FATIMA JINNAH WOMEN UNIVERSITY

Department of Software Engineering

LAB #12

SUBJECT: CLOUD COMPUTING

SUBMITTED TO: SIR MUHAMMAD SHOAIB

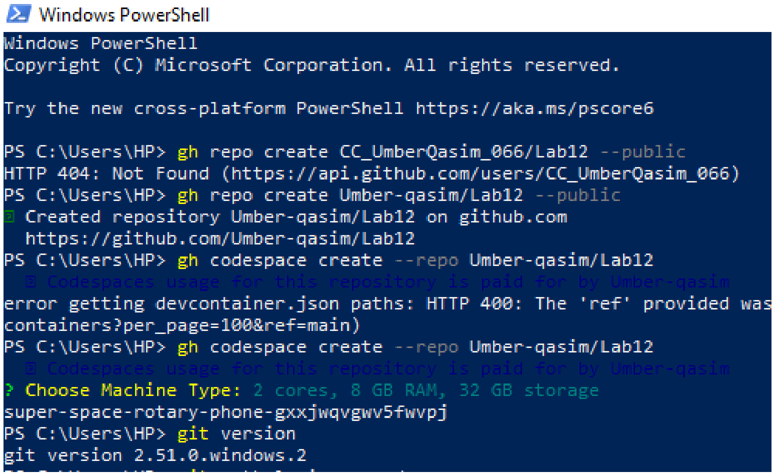
SUBMITTED BY: UMBER QASIM

REGISTRATION NO: 2023-BSE-066

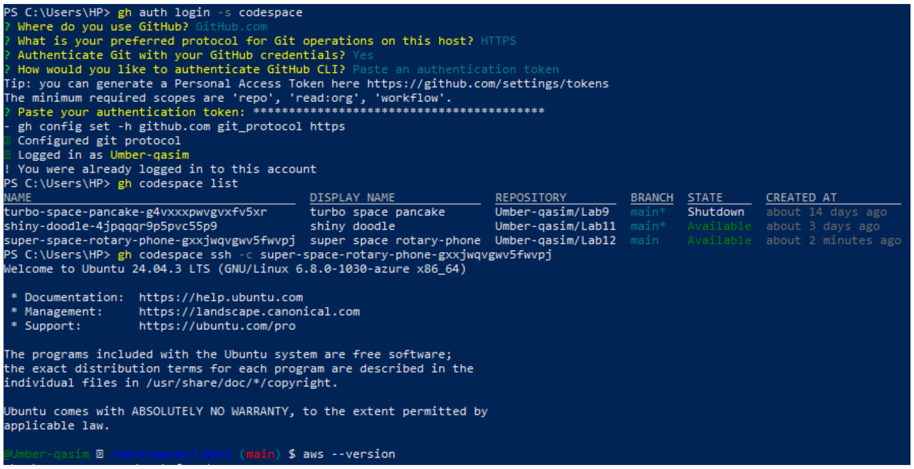
CLASS: BSSE V-B

Terraform Provisioners, Modules & Nginx Reverse Proxy/Load Balancer

Task#0: Lab Setup (Codespace & GH CLI)

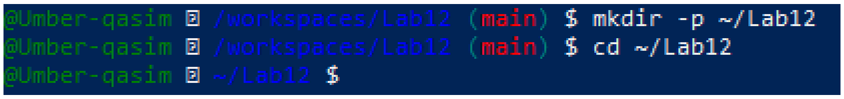


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\HP> gh repo create CC_UmberQasim_066/Lab12 --public
HTTP 404: Not Found (https://api.github.com/users/CC_UmberQasim_066)
PS C:\Users\HP> gh repo create Umber-qasim/Lab12 --public
☐ Created repository Umber-qasim/Lab12 on github.com
  https://github.com/Umber-qasim/Lab12
PS C:\Users\HP> gh codespace create --repo Umber-qasim/Lab12
  ☐ Codespaces usage for this repository is paid for by Umber-qasim
error getting devcontainer.json paths: HTTP 400: The 'ref' provided was
containers?per_page=100&ref=main)
PS C:\Users\HP> gh codespace create --repo Umber-qasim/Lab12
  ☐ Codespaces usage for this repository is paid for by Umber-qasim
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
super-space-rotary-phone-gxxjwqvgwv5fwvpj
PS C:\Users\HP> git version
git version 2.51.0.windows.2
```

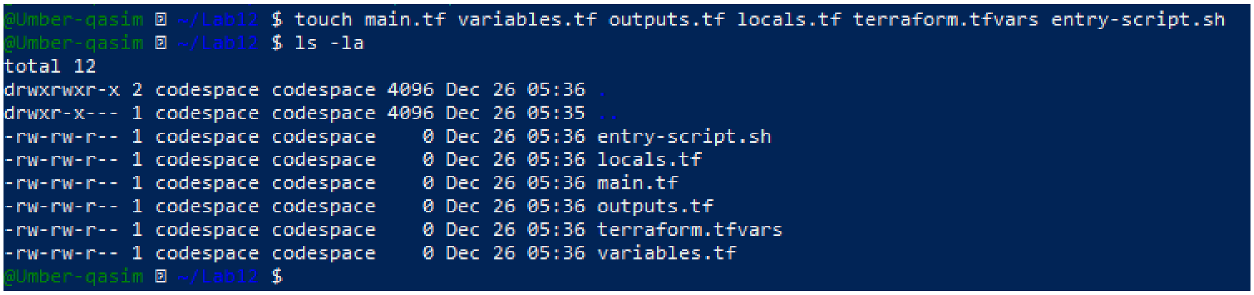


```
PS C:\Users\HP> gh auth login -s codespace
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: ****************************************
- gh config set -h github.com git_protocol https
☐ Configured git protocol
☐ Logged in as Umber-qasim
! You were already logged in to this account
PS C:\Users\HP> gh codespace list
NAME                                       DISPLAY NAME               REPOSITORY          BRANCH  STATE      CREATED AT
turbo-space-pancake-g4vxxxpwvgvxfv5xr      turbo space pancake        Umber-qasim/Lab9    main*   Shutdown   about 14 days ago
shiny-doodle-4jpqqqr9p5pvc55p9             shiny doodle               Umber-qasim/Lab11   main*   Available  about 3 days ago
super-space-rotary-phone-gxxjwqvgwv5fwvpj  super space rotary-phone   Umber-qasim/Lab12   main    Available  about 2 minutes ago
PS C:\Users\HP> gh codespace ssh -c super-space-rotary-phone-gxxjwqvgwv5fwvpj
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.

@Umber-qasim ☐ /workspaces/Lab12 (main) $ aws --version
```

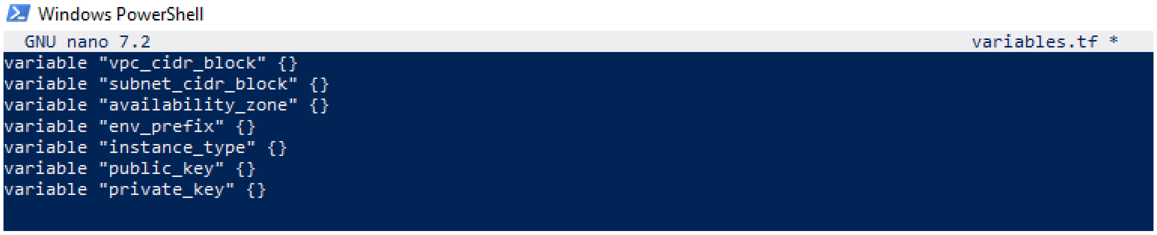
Task#01: Organize Terraform code into separate files



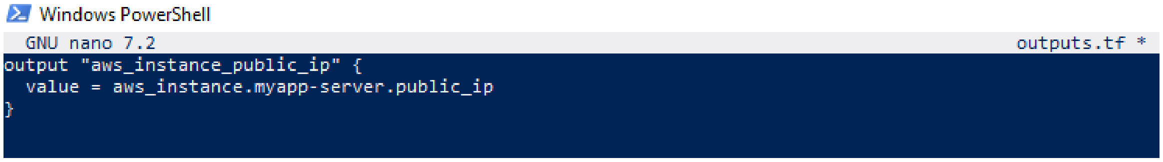
```
@Umber-qasim ☐ /workspaces/Lab12 (main) $ mkdir -p ~/Lab12
@Umber-qasim ☐ /workspaces/Lab12 (main) $ cd ~/Lab12
@Umber-qasim ☐ ~/Lab12 $
```



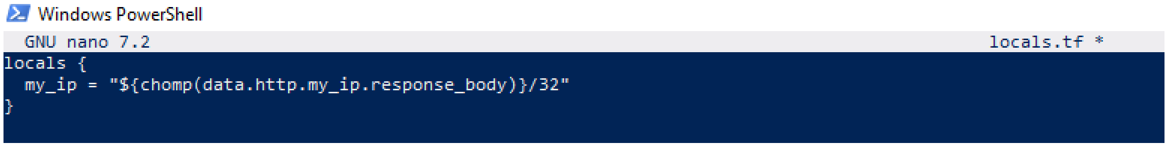
```
@Umber-qasim ☐ ~/Lab12 $ touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
@Umber-qasim ☐ ~/Lab12 $ ls -la
total 12
drwxrwxrwx 2 codespace codespace 4096 Dec 26 05:36 .
drwxr-x--- 1 codespace codespace 4096 Dec 26 05:35 ..
-rw-rw-r-- 1 codespace codespace    0 Dec 26 05:36 entry-script.sh
-rw-rw-r-- 1 codespace codespace    0 Dec 26 05:36 locals.tf
-rw-rw-r-- 1 codespace codespace    0 Dec 26 05:36 main.tf
-rw-rw-r-- 1 codespace codespace    0 Dec 26 05:36 outputs.tf
-rw-rw-r-- 1 codespace codespace    0 Dec 26 05:36 terraform.tfvars
-rw-rw-r-- 1 codespace codespace    0 Dec 26 05:36 variables.tf
@Umber-qasim ☐ ~/Lab12 $
```



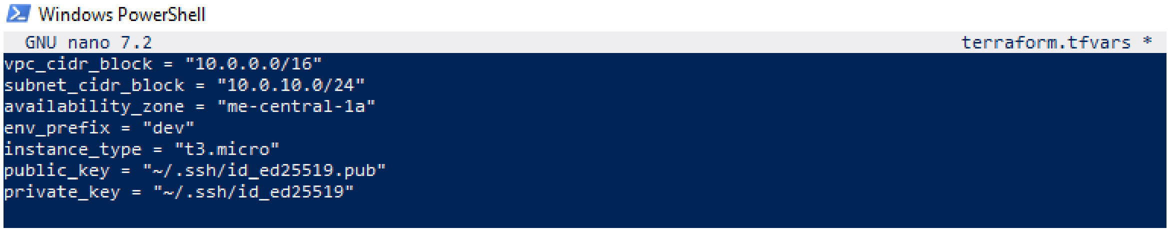
```
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
variable "public_key" {}
variable "private_key" {}
```



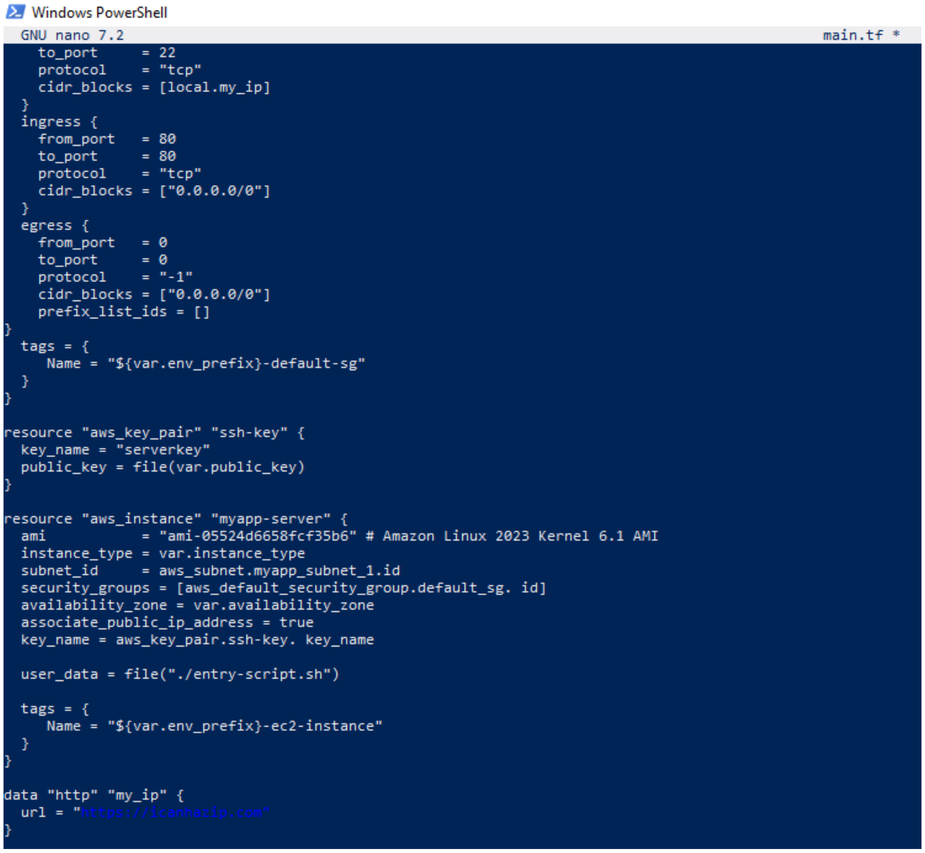
```
output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}
```



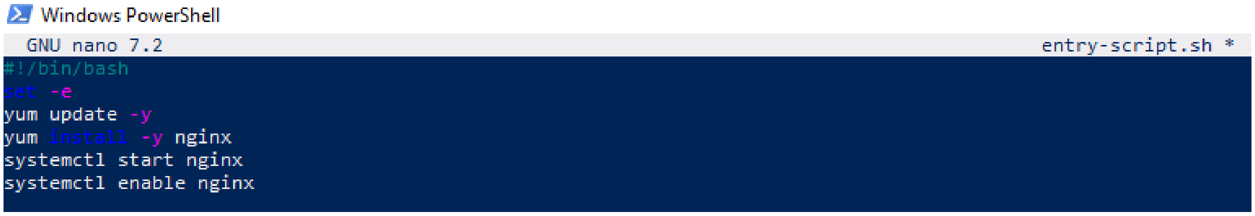
```
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}
```



```
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"
```

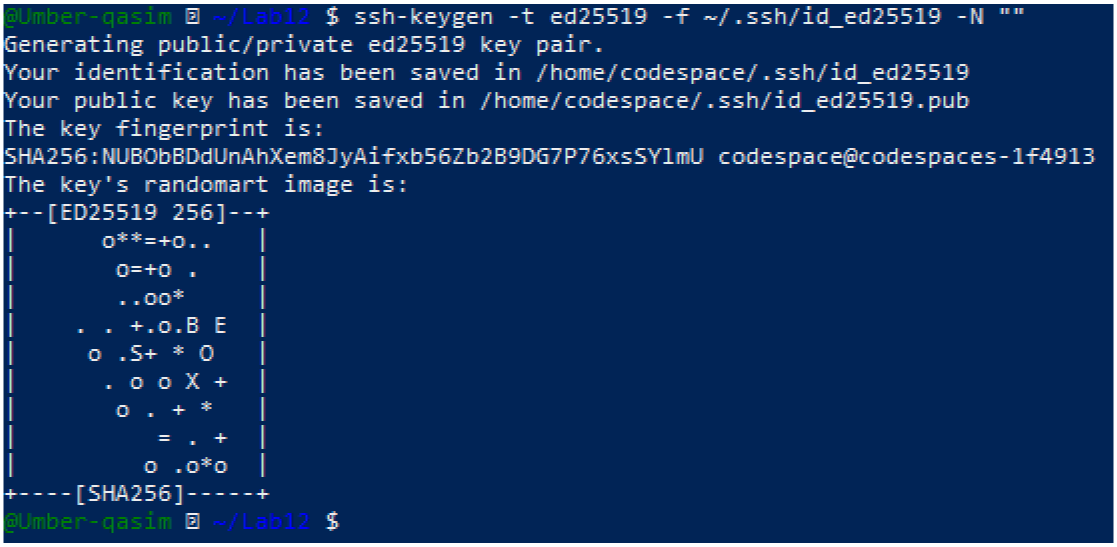


```
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [local.my_ip]
  }
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
    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"
  }
}

