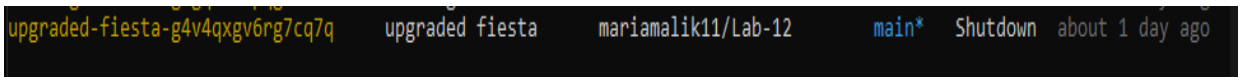
CLOUD COMPUTING LAB

Name: Maria Abdul Malik
Reg No: 2023-BSE-076
Submitted to: Sir Muhammad Shoaib
Class: 5B

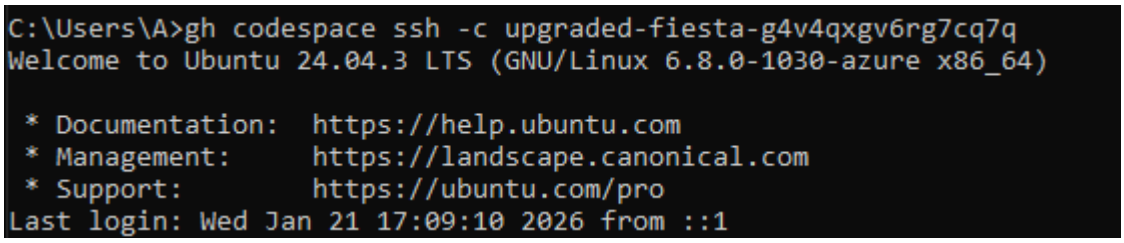
LAB #12

Task 0 Lab Setup (Codespace & GH CLI)

- task0_codespace_create_and_list. png

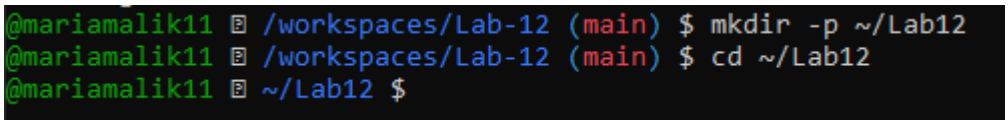


- task0_codespace_ssh_connected.png

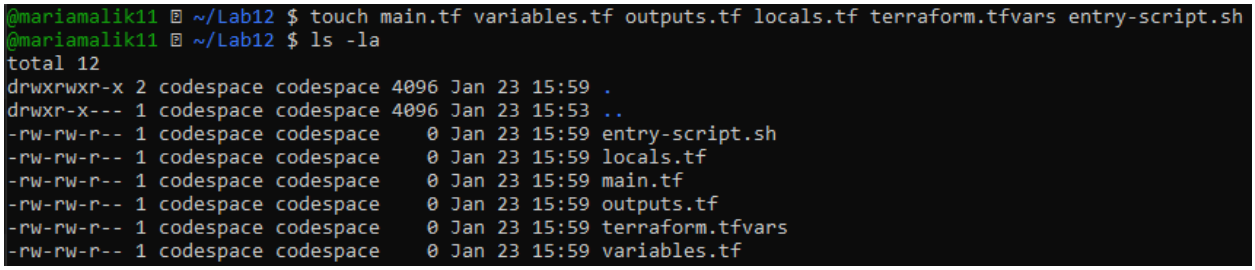


Task 1 — Organize Terraform code into separate files

- task1_project_directory.png



- task1_files_created.png



- task1_variables_tf.png

```
mariamalik11 🖹 ~/Lab12 $ cat variables.tf
ariable "vpc_cidr_block" {}
ariable "subnet_cidr_block" {}
ariable "availability_zone" {}
ariable "env_prefix" {}
ariable "instance_type" {}
ariable "public_key" {}
ariable "private_key" {}
```

- task1_outputs_tf.png

```
mariamalik11 🖹 ~/Lab12 $ cat outputs.tf
utput "aws_instance_public_ip" {
 value = aws_instance.myapp-server.public_ip
```

- task1_locals_tf.png

```
@mariamalik11 🖹 ~/Lab12 $ cat locals.tf
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

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

- task1_terraform_tfvars.png

```
@mariamalik11 🖹 ~/Lab12 $ cat 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"
```

- task1_main_tf.png

```
@mariamalik11 🖹 ~/Lab12 $ cat 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"
  }
}

resource "aws_subnet" "myapp_subnet_1" {
  vpc_id     = aws_vpc.myapp_vpc.id
  cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }
}

resource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.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 = aws_vpc.myapp_vpc.id
    tags = {
     Name = "${var.env_prefix}-igw"
  }
```

