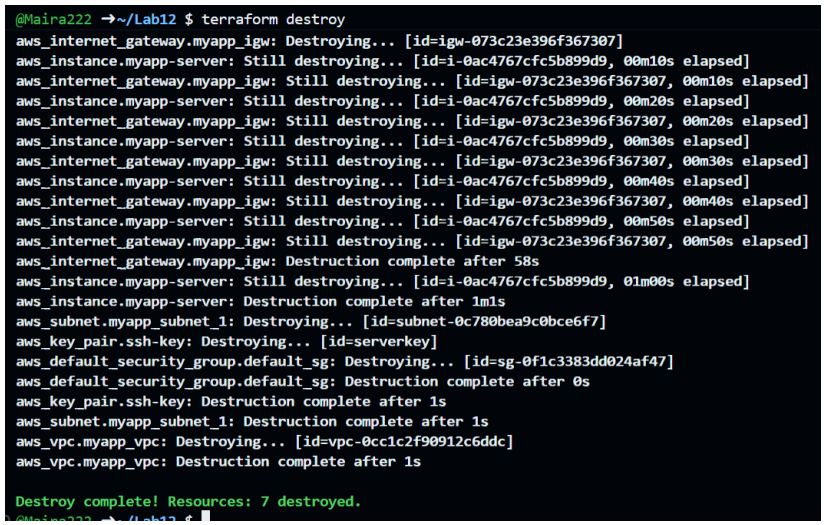
12.

```
@Maira222 ➜~/Lab12 $ terraform output
aws_instance_public_ip = "51.112.180.129"
@Maira222 ➜~/Lab12 $
```

13.



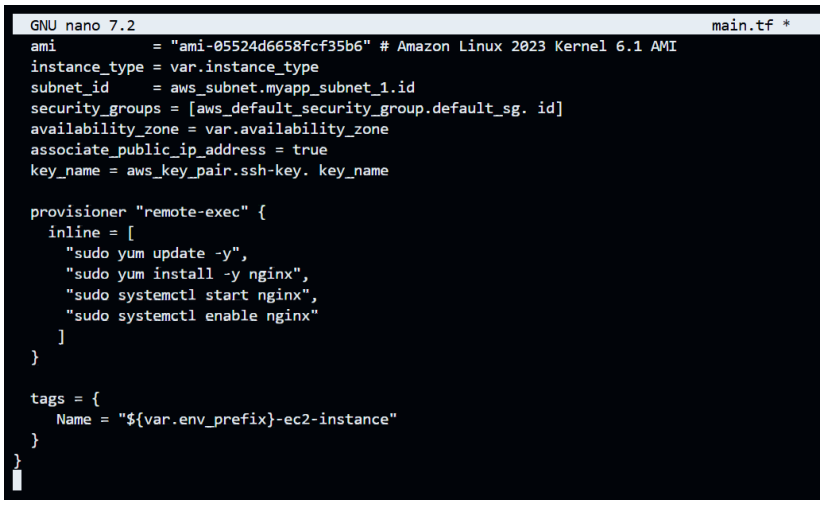
14.



```
@Maira222 →~/Lab12 $ terraform destroy
aws_internet_gateway.myapp_igw: Destroying... [id=igw-073c23e396f367307]
aws_instance.myapp-server: Still destroying... [id=i-0ac4767cfc5b899d9, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-073c23e396f367307, 00m10s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0ac4767cfc5b899d9, 00m20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-073c23e396f367307, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0ac4767cfc5b899d9, 00m30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-073c23e396f367307, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0ac4767cfc5b899d9, 00m40s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-073c23e396f367307, 00m40s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0ac4767cfc5b899d9, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-073c23e396f367307, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 58s
aws_instance.myapp-server: Still destroying... [id=i-0ac4767cfc5b899d9, 01m00s elapsed]
aws_instance.myapp-server: Destruction complete after 1m1s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0c780bea9c0bce6f7]
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-0f1c3383dd024af47]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_key_pair.ssh-key: Destruction complete after 1s
aws_subnet.myapp_subnet_1: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0cc1c2f90912c6ddc]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

TASK 2:

1.

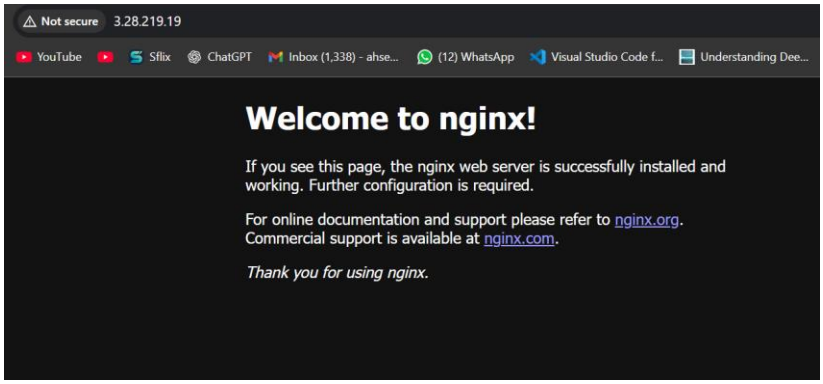