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

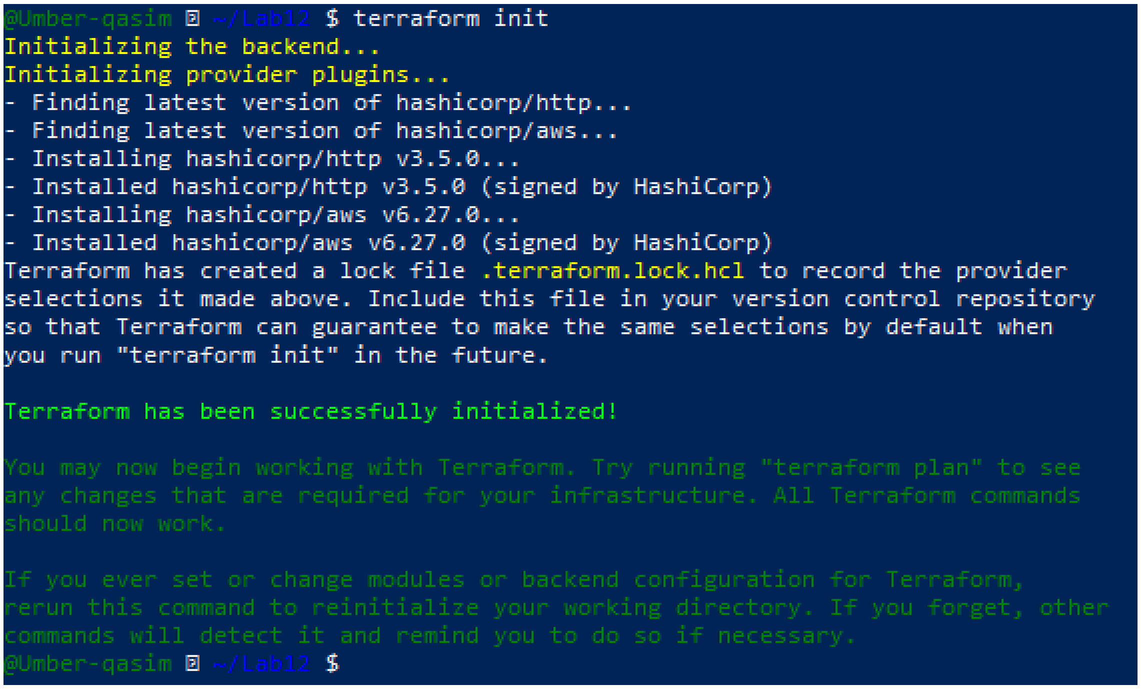


Windows PowerShell

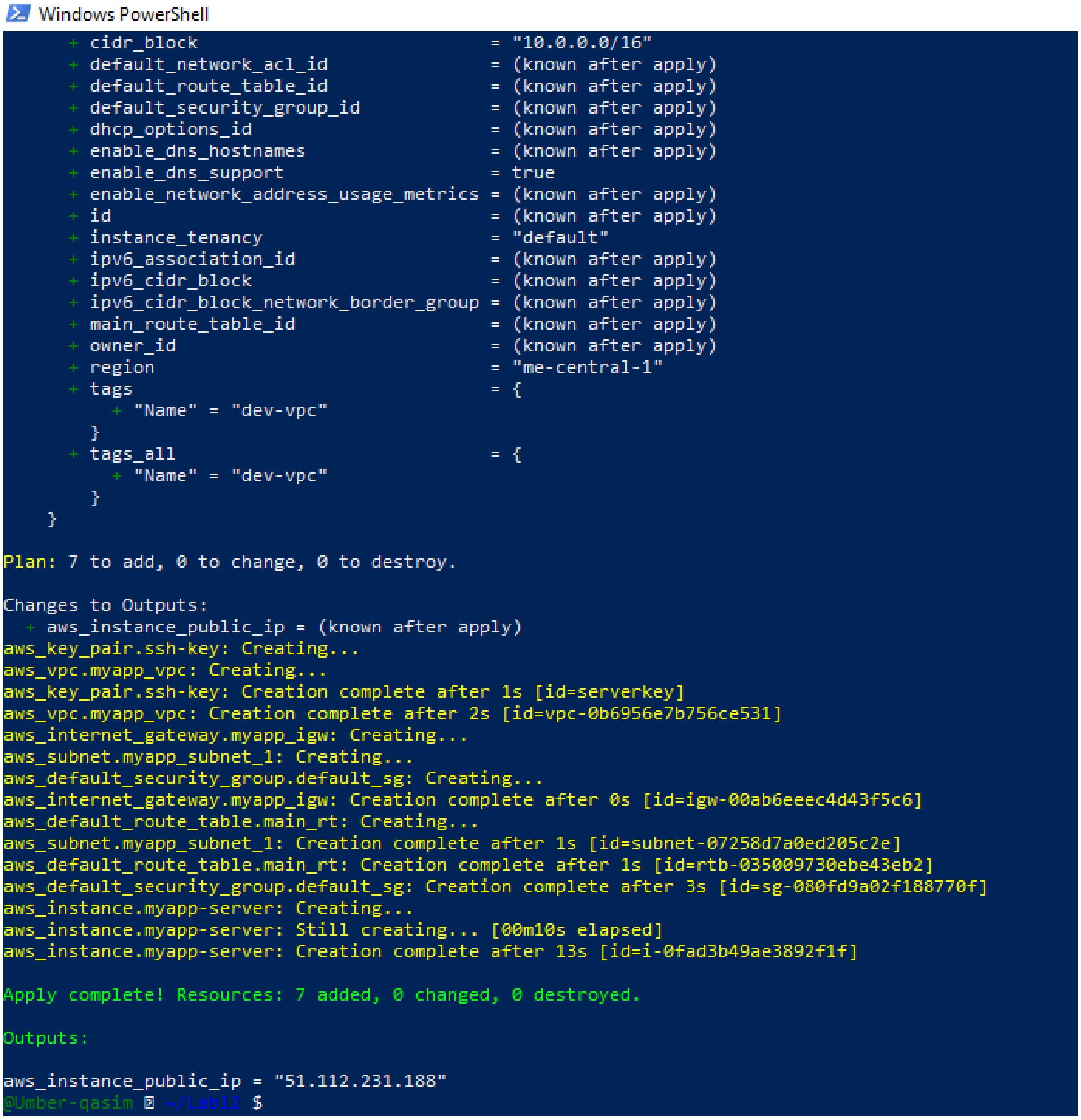
```
  GNU nano 7.2                                                    entry-script.sh *
#!/bin/bash
set -e
yum update -y
yum install -y nginx
systemctl start nginx
systemctl enable nginx
```



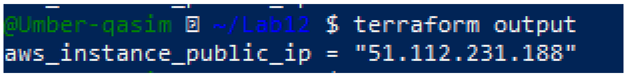
```
@Umber-qasim ⍰ ~/Lab12 $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:NUBObBDdUnAhXem8JyAifxb56Zb2B9DG7P76xsSYlmU codespace@codespaces-1f4913
The key's randomart image is:
+--[ED25519 256]--+
|      o**=+o..   |
|       o=+o .    |
|       ..oo*     |
|    . . +.o.B E  |
|     o .S+ * O   |
|      . o o X +  |
|       o . + *   |
|          = . +  |
|         o .o*o  |
+----[SHA256]-----+
@Umber-qasim ⍰ ~/Lab12 $
```



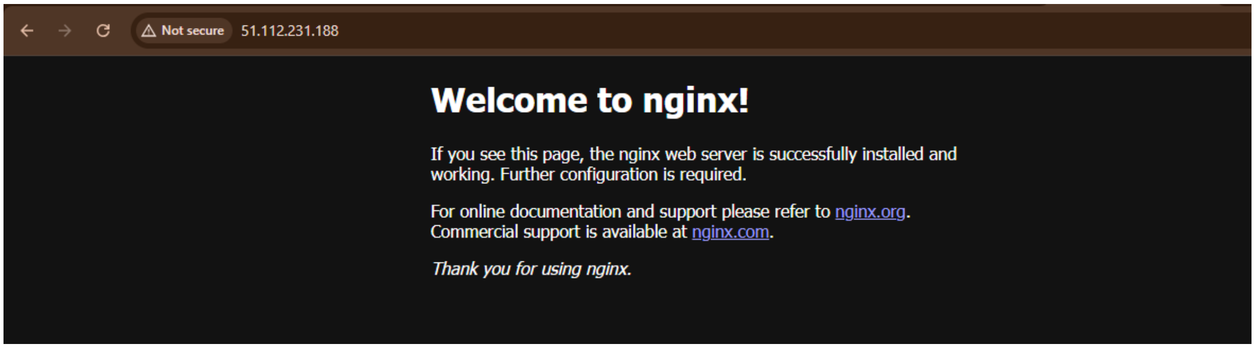
```
@Umber-qasim ⍰ ~/Lab12 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/http...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing 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!

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.
@Umber-qasim ⍰ ~/Lab12 $
```



```
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "me-central-1"
      + tags                                 = {
          + "Name" = "dev-vpc"
        }
      + tags_all                             = {
          + "Name" = "dev-vpc"
        }
    }

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

Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
aws_key_pair.ssh-key: Creation complete after 1s [id=serverkey]
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-0b6956e7b756ce531]
aws_internet_gateway.myapp_igw: Creating...
aws_subnet.myapp_subnet_1: Creating...
aws_default_security_group.default_sg: Creating...
aws_internet_gateway.myapp_igw: Creation complete after 0s [id=igw-00ab6eeec4d43f5c6]
aws_default_route_table.main_rt: Creating...
aws_subnet.myapp_subnet_1: Creation complete after 1s [id=subnet-07258d7a0ed205c2e]
aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-035009730ebe43eb2]
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-080fd9a02f188770f]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 13s [id=i-0fad3b49ae3892f1f]

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

Outputs:

aws_instance_public_ip = "51.112.231.188"
@Umber-qasim □ ~/Lab12 $
```



```
@Umber-qasim □ ~/Lab12 $ terraform output
aws_instance_public_ip = "51.112.231.188"
```

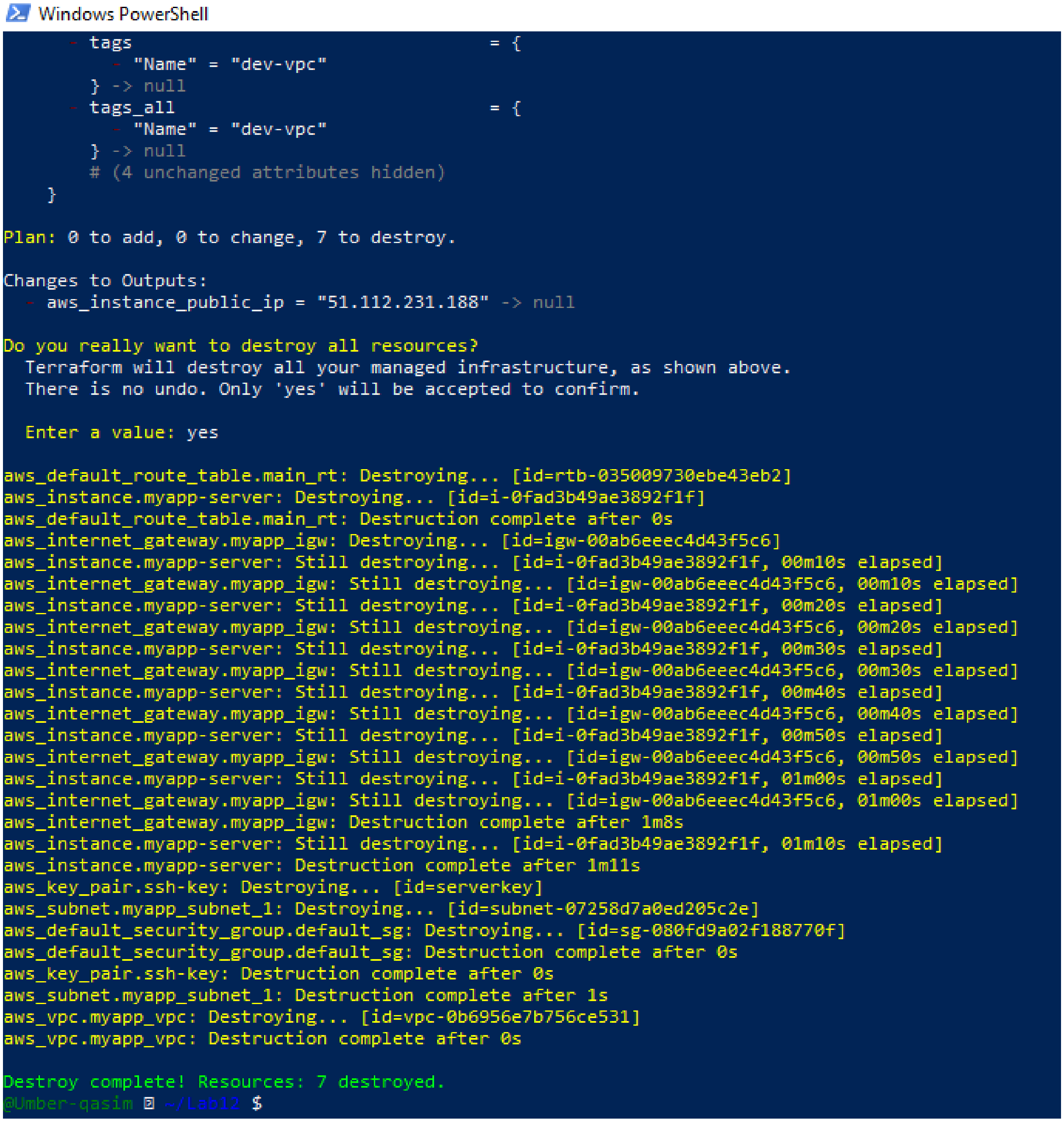


Welcome to nginx!

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

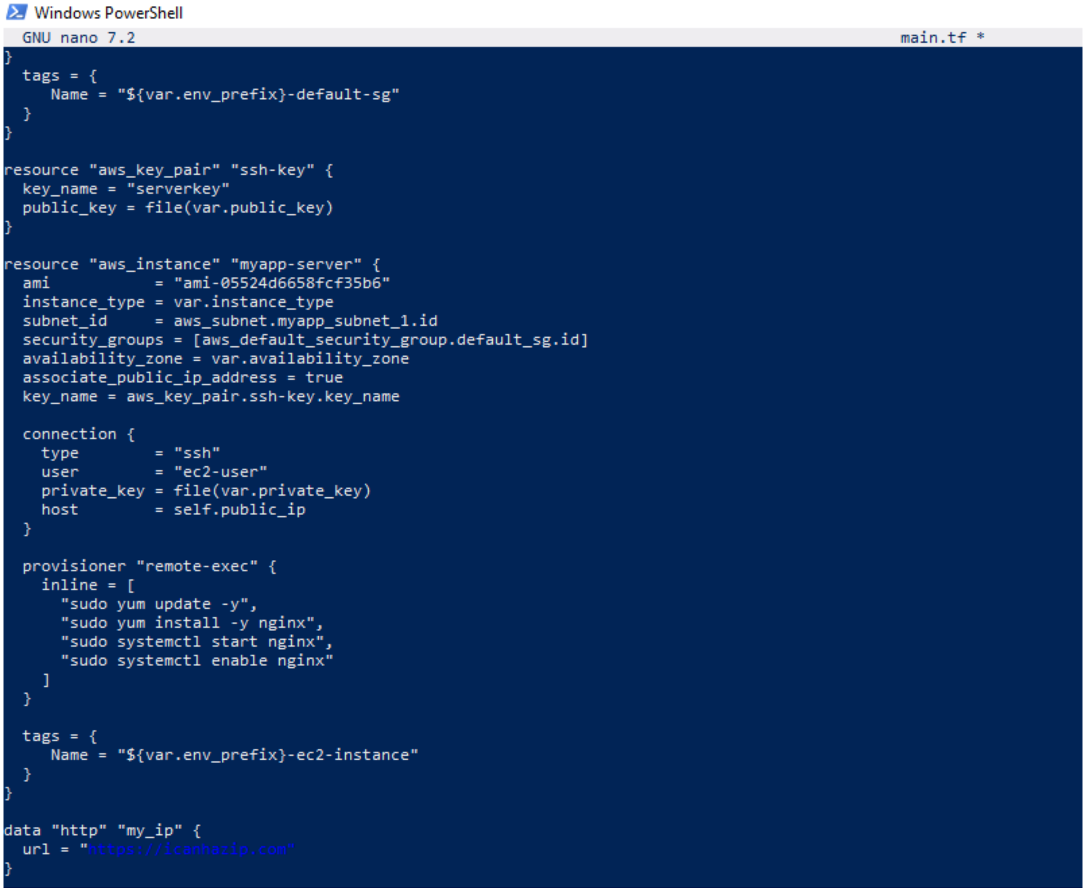
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



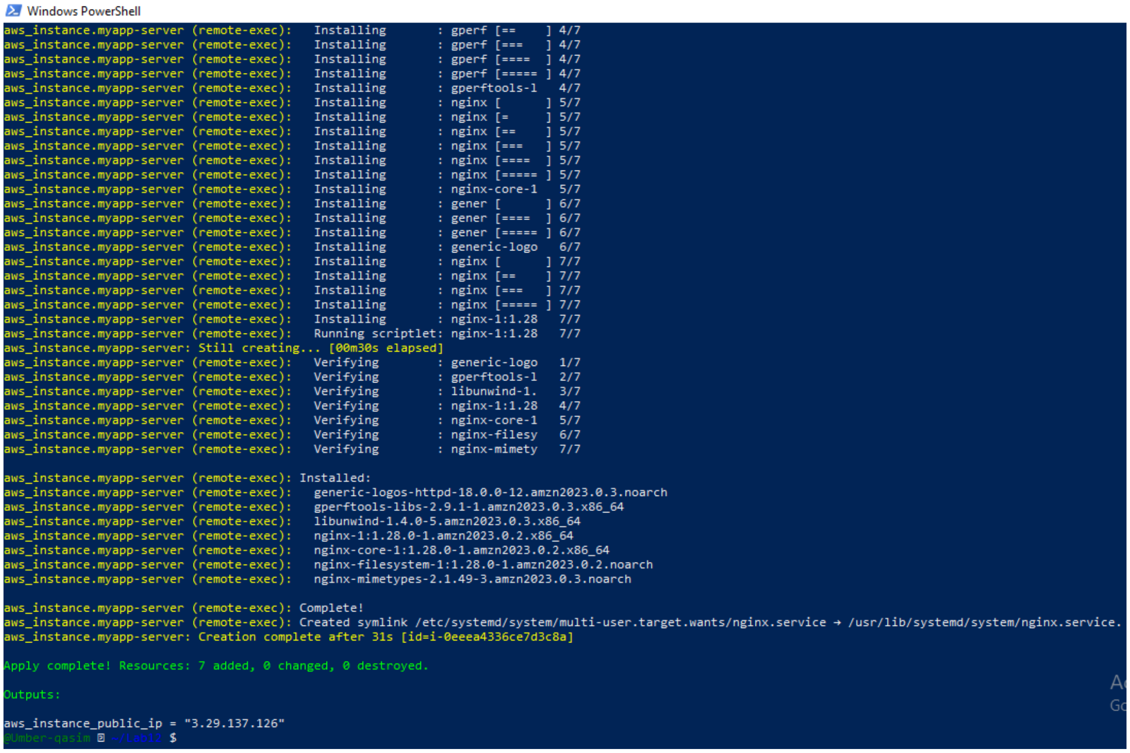
```
      - tags                                = {
          - "Name" = "dev-vpc"
        } -> null
      - tags_all                            = {
          - "Name" = "dev-vpc"
        } -> null
        # (4 unchanged attributes hidden)
    }

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

Changes to Outputs:
  - aws_instance_public_ip = "51.112.231.188" -> 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

aws_default_route_table.main_rt: Destroying... [id=rtb-035009730ebe43eb2]
aws_instance.myapp-server: Destroying... [id=i-0fad3b49ae3892f1f]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_internet_gateway.myapp_igw: Destroying... [id=igw-00ab6eeec4d43f5c6]
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00ab6eeec4d43f5c6, 00m10s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 00m20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00ab6eeec4d43f5c6, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 00m30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00ab6eeec4d43f5c6, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 00m40s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00ab6eeec4d43f5c6, 00m40s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00ab6eeec4d43f5c6, 00m50s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 01m00s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-00ab6eeec4d43f5c6, 01m00s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 1m8s
aws_instance.myapp-server: Still destroying... [id=i-0fad3b49ae3892f1f, 01m10s elapsed]
aws_instance.myapp-server: Destruction complete after 1m11s
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-07258d7a0ed205c2e]
aws_default_security_group.default_sg: Destroying... [id=sg-080fd9a02f188770f]
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-0b6956e7b756ce531]
aws_vpc.myapp_vpc: Destruction complete after 0s

Destroy complete! Resources: 7 destroyed.
@Umber-qasim ▯ ~/Lab12 $
```

