Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

CLOUD COMPUTING LAB
BSE (V-B)

Submitted By:

Musfira Farooq

Roll No:

2023-BSE-045

Submitted To:

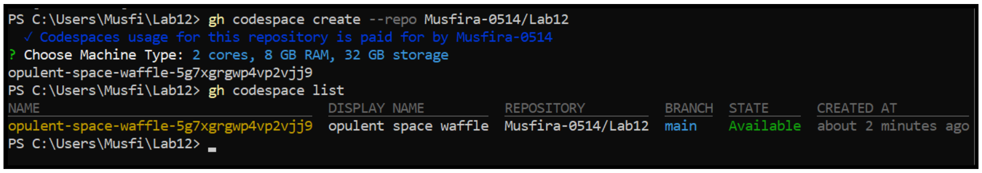
Sir Muhammad Shoaib

LAB 12

Terraform Provisioners, Modules & Nginx Reverse Proxy/Load Balancer

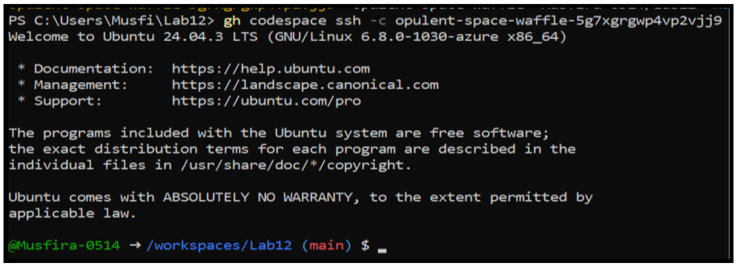
Task 0 – Codespace & GitHub CLI Setup

- Repository Creation and Codespace Listing



```
PS C:\Users\Musfi\Lab12> gh codespace create --repo Musfira-0514/Lab12
  ✓ Codespaces usage for this repository is paid for by Musfira-0514
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
opulent-space-waffle-5g7xgrgwp4vp2vjj9
PS C:\Users\Musfi\Lab12> gh codespace list
NAME                                    DISPLAY NAME          REPOSITORY          BRANCH  STATE      CREATED AT
opulent-space-waffle-5g7xgrgwp4vp2vjj9  opulent space waffle  Musfira-0514/Lab12  main    Available  about 2 minutes ago
PS C:\Users\Musfi\Lab12> _
```

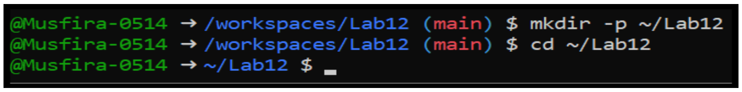
- Successful SSH Connection to Codespace



```
PS C:\Users\Musfi\Lab12> gh codespace ssh -c opulent-space-waffle-5g7xgrgwp4vp2vjj9
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

@Musfira-0514 → /workspaces/Lab12 (main) $ _
```

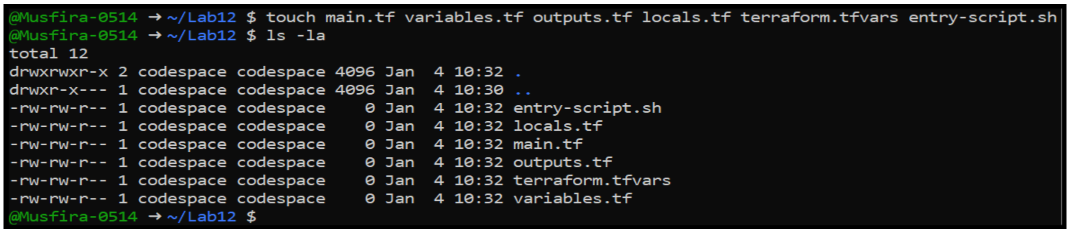
Task 1 – Terraform File Organization

- Project Directory Creation



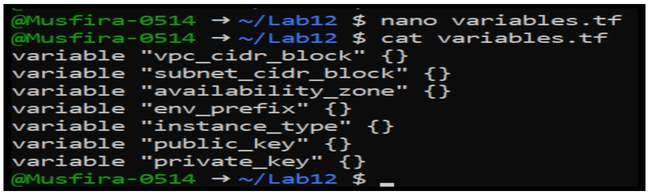
```
@Musfira-0514 → /workspaces/Lab12 (main) $ mkdir -p ~/Lab12
@Musfira-0514 → /workspaces/Lab12 (main) $ cd ~/Lab12
@Musfira-0514 → ~/Lab12 $ _
```

- Creation of Terraform Configuration Files



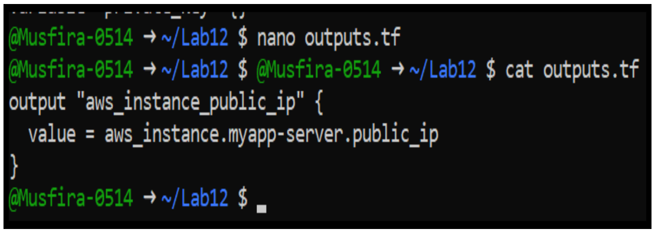
```
@Musfira-0514 → ~/Lab12 $ touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
@Musfira-0514 → ~/Lab12 $ ls -la
total 12
drwxrwxr-x 2 codespace codespace 4096 Jan  4 10:32 .
drwxr-x--- 1 codespace codespace 4096 Jan  4 10:30 ..
-rw-rw-r-- 1 codespace codespace    0 Jan  4 10:32 entry-script.sh
-rw-rw-r-- 1 codespace codespace    0 Jan  4 10:32 locals.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  4 10:32 main.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  4 10:32 outputs.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  4 10:32 terraform.tfvars
-rw-rw-r-- 1 codespace codespace    0 Jan  4 10:32 variables.tf
@Musfira-0514 → ~/Lab12 $
```

- Definition of Input Variables



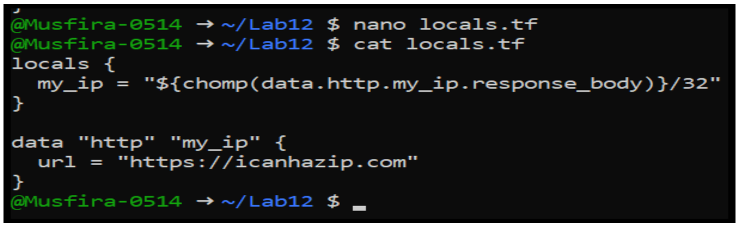
```
@Musfira-0514 → ~/Lab12 $ nano variables.tf
@Musfira-0514 → ~/Lab12 $ cat variables.tf
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
variable "public_key" {}
variable "private_key" {}
@Musfira-0514 → ~/Lab12 $ _
```

- Definition of Output Values



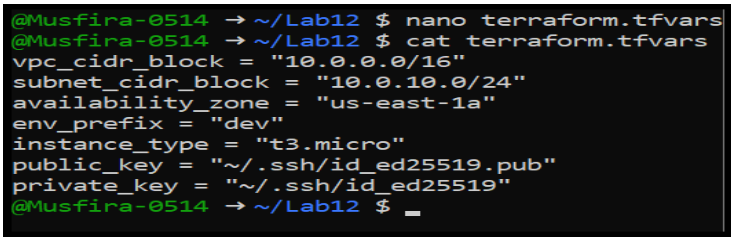
```
@Musfira-0514 → ~/Lab12 $ nano outputs.tf
@Musfira-0514 → ~/Lab12 $ @Musfira-0514 → ~/Lab12 $ cat outputs.tf
output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}
@Musfira-0514 → ~/Lab12 $ _
```

- Local Variables and Public IP Detection



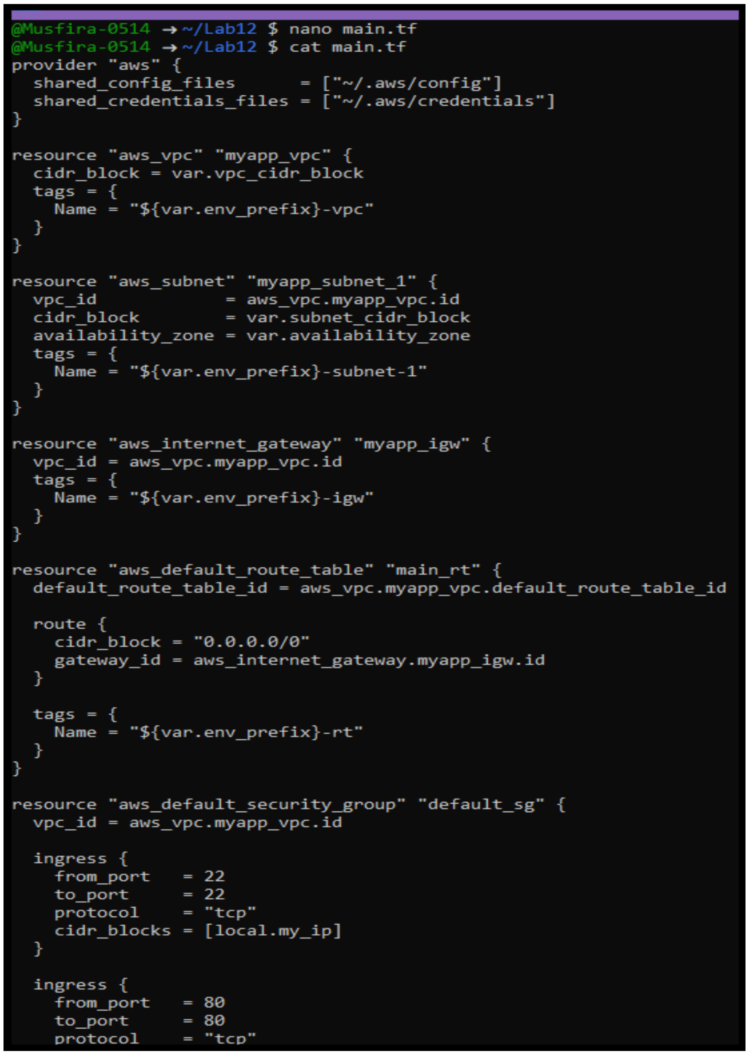
```
@Musfira-0514 →~/Lab12 $ nano locals.tf
@Musfira-0514 →~/Lab12 $ cat locals.tf
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

data "http" "my_ip" {
  url = "https://icanhazip.com"
}
@Musfira-0514 →~/Lab12 $ _
```

- Terraform Variable Values File

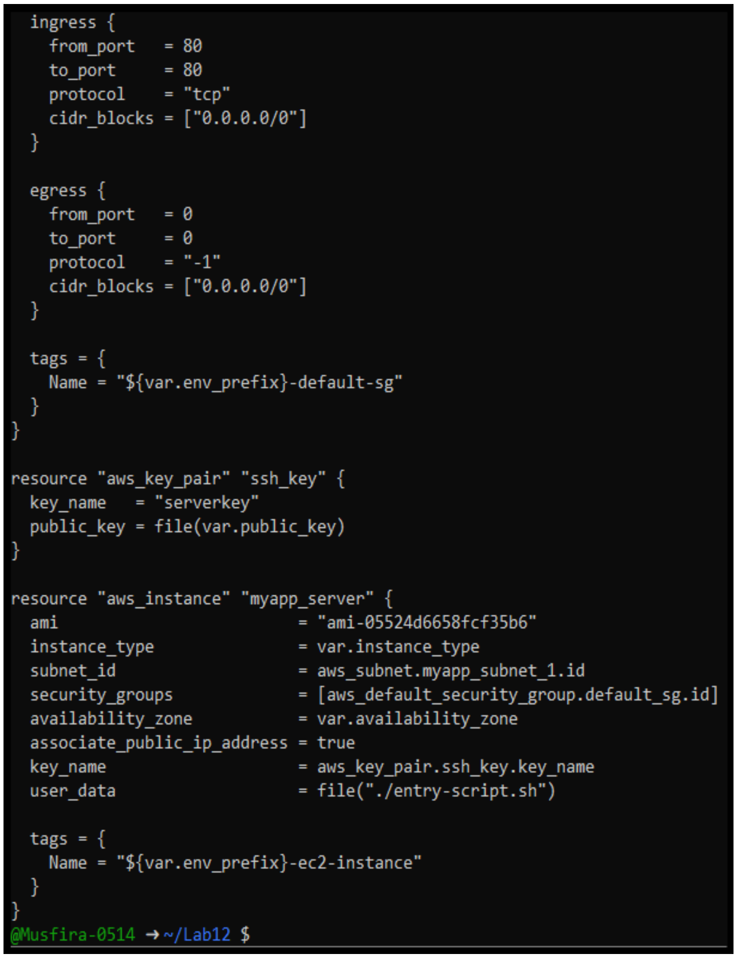


```
@Musfira-0514 →~/Lab12 $ nano terraform.tfvars
@Musfira-0514 →~/Lab12 $ cat terraform.tfvars
vpc_cidr_block = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "us-east-1a"
env_prefix = "dev"
instance_type = "t3.micro"
public_key = "~/.ssh/id_ed25519.pub"
private_key = "~/.ssh/id_ed25519"
@Musfira-0514 →~/Lab12 $ _
```

- Main Terraform Configuration

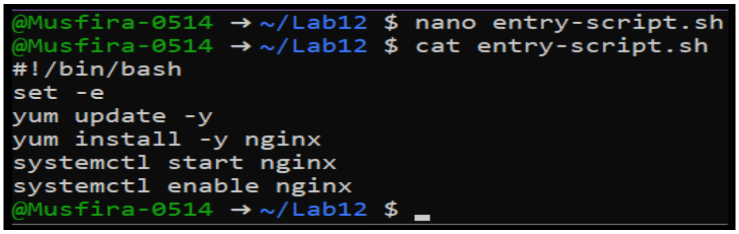


```
@Musfira-0514 →~/Lab12 $ nano main.tf
@Musfira-0514 →~/Lab12 $ cat main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

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_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
  tags = {
    Name = "${var.env_prefix}-igw"
  }
}

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_default_security_group" "default_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [local.my_ip]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
```



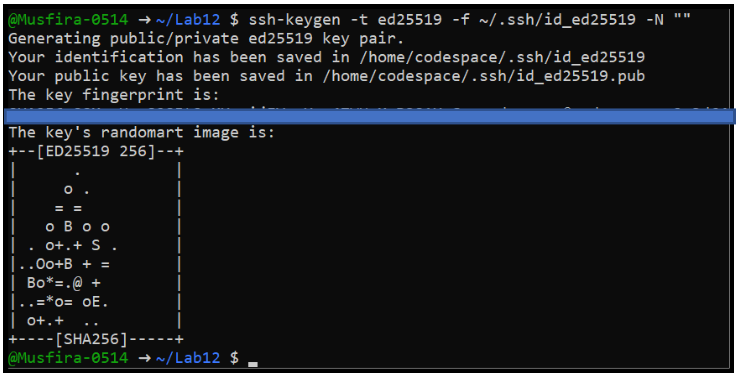
```
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  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"
  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"
  }
}
@Musfira-0514 →~/Lab12 $
```

- User Data Script for Nginx Installation



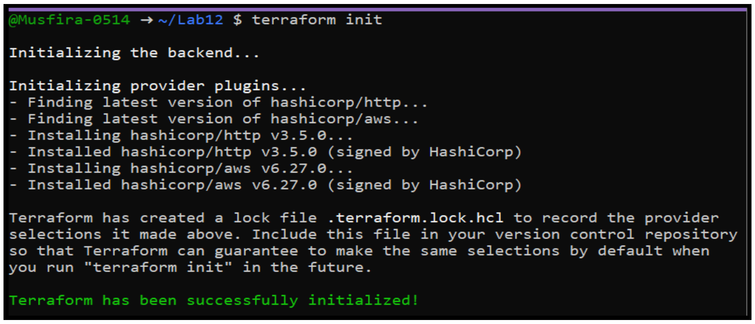
```
@Musfira-0514 →~/Lab12 $ nano entry-script.sh
@Musfira-0514 →~/Lab12 $ cat entry-script.sh
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
@Musfira-0514 →~/Lab12 $
```

- SSH Key Pair Generation



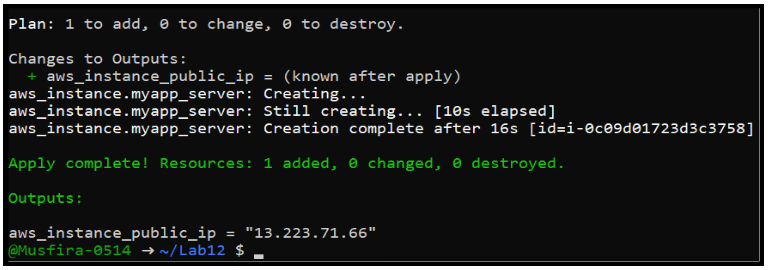
```
@Musfira-0514 →~/Lab12 $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
[redacted]
The key's randomart image is:
+--[ED25519 256]--+
|       .         |
|     o .         |
|    = =          |
|   o B o o       |
| . o+.+ S .      |
|..Oo+B + =       |
| Bo*=.@ +        |
|..=*o= oE.       |
| o+.+  ..        |
+----[SHA256]-----+
@Musfira-0514 →~/Lab12 $
```