```
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
    prefix_list_ids = []
}
  tags = {
     Name = "${var.env_prefix}-default-sg"
  }
}

resource "aws_key_pair" "ssh-key" {
  key_name = "serverkey"
  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     = 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"
  }
}
```

- task1_entry_script.png

```
@mariamalik11 🖹 ~/Lab12 $ cat entry-script.sh
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
```

- task1_ssh_keygen.png

```
@mariamalik11 🖹 ~/Lab12 $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:tfSgr5qJtngIJUIkv3a8uGY43js71nVBF+4JxIkYf4w codespace@codespaces-2240d3
The key's randomart image is:
+--[ED25519 256]--+
|o.   .o o....    |
|.o   ...++..     |
|. .    E.+=.     |
|o .o    .=o+.    |
|.oo o   S oo.    |
|.. o . . o       |
| o..o . . .      |
|o =*oo o .       |
|.=+*B.+..        |
+----[SHA256]-----+
```

- task1_terraform_init.png

```
@mariamalik11 🖹 ~/Lab12 $ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/http...
- Finding latest version of hashicorp/aws...
- 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.
```

- task1_terraform_apply.png

```
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-02fabc07bc
ca308]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-0564ac2723057c4f5]

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

Outputs:

aws_instance_public_ip = "5.21.09.221"
```

- task1_terraform_output.png

```
@mariamalik11 🖹 ~/Lab12 $ terraform output
aws_instance_public_ip = "5.21.09.221"
```

- task1_nginx_browser.png

- task1_terraform_destroy.png

```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_default_route_table.main_rt: Destroying... [id=rtb-008083ed5ada8c4d0]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_instance.myapp-server: Destroying... [id=i-0564ac2723057c4f5]
aws_internet_gateway.myapp_igw: Destroying... [id=igw-0d81ecdb29f68a149]
aws_instance.myapp-server: Still destroying... [id=i-0564ac2723057c4f5, 10s elapsed
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0d81ecdb29f68a149, 10s
apsed]
aws_instance.myapp-server: Still destroying... [id=i-0564ac2723057c4f5, 20s elapsed
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0d81ecdb29f68a149, 20s
apsed]
aws_internet_gateway.myapp_igw: Destruction complete after 28s
aws_instance.myapp-server: Still destroying... [id=i-0564ac2723057c4f5, 30s elapsed
aws_instance.myapp-server: Destruction complete after 31s
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-031b427a769235585]
aws_default_security_group.default_sg: Destroying... [id=sg-02fabc07bc3eca308]
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-071fdc19a582b2f1a]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

Task 2 — Use remote-exec provisioner

- task2_main_tf_remote_exec.png

```
resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6"
  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

  connection {
    type        = "ssh"
    user        = "ec2-user"
    private_key = file(var.private_key)
    host        = self.public_ip
  }

  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"
  }
}
```

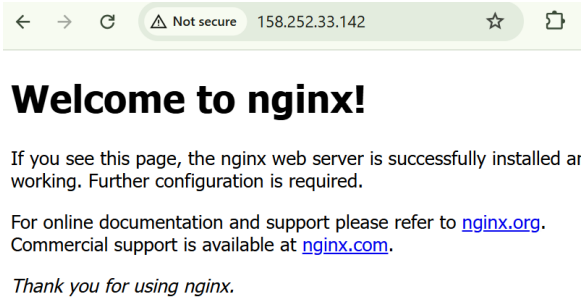
- task2_terraform_apply.png

```
aws_instance.myapp-server (remote-exec):   nginx-mimetypes-2.1.49-3.amzn2023.0.3.no
ch

aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/systemd/system/multi-
er.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
aws_instance.myapp-server: Creation complete after 59s [id=i-0e75a5cd7748d312a]

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

Outputs:

aws_instance_public_ip = "158.252.33.142"
```

- task2_terraform_output.png

```
@mariamalik11 ~/Lab12 $ terraform output
aws_instance_public_ip = "158.252.33.142"
```

- task2_nginx_browser.png



Task 3 — Use file and local-exec provisioners

- task3_main_tf_all_provisioners.png

Command Prompt - gh codespace ssh -c curly-goldfish-jjpj5j6w7vgjcqp5x

```
  associate_public_ip_address = true
  key_name = aws_key_pair.ssh-key.key_name

  connection {
    type        = "ssh"
    user        = "ec2-user"
    private_key = file(var.private_key)
    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"
  }
}
```