Task#02: Use remote-exec provisioner



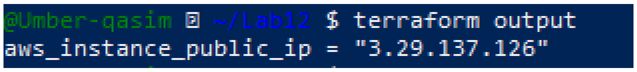
Windows PowerShell

```
GNU nano 7.2                                                                main.tf *
}
  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

  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"
  }
}

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

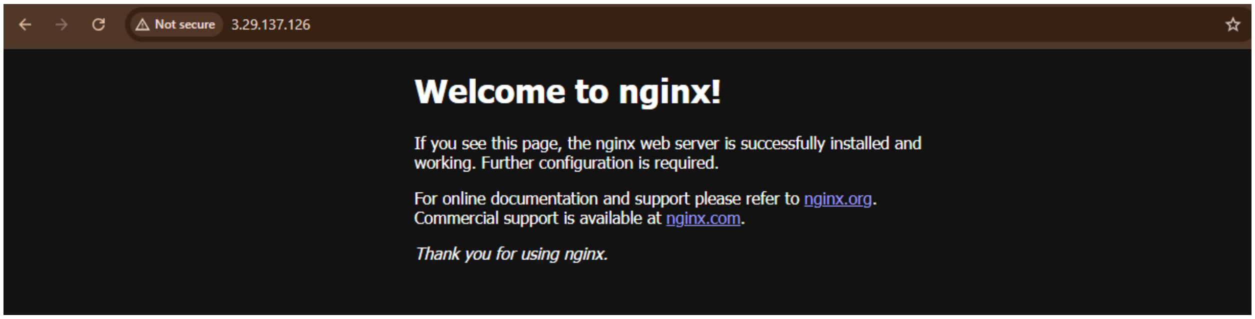


Windows PowerShell

```
aws_instance.myapp-server (remote-exec):   Installing       : gperf [==    ] 4/7
aws_instance.myapp-server (remote-exec):   Installing       : gperf [===   ] 4/7
aws_instance.myapp-server (remote-exec):   Installing       : gperf [====  ] 4/7
aws_instance.myapp-server (remote-exec):   Installing       : gperf [===== ] 4/7
aws_instance.myapp-server (remote-exec):   Installing       : gperftools-l   4/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [      ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [=     ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [==    ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===   ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [====  ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===== ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx-core-1   5/7
aws_instance.myapp-server (remote-exec):   Installing       : gener [      ] 6/7
aws_instance.myapp-server (remote-exec):   Installing       : gener [====  ] 6/7
aws_instance.myapp-server (remote-exec):   Installing       : gener [===== ] 6/7
aws_instance.myapp-server (remote-exec):   Installing       : generic-logo   6/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [      ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [==    ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===   ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===== ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx-1:1.28   7/7
aws_instance.myapp-server (remote-exec):   Running scriptlet: nginx-1:1.28   7/7
aws_instance.myapp-server: Still creating... [00m30s elapsed]
aws_instance.myapp-server (remote-exec):   Verifying        : generic-logo   1/7
aws_instance.myapp-server (remote-exec):   Verifying        : gperftools-l   2/7
aws_instance.myapp-server (remote-exec):   Verifying        : libunwind-1.   3/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-1:1.28   4/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-core-1   5/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-filesy   6/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-mimety   7/7

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

aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
aws_instance.myapp-server: Creation complete after 31s [id=i-0eeea4336ce7d3c8a]

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

Outputs:

aws_instance_public_ip = "3.29.137.126"
@Umber-qasim □ ~/Lab12 $
```



```
@Umber-qasim ⍰ ~/Lab12 $ terraform output
aws_instance_public_ip = "3.29.137.126"
```



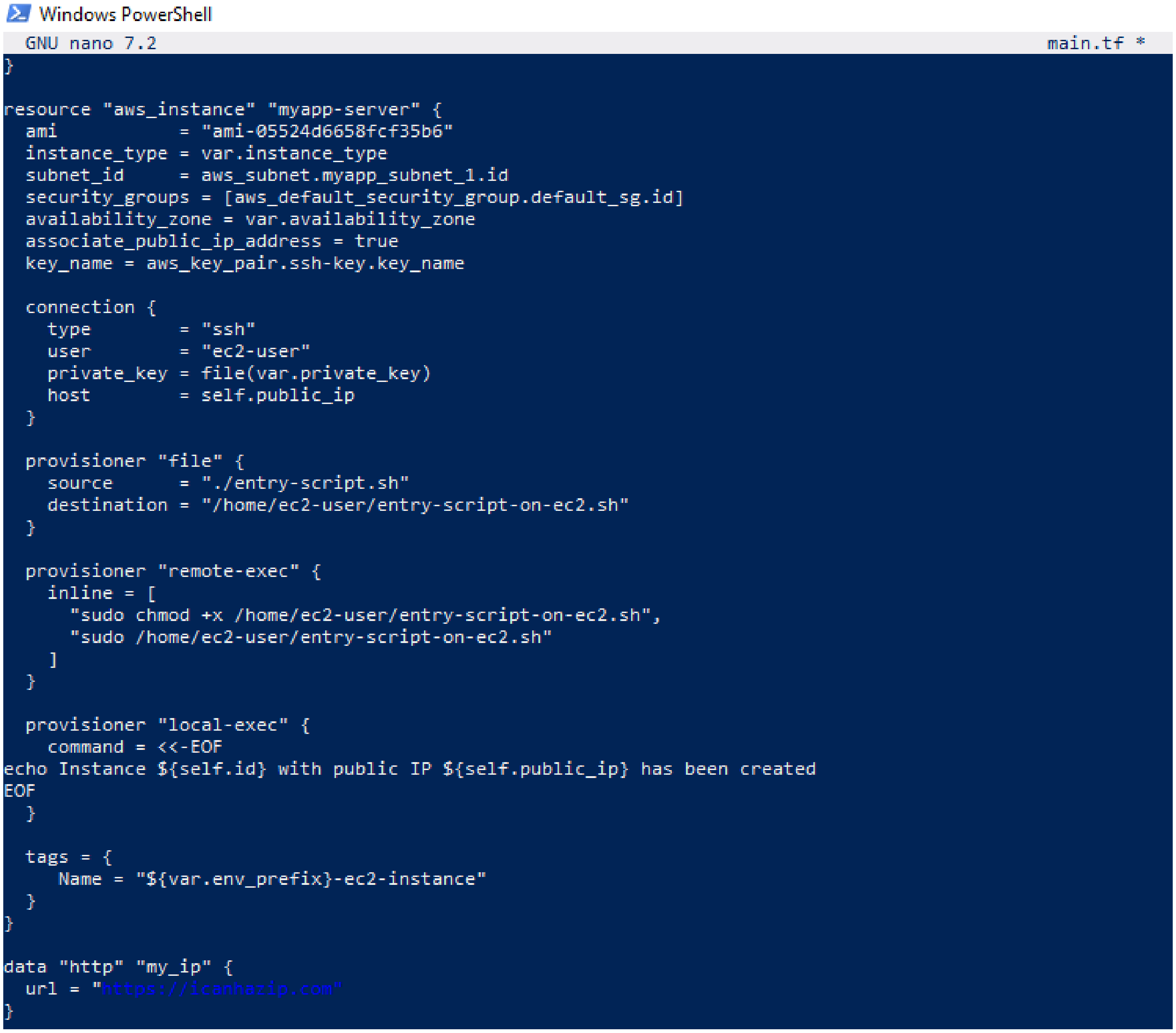
Welcome to nginx!

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

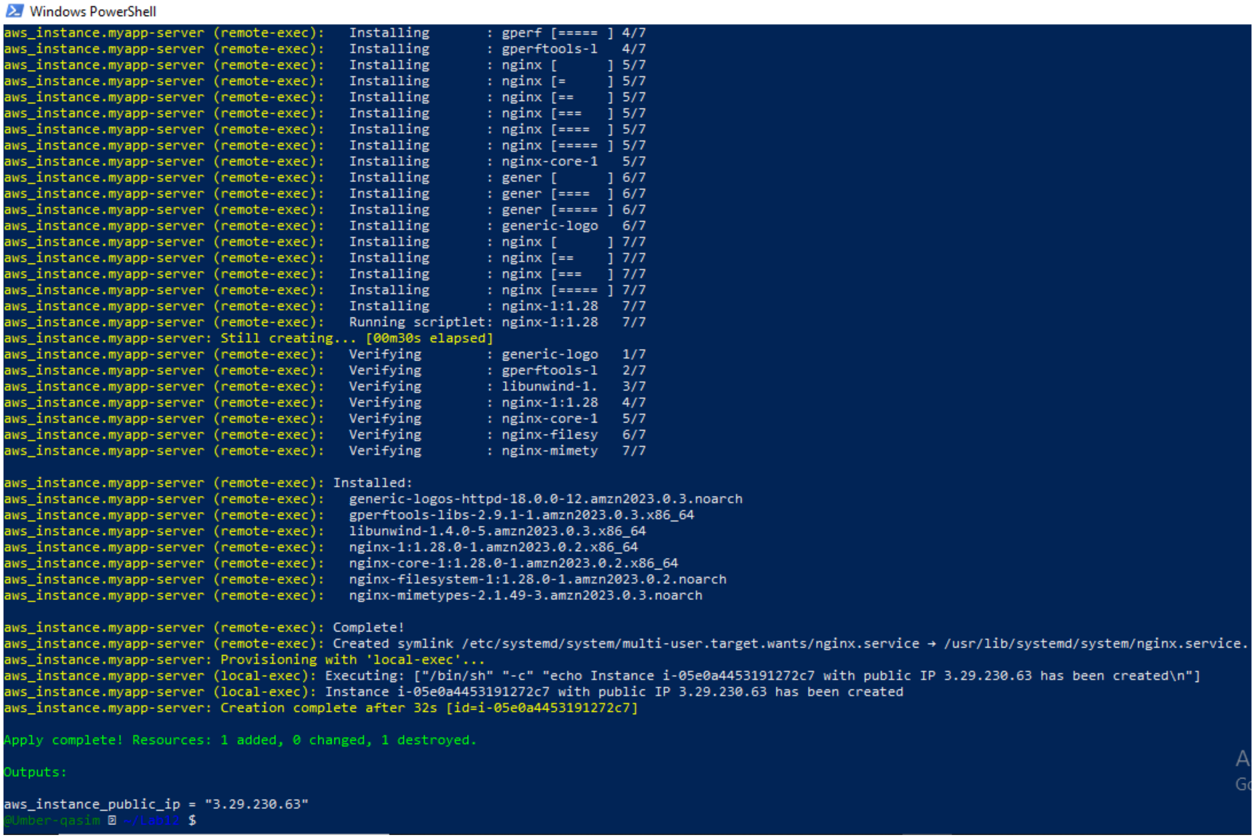
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

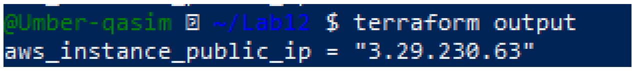
Task#03: Use file and local-exec provisioners



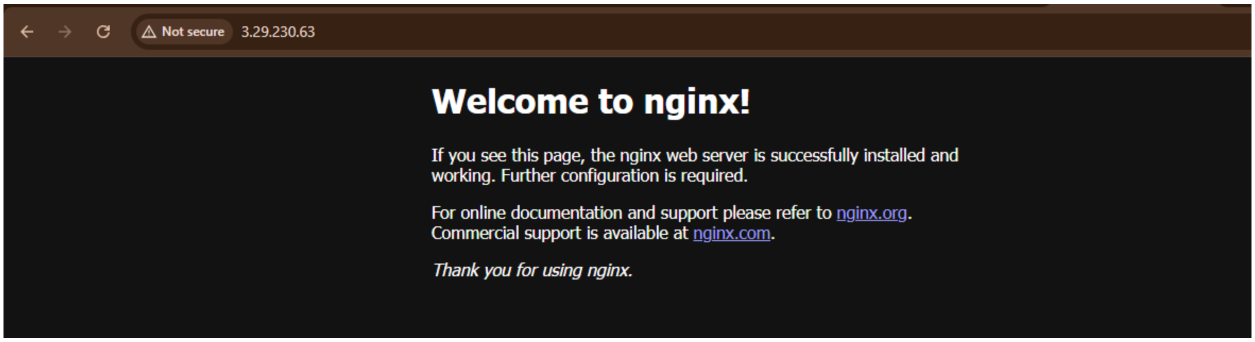
```
}

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 "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"
  }
}

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



```
aws_instance.myapp-server (remote-exec):   Installing       : gperf [===== ] 4/7
aws_instance.myapp-server (remote-exec):   Installing       : gperftools-l   4/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [      ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [=     ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [==    ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===   ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [====  ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===== ] 5/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx-core-1   5/7
aws_instance.myapp-server (remote-exec):   Installing       : gener [      ] 6/7
aws_instance.myapp-server (remote-exec):   Installing       : gener [====  ] 6/7
aws_instance.myapp-server (remote-exec):   Installing       : gener [===== ] 6/7
aws_instance.myapp-server (remote-exec):   Installing       : generic-logo   6/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [      ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [==    ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===   ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx [===== ] 7/7
aws_instance.myapp-server (remote-exec):   Installing       : nginx-1:1.28   7/7
aws_instance.myapp-server (remote-exec):   Running scriptlet: nginx-1:1.28   7/7
aws_instance.myapp-server: Still creating... [00m30s elapsed]
aws_instance.myapp-server (remote-exec):   Verifying        : generic-logo   1/7
aws_instance.myapp-server (remote-exec):   Verifying        : gperftools-l   2/7
aws_instance.myapp-server (remote-exec):   Verifying        : libunwind-1.   3/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-1:1.28   4/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-core-1   5/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-filesy   6/7
aws_instance.myapp-server (remote-exec):   Verifying        : nginx-mimety   7/7

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

aws_instance.myapp-server (remote-exec): Complete!
aws_instance.myapp-server (remote-exec): Created symlink /etc/systemd/system/multi-user.target.wants/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-05e0a4453191272c7 with public IP 3.29.230.63 has been created\n"]
aws_instance.myapp-server (local-exec): Instance i-05e0a4453191272c7 with public IP 3.29.230.63 has been created
aws_instance.myapp-server: Creation complete after 32s [id=i-05e0a4453191272c7]

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

Outputs:

aws_instance_public_ip = "3.29.230.63"
@Umber-qasim ⊠ ~/Lab12 $
```



```
@Umber-qasim ⊠ ~/Lab12 $ terraform output
aws_instance_public_ip = "3.29.230.63"
```

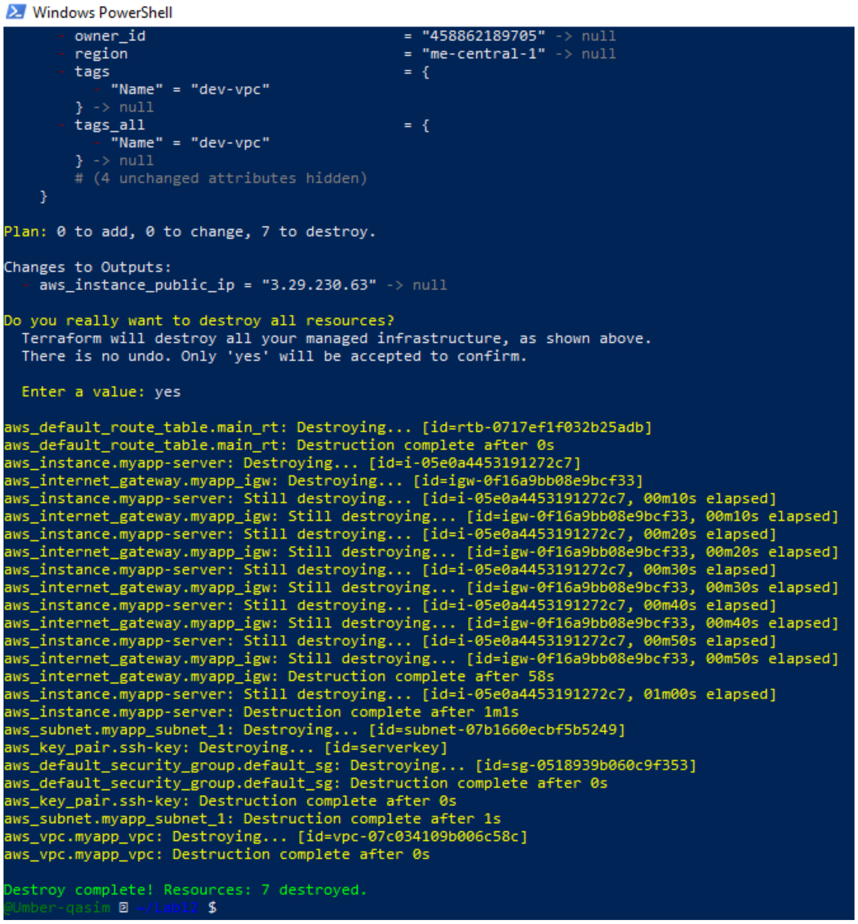


Welcome to nginx!