```
GNU nano 7.2                                                    main.tf *
  ami           = "ami-05524d6658fcf35b6" # Amazon Linux 2023 Kernel 6.1 AMI
  instance_type = var.instance_type
  subnet_id     = aws_subnet.myapp_subnet_1.id
  security_groups = [aws_default_security_group.default_sg. id]
  availability_zone = var.availability_zone
  associate_public_ip_address = true
  key_name = aws_key_pair.ssh-key. key_name

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

  tags = {
     Name = "${var.env_prefix}-ec2-instance"
  }
}
```

2.

```
@Maira222 →~/Lab12 $ terraform apply -auto-approve

aws_instance.myapp-server (remote-exec):     Release notes:
aws_instance.myapp-server (remote-exec):      https://docs.aws.amazon.com/linux/al2023/release-notes/re

aws_instance.myapp-server (remote-exec): ======================================

aws_instance.myapp-server (remote-exec): Installed:
aws_instance.myapp-server (remote-exec):   generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
aws_instance.myapp-server (remote-exec):   gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
aws_instance.myapp-server (remote-exec):   libunwind-1.4.0-5.amzn2023.0.3.x86_64
aws_instance.myapp-server (remote-exec):   nginx-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp-server (remote-exec):   nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp-server (remote-exec):   nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
aws_instance.myapp-server (remote-exec):   nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/ng
aws_instance.myapp-server: Creation complete after 32s [id=i-0e5ba805d96b6f942]

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

Outputs:

aws_instance_public_ip = "3.28.219.19"
@Maira222 →~/Lab12 $
```

3.



TASK 3:

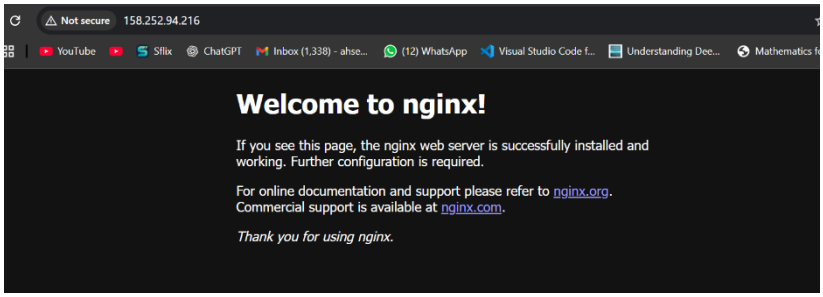
1.

```
GNU nano 7.2                                                    main.tf *
  host        = self.public_ip
}
provisioner "file" {
  source = "./entry-script.sh"
  destination = "/home/ec2-user/entry-script-on-ec2.sh"
}

provisioner "remote-exec" {
  inline = [
    "sudo chmod +x /home/ec2-user/entry-script-on-ec2.sh",
    "sudo /home/ec2-user/entry-script-on-ec2.sh"
  ]
}

provisioner "local-exec" {
  command = <<-EOF
    echo Instance ${self.id} with public IP ${self.public_ip} has been created
  EOF
}

tags = {
  Name = "${var.env_prefix}-ec2-instance"
```

2.