- Terraform Initialization



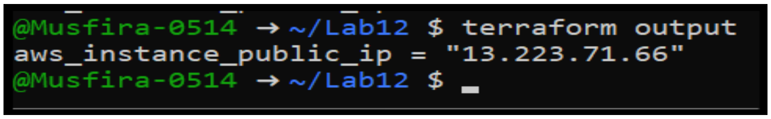
```
@Musfira-0514 → ~/Lab12 $ terraform init

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 v6.27.0...
- 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!
```

- Infrastructure Provisioning Using Terraform



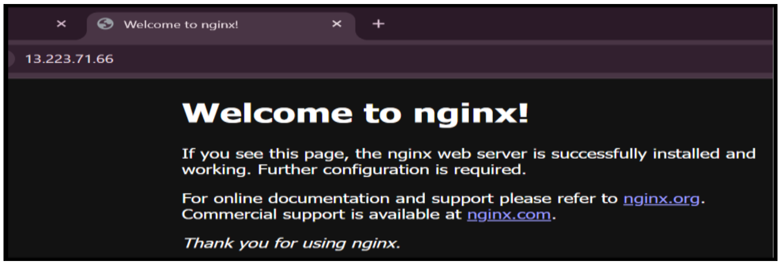
```
Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_instance.myapp_server: Creating...
aws_instance.myapp_server: Still creating... [10s elapsed]
aws_instance.myapp_server: Creation complete after 16s [id=i-0c09d01723d3c3758]

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

Outputs:

aws_instance_public_ip = "13.223.71.66"
@Musfira-0514 → ~/Lab12 $ _
```

- Terraform Output Showing Public IP



```
@Musfira-0514 → ~/Lab12 $ terraform output
aws_instance_public_ip = "13.223.71.66"
@Musfira-0514 → ~/Lab12 $ _
```

- Verification of Nginx Default Page



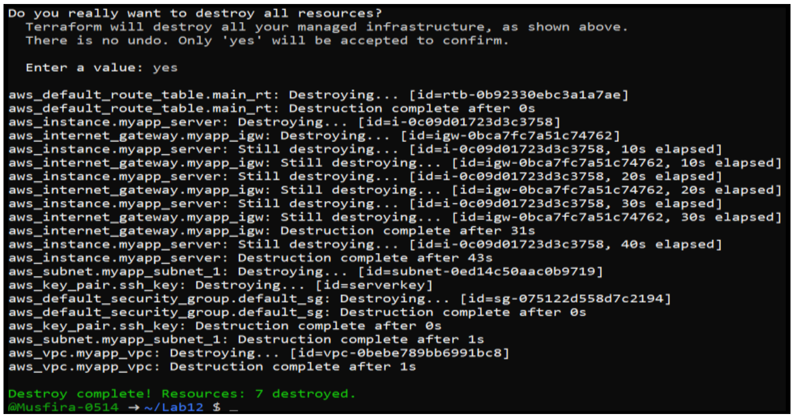
Welcome to nginx!

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.

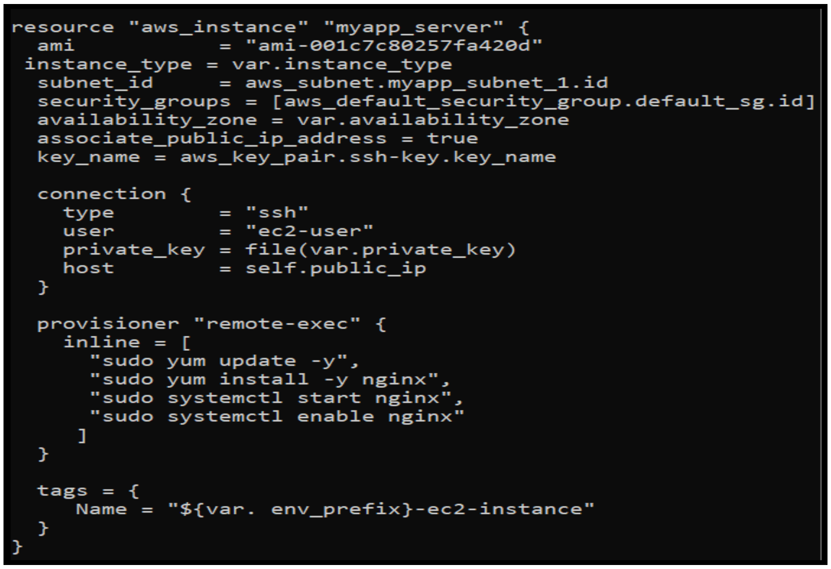
- Destruction of Terraform Resources



```
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-0b92330ebc3a1a7ae]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_instance.myapp_server: Destroying... [id=i-0c09d01723d3c3758]
aws_internet_gateway.myapp_igw: Destroying... [id=igw-0bca7fc7a51c74762]
aws_instance.myapp_server: Still destroying... [id=i-0c09d01723d3c3758, 10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0bca7fc7a51c74762, 10s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-0c09d01723d3c3758, 20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0bca7fc7a51c74762, 20s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-0c09d01723d3c3758, 30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0bca7fc7a51c74762, 30s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 31s
aws_instance.myapp_server: Still destroying... [id=i-0c09d01723d3c3758, 40s elapsed]
aws_instance.myapp_server: Destruction complete after 43s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-0ed14c50aac0b9719]
aws_key_pair.ssh_key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-075122d558d7c2194]
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 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0bebe789bb6991bc8]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Musfira-0514 → ~/Lab12 $ _
```

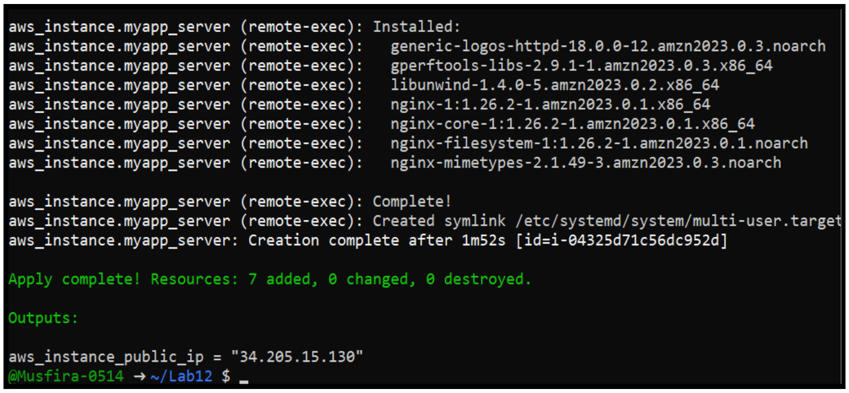
Task 2 – Remote-Exec Provisioner

- Configuration of Remote-Exec Provisioner



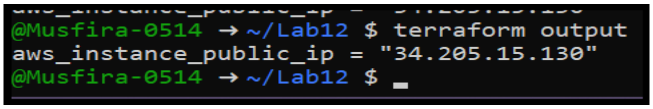
```
resource "aws_instance" "myapp_server" {
  ami           = "ami-001c7c80257fa420d"
 instance_type = var.instance_type
  subnet_id     = aws_subnet.myapp_subnet_1.id
  security_groups = [aws_default_security_group.default_sg.id]
  availability_zone = var.availability_zone
  associate_public_ip_address = 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"
  }
}
```

- Execution of Remote Commands via Terraform



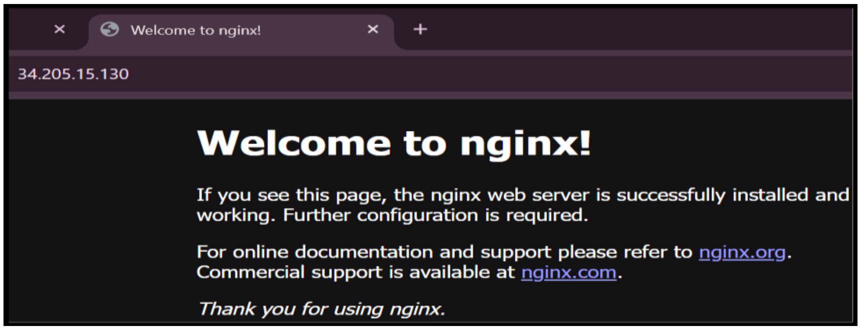
```
aws_instance.myapp_server (remote-exec): Installed:
aws_instance.myapp_server (remote-exec):   generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
aws_instance.myapp_server (remote-exec):   gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
aws_instance.myapp_server (remote-exec):   libunwind-1.4.0-5.amzn2023.0.2.x86_64
aws_instance.myapp_server (remote-exec):   nginx-1:1.26.2-1.amzn2023.0.1.x86_64
aws_instance.myapp_server (remote-exec):   nginx-core-1:1.26.2-1.amzn2023.0.1.x86_64
aws_instance.myapp_server (remote-exec):   nginx-filesystem-1:1.26.2-1.amzn2023.0.1.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
aws_instance.myapp_server: Creation complete after 1m52s [id=i-04325d71c56dc952d]

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

Outputs:

aws_instance_public_ip = "34.205.15.130"
@Musfira-0514 →~/Lab12 $ _
```

- Terraform Output After Remote Execution



```
@Musfira-0514 →~/Lab12 $ terraform output
aws_instance_public_ip = "34.205.15.130"
@Musfira-0514 →~/Lab12 $ _
```

- Browser Verification of Nginx Service



Welcome to nginx!

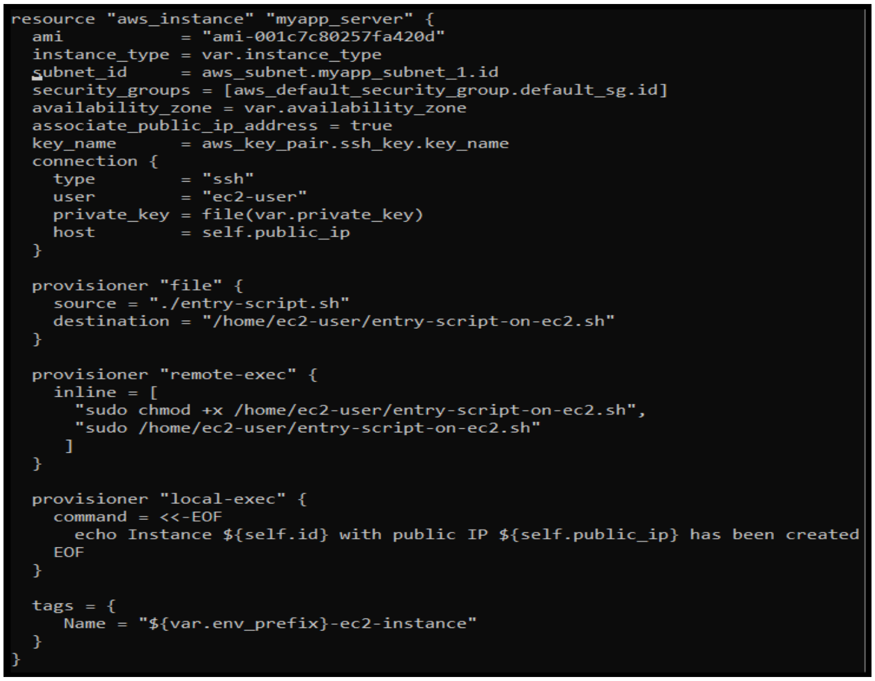
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

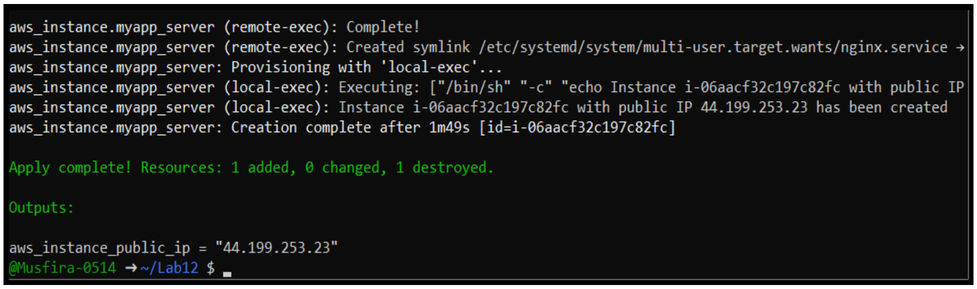
Task 3 – File and Local-Exec Provisioners

- Configuration of File, Remote-Exec and Local-Exec Provisioners



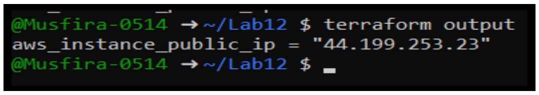
```
resource "aws_instance" "myapp_server" {
  ami           = "ami-001c7c80257fa420d"
  instance_type = var.instance_type
  subnet_id     = aws_subnet.myapp_subnet_1.id
  security_groups = [aws_default_security_group.default_sg.id]
  availability_zone = var.availability_zone
  associate_public_ip_address = 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 "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"
  }
}
```

- Successful Execution of All Provisioners



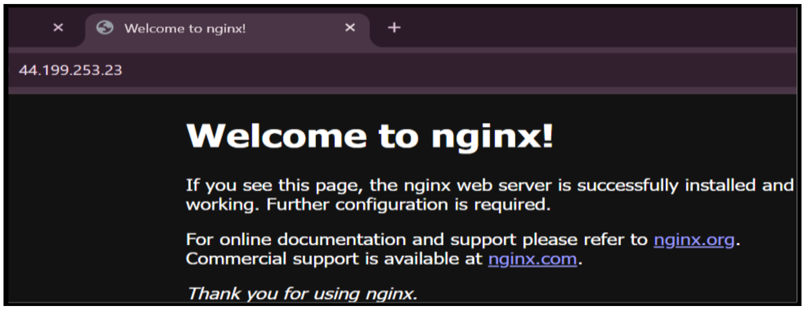
```
aws_instance.myapp_server (remote-exec): Complete!
aws_instance.myapp_server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service →
aws_instance.myapp_server: Provisioning with 'local-exec'...
aws_instance.myapp_server (local-exec): Executing: ["/bin/sh" "-c" "echo Instance i-06aacf32c197c82fc with public IP
aws_instance.myapp_server (local-exec): Instance i-06aacf32c197c82fc with public IP 44.199.253.23 has been created
aws_instance.myapp_server: Creation complete after 1m49s [id=i-06aacf32c197c82fc]

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

Outputs:

aws_instance_public_ip = "44.199.253.23"
@Musfira-0514 →~/Lab12 $ _
```

- Terraform Output Displaying Instance IP



```
@Musfira-0514 →~/Lab12 $ terraform output
aws_instance_public_ip = "44.199.253.23"
@Musfira-0514 →~/Lab12 $ _
```

- Verification of Nginx Deployment



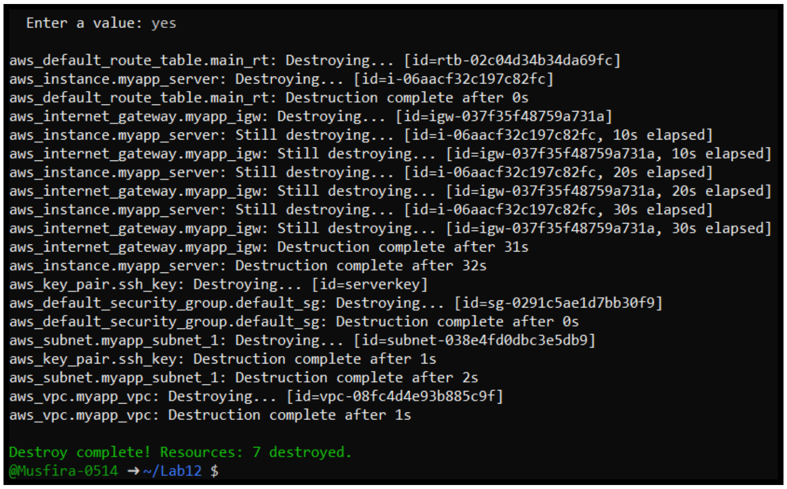
Welcome to nginx!

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.

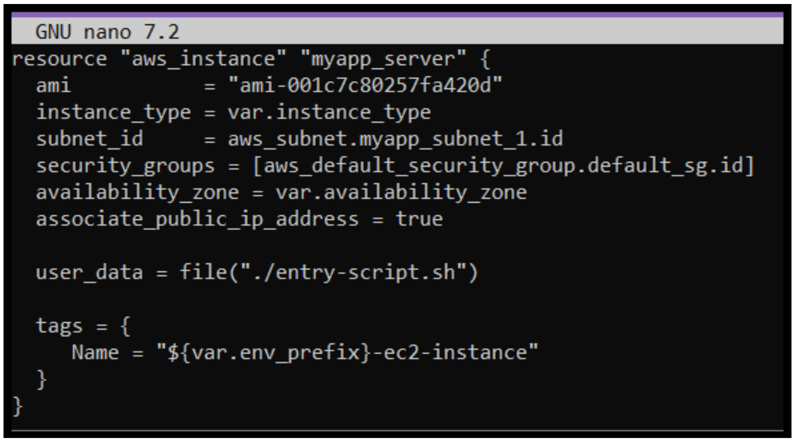
Thank you for using nginx.