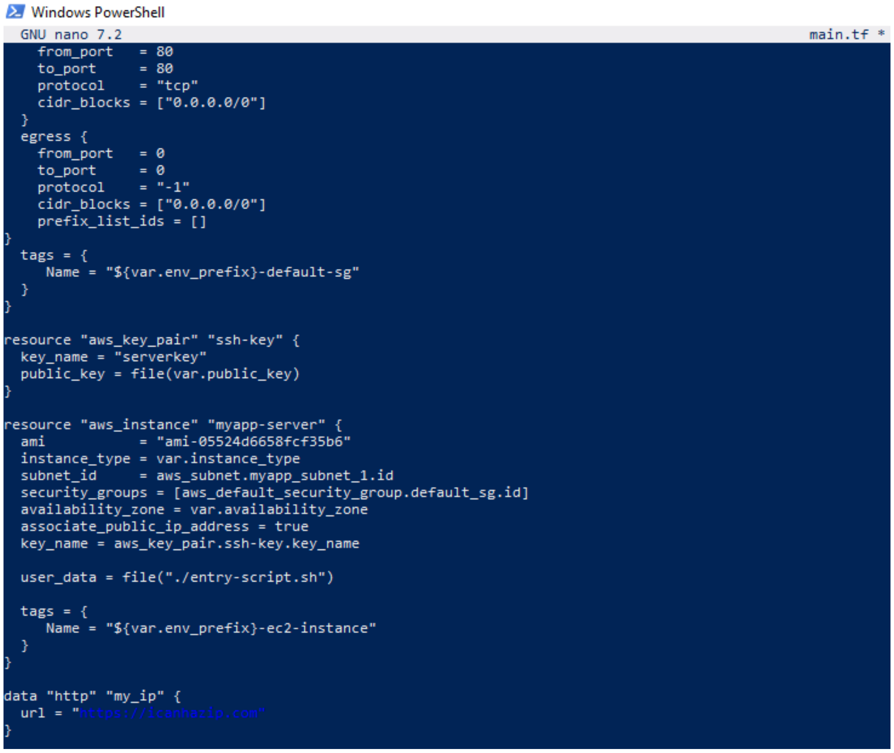
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

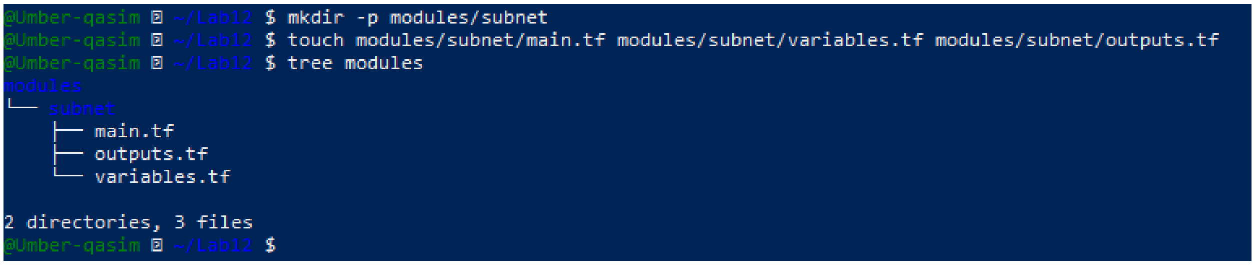


```
      - owner_id                                 = "458862189705" -> null
      - region                                   = "me-central-1" -> null
      - tags                                     = {
          - "Name" = "dev-vpc"
        } -> null
      - tags_all                                 = {
          - "Name" = "dev-vpc"
        } -> null
        # (4 unchanged attributes hidden)
    }

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

Changes to Outputs:
  - aws_instance_public_ip = "3.29.230.63" -> 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

aws_default_route_table.main_rt: Destroying... [id=rtb-0717ef1f032b25adb]
aws_default_route_table.main_rt: Destruction complete after 0s
aws_instance.myapp-server: Destroying... [id=i-05e0a4453191272c7]
aws_internet_gateway.myapp_igw: Destroying... [id=igw-0f16a9bb08e9bcf33]
aws_instance.myapp-server: Still destroying... [id=i-05e0a4453191272c7, 00m10s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f16a9bb08e9bcf33, 00m10s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-05e0a4453191272c7, 00m20s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f16a9bb08e9bcf33, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-05e0a4453191272c7, 00m30s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f16a9bb08e9bcf33, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-05e0a4453191272c7, 00m40s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f16a9bb08e9bcf33, 00m40s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-05e0a4453191272c7, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0f16a9bb08e9bcf33, 00m50s elapsed]
aws_internet_gateway.myapp_igw: Destruction complete after 58s
aws_instance.myapp-server: Still destroying... [id=i-05e0a4453191272c7, 01m00s elapsed]
aws_instance.myapp-server: Destruction complete after 1m1s
aws_subnet.myapp_subnet_1: Destroying... [id=subnet-07b1660ecbf5b5249]
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-0518939b060c9f353]
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-07c034109b006c58c]
aws_vpc.myapp_vpc: Destruction complete after 0s

Destroy complete! Resources: 7 destroyed.
@Umber-qasim ? ~/Lab12 $
```

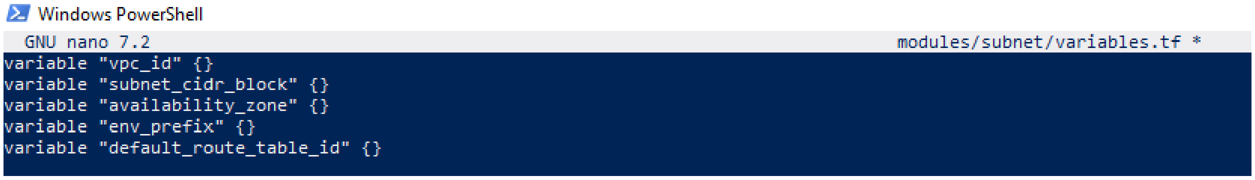


```
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
    prefix_list_ids = []
}
  tags = {
     Name = "${var.env_prefix}-default-sg"
  }
}

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

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

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

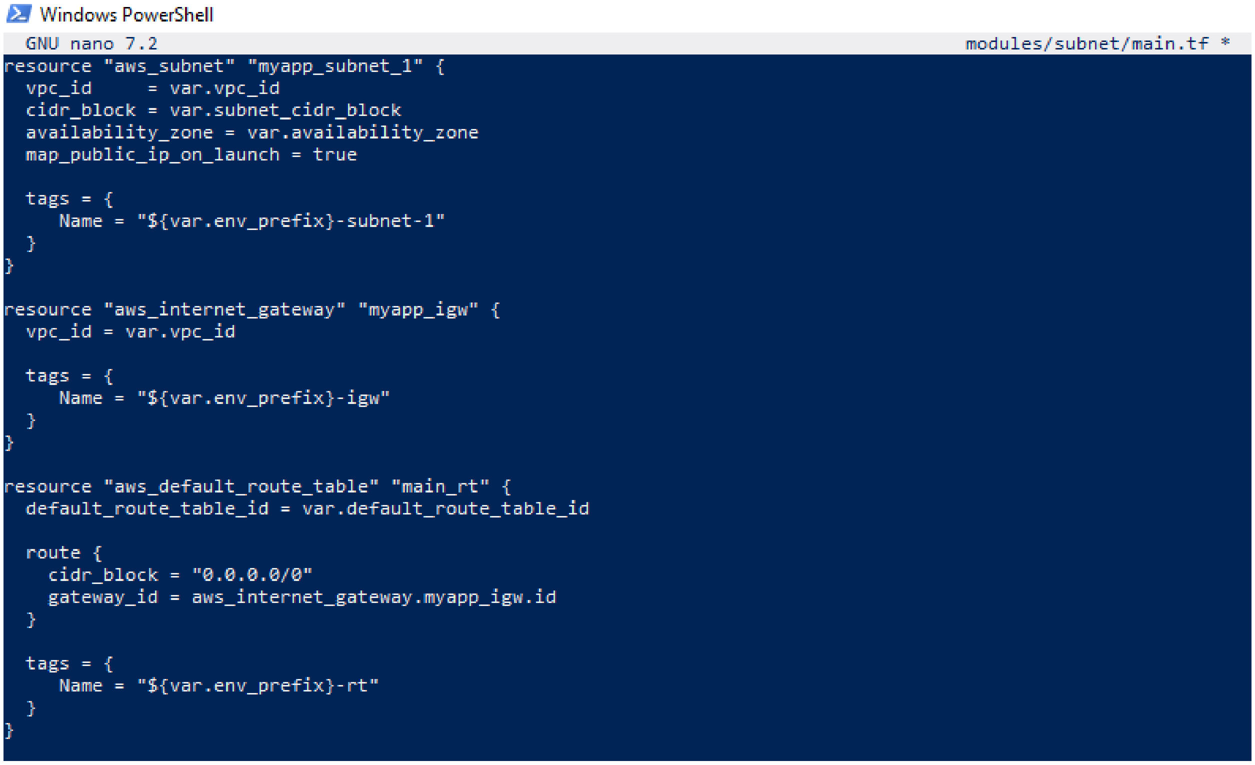
Task#04: Create Terraform modules (subnet module)



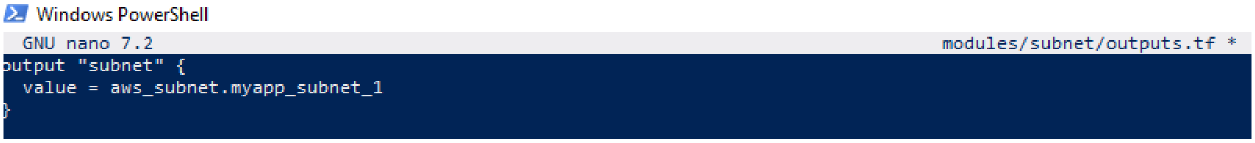
```
@Umber-qasim ⍰ ~/Lab12 $ mkdir -p modules/subnet
@Umber-qasim ⍰ ~/Lab12 $ touch modules/subnet/main.tf modules/subnet/variables.tf modules/subnet/outputs.tf
@Umber-qasim ⍰ ~/Lab12 $ tree modules
modules
└── subnet
    ├── main.tf
    ├── outputs.tf
    └── variables.tf

2 directories, 3 files
@Umber-qasim ⍰ ~/Lab12 $
```



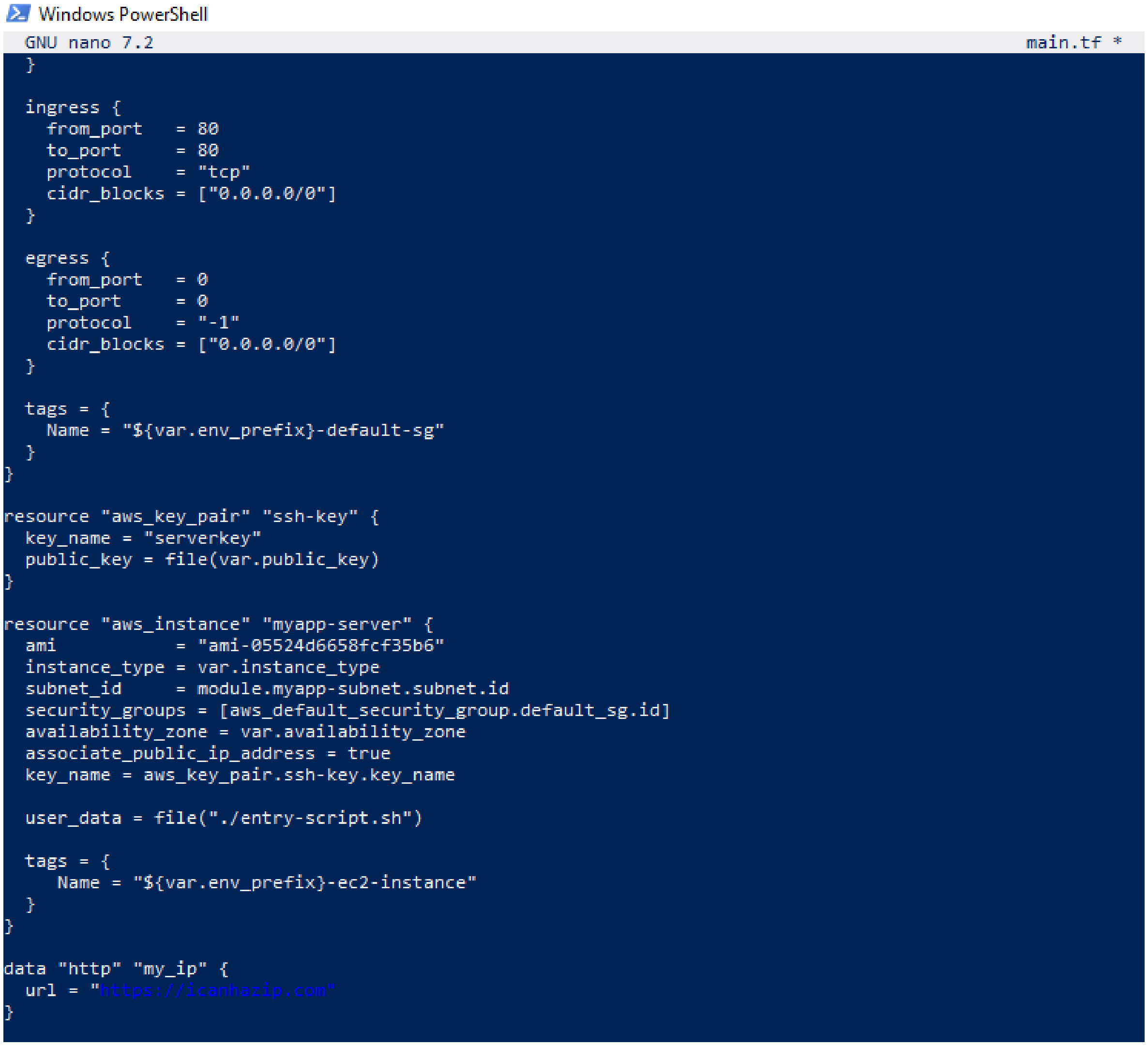
```
  GNU nano 7.2                                                         modules/subnet/variables.tf *
variable "vpc_id" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "default_route_table_id" {}
```



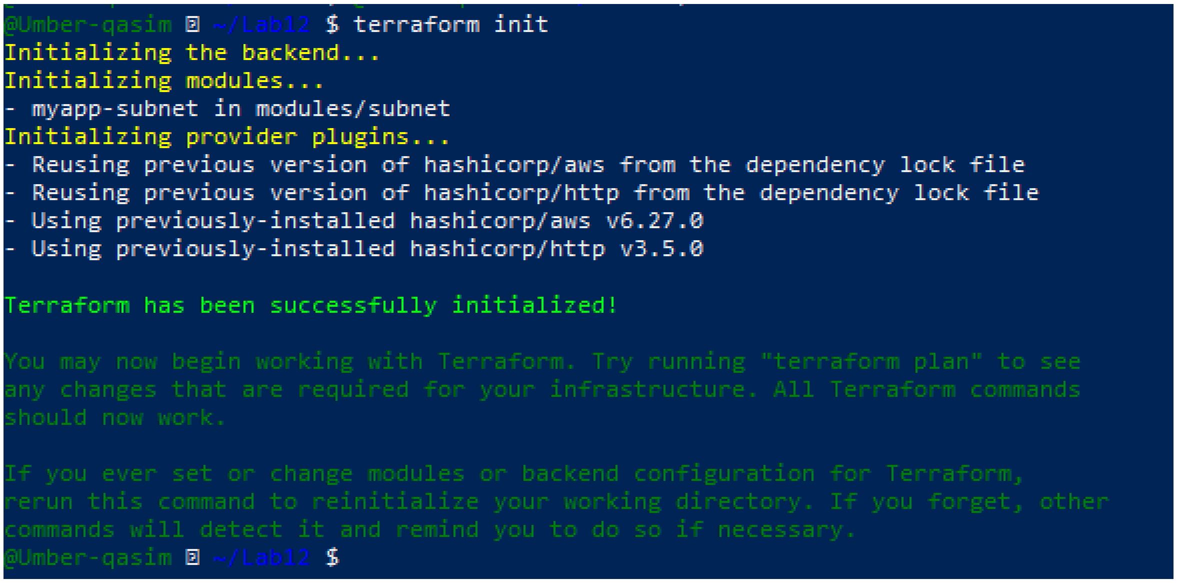
```
  GNU nano 7.2                                                              modules/subnet/main.tf *
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id     = var.vpc_id
  cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  map_public_ip_on_launch = true