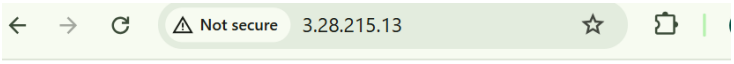
- task3_terraform_apply.png

```
aws_instance.myapp-server (remote-exec):   nginx-core-1:1.28.0-1.amzn2023.0.2.x86_6
aws_instance.myapp-server (remote-exec):   nginx-filesystem-1:1.28.0-1.amzn2023.0.2
oarch
aws_instance.myapp-server (remote-exec):   nginx-mimetypes-2.1.49-3.amzn2023.0.3.no
ch

aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/systemd/system/multi-
er.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
aws_instance.myapp-server: Provisioning with 'local-exec'...
aws_instance.myapp-server (local-exec): Executing: ["/bin/sh" "-c" "echo Instance i
2864176feec54f49 with public IP 3.28.215.13 has been created\n"]
aws_instance.myapp-server (local-exec): Instance i-02864176feec54f49 with public IP
.28.215.13 has been created
aws_instance.myapp-server: Creation complete after 1m0s [id=i-02864176feec54f49]

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

- task3_terraform_output.png

```
@mariamalik11 ~/Lab12 $ terraform output
aws_instance_public_ip = "3.28.215.13"
```

- task3_nginx_browser.png



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.

- task3_terraform_destroy.png

```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_default_route_table.main_rt: Destroying... [id=rtb-002ed38539566e539]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_instance.myapp-server: Destroying... [id=i-02864176feec54f49]
aws_internet_gateway.myapp_igw: Destroying... [id=igw-0175785c9c0c40b40]
aws_instance.myapp-server: Still destroying... [id=i-02864176feec54f49, 10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0175785c9c0c40b40, 10s el
apsed]
aws_internet_gateway.myapp_igw: Destruction complete after 18s
aws_instance.myapp-server: Still destroying... [id=i-02864176feec54f49, 20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-02864176feec54f49, 30s elapsed]
aws_instance.myapp-server: Destruction complete after 30s
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0d2d94df2a0dfbd8b]
aws_default_security_group.default_sg: Destroying... [id=sg-0ae90fe01715b962d]
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-0a811c65e89877015]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

- task3_main_tf_restored.png

```
  key_name = "serverkey"
  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     = 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 — Create Terraform modules (subnet module)

- task4_module_structure.png

```
@mariamalik11 ~/Lab12 $ vim main.tf
@mariamalik11 ~/Lab12 $ mkdir -p modules/subnet
@mariamalik11 ~/Lab12 $ touch modules/subnet/main.tf
@mariamalik11 ~/Lab12 $ touch modules/subnet/variables.tf
@mariamalik11 ~/Lab12 $ touch modules/subnet/outputs.tf
@mariamalik11 ~/Lab12 $ tree
.
├── entry-script.sh
├── locals.tf
├── main.tf
├── modules
│   └── subnet
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
├── tf
└── variables.tf

3 directories, 13 files
```

- task4_subnet_variables.png

```
tf                          cat modules/subnet/variables.tf
variable "vpc_id" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "default_route_table_id" {}
```

- task4_subnet_main.png

```
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id     = var.vpc_id
  cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  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"
  }
}
```

- task4_subnet_outputs.png

```
output "subnet" {
  value = aws_subnet.myapp_subnet_1
}
```

- task4_main_tf_with_module.png

```
  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

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
```

- task4_terraform_init.png

```
Initializing the backend...
Initializing modules...
- myapp-subnet in modules/subnet
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.
```

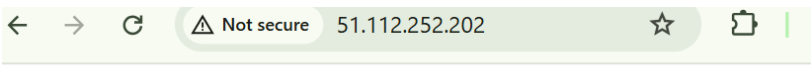
- task4_terraform_apply.png

```
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-04a63db906ae79d4a]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
aws_default_security_group.default_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 1s [id=ig
w-02acc44a19d178e38]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=r
tb-064926c4c8f485837]
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-06d8094f5843
8e536]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 12s [id=subnet
-042e083b5000d4772]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [10s elapsed]
aws_instance.myapp-server: Creation complete after 12s [id=i-0b5e33fcf8936829c]

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

- task4_terraform_output.png

```
@mariamalik11 /~/Lab12 $ terraform output
ws_instance_public_ip = "51.112.252.202"
```

- task4_nginx_browser.png



Welcome to nginx!