- Cleanup of Provisioned Resources



```
  Enter a value: yes

aws_default_route_table.main_rt: Destroying... [id=rtb-02c04d34b34da69fc]
aws_instance.myapp_server: Destroying... [id=i-06aacf32c197c82fc]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_internet_gateway.myapp_igw: Destroying... [id=igw-037f35f48759a731a]
aws_instance.myapp_server: Still destroying... [id=i-06aacf32c197c82fc, 10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-037f35f48759a731a, 10s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-06aacf32c197c82fc, 20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-037f35f48759a731a, 20s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-06aacf32c197c82fc, 30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-037f35f48759a731a, 30s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 31s
aws_instance.myapp_server: Destruction complete after 32s
aws_key_pair.ssh_key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-0291c5ae1d7bb30f9]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-038e4fd0dbc3e5db9]
aws_key_pair.ssh_key: Destruction complete after 1s
aws_subnet.myapp_subnet_1: Destruction complete after 2s
aws_vpc.myapp_vpc: Destroying... [id=vpc-08fc4d4e93b885c9f]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Musfira-0514 ➜~/Lab12 $
```

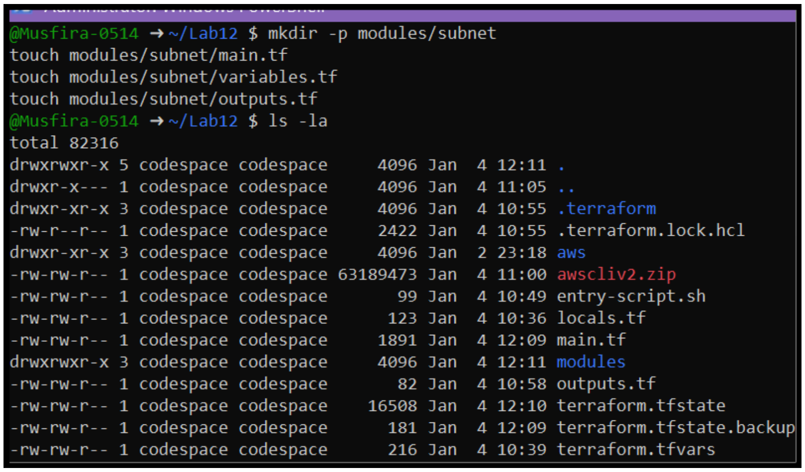
- Restoration of User Data Configuration



```
  GNU nano 7.2
resource "aws_instance" "myapp_server" {
  ami           = "ami-001c7c80257fa420d"
  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

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

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

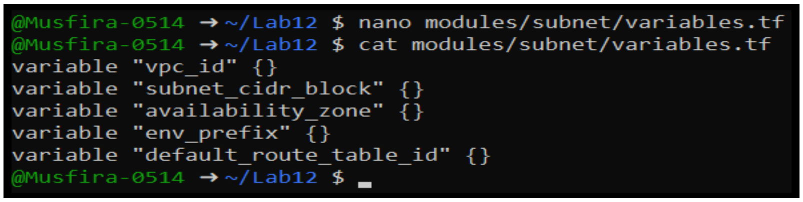
Task 4 – Subnet Module Creation

- Subnet Module Directory Structure



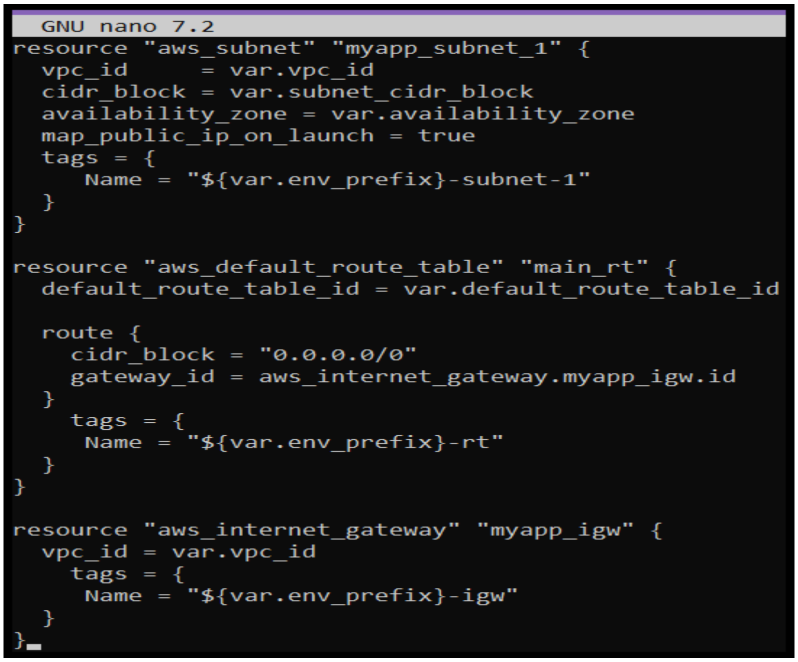
```
@Musfira-0514 ➜~/Lab12 $ mkdir -p modules/subnet
touch modules/subnet/main.tf
touch modules/subnet/variables.tf
touch modules/subnet/outputs.tf
@Musfira-0514 ➜~/Lab12 $ ls -la
total 82316
drwxrwxr-x 5 codespace codespace     4096 Jan  4 12:11 .
drwxr-x--- 1 codespace codespace     4096 Jan  4 11:05 ..
drwxr-xr-x 3 codespace codespace     4096 Jan  4 10:55 .terraform
-rw-r--r-- 1 codespace codespace     2422 Jan  4 10:55 .terraform.lock.hcl
drwxr-xr-x 3 codespace codespace     4096 Jan  2 23:18 aws
-rw-rw-r-- 1 codespace codespace 63189473 Jan  4 11:00 awscliv2.zip
-rw-rw-r-- 1 codespace codespace       99 Jan  4 10:49 entry-script.sh
-rw-rw-r-- 1 codespace codespace      123 Jan  4 10:36 locals.tf
-rw-rw-r-- 1 codespace codespace     1891 Jan  4 12:09 main.tf
drwxrwxr-x 3 codespace codespace     4096 Jan  4 12:11 modules
-rw-rw-r-- 1 codespace codespace       82 Jan  4 10:58 outputs.tf
-rw-rw-r-- 1 codespace codespace    16508 Jan  4 12:10 terraform.tfstate
-rw-rw-r-- 1 codespace codespace      181 Jan  4 12:09 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace      216 Jan  4 10:39 terraform.tfvars
```

- Subnet Module Variables Definition



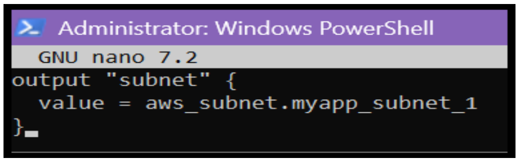
```
@Musfira-0514 ➜ ~/Lab12 $ nano modules/subnet/variables.tf
@Musfira-0514 ➜ ~/Lab12 $ cat modules/subnet/variables.tf
variable "vpc_id" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "default_route_table_id" {}
@Musfira-0514 ➜ ~/Lab12 $
```

- Subnet Module Main Configuration



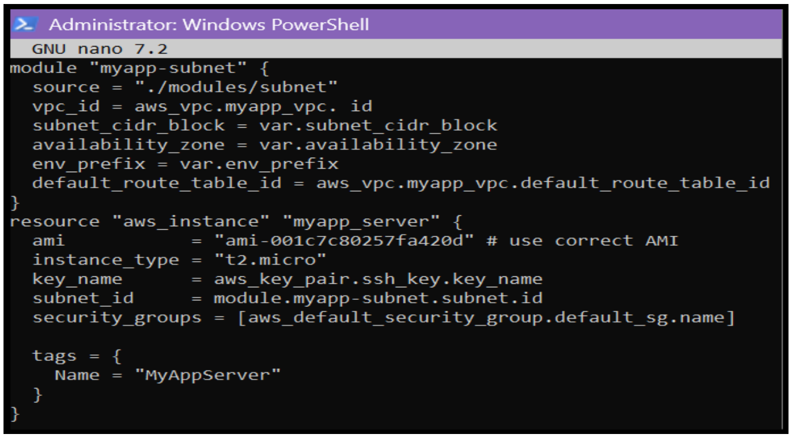
```
  GNU nano 7.2
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id      = var.vpc_id
  cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  map_public_ip_on_launch = true
  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }
}

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"
  }
}
```

- Subnet Module Output Definition



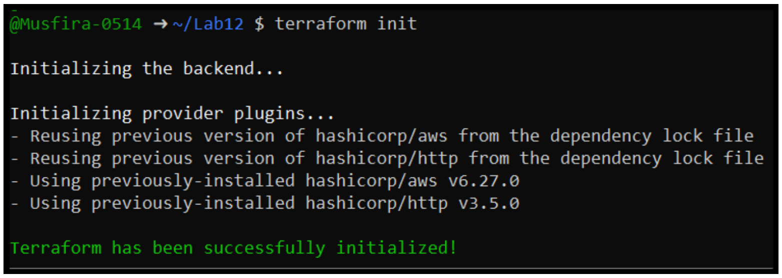
```
  GNU nano 7.2
output "subnet" {
  value = aws_subnet.myapp_subnet_1
}
```

- Integration of Subnet Module in Root Configuration



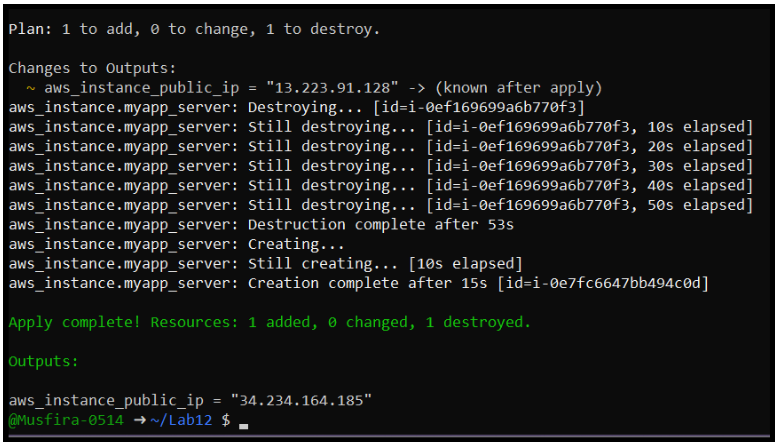
```
  GNU nano 7.2
module "myapp-subnet" {
  source = "./modules/subnet"
  vpc_id = aws_vpc.myapp_vpc. id
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}
resource "aws_instance" "myapp_server" {
  ami           = "ami-001c7c80257fa420d" # use correct AMI
  instance_type = "t2.micro"
  key_name      = aws_key_pair.ssh_key.key_name
  subnet_id     = module.myapp-subnet.subnet.id
  security_groups = [aws_default_security_group.default_sg.name]

  tags = {
    Name = "MyAppServer"
  }
}
```

- Terraform Initialization with Module



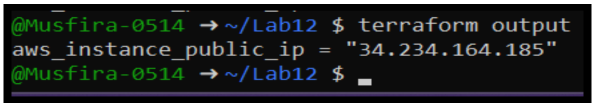
```
@Musfira-0514 ➜ ~/Lab12 $ terraform init

Initializing the backend...

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 v6.27.0
- Using previously-installed hashicorp/http v3.5.0

Terraform has been successfully initialized!
```

- Resource Creation Using Subnet Module



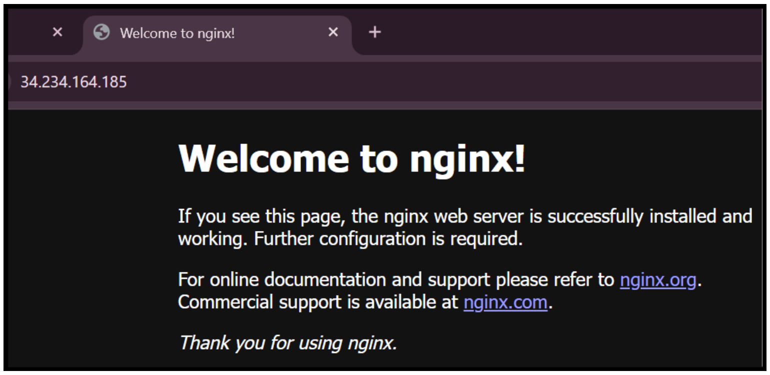
```
Plan: 1 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  ~ aws_instance_public_ip = "13.223.91.128" -> (known after apply)
aws_instance.myapp_server: Destroying... [id=i-0ef169699a6b770f3]
aws_instance.myapp_server: Still destroying... [id=i-0ef169699a6b770f3, 10s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-0ef169699a6b770f3, 20s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-0ef169699a6b770f3, 30s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-0ef169699a6b770f3, 40s elapsed]
aws_instance.myapp_server: Still destroying... [id=i-0ef169699a6b770f3, 50s elapsed]
aws_instance.myapp_server: Destruction complete after 53s
aws_instance.myapp_server: Creating...
aws_instance.myapp_server: Still creating... [10s elapsed]
aws_instance.myapp_server: Creation complete after 15s [id=i-0e7fc6647bb494c0d]

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

Outputs:

aws_instance_public_ip = "34.234.164.185"
@Musfira-0514 ➜ ~/Lab12 $ 
```

- Terraform Output After Module Deployment



```
@Musfira-0514 ➜ ~/Lab12 $ terraform output
aws_instance_public_ip = "34.234.164.185"
@Musfira-0514 ➜ ~/Lab12 $ 
```

- Verification of Nginx Service



Welcome to nginx!

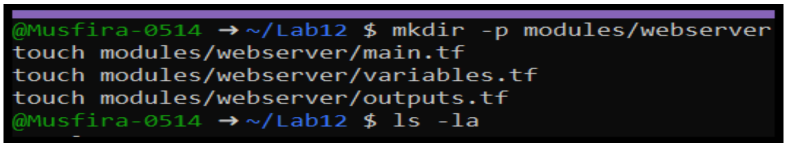
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

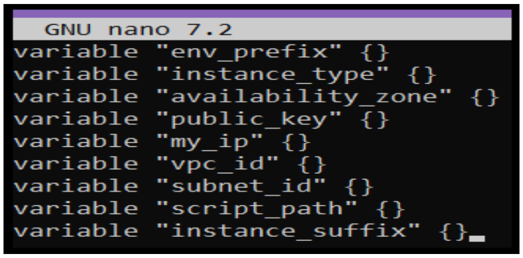
Task 5 – Webserver Module Creation

- Webserver Module Directory Structure



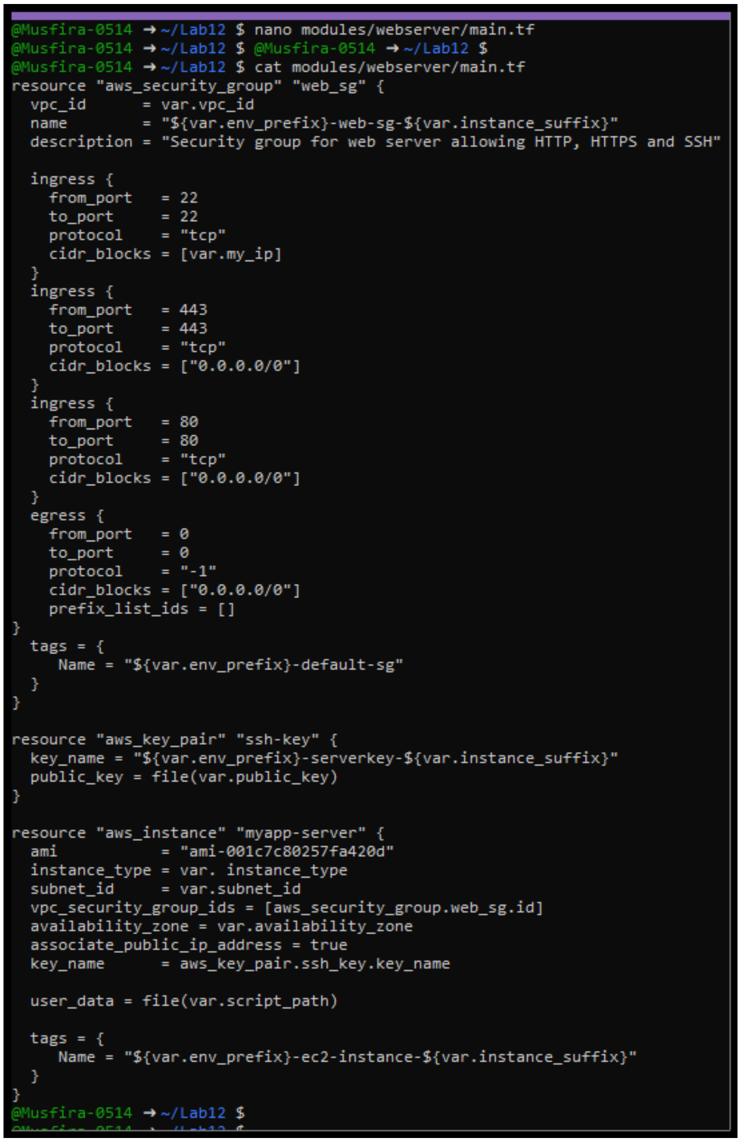
```
@Musfira-0514 ➜ ~/Lab12 $ mkdir -p modules/webserver
touch modules/webserver/main.tf
touch modules/webserver/variables.tf
touch modules/webserver/outputs.tf
@Musfira-0514 ➜ ~/Lab12 $ ls -la
```

- Webserver Module Variable Definitions



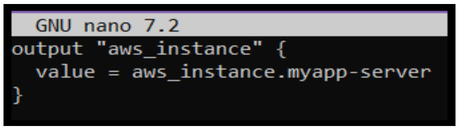
```
  GNU nano 7.2
variable "env_prefix" {}
variable "instance_type" {}
variable "availability_zone" {}
variable "public_key" {}
variable "my_ip" {}
variable "vpc_id" {}
variable "subnet_id" {}
variable "script_path" {}
variable "instance_suffix" {}
```

- Webserver Module Main Configuration



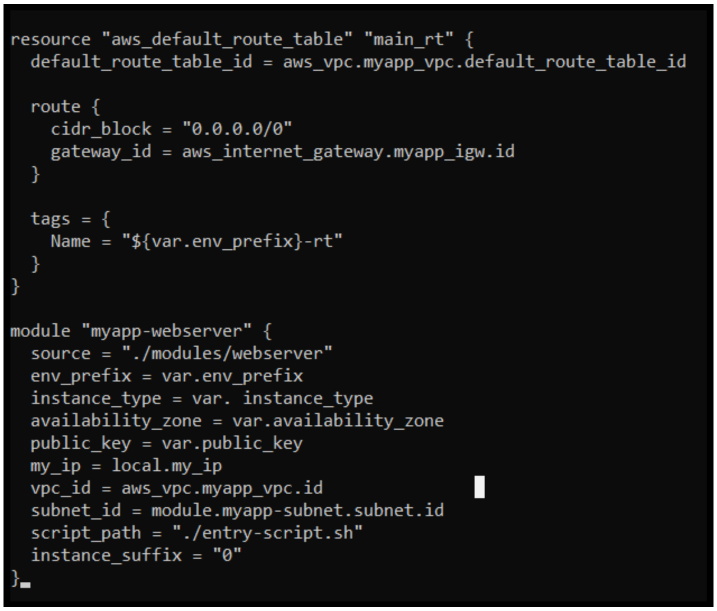
```
@Musfira-0514 ➜ ~/Lab12 $ nano modules/webserver/main.tf
@Musfira-0514 ➜ ~/Lab12 $ @Musfira-0514 ➜ ~/Lab12 $
@Musfira-0514 ➜ ~/Lab12 $ cat modules/webserver/main.tf
resource "aws_security_group" "web_sg" {
  vpc_id      = var.vpc_id
  name        = "${var.env_prefix}-web-sg-${var.instance_suffix}"
  description = "Security group for web server allowing HTTP, HTTPS and SSH"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
  }
  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  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 = "${var.env_prefix}-serverkey-${var.instance_suffix}"
  public_key = file(var.public_key)
}