  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }
}

resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = var.vpc_id

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

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"
  }
}
```



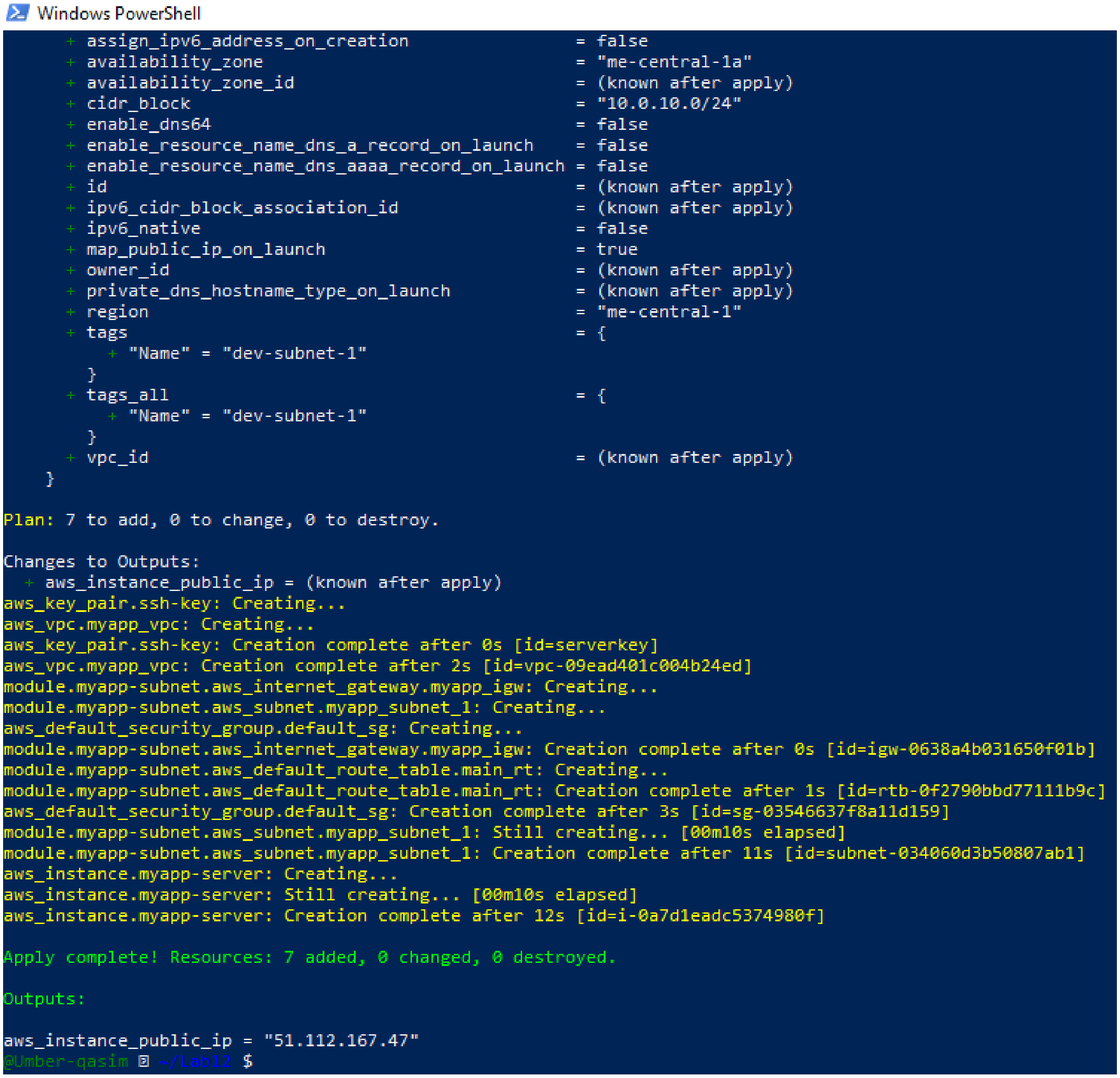
```
  GNU nano 7.2                                                           modules/subnet/outputs.tf *
output "subnet" {
  value = aws_subnet.myapp_subnet_1
}
```



```
  GNU nano 7.2                                                                         main.tf *
  }

  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     = module.myapp-subnet.subnet.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"
  }
}

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

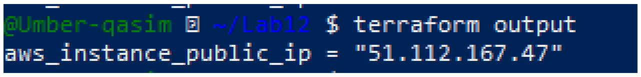


```
@Umber-qasim ⊡ ~/Lab12 $ terraform init
Initializing the backend...
Initializing modules...
- myapp-subnet in modules/subnet
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
- Using previously-installed hashicorp/http v3.5.0

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
@Umber-qasim ⊡ ~/Lab12 $
```

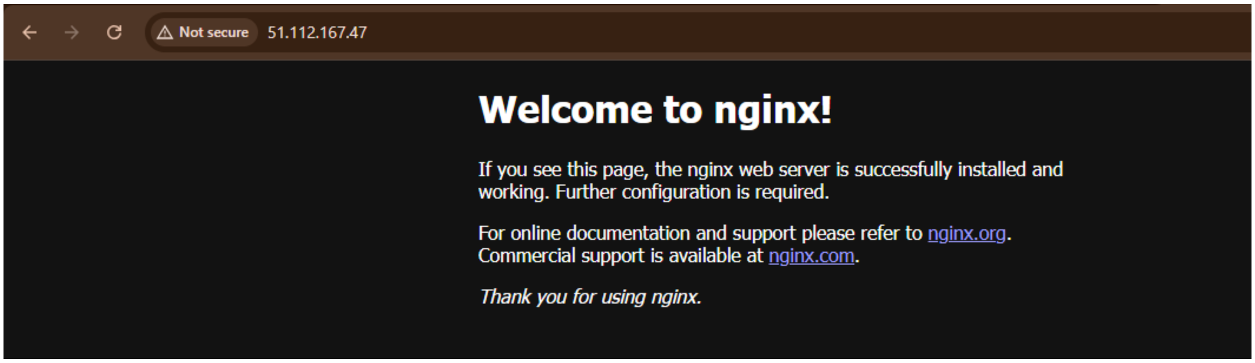


Windows PowerShell

```
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "me-central-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.10.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "me-central-1"
      + tags                                           = {
          + "Name" = "dev-subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "dev-subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }

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

Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
aws_key_pair.ssh-key: Creation complete after 0s [id=serverkey]
aws_vpc.myapp_vpc: Creation complete after 2s [id=vpc-09ead401c004b24ed]
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 0s [id=igw-0638a4b031650f01b]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 1s [id=rtb-0f2790bbd77111b9c]
aws_default_security_group.default_sg: Creation complete after 3s [id=sg-03546637f8a11d159]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-034060d3b50807ab1]
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 12s [id=i-0a7d1eadc5374980f]

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

Outputs:

aws_instance_public_ip = "51.112.167.47"
@Umber-qasim ⊡ ~/Lab12 $
```



```
@Umber-qasim ⊡ ~/Lab12 $ terraform output
aws_instance_public_ip = "51.112.167.47"
```



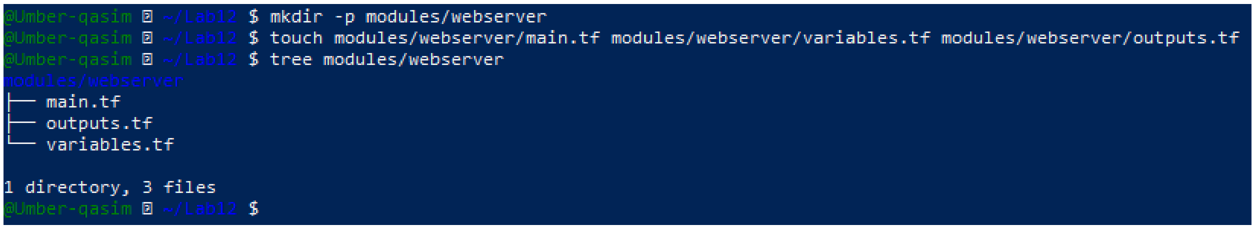
Welcome to nginx!

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

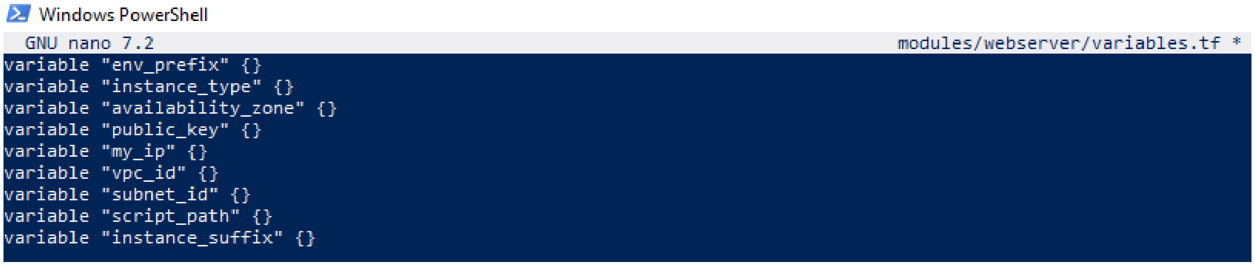
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Task#05: Create webserver module

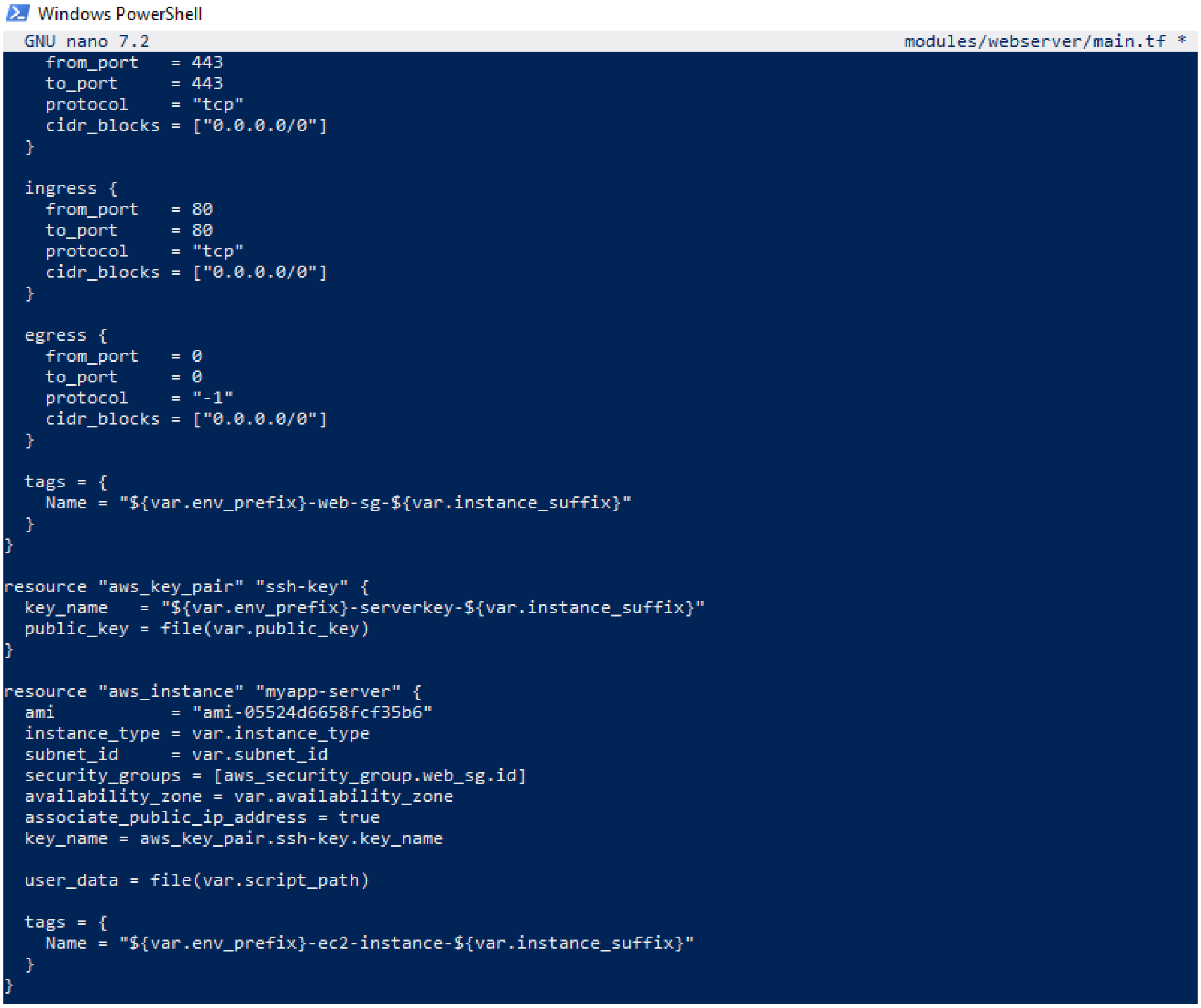


```
@Umber-qasim ⍰ ~/Lab12 $ mkdir -p modules/webserver
@Umber-qasim ⍰ ~/Lab12 $ touch modules/webserver/main.tf modules/webserver/variables.tf modules/webserver/outputs.tf
@Umber-qasim ⍰ ~/Lab12 $ tree modules/webserver
modules/webserver
├── main.tf
├── outputs.tf
└── variables.tf

1 directory, 3 files
@Umber-qasim ⍰ ~/Lab12 $
```



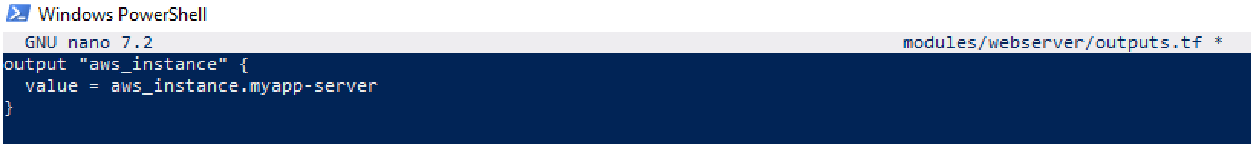
Windows PowerShell

```
  GNU nano 7.2                                                          modules/webserver/variables.tf *
variable "env_prefix" {}
variable "instance_type" {}
variable "availability_zone" {}
variable "public_key" {}
variable "my_ip" {}
variable "vpc_id" {}
variable "subnet_id" {}
variable "script_path" {}
variable "instance_suffix" {}
```

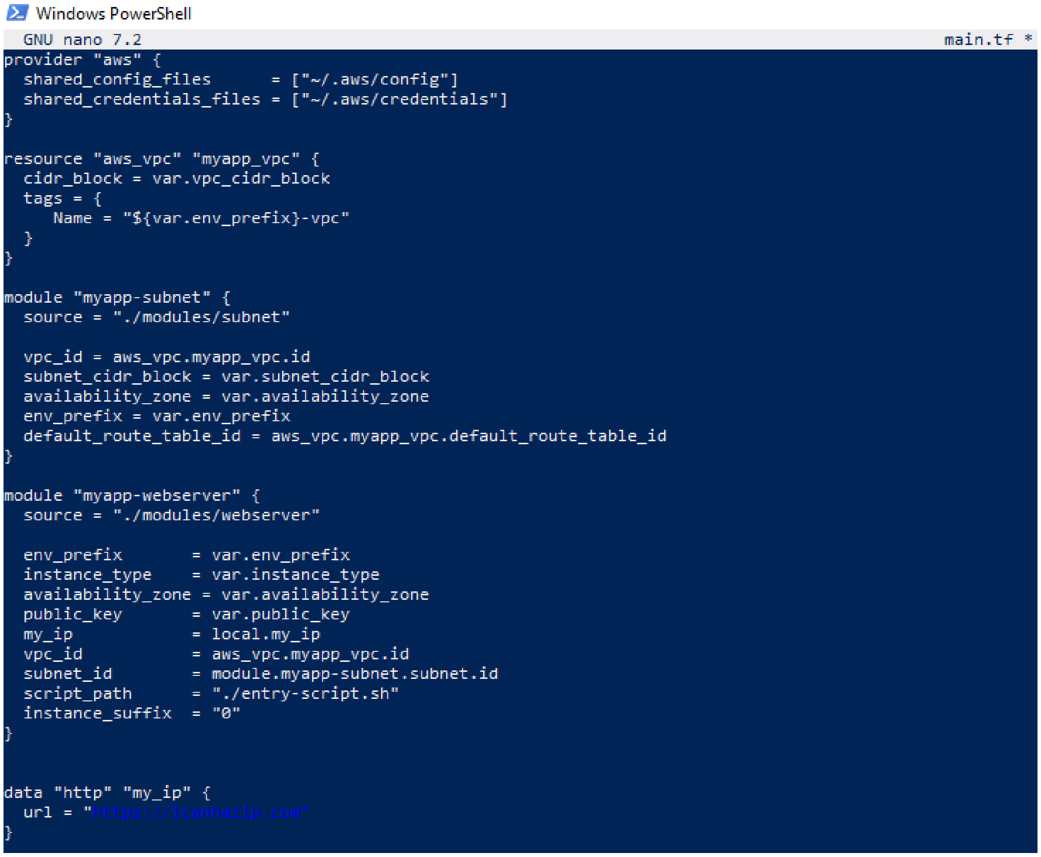


Windows PowerShell

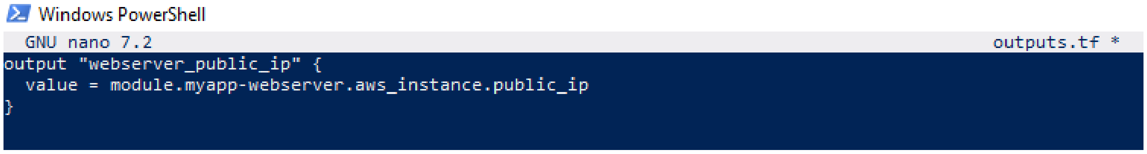
```
  GNU nano 7.2                                                               modules/webserver/main.tf *
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  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"]
  }

  tags = {
    Name = "${var.env_prefix}-web-sg-${var.instance_suffix}"
  }
}

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

resource "aws_instance" "myapp-server" {
  ami           = "ami-05524d6658fcf35b6"
  instance_type = var.instance_type
  subnet_id     = var.subnet_id
  security_groups = [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}"
  }
}
```