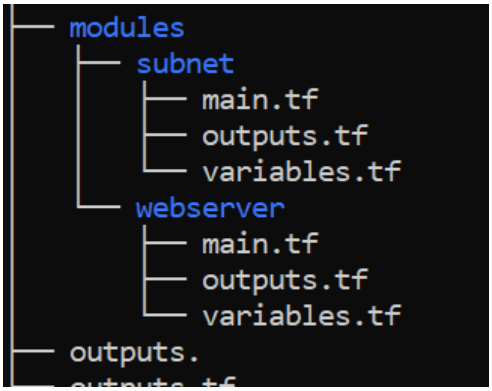
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

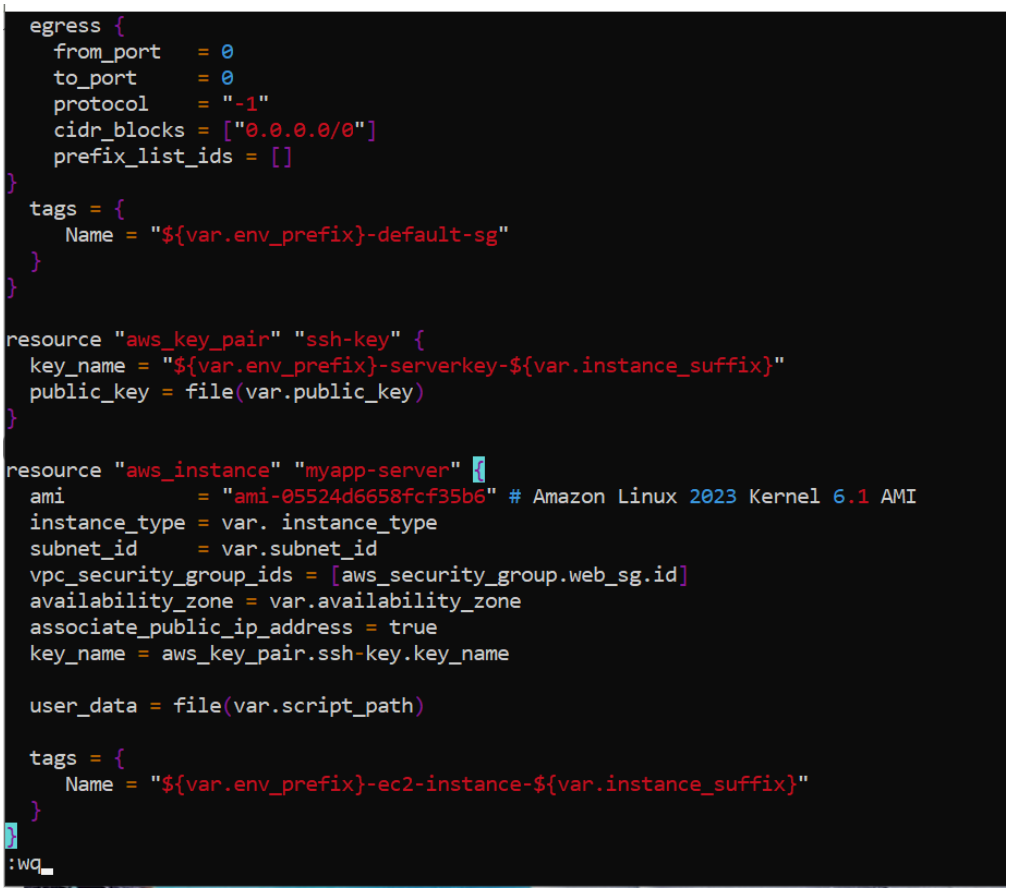
Task 5 — Create webserver module

- task5_webserver_module_structure.png



- task5_webserver_variables.png

- task5_webserver_main.png



- task5_webserver_outputs.png

```
@mariamalik11 ~/Lab12 $ cat modules/webserver/outputs.tf
output "aws_instance" {
  value = aws_instance.myapp-server
}
```

- task5_main_tf_webserver_module.png

```
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"
}
~
~
~
~
~
:wq
```