resource "aws_instance" "myapp-server" {
  ami           = "ami-001c7c80257fa420d"
  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}"
  }
}
@Musfira-0514 ➜ ~/Lab12 $
@Musfira-0514 ➜ ~/Lab12 $
```

- Webserver Module Output Definition



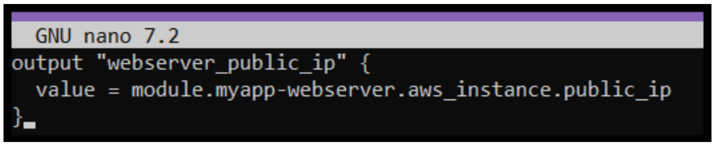
```
  GNU nano 7.2
output "aws_instance" {
  value = aws_instance.myapp-server
}
```

- Integration of Webserver Module in Root Configuration



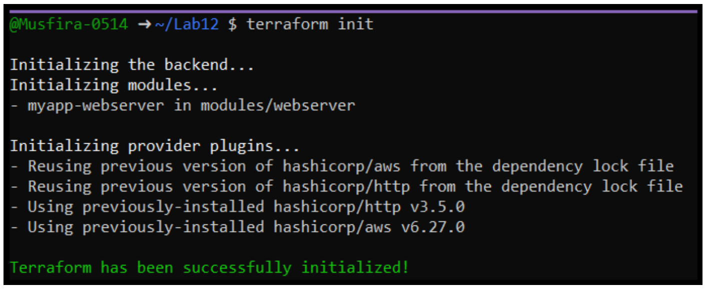
```
resource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }

  tags = {
    Name = "${var.env_prefix}-rt"
  }
}

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"
}
```

- Updated Terraform Outputs File



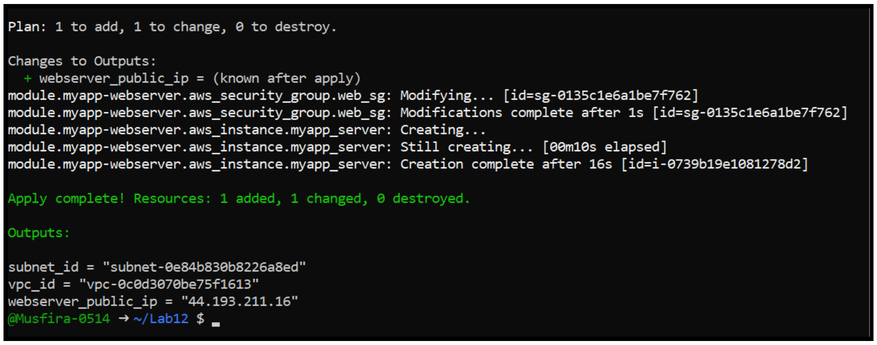
```
  GNU nano 7.2
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}
```

- Terraform Initialization



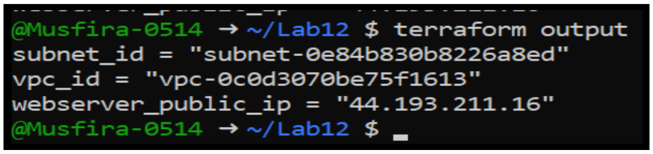
```
@Musfira-0514 ➜ ~/Lab12 $ terraform init

Initializing the backend...
Initializing modules...
- myapp-webserver in modules/webserver

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/http v3.5.0
- Using previously-installed hashicorp/aws v6.27.0

Terraform has been successfully initialized!
```

- Deployment of Webserver Module



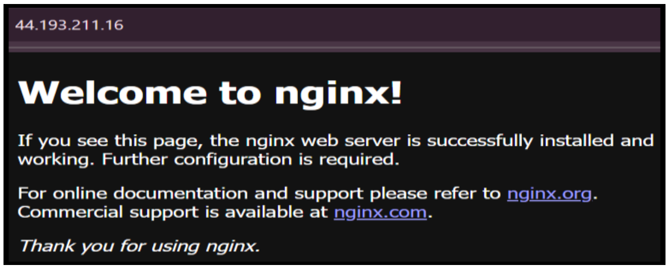
```
Plan: 1 to add, 1 to change, 0 to destroy.

Changes to Outputs:
  + webserver_public_ip = (known after apply)
module.myapp-webserver.aws_security_group.web_sg: Modifying... [id=sg-0135c1e6a1be7f762]
module.myapp-webserver.aws_security_group.web_sg: Modifications complete after 1s [id=sg-0135c1e6a1be7f762]
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 16s [id=i-0739b19e1081278d2]

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

Outputs:

subnet_id = "subnet-0e84b830b8226a8ed"
vpc_id = "vpc-0c0d3070be75f1613"
webserver_public_ip = "44.193.211.16"
@Musfira-0514 ➜ ~/Lab12 $
```

- Terraform Output Showing Webserver IP



```
@Musfira-0514 ➜ ~/Lab12 $ terraform output
subnet_id = "subnet-0e84b830b8226a8ed"
vpc_id = "vpc-0c0d3070be75f1613"
webserver_public_ip = "44.193.211.16"
@Musfira-0514 ➜ ~/Lab12 $ _
```

- Verification of Webserver in Browser



44.193.211.16

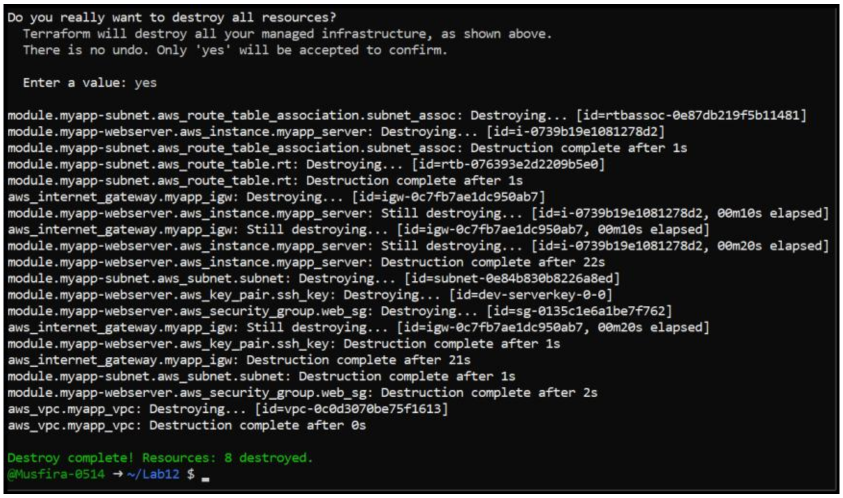
Welcome to nginx!

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.

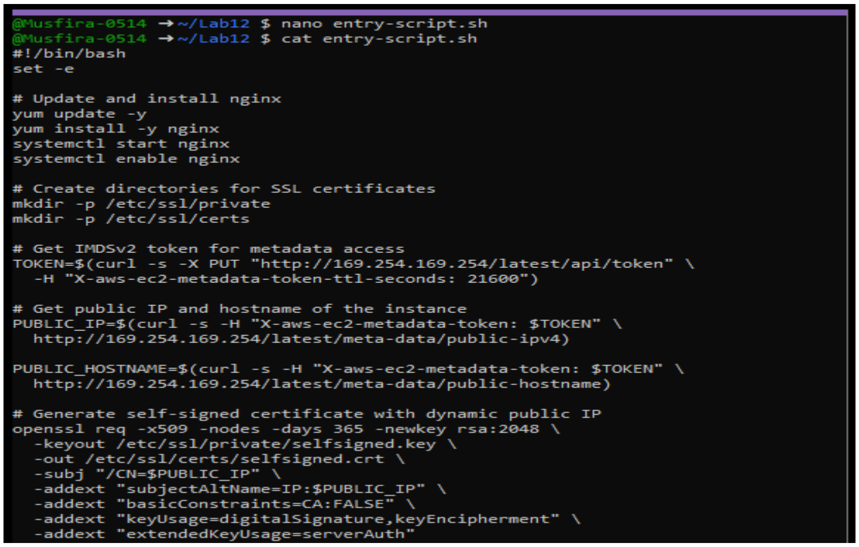
- Resource Cleanup



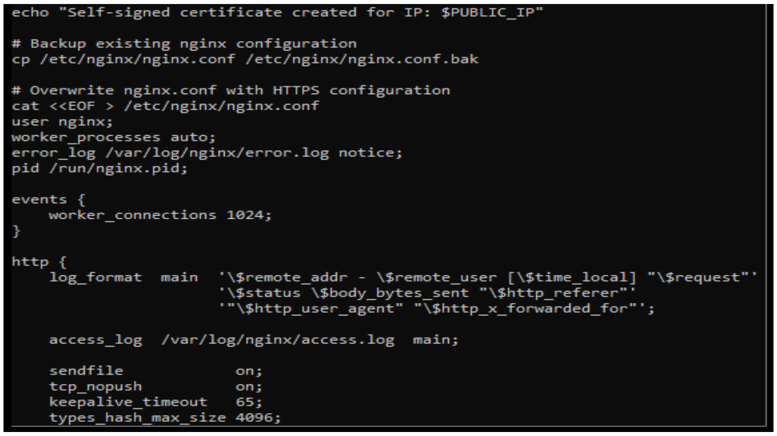
```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

module.myapp-subnet.aws_route_table_association.subnet_assoc: Destroying... [id=rtbassoc-0e87db219f5b11481]
module.myapp-webserver.aws_instance.myapp_server: Destroying... [id=i-0739b19e1081278d2]
module.myapp-subnet.aws_route_table_association.subnet_assoc: Destruction complete after 1s
module.myapp-subnet.aws_route_table.rt: Destroying... [id=rtb-076393e2d2209b5e0]
module.myapp-subnet.aws_route_table.rt: Destruction complete after 1s
aws_internet_gateway.myapp_igw: Destroying... [id=igw-0c7fb7ae1dc950ab7]
module.myapp-webserver.aws_instance.myapp_server: Still destroying... [id=i-0739b19e1081278d2, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0c7fb7ae1dc950ab7, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still destroying... [id=i-0739b19e1081278d2, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Destruction complete after 22s
module.myapp-subnet.aws_subnet.subnet: Destroying... [id=subnet-0e84b830b8226a8ed]
module.myapp-webserver.aws_key_pair.ssh_key: Destroying... [id=dev-serverkey-0-0]
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-0135c1e6a1be7f762]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0c7fb7ae1dc950ab7, 00m20s elapsed]
module.myapp-webserver.aws_key_pair.ssh_key: Destruction complete after 1s
aws_internet_gateway.myapp_igw: Destruction complete after 21s
module.myapp-subnet.aws_subnet.subnet: Destruction complete after 1s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 2s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0c0d3070be75f1613]
aws_vpc.myapp_vpc: Destruction complete after 0s

Destroy complete! Resources: 8 destroyed.
@Musfira-0514 ➜ ~/Lab12 $ _
```

Task 6 – HTTPS with Self-Signed Certificates

- SSL-Enabled Nginx Configuration Script

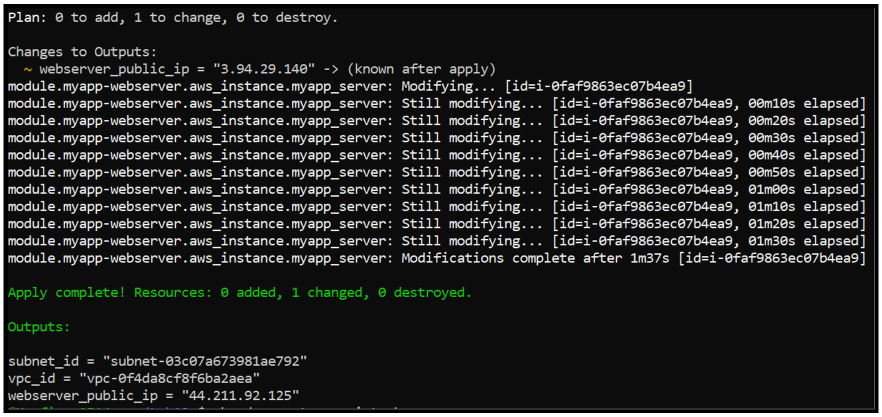


```
@Musfira-0514 ➜ ~/Lab12 $ nano entry-script.sh
@Musfira-0514 ➜ ~/Lab12 $ cat entry-script.sh
#!/bin/bash
set -e

# Update and install nginx
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx

# Create directories for SSL certificates
mkdir -p /etc/ssl/private
mkdir -p /etc/ssl/certs

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

# Get public IP and hostname of the instance
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 public IP
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/private/selfsigned.key \
  -out /etc/ssl/certs/selfsigned.crt \
  -subj "/CN=$PUBLIC_IP" \
  -addext "subjectAltName=IP:$PUBLIC_IP" \
  -addext "basicConstraints=CA:FALSE" \
  -addext "keyUsage=digitalSignature,keyEncipherment" \
  -addext "extendedKeyUsage=serverAuth"
```



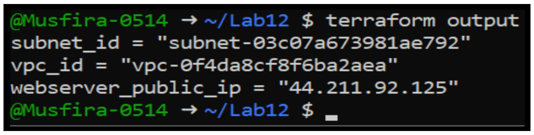
```
echo "Self-signed certificate created for IP: $PUBLIC_IP"

# Backup existing nginx configuration
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak

# Overwrite nginx.conf with HTTPS configuration
cat <<EOF > /etc/nginx/nginx.conf
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    log_format  main  '\$remote_addr - \$remote_user [\$time_local] "\$request"'
                      '\$status \$body_bytes_sent "\$http_referer"'
                      '"\$http_user_agent" "\$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;
```

- Terraform Apply for HTTPS Setup



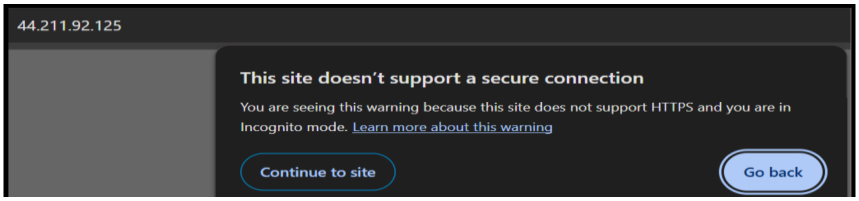
```
Plan: 0 to add, 1 to change, 0 to destroy.