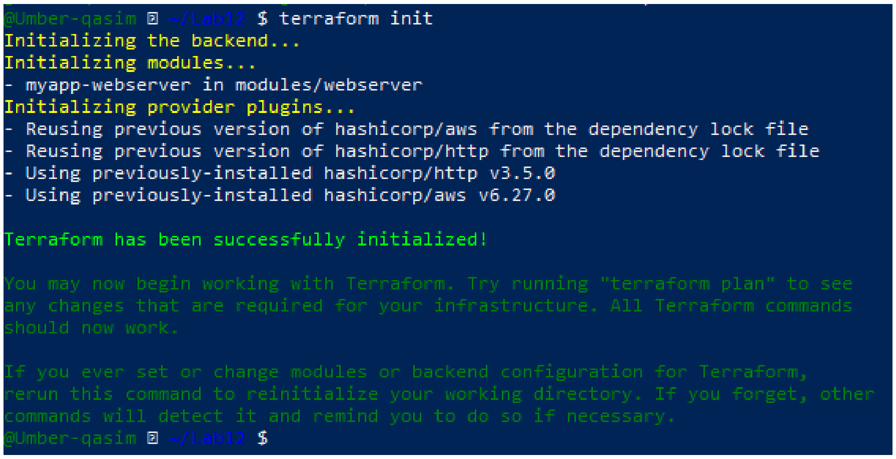
```
  GNU nano 7.2                                                    modules/webserver/outputs.tf *
output "aws_instance" {
  value = aws_instance.myapp-server
}
```



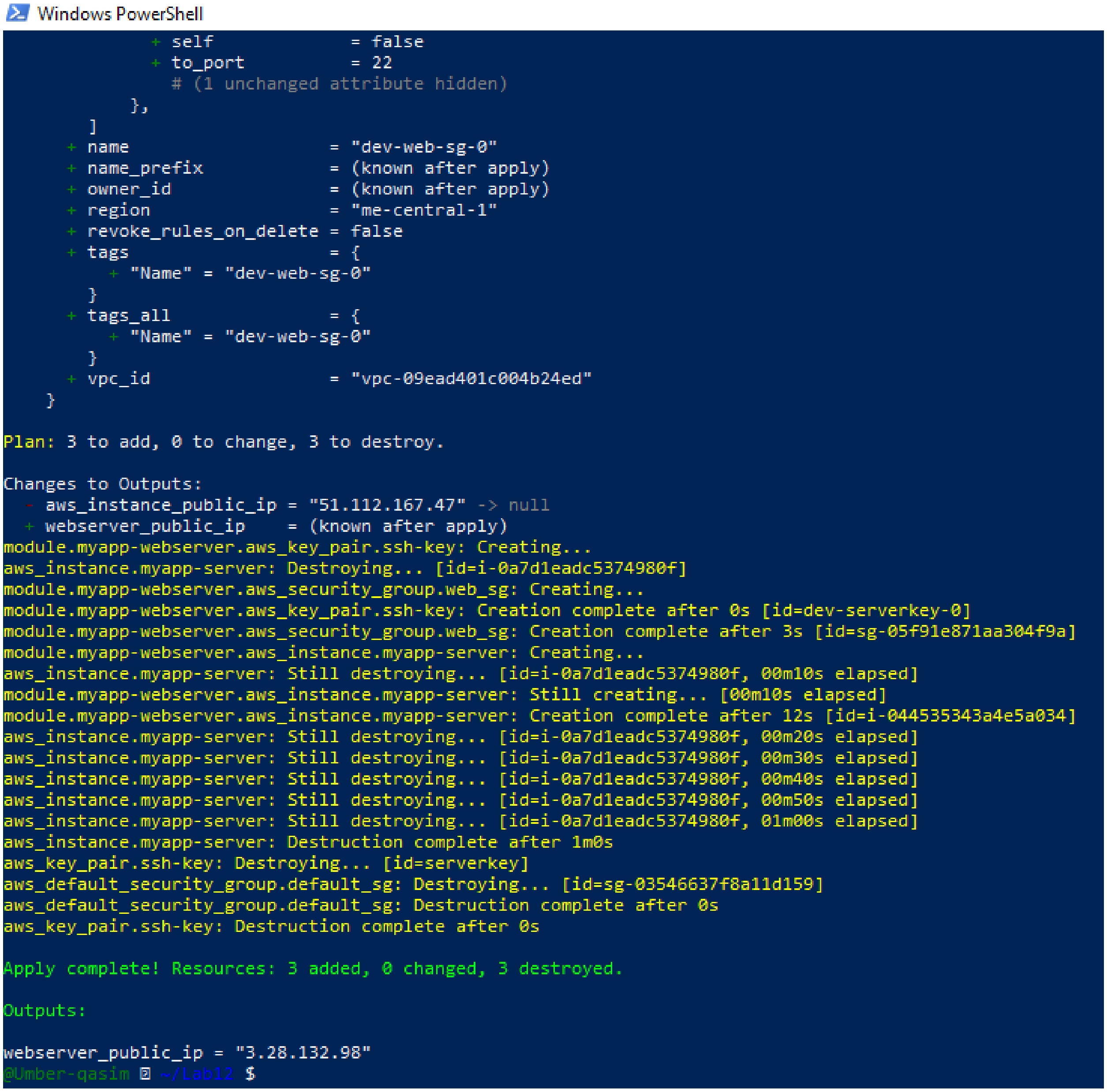
```
  GNU nano 7.2                                                                    main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

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

module "myapp-subnet" {
  source = "./modules/subnet"

  vpc_id = aws_vpc.myapp_vpc.id
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}

module "myapp-webserver" {
  source = "./modules/webserver"

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


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



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

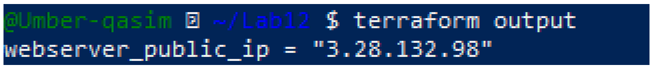


```
@Umber-qasim ➜ ~/Lab12 $ terraform init
Initializing the backend...
Initializing modules...
- myapp-webserver in modules/webserver
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/http v3.5.0
- Using previously-installed hashicorp/aws v6.27.0

Terraform has been successfully initialized!

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.
@Umber-qasim ➜ ~/Lab12 $
```

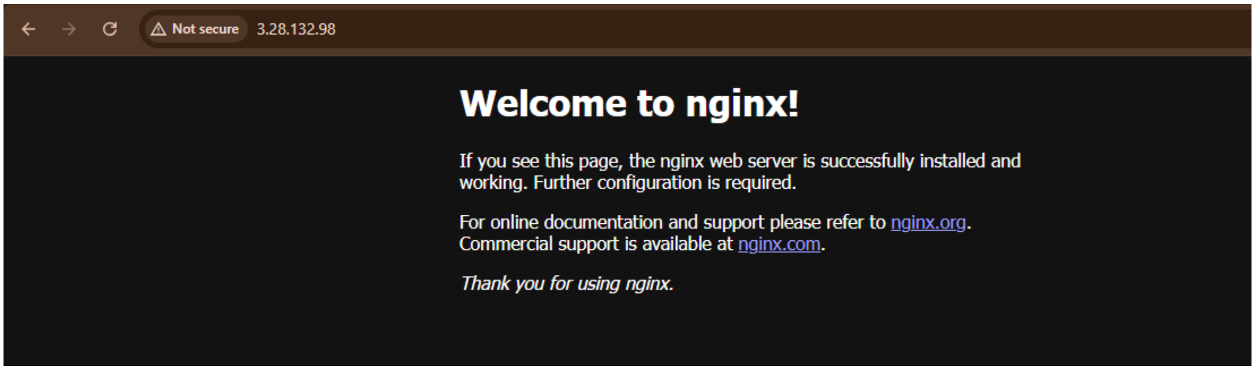


Windows PowerShell

```
               + self             = false
               + to_port          = 22
                 # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "dev-web-sg-0"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "me-central-1"
      + revoke_rules_on_delete = false
      + tags                   = {
          + "Name" = "dev-web-sg-0"
        }
      + tags_all               = {
          + "Name" = "dev-web-sg-0"
        }
      + vpc_id                 = "vpc-09ead401c004b24ed"
    }

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

Changes to Outputs:
  - aws_instance_public_ip = "51.112.167.47" -> null
  + webserver_public_ip    = (known after apply)
module.myapp-webserver.aws_key_pair.ssh-key: Creating...
aws_instance.myapp-server: Destroying... [id=i-0a7d1eadc5374980f]
module.myapp-webserver.aws_security_group.web_sg: Creating...
module.myapp-webserver.aws_key_pair.ssh-key: Creation complete after 0s [id=dev-serverkey-0]
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-05f91e871aa304f9a]
module.myapp-webserver.aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still destroying... [id=i-0a7d1eadc5374980f, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 12s [id=i-044535343a4e5a034]
aws_instance.myapp-server: Still destroying... [id=i-0a7d1eadc5374980f, 00m20s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0a7d1eadc5374980f, 00m30s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0a7d1eadc5374980f, 00m40s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0a7d1eadc5374980f, 00m50s elapsed]
aws_instance.myapp-server: Still destroying... [id=i-0a7d1eadc5374980f, 01m00s elapsed]
aws_instance.myapp-server: Destruction complete after 1m0s
aws_key_pair.ssh-key: Destroying... [id=serverkey]
aws_default_security_group.default_sg: Destroying... [id=sg-03546637f8a11d159]
aws_default_security_group.default_sg: Destruction complete after 0s
aws_key_pair.ssh-key: Destruction complete after 0s

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

Outputs:

webserver_public_ip = "3.28.132.98"
@Umber-qasim ⊠ ~/Lab12 $
```



```
@Umber-qasim ⊠ ~/Lab12 $ terraform output
webserver_public_ip = "3.28.132.98"
```

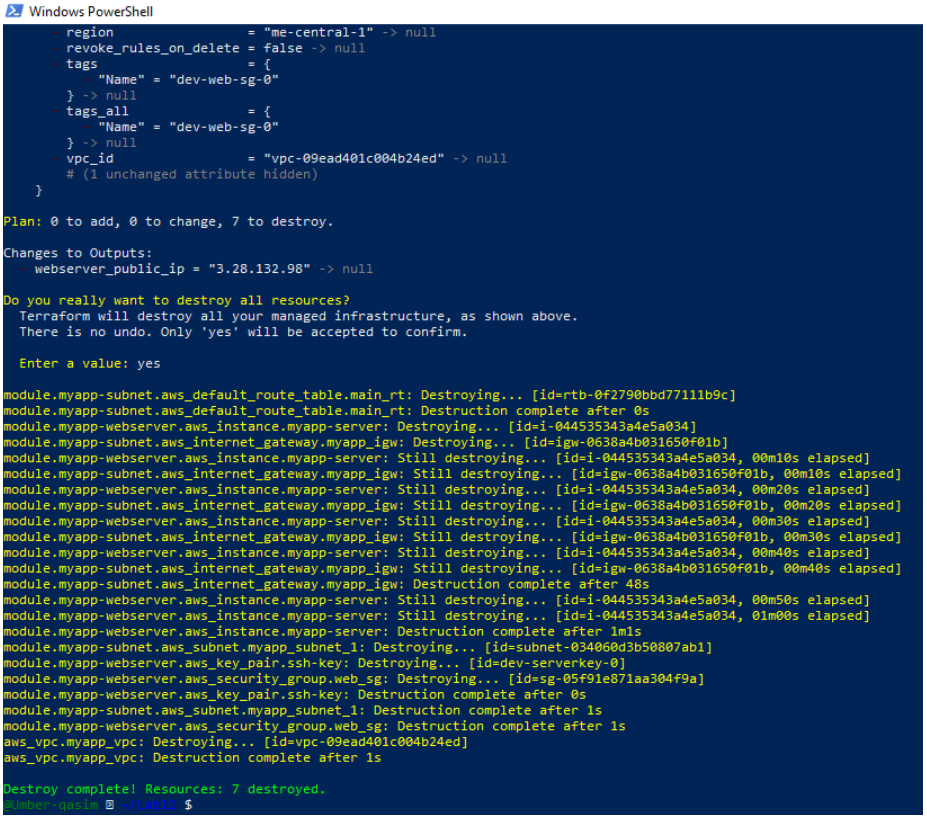


Welcome to nginx!

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

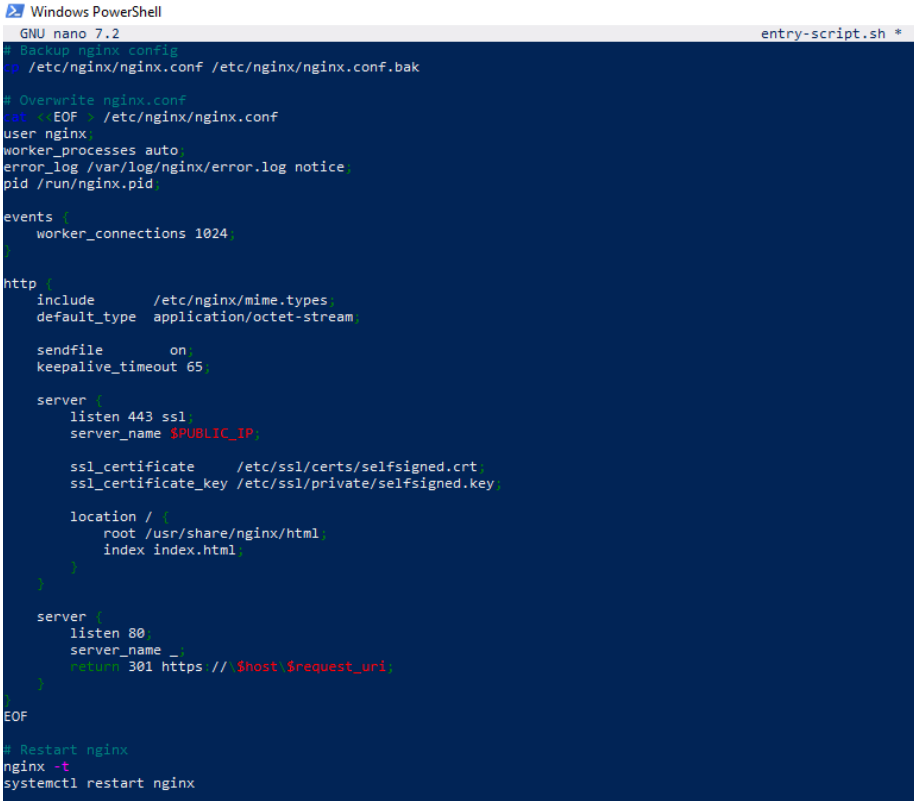
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

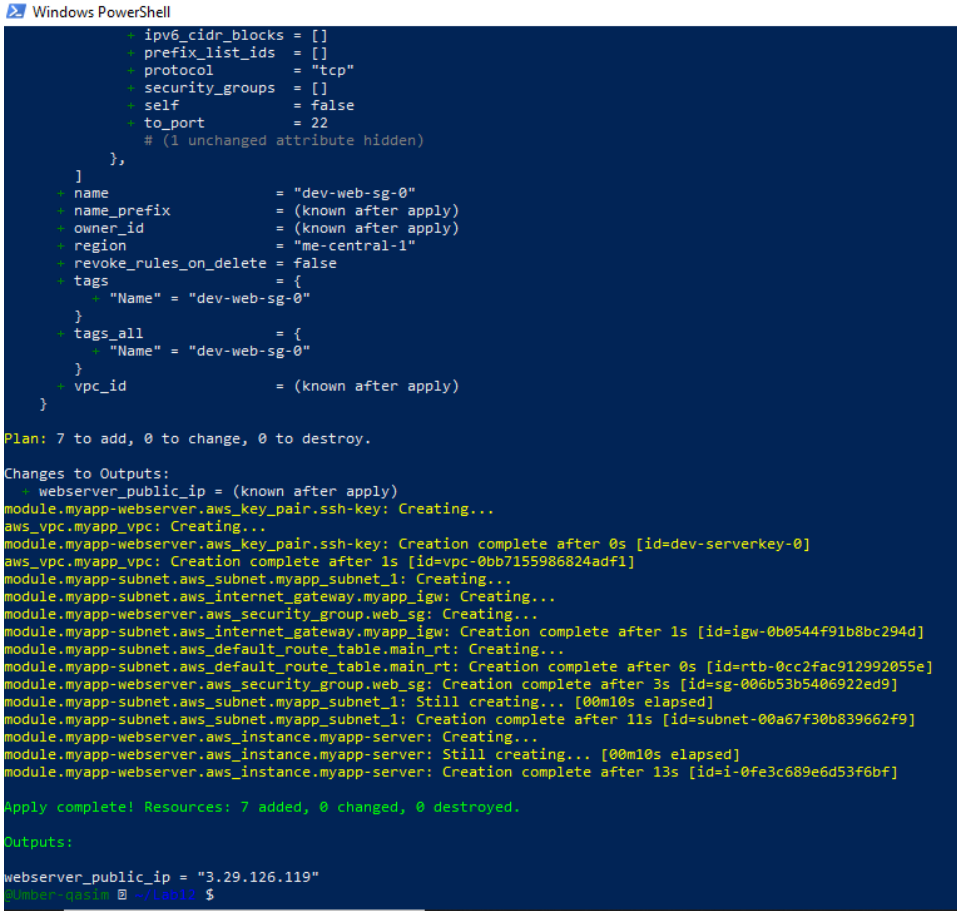


```
      - region                 = "me-central-1" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {
          - "Name" = "dev-web-sg-0"
        } -> null
      - tags_all               = {
          - "Name" = "dev-web-sg-0"
        } -> null
      - vpc_id                 = "vpc-09ead401c004b24ed" -> null
        # (1 unchanged attribute hidden)
    }

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

Changes to Outputs:
  - webserver_public_ip = "3.28.132.98" -> 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

module.myapp-subnet.aws_default_route_table.main_rt: Destroying... [id=rtb-0f2790bbd77111b9c]
module.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp-webserver.aws_instance.myapp-server: Destroying... [id=i-044535343a4e5a034]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-0638a4b031650f01b]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-044535343a4e5a034, 00m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0638a4b031650f01b, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-044535343a4e5a034, 00m20s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0638a4b031650f01b, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-044535343a4e5a034, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0638a4b031650f01b, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-044535343a4e5a034, 00m40s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0638a4b031650f01b, 00m40s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 48s
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-044535343a4e5a034, 00m50s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-044535343a4e5a034, 01m00s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 1m1s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-034060d3b50807ab1]
module.myapp-webserver.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-05f91e871aa304f9a]
module.myapp-webserver.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-09ead401c004b24ed]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
@Umber-qasim ⊡ ~/Lab12 $
```

<u>Task#06:</u> Configure HTTPS with self-signed certificates

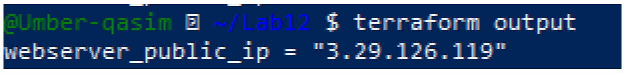