- task5_outputs_updated.png

```
@mariamalik11 ~/Lab12 $ cat outputs.tf

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

- task5_terraform_init.png

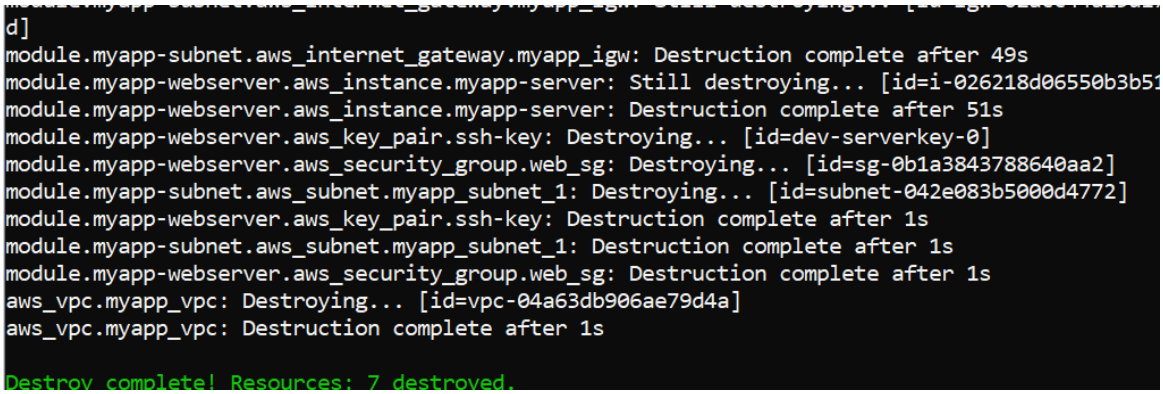
```
@mariamalik11 ~/Lab12 $ terraform init
itializing the backend...
itializing modules...
 myapp-webserver in modules/webserver
itializing provider plugins...
- Reusing previous version of hashicorp/http from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/http v3.5.0
- Using previously-installed hashicorp/aws v6.28.0

rraform has been successfully initialized!

u may now begin working with Terraform. Try running "terraform plan" to see
y changes that are required for your infrastructure. All Terraform commands
ould now work.

 you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
mmands will detect it and remind you to do so if necessary.
```

- task5_terraform_apply.png

```
aws_instance.myapp-server: Still destroying... [id=i-0b5e33fcf8936829c, 50s ela
aws_instance.myapp-server: Destruction complete after 50s
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-06d8094f58438e536]
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:
```

- task5_nginx_browser.png

- task5_terraform_destroy.png



Task 6 — Configure HTTPS with self-signed certificates

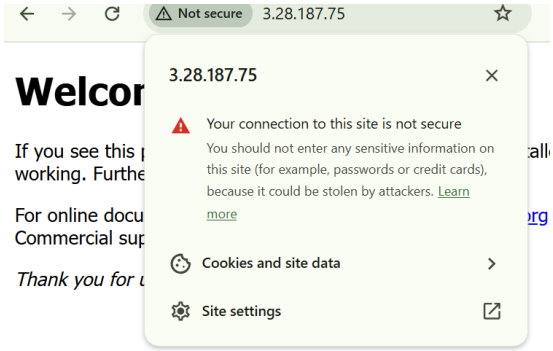
- task6_entry_script_https.png

```
    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    upstream backend_servers {
            server 158.252.94.241:80;
        server 158.252.94.242:80 backup;
    }

    server {
        listen 443 ssl;
        server_name $PUBLIC_IP;
        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;

        location / {
            root /usr/share/nginx/html;
            index index.html;
           #proxy_pass http://158.252.94.241:80;
           #proxy_pass http://backend_servers
        }
     }

    server {
        listen 80;
        server_name _;
                return 301 https://\$host\$request_uri;
                    }
            }
EOF

systemctl restart nginx
:wq
```

- task6_terraform_apply.png

```
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-090a7e35bdc3
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-0f123111e126631e
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-0cbcf80301f337
module.myapp-webserver.aws_instance.myapp-server: Creating...
module.myapp-webserver.aws_instance.myapp-server: Still creating... [10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 13s [id=i-0169035e4129e88d

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

Outputs:
```

- task6_browser_security_warning.png



- task6_nginx_https_browser.png

Task 7 — Configure Nginx as reverse proxy

- task7_apache_script.png

```
@mariamalik11 /workspaces/Lab-12 (main) $ cat > apache.sh << 'EOF'
> #!/bin/bash
> yum update -y
> yum install httpd -y
> systemctl start httpd
> systemctl enable httpd
>
> # Get IMDSv2 token for metadata
> TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds:
21600")
>
> # Create the webpage
> echo "<h1>Welcome to My Backend Web Server</h1>" > /var/www/html/index.html
> echo "<h2>Hostname: $(hostname)</h2>" >> /var/www/html/index.html
> echo "<h2>Private IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-da
ta/local-ipv4)</h2>" >> /var/www/html/index.html
> echo "<h2>Public IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-dat
a/public-ipv4)</h2>" >> /var/www/html/index.html
> echo "<h2>Deployed via Terraform</h2>" >> /var/www/html/index.html
> EOF
@mariamalik11 /workspaces/Lab-12 (main) $
```

- task7_main_tf_web1.png