Changes to Outputs:
  ~ webserver_public_ip = "3.94.29.140" -> (known after apply)
module.myapp-webserver.aws_instance.myapp_server: Modifying... [id=i-0faf9863ec07b4ea9]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m50s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 01m00s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 01m10s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 01m20s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 01m30s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Modifications complete after 1m37s [id=i-0faf9863ec07b4ea9]

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

Outputs:

subnet_id = "subnet-03c07a673981ae792"
vpc_id = "vpc-0f4da8cf8f6ba2aea"
webserver_public_ip = "44.211.92.125"
```

- Terraform Output After HTTPS Deployment



```
@Musfira-0514 ➜ ~/Lab12 $ terraform output
subnet_id = "subnet-03c07a673981ae792"
vpc_id = "vpc-0f4da8cf8f6ba2aea"
webserver_public_ip = "44.211.92.125"
@Musfira-0514 ➜ ~/Lab12 $ _
```

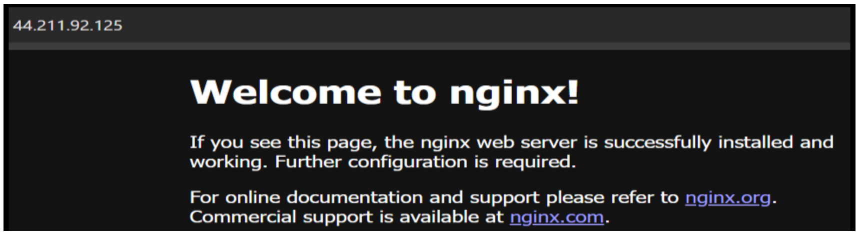
- Browser Security Warning for Self-Signed Certificate



This site doesn’t support a secure connection

You are seeing this warning because this site does not support HTTPS and you are in Incognito mode. Learn more about this warning

- Successful HTTPS Access to Nginx



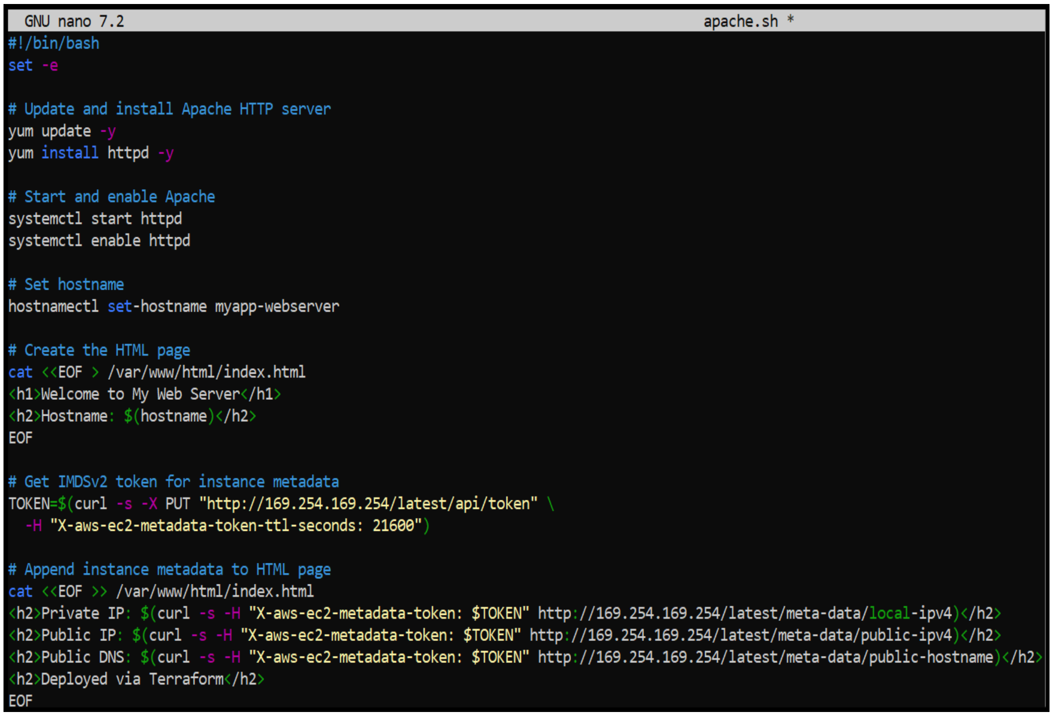
Welcome to nginx!

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.

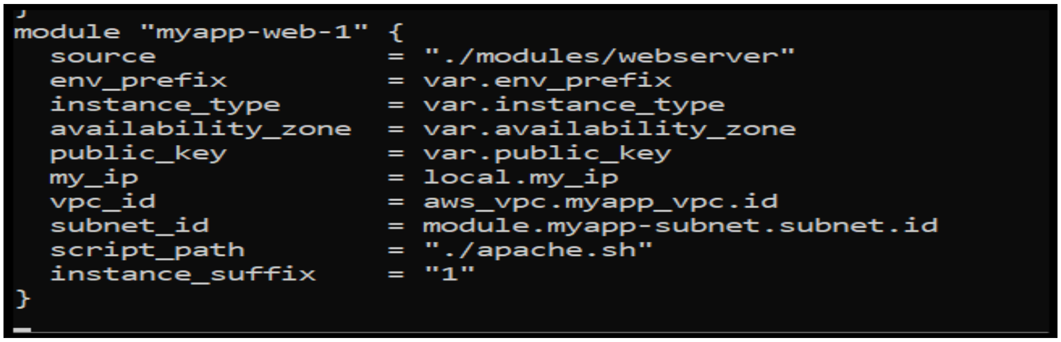
Task 7 — Configure Nginx as Reverse Proxy

- Apache Installation and Web Server Script



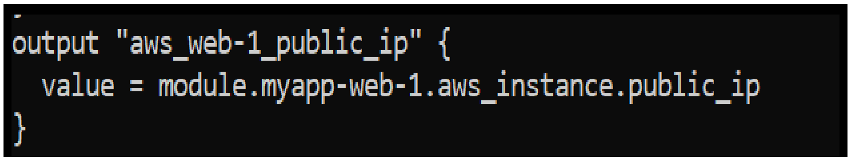
```
#!/bin/bash
set -e

# Update and install Apache HTTP server
yum update -y
yum install httpd -y

# Start and enable Apache
systemctl start httpd
systemctl enable httpd

# Set hostname
hostnamectl set-hostname myapp-webserver

# Create the HTML page
cat <<EOF > /var/www/html/index.html
<h1>Welcome to My Web Server</h1>
<h2>Hostname: $(hostname)</h2>
EOF

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

# Append instance metadata to HTML page
cat <<EOF >> /var/www/html/index.html
<h2>Private IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/local-ipv4)</h2>
<h2>Public IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-ipv4)</h2>
<h2>Public DNS: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-hostname)</h2>
<h2>Deployed via Terraform</h2>
EOF
```

- Terraform Module for Backend Web Server (web-1) in main.tf



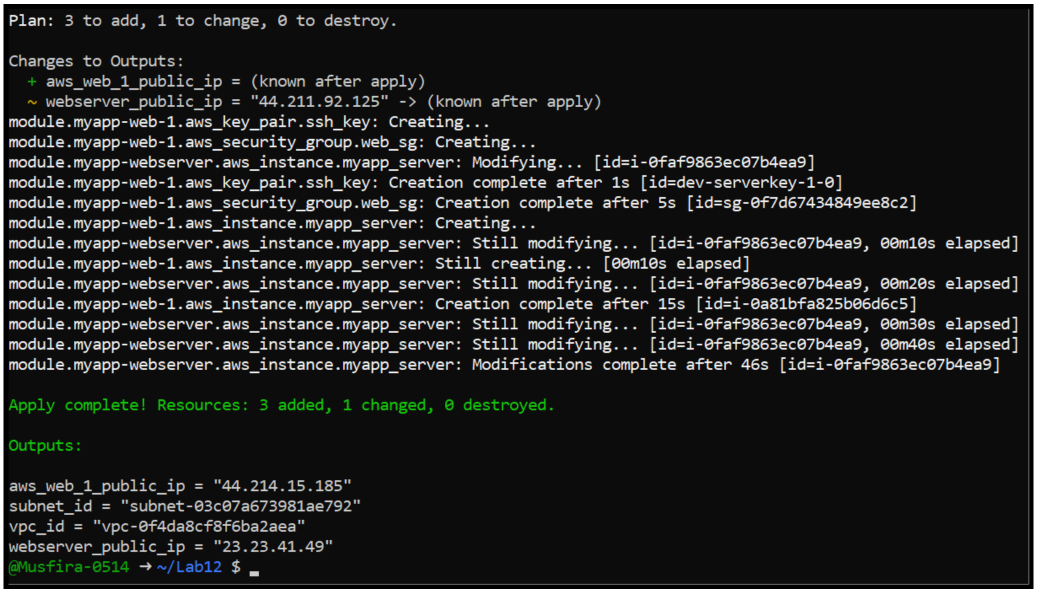
```
module "myapp-web-1" {
   source               = "./modules/webserver"
   env_prefix           = var.env_prefix
   instance_type        = var.instance_type
   availability_zone    = var.availability_zone
   public_key           = var.public_key
   my_ip                = local.my_ip
   vpc_id               = aws_vpc.myapp_vpc.id
   subnet_id            = module.myapp-subnet.subnet.id
   script_path          = "./apache.sh"
   instance_suffix      = "1"
}
```

- Output Configuration for Web-1 Public IP



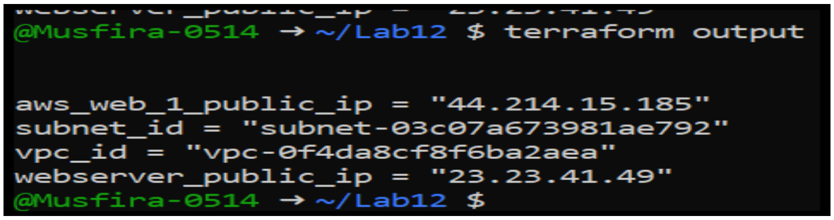
```
output "aws_web-1_public_ip" {
  value = module.myapp-web-1.aws_instance.public_ip
}
```

- Terraform Apply Output Showing Web-1 Creation



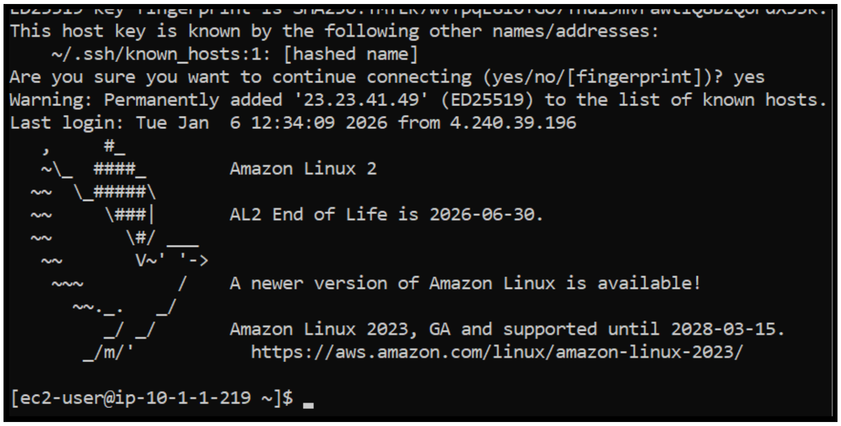
```
Plan: 3 to add, 1 to change, 0 to destroy.

Changes to Outputs:
  + aws_web_1_public_ip = (known after apply)
  ~ webserver_public_ip = "44.211.92.125" -> (known after apply)
module.myapp-web-1.aws_key_pair.ssh_key: Creating...
module.myapp-web-1.aws_security_group.web_sg: Creating...
module.myapp-webserver.aws_instance.myapp_server: Modifying... [id=i-0faf9863ec07b4ea9]
module.myapp-web-1.aws_key_pair.ssh_key: Creation complete after 1s [id=dev-serverkey-1-0]
module.myapp-web-1.aws_security_group.web_sg: Creation complete after 5s [id=sg-0f7d67434849ee8c2]
module.myapp-web-1.aws_instance.myapp_server: Creating...
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m10s elapsed]
module.myapp-web-1.aws_instance.myapp_server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m20s elapsed]
module.myapp-web-1.aws_instance.myapp_server: Creation complete after 15s [id=i-0a81bfa825b06d6c5]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still modifying... [id=i-0faf9863ec07b4ea9, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Modifications complete after 46s [id=i-0faf9863ec07b4ea9]

Apply complete! Resources: 3 added, 1 changed, 0 destroyed.

Outputs:

aws_web_1_public_ip = "44.214.15.185"
subnet_id = "subnet-03c07a673981ae792"
vpc_id = "vpc-0f4da8cf8f6ba2aea"
webserver_public_ip = "23.23.41.49"
@Musfira-0514 → ~/Lab12 $
```

- Terraform Output Displaying Web-1 Public IP



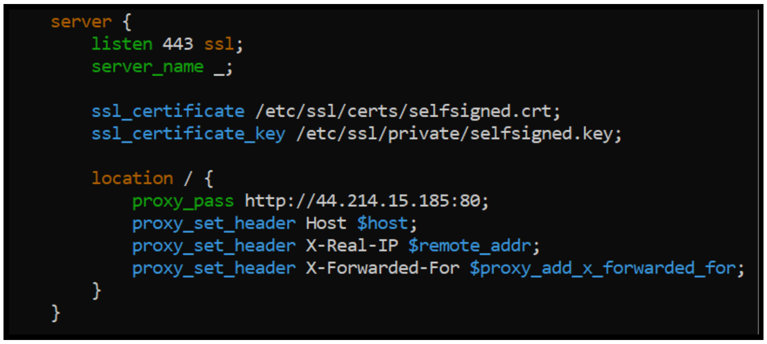
```
@Musfira-0514 → ~/Lab12 $ terraform output

aws_web_1_public_ip = "44.214.15.185"
subnet_id = "subnet-03c07a673981ae792"
vpc_id = "vpc-0f4da8cf8f6ba2aea"
webserver_public_ip = "23.23.41.49"
@Musfira-0514 → ~/Lab12 $
```

- SSH Session to Nginx Webserver



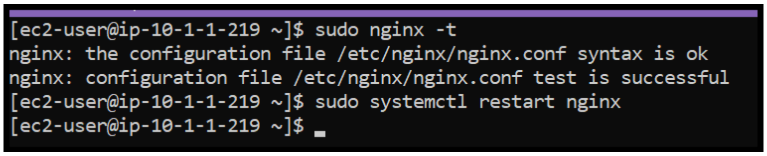
```
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '23.23.41.49' (ED25519) to the list of known hosts.
Last login: Tue Jan  6 12:34:09 2026 from 4.240.39.196
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-1-1-219 ~]$
```

- Nginx Configuration with Reverse Proxy to Web-1



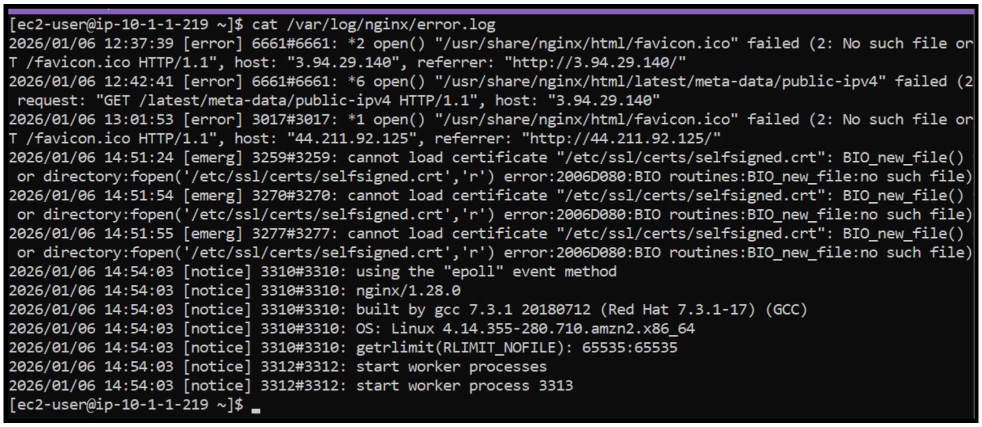
```
server {
    listen 443 ssl;
    server_name _;

    ssl_certificate /etc/ssl/certs/selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

    location / {
        proxy_pass http://44.214.15.185:80;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

- Restarting Nginx Service



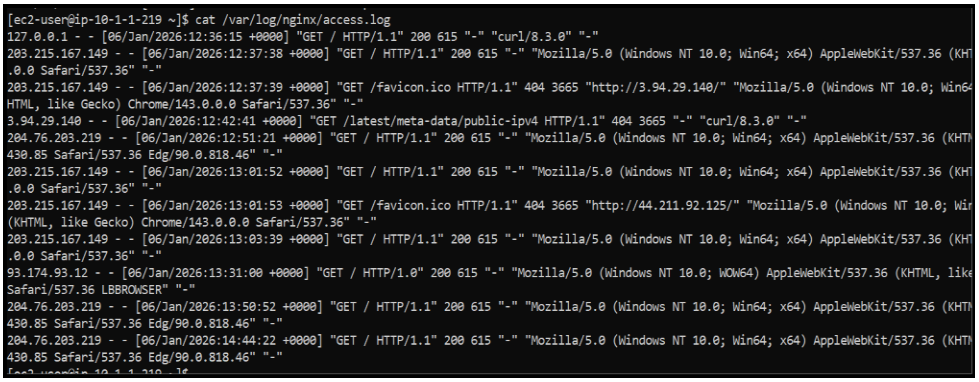
```
[ec2-user@ip-10-1-1-219 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-1-1-219 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-1-1-219 ~]$ _
```