```
@Maira222 →~/Lab12 $ terraform apply -auto-approve

aws_instance.myapp-server (remote-exec): ======================================

aws_instance.myapp-server (remote-exec): Installed:
aws_instance.myapp-server (remote-exec):   generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
aws_instance.myapp-server (remote-exec):   gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
aws_instance.myapp-server (remote-exec):   libunwind-1.4.0-5.amzn2023.0.3.x86_64
aws_instance.myapp-server (remote-exec):   nginx-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp-server (remote-exec):   nginx-core-1:1.28.0-1.amzn2023.0.2.x86_64
aws_instance.myapp-server (remote-exec):   nginx-filesystem-1:1.28.0-1.amzn2023.0.2.noarch
aws_instance.myapp-server (remote-exec):   nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/
aws_instance.myapp-server: Provisioning with 'local-exec'...
aws_instance.myapp-server (local-exec): Executing: ["/bin/sh" "-c" "echo Instance i-0303405870d484677
aws_instance.myapp-server (local-exec): Instance i-0303405870d484677 with public IP 158.252.94.216 ha
aws_instance.myapp-server: Creation complete after 59s [id=i-0303405870d484677]

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

Outputs:

aws_instance_public_ip = "158.252.94.216"
```

3.



4.

```
@Maira222 →~/Lab12 $ terraform destroy
aws_default_route_table.main_rt: Destruction complete after 0s
aws_internet_gateway.myapp_igw: Destroying... [id=igw-00fc407d1d5bd7da7]
aws_instance.myapp-server: Still destroying... [id=i-0303405870d484677, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00fc407d1d5bd7da7, 00m10s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0303405870d484677, 00m20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00fc407d1d5bd7da7, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0303405870d484677, 00m30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00fc407d1d5bd7da7, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0303405870d484677, 00m40s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00fc407d1d5bd7da7, 00m40s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 48s
aws_instance.myapp-server: Still destroying... [id=i-0303405870d484677, 00m50s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0303405870d484677, 01m00s elapsed]
aws_instance.myapp-server: Destruction complete after 1m1s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-04d3bc2aa8791f0af]
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-04c635071c9fcf936]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_key_pair.ssh-key: Destruction complete after 0s
aws_subnet.myapp_subnet_1: Destruction complete after 0s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0ccfcbafa191c4ce9]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

5.

```
  GNU nano 7.2                                                    main.tf *
resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6" # Amazon Linux 2023 Kernel 6.1 AMI
  instance_type = var.instance_type
  subnet_id     = aws_subnet.myapp_subnet_1.id
  security_groups = [aws_default_security_group.default_sg. id]
  availability_zone = var.availability_zone
  associate_public_ip_address = true
  key_name = aws_key_pair.ssh-key. key_name

  user_data = file("./entry-script.sh")

  tags = {
     Name = "${var.env_prefix}-ec2-instance"
  }
}
```

TASK 4:

1.

```
├── awscliv2.zip
├── entry-script.sh
├── locals.tf
├── main.tf
├── modules
│   └── subnet
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
└── variables.tf
```

2.

```
@Maira222 →~/Lab12/modules/subnet $ cat variables.tf
variable "vpc_id" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "default_route_table_id" {}
```

3.

```
@Maira222 →~/Lab12/modules/subnet $ cat main.tf
  map_public_ip_on_launch = true
  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }
}

resource "aws_default_route_table" "main_rt" {
  default_route_table_id = var.default_route_table_id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }
    tags = {
     Name = "${var.env_prefix}-rt"
  }
}

resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = var.vpc_id
    tags = {
     Name = "${var.env_prefix}-igw"
  }
}
```

4.

```
@Maira222 →~/Lab12/modules/subnet $ cat outputs.tf
output "subnet" {
  value = aws_subnet.myapp_subnet_1
}
```

5.

```
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
     Name = "${var.env_prefix}-vpc"
  }
}
module "myapp-subnet" {
  source = "./modules/subnet"
  vpc_id = aws_vpc.myapp_vpc. id
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}

resource "aws_default_security_group" "default_sg" {
  vpc_id      = aws_vpc.myapp_vpc.id

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify
espaces: refactored broccoli  main*  ⊗ 0 ⚠ 0  0
```