```
Command Prompt - gh codespace ssh -c curly-goldfish-jjpj5j6w7vgjcqp5x
  cidr_blocks = ["0.0.0.0/0"]
 }
ngress {
  from_port   = 80
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"] # This allows anyone to view your website
 }
 # Allow all outbound traffic
 egress {
  from_port   = 0
  to_port     = 0
  protocol    = "-1"
  cidr_blocks = ["0.0.0.0/0"]
 }

esource "aws_key_pair" "ssh_key" {
 key_name   = "serverkey"
 public_key = file("~/.ssh/id_ed25519.pub")

odule "myapp-web-1" {
 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 = "./apache.sh"
 instance_suffix = "1"

wq
```

- task7_outputs_web1.png

```
  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 = "./apache.sh"
  instance_suffix = "1"
}

output "aws_web-1_public_ip" {
  value = module.myapp-web-1.aws_instance.public_ip
}
:wq
```

- task7_terraform_apply.png

```
Plan: 2 to add, 0 to change, 0 to destroy.
module.backend-server.aws_instance.myapp-server: Creating...
module.proxy-server.aws_instance.myapp-server: Creating...
module.proxy-server.aws_instance.myapp-server: Still creating... [10s elapsed]
module.backend-server.aws_instance.myapp-server: Still creating... [10s elapsed]
module.proxy-server.aws_instance.myapp-server: Creation complete after 13s [id=i-04a69a7aae4719296]
module.backend-server.aws_instance.myapp-server: Creation complete after 13s [id=i-0354ea417ed240cc3]

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

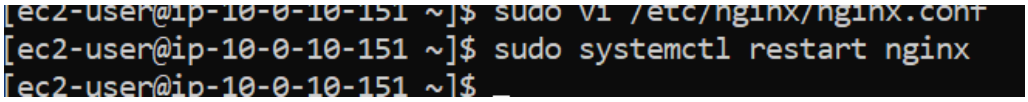
- task7_terraform_output.png

```
backend_public_ip = "51.112.51.166"
proxy_public_ip = "158.252.85.18"
```