- Nginx Error Log Content



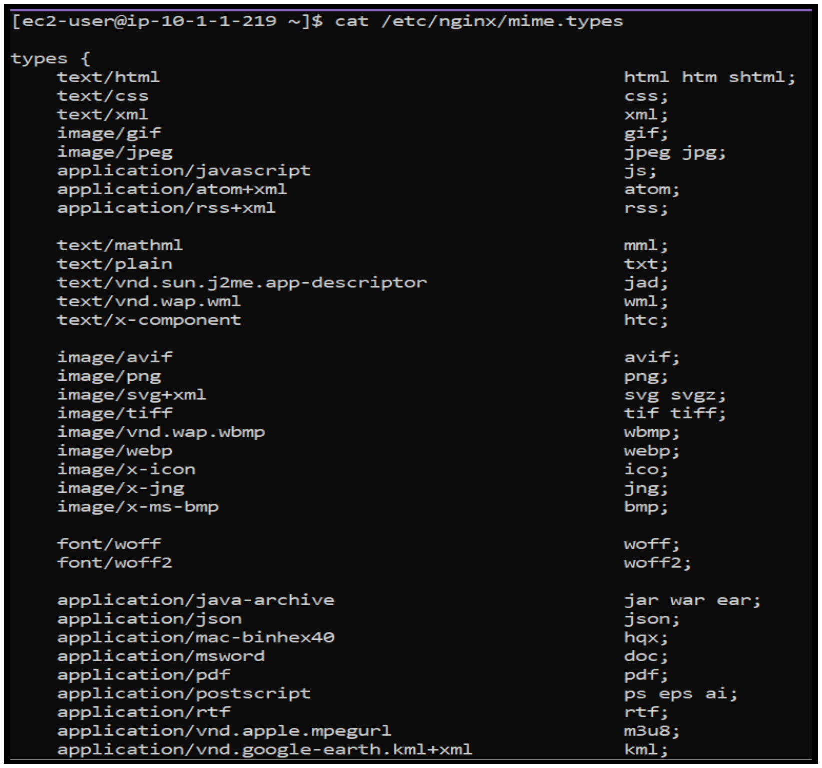
```
[ec2-user@ip-10-1-1-219 ~]$ cat /var/log/nginx/error.log
2026/01/06 12:37:39 [error] 6661#6661: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or
T /favicon.ico HTTP/1.1", host: "3.94.29.140", referrer: "http://3.94.29.140/"
2026/01/06 12:42:41 [error] 6661#6661: *6 open() "/usr/share/nginx/html/latest/meta-data/public-ipv4" failed (2
 request: "GET /latest/meta-data/public-ipv4 HTTP/1.1", host: "3.94.29.140"
2026/01/06 13:01:53 [error] 3017#3017: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or
T /favicon.ico HTTP/1.1", host: "44.211.92.125", referrer: "http://44.211.92.125/"
2026/01/06 14:51:24 [emerg] 3259#3259: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file()
 or directory:fopen('/etc/ssl/certs/selfsigned.crt','r') error:2006D080:BIO routines:BIO_new_file:no such file)
2026/01/06 14:51:54 [emerg] 3270#3270: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file()
 or directory:fopen('/etc/ssl/certs/selfsigned.crt','r') error:2006D080:BIO routines:BIO_new_file:no such file)
2026/01/06 14:51:55 [emerg] 3277#3277: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file()
 or directory:fopen('/etc/ssl/certs/selfsigned.crt','r') error:2006D080:BIO routines:BIO_new_file:no such file)
2026/01/06 14:54:03 [notice] 3310#3310: using the "epoll" event method
2026/01/06 14:54:03 [notice] 3310#3310: nginx/1.28.0
2026/01/06 14:54:03 [notice] 3310#3310: built by gcc 7.3.1 20180712 (Red Hat 7.3.1-17) (GCC)
2026/01/06 14:54:03 [notice] 3310#3310: OS: Linux 4.14.355-280.710.amzn2.x86_64
2026/01/06 14:54:03 [notice] 3310#3310: getrlimit(RLIMIT_NOFILE): 65535:65535
2026/01/06 14:54:03 [notice] 3312#3312: start worker processes
2026/01/06 14:54:03 [notice] 3312#3312: start worker process 3313
[ec2-user@ip-10-1-1-219 ~]$ _
```

- Nginx Access Log Content



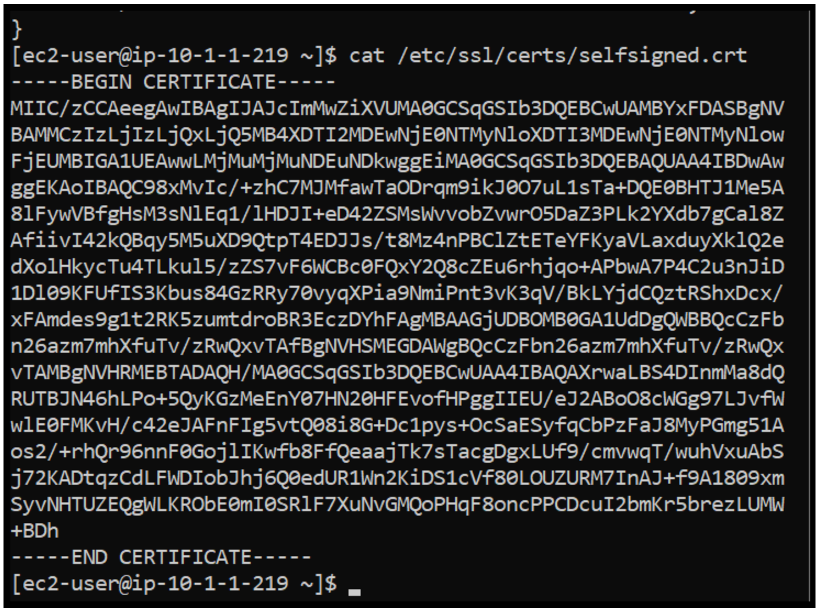
```
[ec2-user@ip-10-1-1-219 ~]$ cat /var/log/nginx/access.log
127.0.0.1 - - [06/Jan/2026:12:36:15 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.3.0" "-"
203.215.167.149 - - [06/Jan/2026:12:37:38 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
.0.0 Safari/537.36" "-"
203.215.167.149 - - [06/Jan/2026:12:37:39 +0000] "GET /favicon.ico HTTP/1.1" 404 3665 "http://3.94.29.140/" "Mozilla/5.0 (Windows NT 10.0; Win64
HTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
3.94.29.140 - - [06/Jan/2026:12:42:41 +0000] "GET /latest/meta-data/public-ipv4 HTTP/1.1" 404 3665 "-" "curl/8.3.0" "-"
204.76.203.219 - - [06/Jan/2026:12:51:21 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
430.85 Safari/537.36 Edg/90.0.818.46" "-"
203.215.167.149 - - [06/Jan/2026:13:01:52 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
.0.0 Safari/537.36" "-"
203.215.167.149 - - [06/Jan/2026:13:01:53 +0000] "GET /favicon.ico HTTP/1.1" 404 3665 "http://44.211.92.125/" "Mozilla/5.0 (Windows NT 10.0; Wir
(KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
203.215.167.149 - - [06/Jan/2026:13:03:39 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
.0.0 Safari/537.36" "-"
93.174.93.12 - - [06/Jan/2026:13:31:00 +0000] "GET / HTTP/1.0" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Safari/537.36 LBBROWSER" "-"
204.76.203.219 - - [06/Jan/2026:13:50:52 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
430.85 Safari/537.36 Edg/90.0.818.46" "-"
204.76.203.219 - - [06/Jan/2026:14:44:22 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTM
430.85 Safari/537.36 Edg/90.0.818.46" "-"
[ec2-user@ip-10-1-1-219 ~]$
```

- Nginx MIME Types Configuration



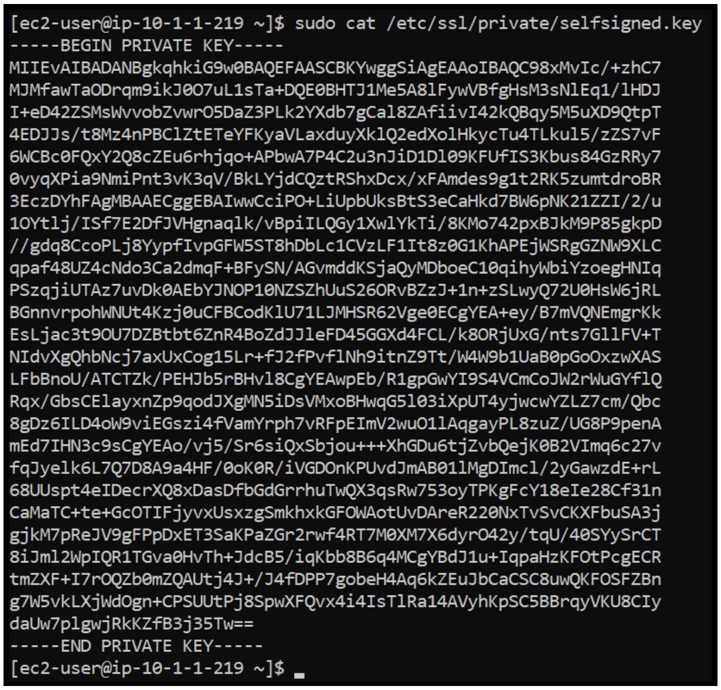
```
[ec2-user@ip-10-1-1-219 ~]$ cat /etc/nginx/mime.types

types {
    text/html                                        html htm shtml;
    text/css                                         css;
    text/xml                                         xml;
    image/gif                                        gif;
    image/jpeg                                       jpeg jpg;
    application/javascript                           js;
    application/atom+xml                             atom;
    application/rss+xml                              rss;

    text/mathml                                      mml;
    text/plain                                       txt;
    text/vnd.sun.j2me.app-descriptor                 jad;
    text/vnd.wap.wml                                 wml;
    text/x-component                                 htc;

    image/avif                                       avif;
    image/png                                        png;
    image/svg+xml                                    svg svgz;
    image/tiff                                       tif tiff;
    image/vnd.wap.wbmp                               wbmp;
    image/webp                                       webp;
    image/x-icon                                     ico;
    image/x-jng                                      jng;
    image/x-ms-bmp                                   bmp;

    font/woff                                        woff;
    font/woff2                                       woff2;

    application/java-archive                         jar war ear;
    application/json                                 json;
    application/mac-binhex40                         hqx;
    application/msword                               doc;
    application/pdf                                  pdf;
    application/postscript                           ps eps ai;
    application/rtf                                  rtf;
    application/vnd.apple.mpegurl                    m3u8;
    application/vnd.google-earth.kml+xml             kml;
```

- Self-Signed SSL Certificate Content



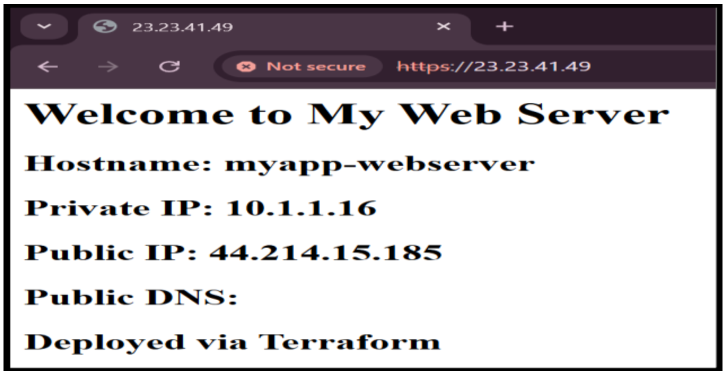
```
}
[ec2-user@ip-10-1-1-219 ~]$ cat /etc/ssl/certs/selfsigned.crt
-----BEGIN CERTIFICATE-----
MIIC/zCCAeegAwIBAgIJAJcImMwZiXVUMA0GCSqGSIb3DQEBCwUAMBYxFDASBgNV
BAMMCzIzLjIzLjQxLjQ5MB4XDTI2MDEwNjE0NTMyNloXDTI3MDEwNjE0NTMyNlow
FjEUMBIGA1UEAwwLMjMuMjMuNDEuNDkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQC98xMvIc/+zhC7MJMfawTaODrqm9ikJ0O7uL1sTa+DQE0BHTJ1Me5A
8lFywVBfgHsM3sNlEq1/lHDJI+eD42ZSMsWvvobZvwrO5DaZ3PLk2YXdb7gCal8Z
AfiivI42kQBqy5M5uXD9QtpT4EDJJs/t8Mz4nPBClZtETeYFKyaVLaxduyXklQ2e
dXolHkycTu4TLkul5/zZS7vF6WCBc0FQxY2Q8cZEu6rhjqo+APbwA7P4C2u3nJiD
1Dl09KFUfIS3Kbus84GzRRy70vyqXPia9NmiPnt3vK3qV/BkLYjdCQztRShxDcx/
xFAmdes9g1t2RK5zumtdroBR3EczDYhFAgMBAAGjUDBOMB0GA1UdDgQWBBQcCzFb
n26azm7mhXfuTv/zRwQxvTAfBgNVHSMEGDAWgBQcCzFbn26azm7mhXfuTv/zRwQx
vTAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQAXrwaLBS4DInmMa8dQ
RUTBJN46hLPo+5QyKGzMeEnY07HN20HFEvofHPggIIEU/eJ2ABoO8cWGg97LJvfW
wlE0FMKvH/c42eJAFnFIg5vtQ08i8G+Dc1pys+OcSaESyfqCbPzFaJ8MyPGmg51A
os2/+rhQr96nnF0GojlIKwfb8FfQeaajTk7sTacgDgxLUf9/cmvwqT/wuhVxuAbS
j72KADtqzCdLFWDIobJhj6Q0edUR1Wn2KiDS1cVf80LOUZURM7InAJ+f9A1809xm
SyvNHTUZEQgWLKRObE0mI0SRlF7XuNvGMQoPHqF8oncPPCDcuI2bmKr5brezLUMW
+BDh
-----END CERTIFICATE-----
[ec2-user@ip-10-1-1-219 ~]$
```

- SSL Private Key Content



```
[ec2-user@ip-10-1-1-219 ~]$ sudo cat /etc/ssl/private/selfsigned.key
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQC98xMvIc/+zhC7
MJMfawTaODrqm9ikJ0O7uL1sTa+DQE0BHTJ1Me5A8lFywVBfgHsM3sNlEq1/lHDJ
I+eD42ZSMsWvvobZvwrO5DaZ3PLk2YXdb7gCal8ZAfiivI42kQBqy5M5uXD9QtpT
4EDJJs/t8Mz4nPBClZtETeYFKyaVLaxduyXklQ2edXolHkycTu4TLkul5/zZS7vF
6WCBc0FQxY2Q8cZEu6rhjqo+APbwA7P4C2u3nJiD1Dl09KFUfIS3Kbus84GzRRy7
0vyqXPia9NmiPnt3vK3qV/BkLYjdCQztRShxDcx/xFAmdes9g1t2RK5zumtdroBR
3EczDYhFAgMBAAECggEBAIwwCciPO+LiUpbUksBtS3eCaHkd7BW6pNK21ZZI/2/u
1OYtlj/ISf7E2DfJVHgnaqlk/vBpiILQGy1XwlYkTi/8KMo742pxBJkM9P85gkpD
//gdq8CcoPLj8YypfIvpGFW5ST8hDbLc1CVzLF1It8z0G1KhAPEjWSRgGZNW9XLC
qpaf48UZ4cNdo3Ca2dmqF+BFySN/AGvmddKSjaQyMDboeC10qihyWbiYzoegHNIq
PSzqjiUTAz7uvDk0AEbYJNOP10NZSZhUuS26ORvBZzJ+1n+zSLwyQ72U0HsW6jRL
BGnnvrpohWNUt4Kzj0uCFBCodKlU71LJMHSR62Vge0ECgYEA+ey/B7mVQNEmgrKk
EsLjac3t9OU7DZBtbt6ZnR4BoZdJJleFD45GGXd4FCL/k8ORjUxG/nts7GllFV+T
NIdvXgQhbNcj7axUxCog15Lr+fJ2fPvflNh9itnZ9Tt/W4W9b1UaB0pGoOxzwXAS
LFbBnoU/ATCTZk/PEHJb5rBHvl8CgYEAwpEb/R1gpGwYI9S4VCmCoJW2rWuGYflQ
Rqx/GbsCElayxnZp9qodJXgMN5iDsVMxoBHwqG5l03iXpUT4yjwcwYZLZ7cm/Qbc
8gDz6ILD4oW9viEGszi4fVamYrph7vRFpEImV2wuO1lAqgayPL8zuZ/UG8P9penA
mEd7IHN3c9sCgYEAo/vj5/Sr6siQxSbjou+++XhGDu6tjZvbQejK0B2VImq6c27v
fqJyelk6L7Q7D8A9a4HF/0oK0R/iVGDOnKPUvdJmAB01lMgDImcl/2yGawzdE+rL
68UUspt4eIDecrXQ8xDasDfbGdGrrhuTwQX3qsRw753oyTPKgFcY18eIe28Cf31n
CaMaTC+te+GcOTIFjyvxUsxzgSmkhxkGFOWAotUvDAreR220NxTvSvCKXFbuSA3j
gjkM7pReJV9gFPpDxET3SaKPaZGr2rwf4RT7M0XM7X6dyrO42y/tqU/40SYySrCT
8iJml2WpIQR1TGva0HvTh+JdcB5/iqKbb8B6q4MCgYBdJ1u+IqpaHzKFOtPcgECR
tmZXF+I7rOQZb0mZQAUtj4J+/J4fDPP7gobeH4Aq6kZEuJbCaCSC8uwQKFOSFZBn
g7W5vkLXjWdOgn+CPSUUtPj8SpwXFQvx4i4IsTlRa14AVyhKpSC5BBrqyVKU8CIy
daUw7plgwjRkKZfB3j35Tw==
-----END PRIVATE KEY-----
[ec2-user@ip-10-1-1-219 ~]$ 
```