```
# Backup nginx config
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak

# Overwrite nginx.conf
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 {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    sendfile        on;
    keepalive_timeout 65;

    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;
        }
    }

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

# Restart nginx
nginx -t
systemctl restart nginx
```

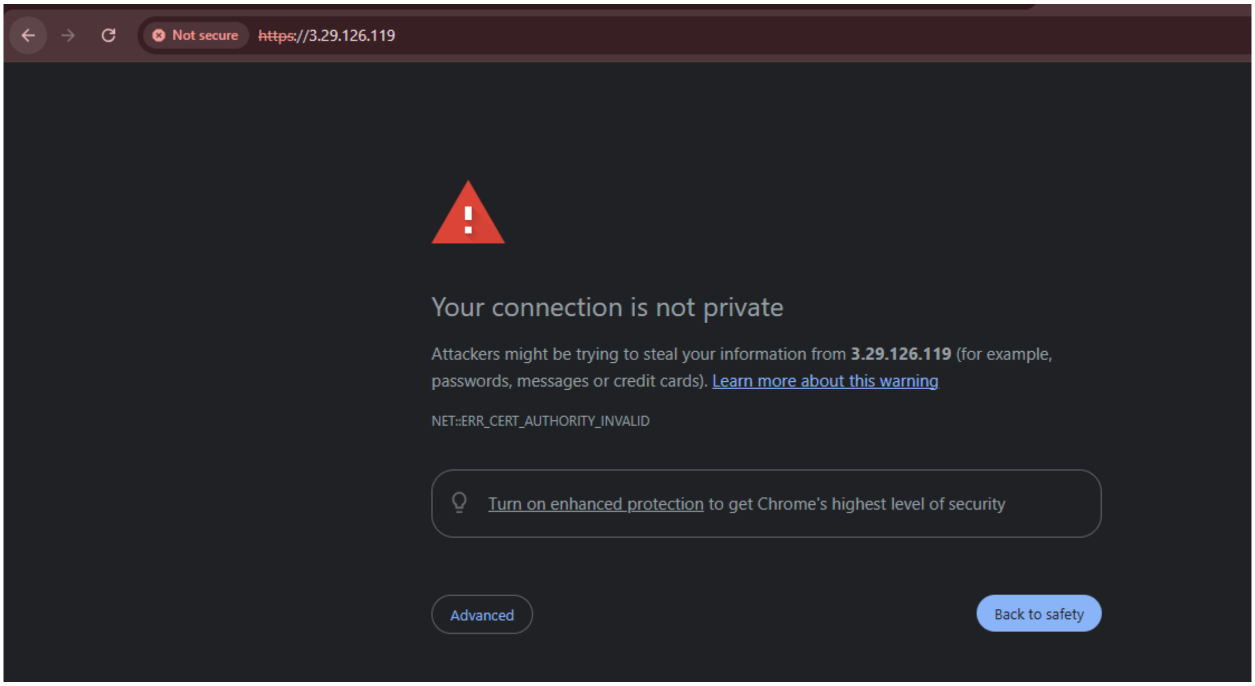


Windows PowerShell

```
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
              + protocol         = "tcp"
              + security_groups  = []
              + self             = false
              + to_port          = 22
                # (1 unchanged attribute hidden)
            },
        ]
      + name                   = "dev-web-sg-0"
      + name_prefix            = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "me-central-1"
      + revoke_rules_on_delete = false
      + tags                   = {
          + "Name" = "dev-web-sg-0"
        }
      + tags_all               = {
          + "Name" = "dev-web-sg-0"
        }
      + vpc_id                 = (known after apply)
    }

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

Changes to Outputs:
  + webserver_public_ip = (known after apply)
module.myapp-webserver.aws_key_pair.ssh-key: Creating...
aws_vpc.myapp_vpc: Creating...
module.myapp-webserver.aws_key_pair.ssh-key: Creation complete after 0s [id=dev-serverkey-0]
aws_vpc.myapp_vpc: Creation complete after 1s [id=vpc-0bb7155986824adf1]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creating...
module.myapp-webserver.aws_security_group.web_sg: Creating...
module.myapp-subnet.aws_internet_gateway.myapp_igw: Creation complete after 1s [id=igw-0b0544f91b8bc294d]
module.myapp-subnet.aws_default_route_table.main_rt: Creating...
module.myapp-subnet.aws_default_route_table.main_rt: Creation complete after 0s [id=rtb-0cc2fac912992055e]
module.myapp-webserver.aws_security_group.web_sg: Creation complete after 3s [id=sg-006b53b5406922ed9]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Still creating... [00m10s elapsed]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Creation complete after 11s [id=subnet-00a67f30b839662f9]
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 13s [id=i-0fe3c689e6d53f6bf]

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

Outputs:

webserver_public_ip = "3.29.126.119"
@Umber-qasim ⊡ ~/Lab12 $
```



```
@Umber-qasim ⊡ ~/Lab12 $ terraform output
webserver_public_ip = "3.29.126.119"
```

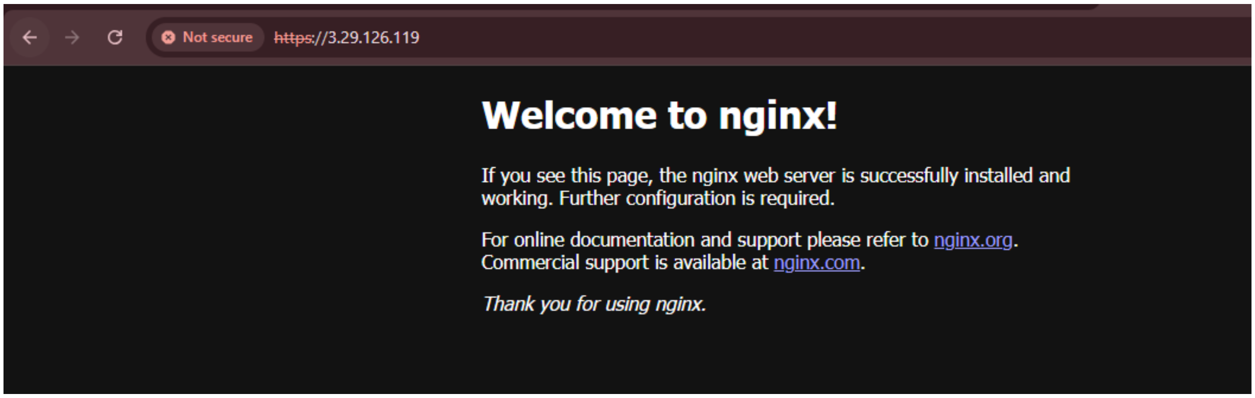


Your connection is not private

Attackers might be trying to steal your information from **3.29.126.119** (for example, passwords, messages or credit cards). Learn more about this warning

NET::ERR_CERT_AUTHORITY_INVALID

Turn on enhanced protection to get Chrome's highest level of security

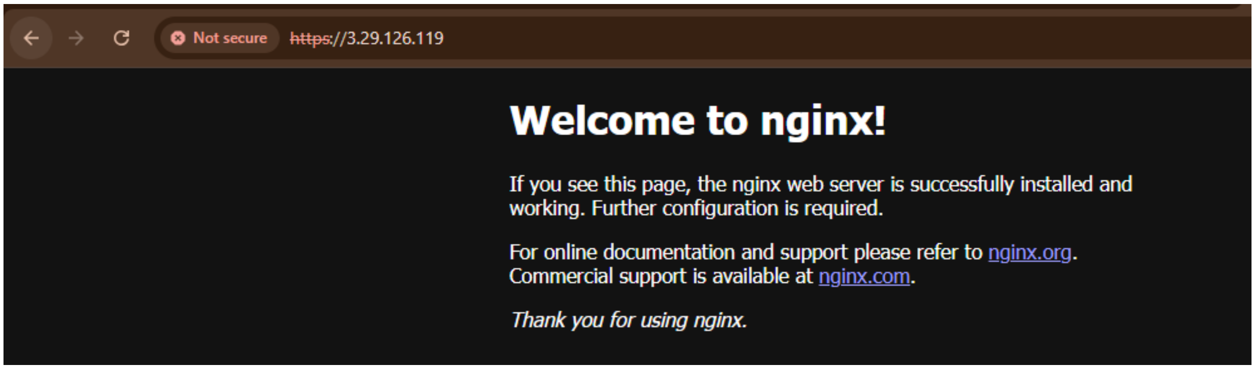


Welcome to nginx!

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.



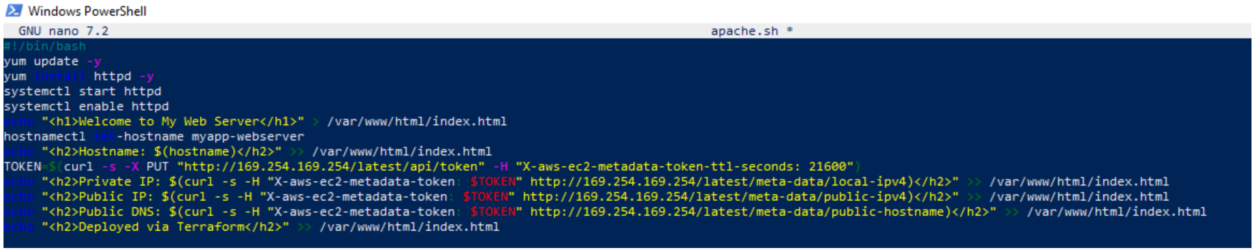
Welcome to nginx!

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

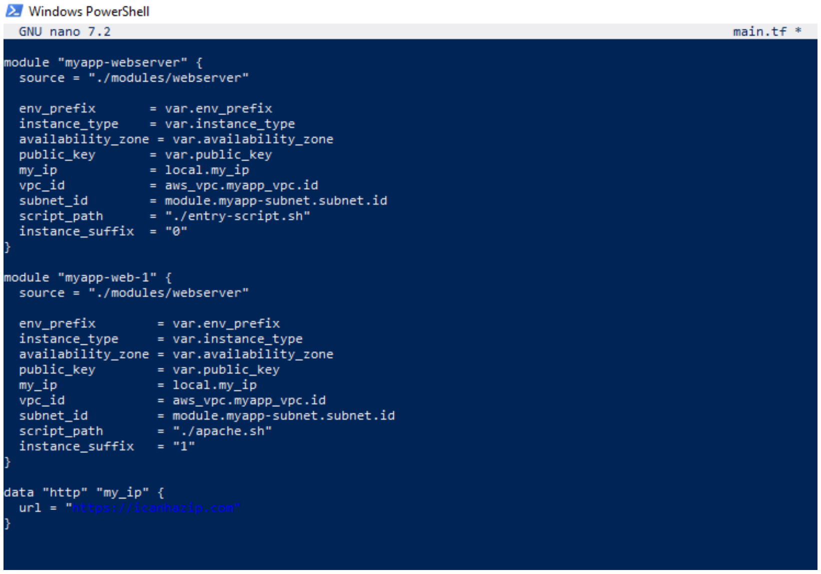
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

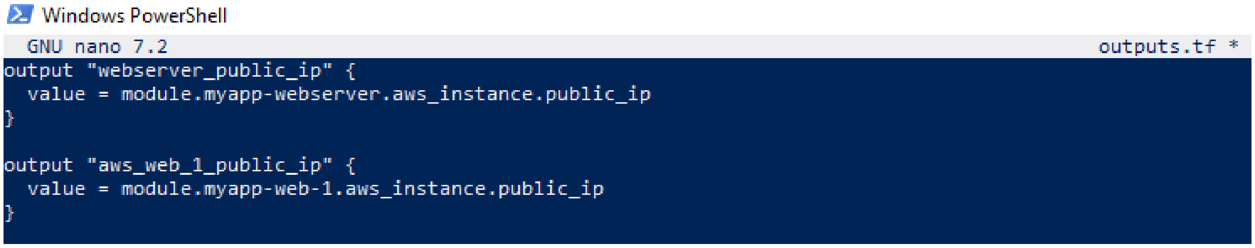
Task#07: Configure Nginx as reverse proxy



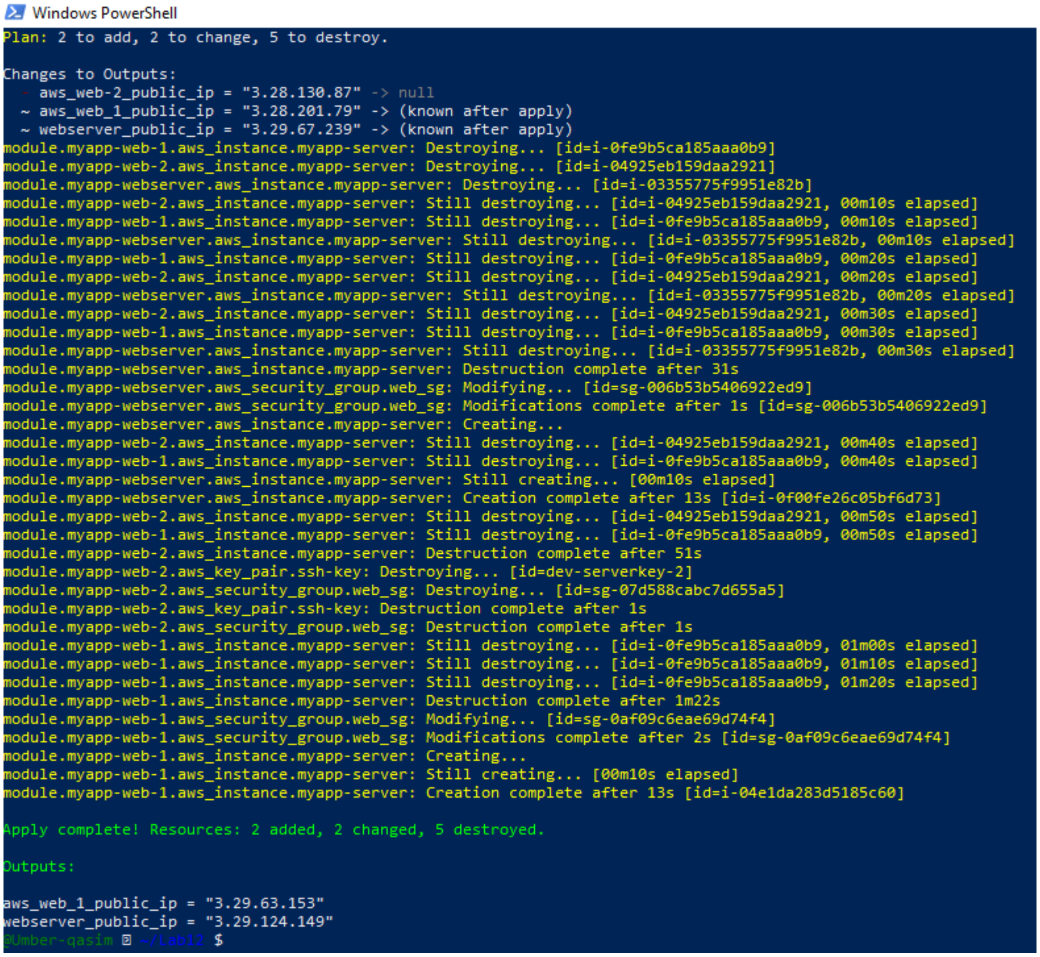
```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "<h1>Welcome to My Web Server</h1>" > /var/www/html/index.html
hostnamectl set-hostname myapp-webserver
echo "<h2>Hostname: $(hostname)</h2>" >> /var/www/html/index.html
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
echo "<h2>Private IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/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-data/public-ipv4)</h2>" >> /var/www/html/index.html
echo "<h2>Public DNS: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-hostname)</h2>" >> /var/www/html/index.html
echo "<h2>Deployed via Terraform</h2>" >> /var/www/html/index.html
```



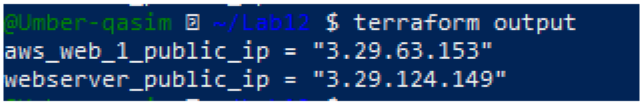
```
module "myapp-webserver" {
  source = "./modules/webserver"

  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"
}

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"
}

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



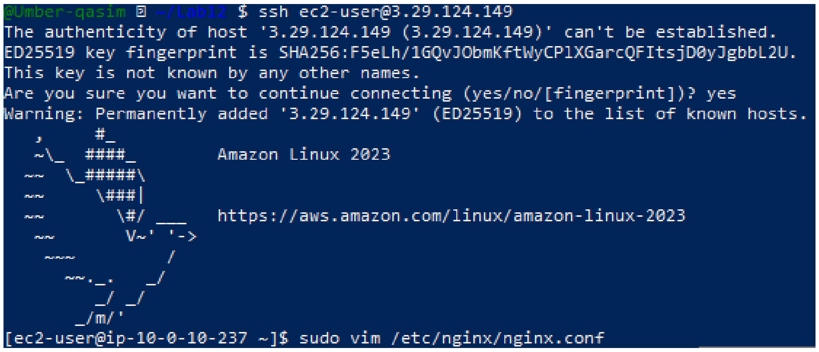
```
  GNU nano 7.2                                                                                   outputs.tf *
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}

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



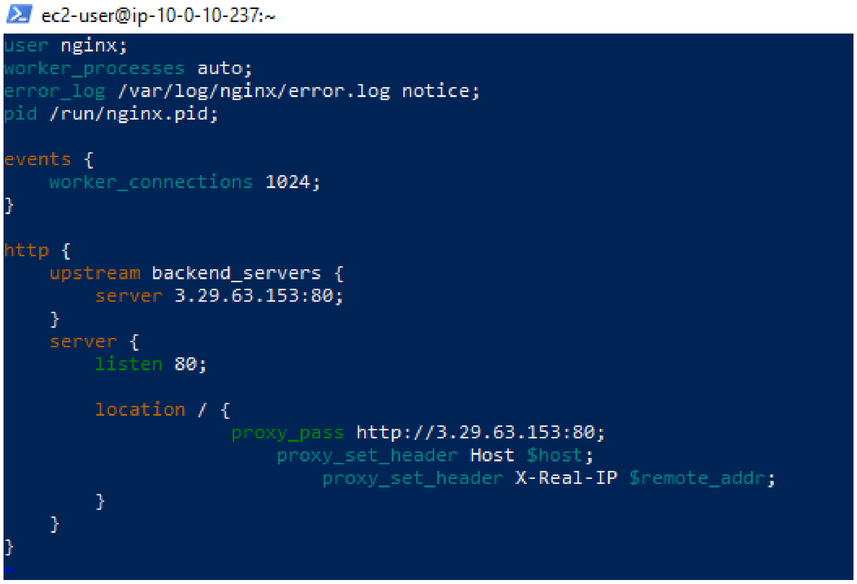
```
Plan: 2 to add, 2 to change, 5 to destroy.