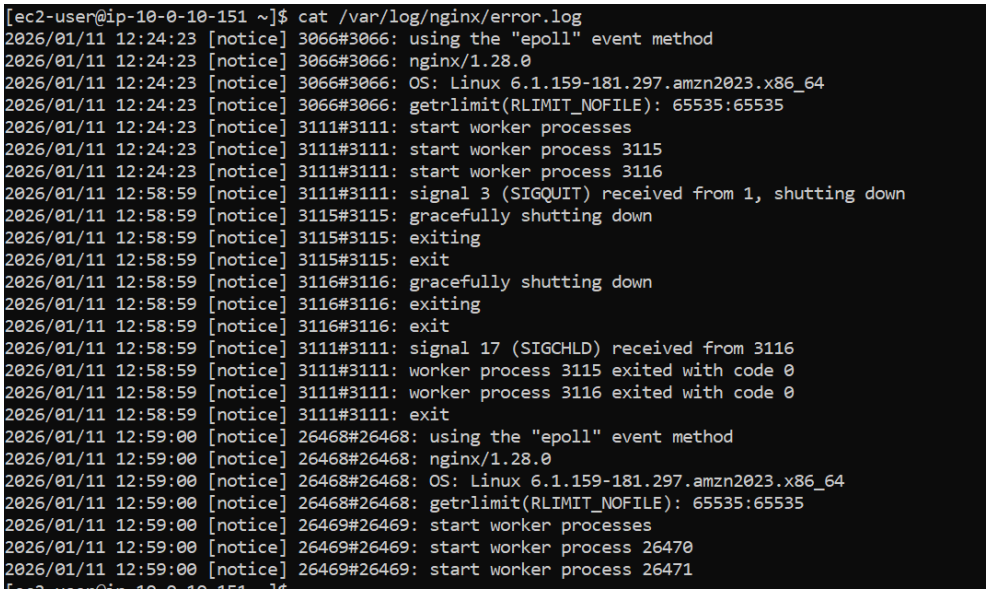
- task7_ssh_webserver.png

- task7_nginx_conf_reverse_proxy.png

- task7_nginx_restart.png



- task7_error_log.png



- task7_access_log.png

- task7_mime_types.png

```
[ec2-user@ip-10-0-10-151 ~]$ cat /etc/nginx/mime.types
types {
application/A2L                         a2l;
application/AML                         aml;
application/andrew-inset                ez;
application/ATF                         atf;
application/ATFX                        atfx;
application/ATXML                       atxml;
application/atom+xml                    atom;
application/atomcat+xml                 atomcat;
application/atomdeleted+xml             atomdeleted;
application/atomsvc+xml                 atomsvc;
application/atsc-dwd+xml                dwd;
application/atsc-held+xml               held;
application/atsc-rsat+xml               rsat;
application/auth-policy+xml             apxml;
application/bacnet-xdd+zip              xdd;
application/calendar+xml                xcs;
application/cbor                        cbor;
application/cccex                       c3ex;
application/ccmp+xml                    ccmp;
application/ccxml+xml                   ccxml;
application/CDFX+XML                    cdfx;
application/cdmi-capability             cdmia;
application/cdmi-container              cdmic;
```

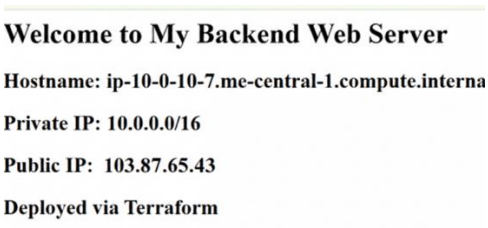
- task7_ssl_cert.png

```
[ec2-user@ip-10-0-10-151 ~]$ cat /etc/ssl/certs/selfsigned.crt
```

- task7_ssl_key.png

```
[ec2-user@ip-10-0-10-151 ~]$ sudo cat /etc/ssl/private/selfsigned.key
cat: /etc/ssl/private/selfsigned.key: No such file or directory
```

- task7_reverse_proxy_browser.png



Task 8 — Configure Nginx as load balancer

- task8_main_tf_web2.png

```
module "backend-server-2" {
  source            = "./modules/webserver"
  vpc_id            = aws_vpc.myapp_vpc.id
  subnet_id         = aws_subnet.myapp_subnet_1.id
  instance_type     = var.instance_type
  availability_zone = var.availability_zone
  env_prefix        = var.env_prefix
  key_name          = aws_key_pair.ssh_key.key_name
  public_key        = var.public_key
  my_ip             = var.my_ip
  script_path       = "./apache.sh"
  instance_suffix   = "backend-2"
}
:wq
```

- task8_outputs_web2.png

```
}

output "backend_2_public_ip" {
  value = module.backend-server-2.instance_ip
}

output "backend_2_private_ip" {
  value = module.backend-server-2.instance_private_ip
}
```

- task8_terraform_apply.png

```
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + backend_2_private_ip = (known after apply)
  + backend_2_public_ip  = (known after apply)
module.backend-server-2.aws_security_group.myapp-sg: Creating...
module.backend-server-2.aws_security_group.myapp-sg: Creation complete after 3s [id=sg-0a
module.backend-server-2.aws_instance.myapp-server: Creating...
module.backend-server-2.aws_instance.myapp-server: Still creating... [10s elapsed]
module.backend-server-2.aws_instance.myapp-server: Creation complete after 13s [id=i-0d56

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

- task8_terraform_output.png

```
backend_2_private_ip = "10.0.10.144"
backend_2_public_ip = "51.112.231.71"
backend_private_ip = "10.0.10.7"
backend_public_ip = "51.112.51.166"
proxy_public_ip = "158.252.85.18"
```

- task8_nginx_conf_load_balancer.png

```
        root         /usr/share/nginx/html;

    location / {
            proxy_pass http://backend_servers;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
                    }
        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
        location = /404.html {
        }
```

- task8_nginx_restart.png

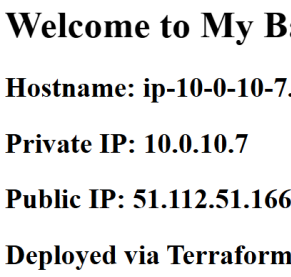
```
[ec2-user@ip-10-0-10-151 ~]$ [ec2-user@ip-10-0-10-151 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-151 ~]$
```

Task 9 — Configure high availability with backup servers

- task9_nginx_conf_ha_web1_primary.png