- Browser Showing Web-1 Content Through Nginx Proxy



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.1.1.16

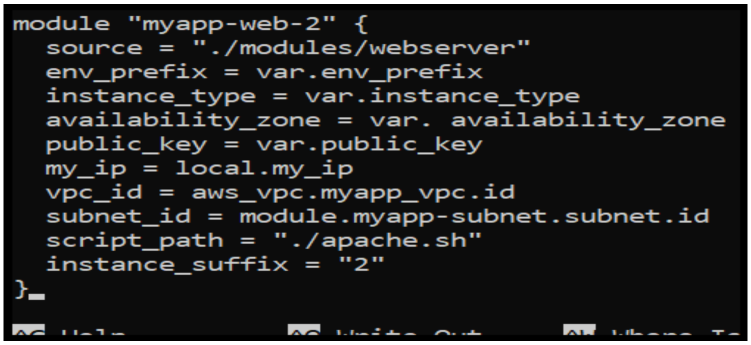
Public IP: 44.214.15.185

Public DNS:

Deployed via Terraform

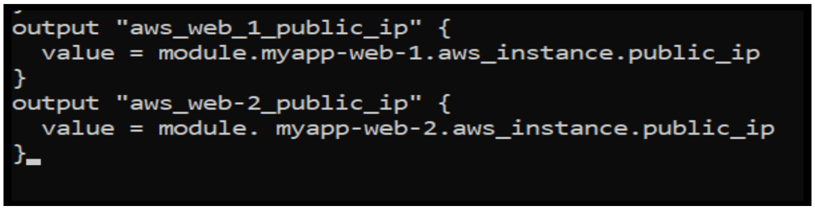
Task 8 — Configure Nginx as Load Balancer

- Terraform Module for Backend Web Server (web-2) in main.tf



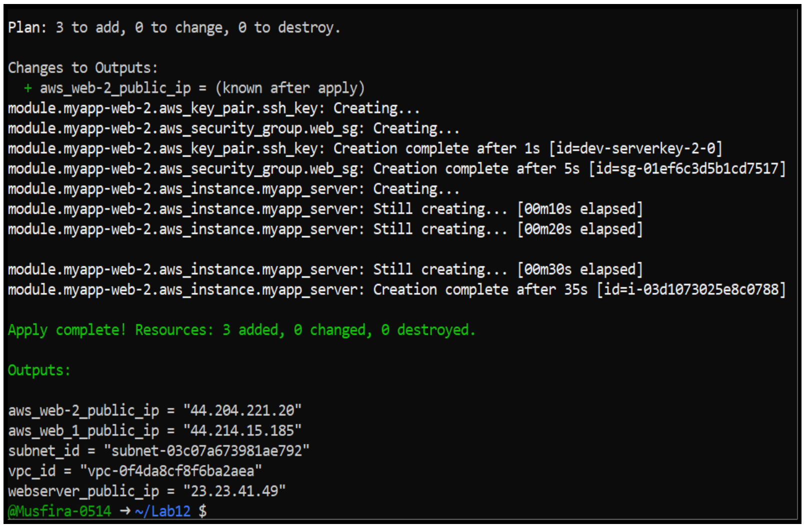
```
module "myapp-web-2" {
  source = "./modules/webserver"
  env_prefix = var.env_prefix
  instance_type = var.instance_type
  availability_zone = var. availability_zone
  public_key = var.public_key
  my_ip = local.my_ip
  vpc_id = aws_vpc.myapp_vpc.id
  subnet_id = module.myapp-subnet.subnet.id
  script_path = "./apache.sh"
  instance_suffix = "2"
}
```

- Output Configuration for Web-2 Public IP



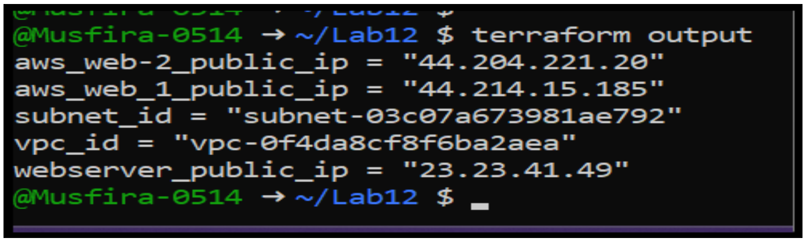
```
output "aws_web_1_public_ip" {
  value = module.myapp-web-1.aws_instance.public_ip
}
output "aws_web-2_public_ip" {
  value = module. myapp-web-2.aws_instance.public_ip
}
```

- Terraform Apply Output Showing Web-2 Creation



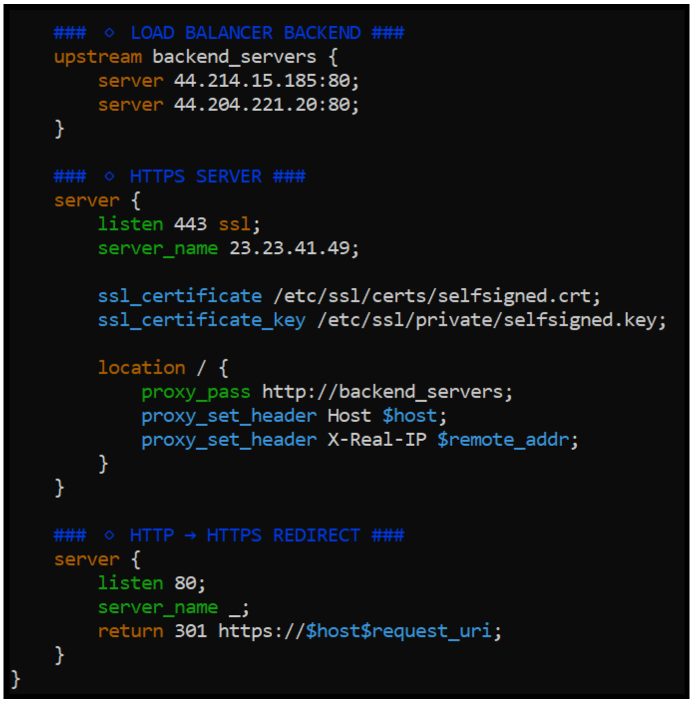
```
Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + aws_web-2_public_ip = (known after apply)
module.myapp-web-2.aws_key_pair.ssh_key: Creating...
module.myapp-web-2.aws_security_group.web_sg: Creating...
module.myapp-web-2.aws_key_pair.ssh_key: Creation complete after 1s [id=dev-serverkey-2-0]
module.myapp-web-2.aws_security_group.web_sg: Creation complete after 5s [id=sg-01ef6c3d5b1cd7517]
module.myapp-web-2.aws_instance.myapp_server: Creating...
module.myapp-web-2.aws_instance.myapp_server: Still creating... [00m10s elapsed]
module.myapp-web-2.aws_instance.myapp_server: Still creating... [00m20s elapsed]

module.myapp-web-2.aws_instance.myapp_server: Still creating... [00m30s elapsed]
module.myapp-web-2.aws_instance.myapp_server: Creation complete after 35s [id=i-03d1073025e8c0788]

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

Outputs:

aws_web-2_public_ip = "44.204.221.20"
aws_web_1_public_ip = "44.214.15.185"
subnet_id = "subnet-03c07a673981ae792"
vpc_id = "vpc-0f4da8cf8f6ba2aea"
webserver_public_ip = "23.23.41.49"
@Musfira-0514 → ~/Lab12 $
```

- Terraform Output Displaying All Public IPs



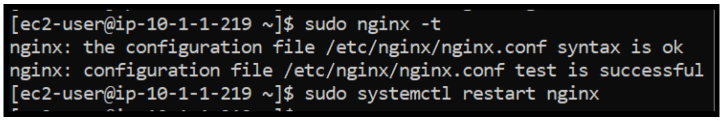
```
@Musfira-0514 → ~/Lab12 $ terraform output
aws_web-2_public_ip = "44.204.221.20"
aws_web_1_public_ip = "44.214.15.185"
subnet_id = "subnet-03c07a673981ae792"
vpc_id = "vpc-0f4da8cf8f6ba2aea"
webserver_public_ip = "23.23.41.49"
@Musfira-0514 → ~/Lab12 $
```

- Nginx Configuration for Load Balancing Between Web-1 and Web-2



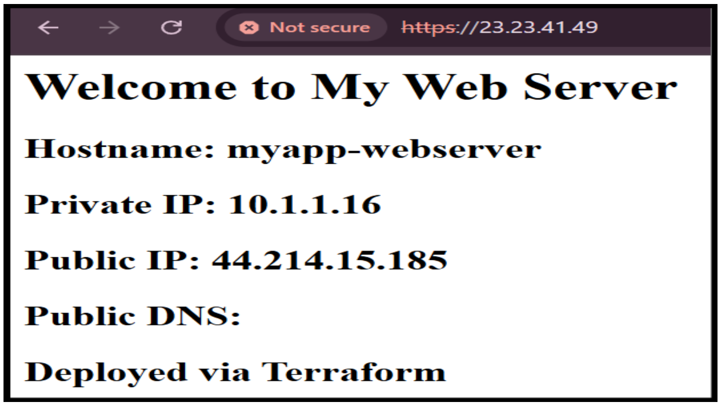
```
    ### ◇ LOAD BALANCER BACKEND ###
    upstream backend_servers {
        server 44.214.15.185:80;
        server 44.204.221.20:80;
    }

    ### ◇ HTTPS SERVER ###
    server {
        listen 443 ssl;
        server_name 23.23.41.49;

        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;

        location / {
            proxy_pass http://backend_servers;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }

    ### ◇ HTTP → HTTPS REDIRECT ###
    server {
        listen 80;
        server_name _;
        return 301 https://$host$request_uri;
    }
}
```

- Restarting Nginx Service After Load Balancer Configuration



```
[ec2-user@ip-10-1-1-219 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-1-1-219 ~]$ sudo systemctl restart nginx
```

- Browser Showing Web-1 Content via Load Balancer



Welcome to My Web Server

Hostname: myapp-webserver

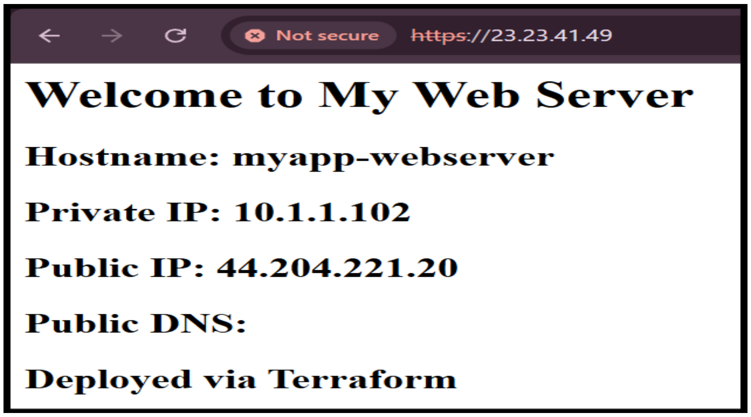
Private IP: 10.1.1.16

Public IP: 44.214.15.185

Public DNS:

Deployed via Terraform

- Browser Showing Web-2 Content via Load Balancer



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.1.1.102

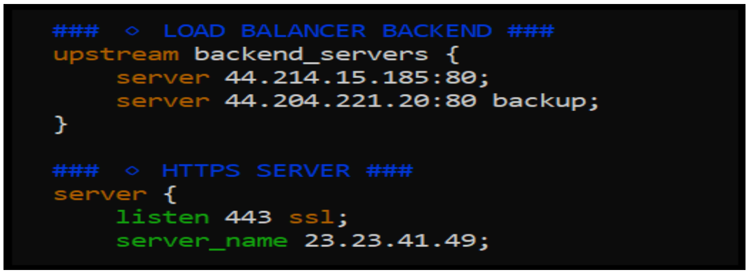
Public IP: 44.204.221.20

Public DNS:

Deployed via Terraform

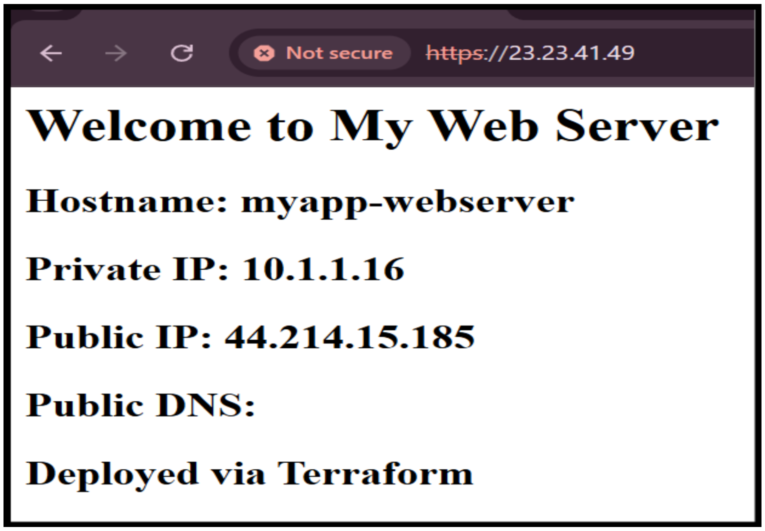
Task 9 — Configure High Availability with Backup Servers

- Nginx Configuration with Web-1 as Primary and Web-2 as Backup



```
### ◇ LOAD BALANCER BACKEND ###
upstream backend_servers {
    server 44.214.15.185:80;
    server 44.204.221.20:80 backup;
}

### ◇ HTTPS SERVER ###
server {
    listen 443 ssl;
    server_name 23.23.41.49;
```

- Browser Showing Only Web-1 Content (Primary Server)



Welcome to My Web Server

Hostname: myapp-webserver

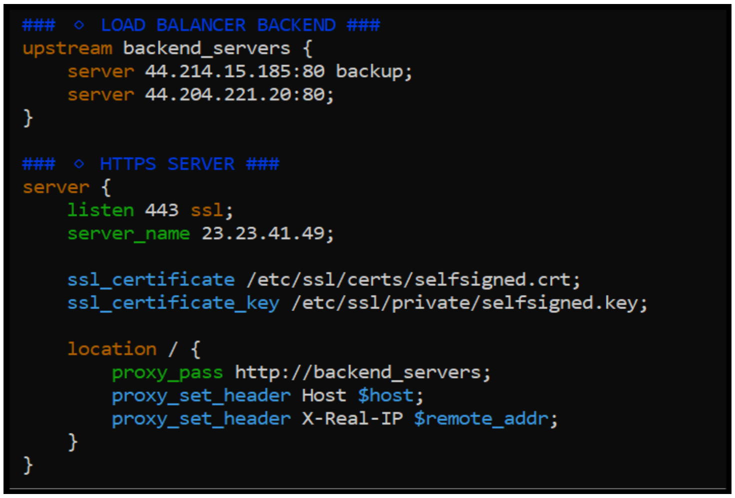
Private IP: 10.1.1.16

Public IP: 44.214.15.185

Public DNS:

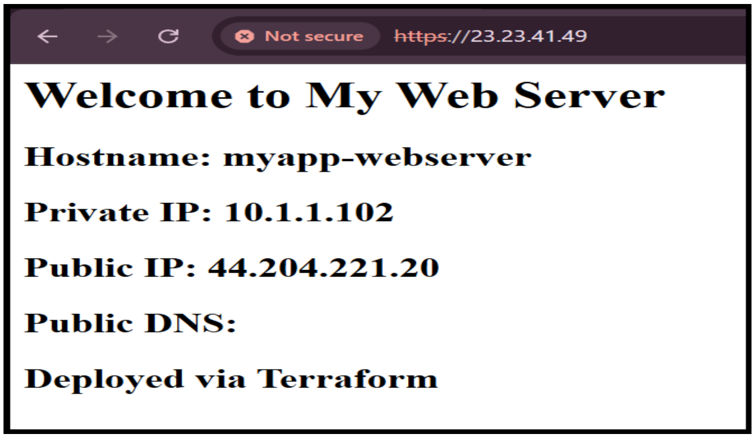
Deployed via Terraform

- Nginx Configuration with Web-2 as Primary and Web-1 as Backup



```
### ◇ LOAD BALANCER BACKEND ###
upstream backend_servers {
    server 44.214.15.185:80 backup;
    server 44.204.221.20:80;
}

### ◇ HTTPS SERVER ###
server {
    listen 443 ssl;
    server_name 23.23.41.49;

    ssl_certificate /etc/ssl/certs/selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

    location / {
        proxy_pass http://backend_servers;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

- Browser Showing Only Web-2 Content (Primary Server)



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.1.1.102

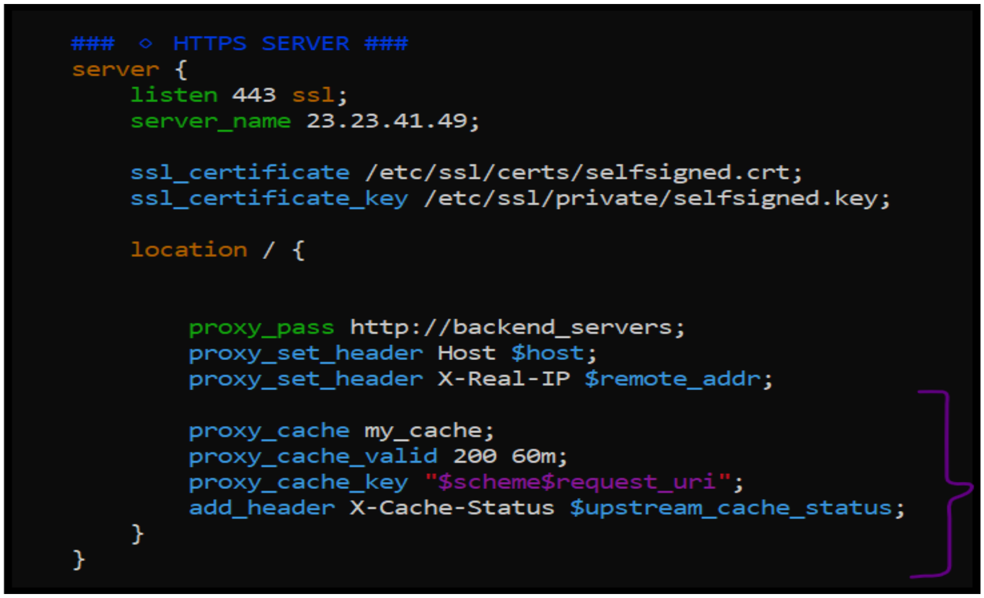
Public IP: 44.204.221.20

Public DNS:

Deployed via Terraform

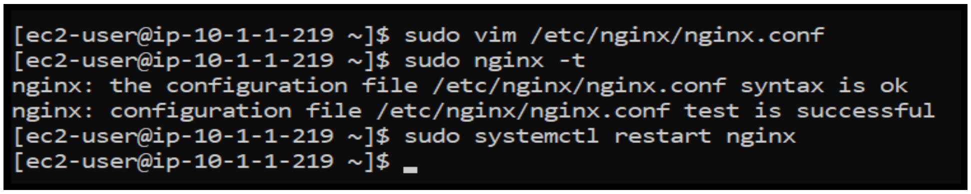
Task 10 — Enable Nginx Caching

- Nginx Configuration with Proxy Cache Enabled



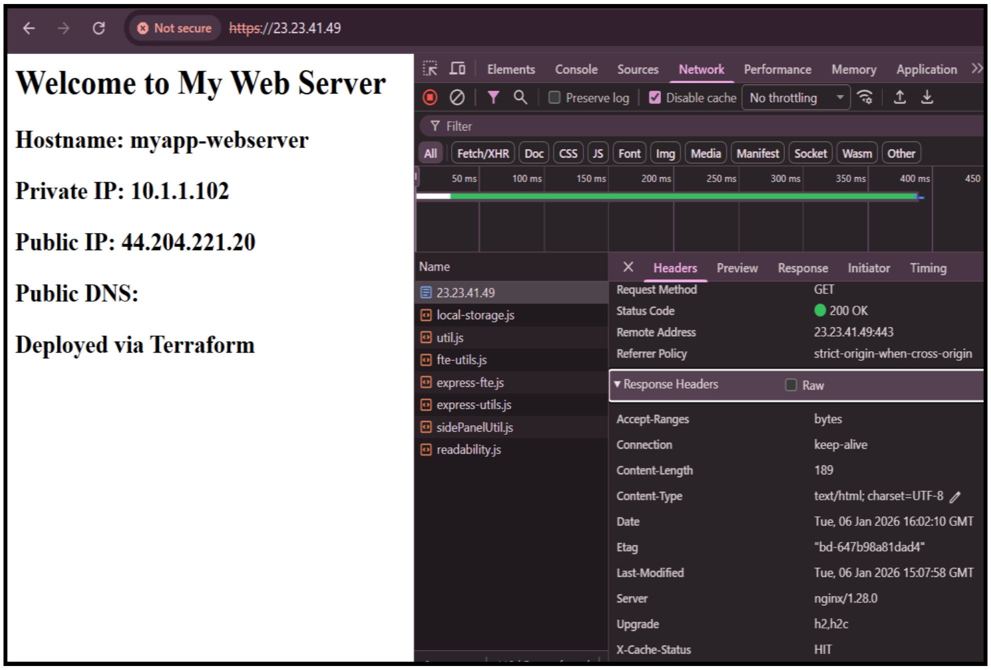
```
### ◇ HTTPS SERVER ###
server {
    listen 443 ssl;
    server_name 23.23.41.49;

    ssl_certificate /etc/ssl/certs/selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/selfsigned.key;

    location / {


        proxy_pass http://backend_servers;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;

        proxy_cache my_cache;
        proxy_cache_valid 200 60m;
        proxy_cache_key "$scheme$request_uri";
        add_header X-Cache-Status $upstream_cache_status;
    }
}
```

- Restarting Nginx Service After Cache Configuration



```
[ec2-user@ip-10-1-1-219 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-1-1-219 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-1-1-219 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-1-1-219 ~]$ _
```

- Browser Developer Tools Showing Cache HIT on Subsequent Request



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.1.1.102

Public IP: 44.204.221.20

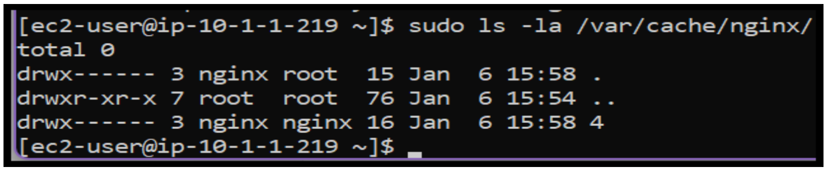
Public DNS:

Deployed via Terraform

```
23.23.41.49
local-storage.js
util.js
fte-utils.js
express-fte.js
express-utils.js
sidePanelUtil.js
readability.js
```

```
Request Method        GET
Status Code           ● 200 OK
Remote Address        23.23.41.49:443
Referrer Policy       strict-origin-when-cross-origin

▾ Response Headers    □ Raw

Accept-Ranges         bytes
Connection            keep-alive
Content-Length        189
Content-Type          text/html; charset=UTF-8
Date                  Tue, 06 Jan 2026 16:02:10 GMT
Etag                  "bd-647b98a81dad4"
Last-Modified         Tue, 06 Jan 2026 15:07:58 GMT
Server                nginx/1.28.0
Upgrade               h2,h2c
X-Cache-Status        HIT
```

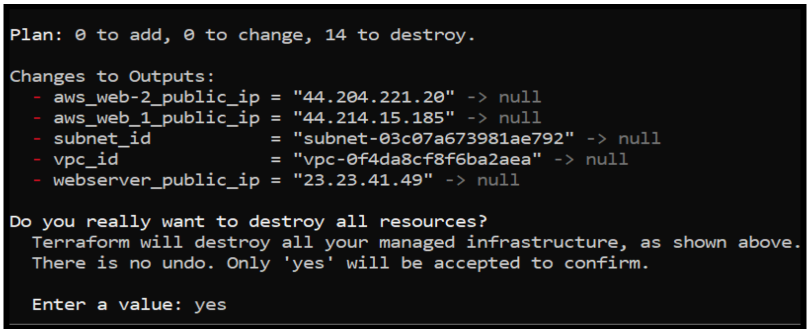
- Nginx Cache Directory Contents



```
[ec2-user@ip-10-1-1-219 ~]$ sudo ls -la /var/cache/nginx/
total 0
drwx------ 3 nginx root   15 Jan  6 15:58 .
drwxr-xr-x 7 root  root   76 Jan  6 15:54 ..
drwx------ 3 nginx nginx 16 Jan  6 15:58 4
[ec2-user@ip-10-1-1-219 ~]$ _
```

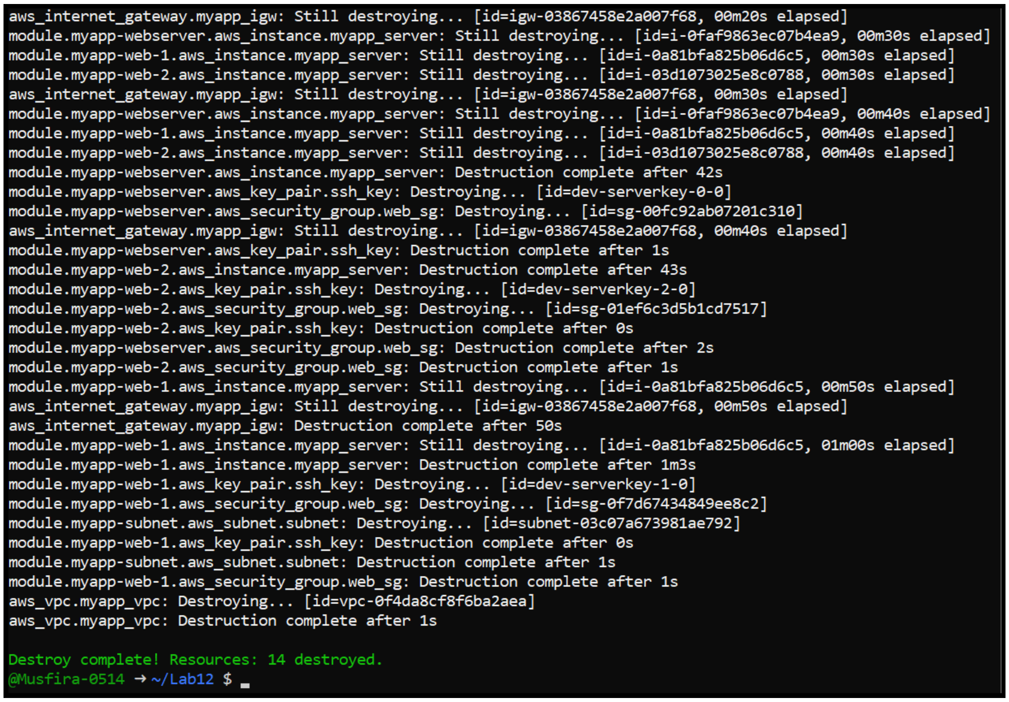
Cleanup

- Terraform Destroy Confirmation



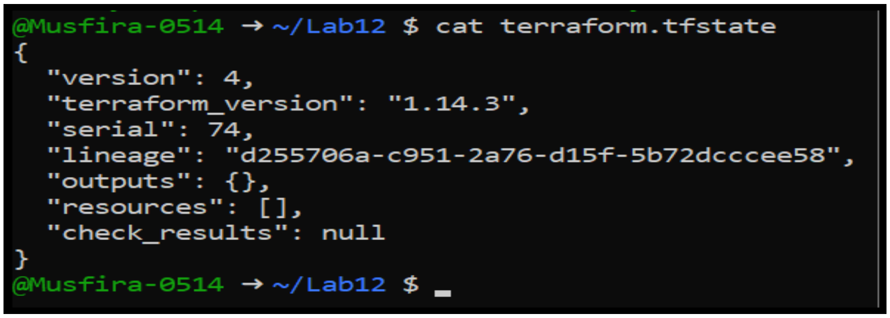
```
Plan: 0 to add, 0 to change, 14 to destroy.

Changes to Outputs:
  - aws_web-2_public_ip = "44.204.221.20" -> null
  - aws_web_1_public_ip = "44.214.15.185" -> null
  - subnet_id           = "subnet-03c07a673981ae792" -> null
  - vpc_id              = "vpc-0f4da8cf8f6ba2aea" -> null
  - webserver_public_ip = "23.23.41.49" -> null

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
```

- Successful Resource Deletion



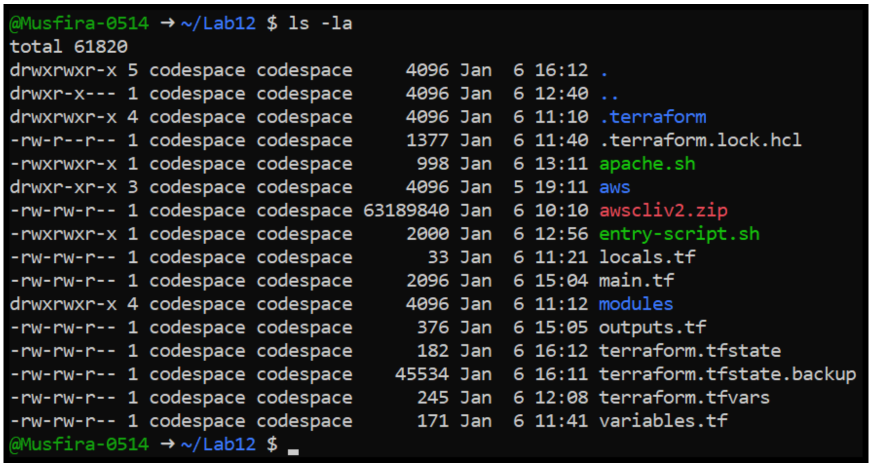
```
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-03867458e2a007f68, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still destroying... [id=i-0faf9863ec07b4ea9, 00m30s elapsed]
module.myapp-web-1.aws_instance.myapp_server: Still destroying... [id=i-0a81bfa825b06d6c5, 00m30s elapsed]
module.myapp-web-2.aws_instance.myapp_server: Still destroying... [id=i-03d1073025e8c0788, 00m30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-03867458e2a007f68, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Still destroying... [id=i-0faf9863ec07b4ea9, 00m40s elapsed]
module.myapp-web-1.aws_instance.myapp_server: Still destroying... [id=i-0a81bfa825b06d6c5, 00m40s elapsed]
module.myapp-web-2.aws_instance.myapp_server: Still destroying... [id=i-03d1073025e8c0788, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp_server: Destruction complete after 42s
module.myapp-webserver.aws_key_pair.ssh_key: Destroying... [id=dev-serverkey-0-0]
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-00fc92ab07201c310]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-03867458e2a007f68, 00m40s elapsed]
module.myapp-webserver.aws_key_pair.ssh_key: Destruction complete after 1s
module.myapp-web-2.aws_instance.myapp_server: Destruction complete after 43s
module.myapp-web-2.aws_key_pair.ssh_key: Destroying... [id=dev-serverkey-2-0]
module.myapp-web-2.aws_security_group.web_sg: Destroying... [id=sg-01ef6c3d5b1cd7517]
module.myapp-web-2.aws_key_pair.ssh_key: Destruction complete after 0s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 2s
module.myapp-web-2.aws_security_group.web_sg: Destruction complete after 1s
module.myapp-web-1.aws_instance.myapp_server: Still destroying... [id=i-0a81bfa825b06d6c5, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-03867458e2a007f68, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 50s
module.myapp-web-1.aws_instance.myapp_server: Still destroying... [id=i-0a81bfa825b06d6c5, 01m00s elapsed]
module.myapp-web-1.aws_instance.myapp_server: Destruction complete after 1m3s
module.myapp-web-1.aws_key_pair.ssh_key: Destroying... [id=dev-serverkey-1-0]
module.myapp-web-1.aws_security_group.web_sg: Destroying... [id=sg-0f7d67434849ee8c2]
module.myapp-subnet.aws_subnet.subnet: Destroying... [id=subnet-03c07a673981ae792]
module.myapp-web-1.aws_key_pair.ssh_key: Destruction complete after 0s
module.myapp-subnet.aws_subnet.subnet: Destruction complete after 1s
module.myapp-web-1.aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0f4da8cf8f6ba2aea]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 14 destroyed.
@Musfira-0514 → ~/Lab12 $ _
```

- Verification of Empty Terraform State



```
@Musfira-0514 → ~/Lab12 $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 74,
  "lineage": "d255706a-c951-2a76-d15f-5b72dcccee58",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Musfira-0514 → ~/Lab12 $ _
```

- Final Project File Structure



```
@Musfira-0514 → ~/Lab12 $ ls -la
total 61820
drwxrwxr-x 5 codespace codespace     4096 Jan  6 16:12 .
drwxr-x--- 1 codespace codespace     4096 Jan  6 12:40 ..
drwxrwxr-x 4 codespace codespace     4096 Jan  6 11:10 .terraform
-rw-r--r-- 1 codespace codespace     1377 Jan  6 11:40 .terraform.lock.hcl
-rwxrwxr-x 1 codespace codespace      998 Jan  6 13:11 apache.sh
drwxr-xr-x 3 codespace codespace     4096 Jan  5 19:11 aws
-rw-rw-r-- 1 codespace codespace 63189840 Jan  6 10:10 awscliv2.zip
-rwxrwxr-x 1 codespace codespace     2000 Jan  6 12:56 entry-script.sh
-rw-rw-r-- 1 codespace codespace       33 Jan  6 11:21 locals.tf
-rw-rw-r-- 1 codespace codespace     2096 Jan  6 15:04 main.tf
drwxrwxr-x 4 codespace codespace     4096 Jan  6 11:12 modules
-rw-rw-r-- 1 codespace codespace      376 Jan  6 15:05 outputs.tf
-rw-rw-r-- 1 codespace codespace      182 Jan  6 16:12 terraform.tfstate
-rw-rw-r-- 1 codespace codespace    45534 Jan  6 16:11 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace      245 Jan  6 12:08 terraform.tfvars
-rw-rw-r-- 1 codespace codespace      171 Jan  6 11:41 variables.tf
@Musfira-0514 → ~/Lab12 $ _
```