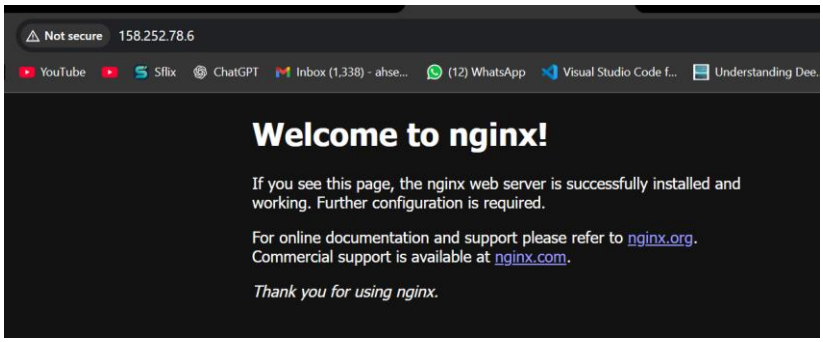
6.

```
@Maira222 →~/Lab12 $ terraform init
Initializing the backend...
Initializing modules...
- myapp-subnet in modules/subnet
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/aws v6.28.0
- Using previously-installed hashicorp/http v3.5.0

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.
@Maira222 →~/Lab12 $
```

7.

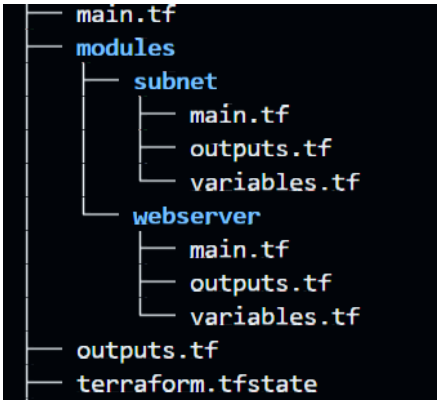
```
@Maira222 →~/Lab12 $ terraform apply -auto-approve
Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
aws_key_pair.ssh-key: Creation complete after 1s [id=serverkey]
aws_vpc.myapp_vpc: Creation complete after 1s [id=vpc-078d8836724950054]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
aws_default_security_group.default_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-05543703669ab4c78]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-0d9851fbe61e1d41b]
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-06d540f89be43f112]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-003b3a5e8dfc2194a]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-04eb8d46c6041031c]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

aws_instance_public_ip = "158.252.78.6"
```

8.



TASK 5:

1.



2.

```
@Maira222 →~/Lab12/modules/webserver $ cat variables.tf
variable "env_prefix" {}
variable "instance_type" {}
variable "availability_zone" {}
variable "public_key" {}
variable "my_ip" {}
variable "vpc_id" {}
variable "subnet_id" {}
variable "script_path" {}
variable "instance_suffix" {}
@Maira222 →~/Lab12/modules/webserver $
```

3.

```
@Maira222 →~/Lab12/modules/webserver $ cat main.tf
    Name = "${var.env_prefix}-default-sg"
  }
}

resource "aws_key_pair" "ssh-key" {
  key_name = "${var.env_prefix}-serverkey-${var.instance_suffix}"
  public_key = file(var.public_key)
}

resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6" # Amazon Linux 2023 Kernel 6.1 AMI
  instance_type = var. instance_type
  subnet_id     = var.subnet_id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  availability_zone = var.availability_zone
  associate_public_ip_address = true
  key_name = aws_key_pair.ssh-key.key_name

  user_data = file(var.script_path)

  tags = {
    Name = "${var.env_prefix}-ec2-instance-${var.instance_suffix}"
```

4.

```
@Maira222 →~/Lab12/modules/webserver $ cat outputs.tf
output "aws_instance" {
  value = aws_instance.myapp-server
}
@Maira222 →~/Lab12/modules/webserver $
```

5.

```
 GNU nano 7.2                                                    main.tf
}
module "myapp-subnet" {
  source = "./modules/subnet"
  vpc_id = aws_vpc.myapp_vpc. id
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}

module "myapp-webserver" {
  source = "./modules/webserver"
  env_prefix = var.env_prefix
  instance_type = var. instance_type
  availability_zone = var.availability_zone
  public_key = var.public_key
  my_ip = local.my_ip
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id
  script_path = "./entry-script.sh"
  instance_suffix = "0"
}
```