```
}

http {
    upstream backend_servers {
        server 10.0.10.7:80;
        server 10.0.10.144:80 backup;}

    log_format  main  '$remote_addr - $remote_user [$t
                      '$status $body_bytes_sent "$http
                      '"$http_user_agent" "$http_x_for
```

- task9_ha_web1_only.png



Welcome to My Backend Web Server

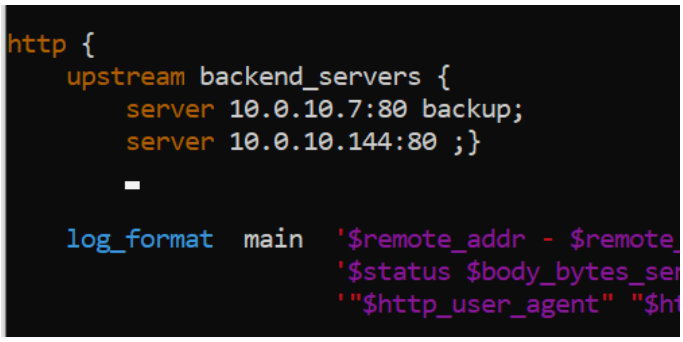
Hostname: ip-10-0-10-7.me-central-1.compute.internal

Private IP: 10.0.10.7

Public IP: 51.112.51.166

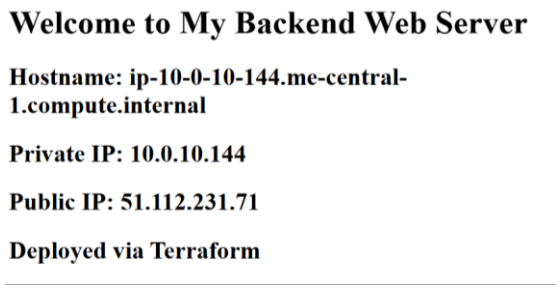
Deployed via Terraform

- task9_nginx_conf_ha_web2_primary.png



```
http {
    upstream backend_servers {
        server 10.0.10.7:80 backup;
        server 10.0.10.144:80 ;}

    log_format  main  '$remote_addr - $remote_
                      '$status $body_bytes_se
                      '"$http_user_agent" "$ht
```

- task9_ha_web2_only.png



Welcome to My Backend Web Server

Hostname: ip-10-0-10-144.me-central-1.compute.internal

Private IP: 10.0.10.144

Public IP: 51.112.231.71

Deployed via Terraform

Task 10 — Enable Nginx caching

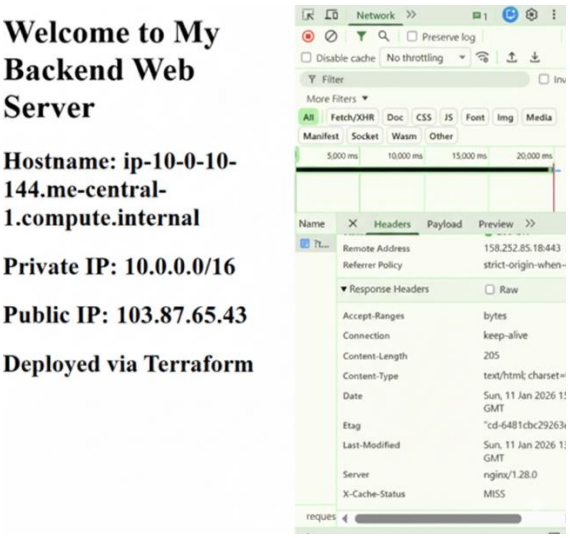
- task10_nginx_conf_cache.png

```
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

location / {
        proxy_pass http://backend_servers;
        proxy_cache my_cache;
        proxy_cache_valid 200 60m;
        proxy_cache_key "$scheme$request_uri";
        add_header X-Cache-Status $upstream_cache_status;
                }
    # Load configuration files for the default server block
```

- task10_nginx_restart.png

```
[ec2-user@ip-10-0-10-151 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-151 ~]$
```

- task10_cache_miss.png



- task10_cache_directory.png

```
ls: cannot open directory '/var/cache/nginx/': Permission denied
[ec2-user@ip-10-0-10-151 ~]$ sudo ls -la /var/cache/nginx/
total 0
drwx------. 4 nginx root   24 Jan 11 15:04 .
drwxr-xr-x. 9 root  root  101 Jan 11 14:44 ..
drwx------. 3 nginx nginx  16 Jan 11 14:46 4
drwx------. 3 nginx nginx  16 Jan 11 15:04 f
[ec2-user@ip-10-0-10-151 ~]$
```