6.

```
 GNU nano 7.2
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}
```

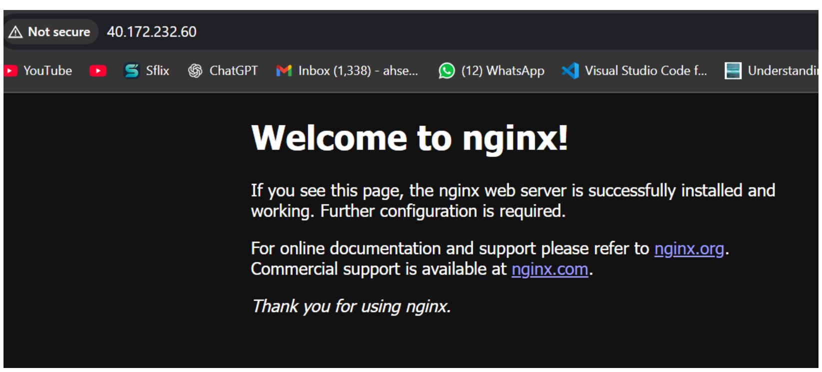
7.

```
@Maira222 →~/Lab12 $ terraform init
Initializing the backend...
Initializing modules...
- myapp-webserver in modules/webserver
Initializing provider plugins...
- Reusing previous version of hashicorp/http from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/http v3.5.0
- Using previously-installed hashicorp/aws v6.28.0

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.
@Maira222 →~/Lab12 $
```

8.

```
@Maira222 →~/Lab12 $ terraform apply -auto-approve
aws_instance.myapp-server: Destroying... [id=i-04eb8d46c6041031c]
module.myapp-webserver.aws_security_group.web_sg: Creating...
module.myapp-webserver.aws_key_pair.ssh-key: Creation complete after 1s [id=dev-serverkey-0]
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-03ffda6917959f30e]
module.myapp-webserver.aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still destroying... [id=i-04eb8d46c6041031c, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 14s [id=i-0cf064d9c3e3b042a]
aws_instance.myapp-server: Still destroying... [id=i-04eb8d46c6041031c, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-04eb8d46c6041031c, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-04eb8d46c6041031c, 00m40s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-04eb8d46c6041031c, 00m50s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-04eb8d46c6041031c, 01m00s elapsed]
aws_instance.myapp-server: Destruction complete after 1m1s
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-06d540f89be43f112]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_key_pair.ssh-key: Destruction complete after 0s

Apply complete! Resources: 3 added, 0 changed, 3 destroyed.

Outputs:

webserver_public_ip = "40.172.232.60"
```

9.



10.

```
@Maira222 →~/Lab12 $ terraform destroy
module.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp-webserver.aws_instance.myapp-server: Destroying... [id=i-0cf064d9c3e3b042a]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-05543703669ab4c78]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0cf064d9c3e3b042a
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05543703669ab
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0cf064d9c3e3b042a
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05543703669ab
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0cf064d9c3e3b042a
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05543703669ab
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0cf064d9c3e3b042a
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-05543703669ab
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 47s
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0cf064d9c3e3b042a
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 50s
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-03ffda6917959f30e]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-003b3a5e8dfc2194a]
module.myapp-webserver.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-webserver.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-078d8836724950054]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Maira222 →~/Lab12 $
```

TASK 6:

1.

```
  GNU nano 7.2                                                  entry-script.sh
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
# Create directories for SSL certificates if they don't exist
mkdir -p /etc/ssl/private
mkdir -p /etc/ssl/certs

# Get IMDSv2 token
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \
  -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Get current public IP
PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
  http://169.254.169.254/latest/meta-data/public-ipv4)

PUBLIC_HOSTNAME=$(curl -s -H "X-aws-ec2-metadata-token:  $TOKEN" \
  http://169.254.169.254/latest/meta-data/public-hostname)

# Generate self-signed certificate with dynamic IP
```

2.

```
@Maira222 →~/Lab12 $ terraform apply -auto-approve
Changes to Outputs:
  + webserver_public_ip = (known after apply)
module.myapp-webserver.aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
module.myapp-webserver.aws_key_pair.ssh-key: Creation complete after 1s [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-0fca85775e706a268]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-webserver.aws_security_group.web_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-0195e09ba578
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-09eb8379bb8
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-0272eb968e64f2c
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-0609cb0b672be
module.myapp-webserver.aws_instance.myapp-server: Creating...
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 12s [id=i-08fb597c792ce8b

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

Outputs:

webserver_public_ip = "3.28.46.199"
```