Changes to Outputs:
  - aws_web-2_public_ip = "3.28.130.87" -> null
  ~ aws_web_1_public_ip = "3.28.201.79" -> (known after apply)
  ~ webserver_public_ip = "3.29.67.239" -> (known after apply)
module.myapp-web-1.aws_instance.myapp-server: Destroying... [id=i-0fe9b5ca185aaa0b9]
module.myapp-web-2.aws_instance.myapp-server: Destroying... [id=i-04925eb159daa2921]
module.myapp-webserver.aws_instance.myapp-server: Destroying... [id=i-03355775f9951e82b]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-04925eb159daa2921, 00m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-03355775f9951e82b, 00m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 00m20s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-04925eb159daa2921, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-03355775f9951e82b, 00m20s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-04925eb159daa2921, 00m30s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-03355775f9951e82b, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 31s
module.myapp-webserver.aws_security_group.web_sg: Modifying... [id=sg-006b53b5406922ed9]
module.myapp-webserver.aws_security_group.web_sg: Modifications complete after 1s [id=sg-006b53b5406922ed9]
module.myapp-webserver.aws_instance.myapp-server: Creating...
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-04925eb159daa2921, 00m40s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Creation complete after 13s [id=i-0f00fe26c05bf6d73]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-04925eb159daa2921, 00m50s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 00m50s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Destruction complete after 51s
module.myapp-web-2.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-2]
module.myapp-web-2.aws_security_group.web_sg: Destroying... [id=sg-07d588cabc7d655a5]
module.myapp-web-2.aws_key_pair.ssh-key: Destruction complete after 1s
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-0fe9b5ca185aaa0b9, 01m00s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 01m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-0fe9b5ca185aaa0b9, 01m20s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Destruction complete after 1m22s
module.myapp-web-1.aws_security_group.web_sg: Modifying... [id=sg-0af09c6eae69d74f4]
module.myapp-web-1.aws_security_group.web_sg: Modifications complete after 2s [id=sg-0af09c6eae69d74f4]
module.myapp-web-1.aws_instance.myapp-server: Creating...
module.myapp-web-1.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Creation complete after 13s [id=i-04e1da283d5185c60]

Apply complete! Resources: 2 added, 2 changed, 5 destroyed.

Outputs:

aws_web_1_public_ip = "3.29.63.153"
webserver_public_ip = "3.29.124.149"
@Umber-qasim ⊡ ~/Lab12 $
```



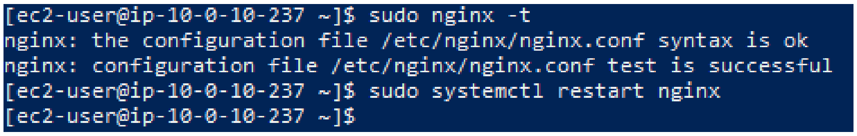
```
@Umber-qasim ⊡ ~/Lab12 $ terraform output
aws_web_1_public_ip = "3.29.63.153"
webserver_public_ip = "3.29.124.149"
```



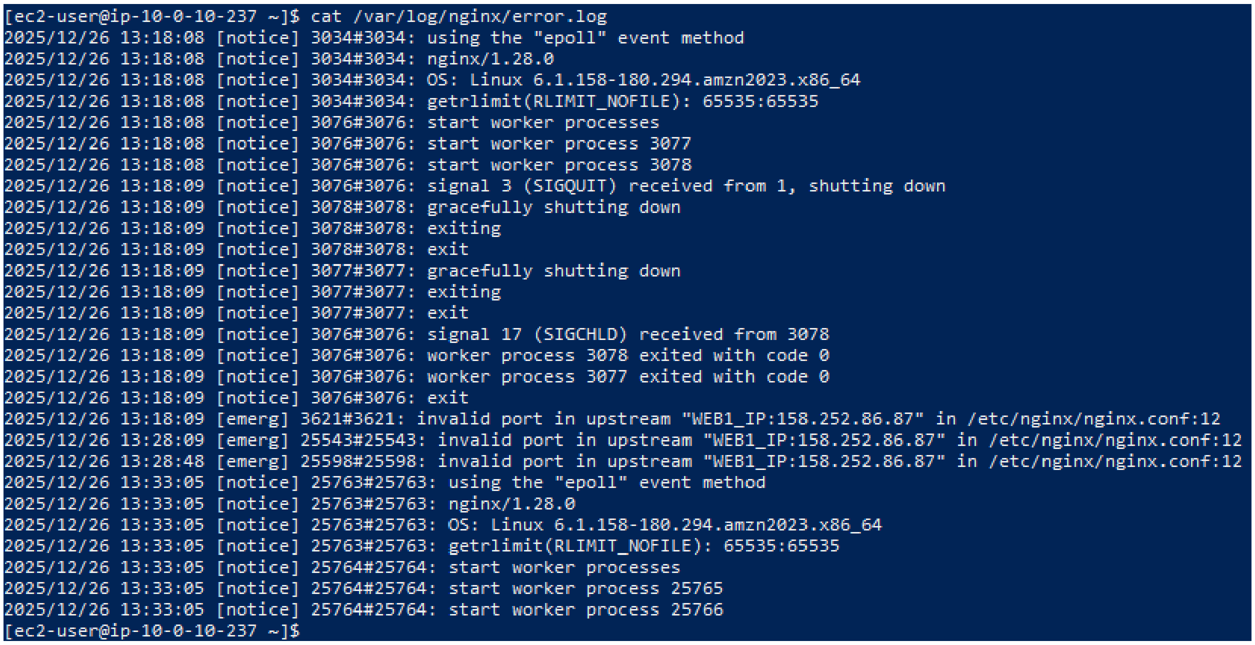
```
@Umber-qasim ⊡ ~/Lab12 $ ssh ec2-user@3.29.124.149
The authenticity of host '3.29.124.149 (3.29.124.149)' can't be established.
ED25519 key fingerprint is SHA256:F5eLh/1GQvJObmKftWyCPlXGarcQFItsjD0yJgbbL2U.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.29.124.149' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-10-237 ~]$ sudo vim /etc/nginx/nginx.conf
```



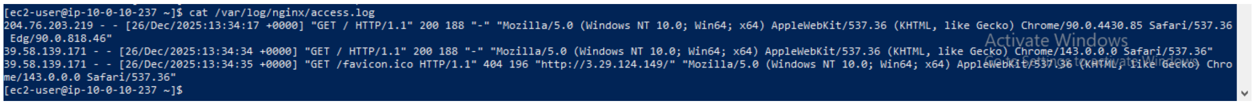
```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    upstream backend_servers {
        server 3.29.63.153:80;
    }
    server {
        listen 80;

        location / {
                    proxy_pass http://3.29.63.153:80;
                        proxy_set_header Host $host;
                            proxy_set_header X-Real-IP $remote_addr;
        }
    }
}
~
```



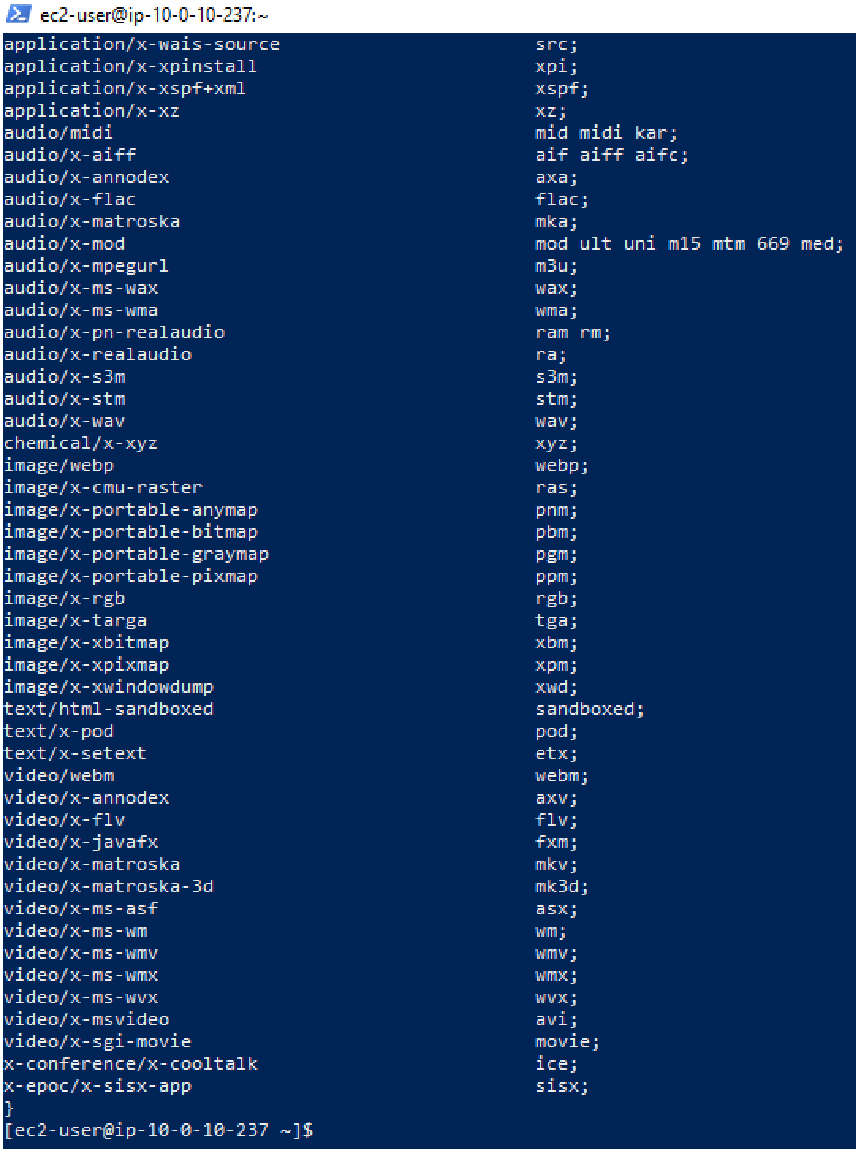
```
[ec2-user@ip-10-0-10-237 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-237 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-237 ~]$
```



```
[ec2-user@ip-10-0-10-237 ~]$ cat /var/log/nginx/error.log
2025/12/26 13:18:08 [notice] 3034#3034: using the "epoll" event method
2025/12/26 13:18:08 [notice] 3034#3034: nginx/1.28.0
2025/12/26 13:18:08 [notice] 3034#3034: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/26 13:18:08 [notice] 3034#3034: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/26 13:18:08 [notice] 3076#3076: start worker processes
2025/12/26 13:18:08 [notice] 3076#3076: start worker process 3077
2025/12/26 13:18:08 [notice] 3076#3076: start worker process 3078
2025/12/26 13:18:09 [notice] 3076#3076: signal 3 (SIGQUIT) received from 1, shutting down
2025/12/26 13:18:09 [notice] 3078#3078: gracefully shutting down
2025/12/26 13:18:09 [notice] 3078#3078: exiting
2025/12/26 13:18:09 [notice] 3078#3078: exit
2025/12/26 13:18:09 [notice] 3077#3077: gracefully shutting down
2025/12/26 13:18:09 [notice] 3077#3077: exiting
2025/12/26 13:18:09 [notice] 3077#3077: exit
2025/12/26 13:18:09 [notice] 3076#3076: signal 17 (SIGCHLD) received from 3078
2025/12/26 13:18:09 [notice] 3076#3076: worker process 3078 exited with code 0
2025/12/26 13:18:09 [notice] 3076#3076: worker process 3077 exited with code 0
2025/12/26 13:18:09 [notice] 3076#3076: exit
2025/12/26 13:18:09 [emerg] 3621#3621: invalid port in upstream "WEB1_IP:158.252.86.87" in /etc/nginx/nginx.conf:12
2025/12/26 13:28:09 [emerg] 25543#25543: invalid port in upstream "WEB1_IP:158.252.86.87" in /etc/nginx/nginx.conf:12
2025/12/26 13:28:48 [emerg] 25598#25598: invalid port in upstream "WEB1_IP:158.252.86.87" in /etc/nginx/nginx.conf:12
2025/12/26 13:33:05 [notice] 25763#25763: using the "epoll" event method
2025/12/26 13:33:05 [notice] 25763#25763: nginx/1.28.0
2025/12/26 13:33:05 [notice] 25763#25763: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/26 13:33:05 [notice] 25763#25763: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/26 13:33:05 [notice] 25764#25764: start worker processes
2025/12/26 13:33:05 [notice] 25764#25764: start worker process 25765
2025/12/26 13:33:05 [notice] 25764#25764: start worker process 25766
[ec2-user@ip-10-0-10-237 ~]$
```

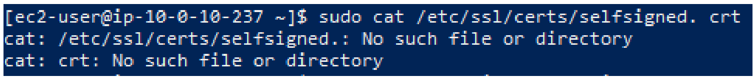


```
[ec2-user@ip-10-0-10-237 ~]$ cat /var/log/nginx/access.log
204.76.203.219 - - [26/Dec/2025:13:34:17 +0000] "GET / HTTP/1.1" 200 188 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36
 Edg/90.0.818.46"
39.58.139.171 - - [26/Dec/2025:13:34:34 +0000] "GET / HTTP/1.1" 200 188 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
39.58.139.171 - - [26/Dec/2025:13:34:35 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://3.29.124.149/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chro
me/143.0.0.0 Safari/537.36"
[ec2-user@ip-10-0-10-237 ~]$
```

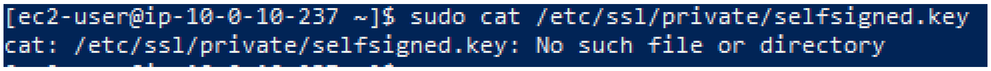


ec2-user@ip-10-0-10-237:~

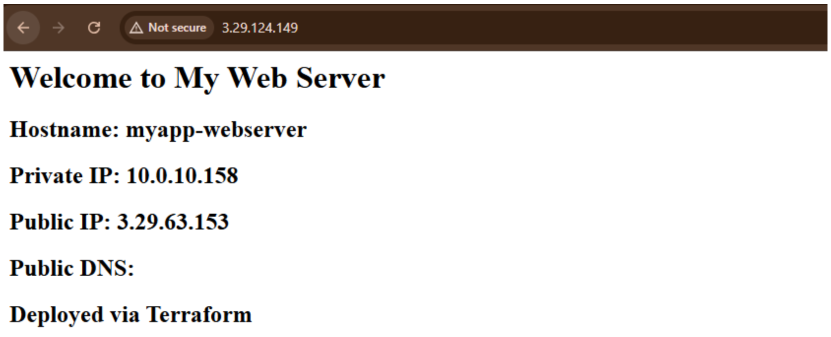
```
application/x-wais-source                        src;
application/x-xpinstall                          xpi;
application/x-xspf+xml                           xspf;
application/x-xz                                 xz;
audio/midi                                       mid midi kar;
audio/x-aiff                                     aif aiff aifc;
audio/x-annodex                                  axa;
audio/x-flac                                     flac;
audio/x-matroska                                 mka;
audio/x-mod                                      mod ult uni m15 mtm 669 med;
audio/x-mpegurl                                  m3u;
audio/x-ms-wax                                   wax;
audio/x-ms-wma                                   wma;
audio/x-pn-realaudio                             ram rm;
audio/x-realaudio                                ra;
audio/x-s3m                                      s3m;
audio/x-stm                                      stm;
audio/x-wav                                      wav;
chemical/x-xyz                                   xyz;
image/webp                                       webp;
image/x-cmu-raster                               ras;
image/x-portable-anymap                          pnm;
image/x-portable-bitmap                          pbm;
image/x-portable-graymap                         pgm;
image/x-portable-pixmap                          ppm;
image/x-rgb                                      rgb;
image/x-targa                                    tga;
image/x-xbitmap                                  xbm;
image/x-xpixmap                                  xpm;
image/x-xwindowdump                              xwd;
text/html-sandboxed                              sandboxed;
text/x-pod                                       pod;
text/x-setext                                    etx;
video/webm                                       webm;
video/x-annodex                                  axv;
video/x-flv                                      flv;
video/x-javafx                                   fxm;
video/x-matroska                                 mkv;
video/x-matroska-3d                              mk3d;
video/x-ms-asf                                   asx;
video/x-ms-wm                                    wm;
video/x-ms-wmv                                   wmv;
video/x-ms-wmx                                   wmx;
video/x-ms-wvx                                   wvx;
video/x-msvideo                                  avi;
video/x-sgi-movie                                movie;
x-conference/x-cooltalk                          ice;
x-epoc/x-sisx-app                                sisx;
}
[ec2-user@ip-10-0-10-237 ~]$
```



```
[ec2-user@ip-10-0-10-237 ~]$ sudo cat /etc/ssl/certs/selfsigned. crt
cat: /etc/ssl/certs/selfsigned.: No such file or directory
cat: crt: No such file or directory
```



```
[ec2-user@ip-10-0-10-237 ~]$ sudo cat /etc/ssl/private/selfsigned.key
cat: /etc/ssl/private/selfsigned.key: No such file or directory
```



Welcome to My Web Server

Hostname: myapp-webserver

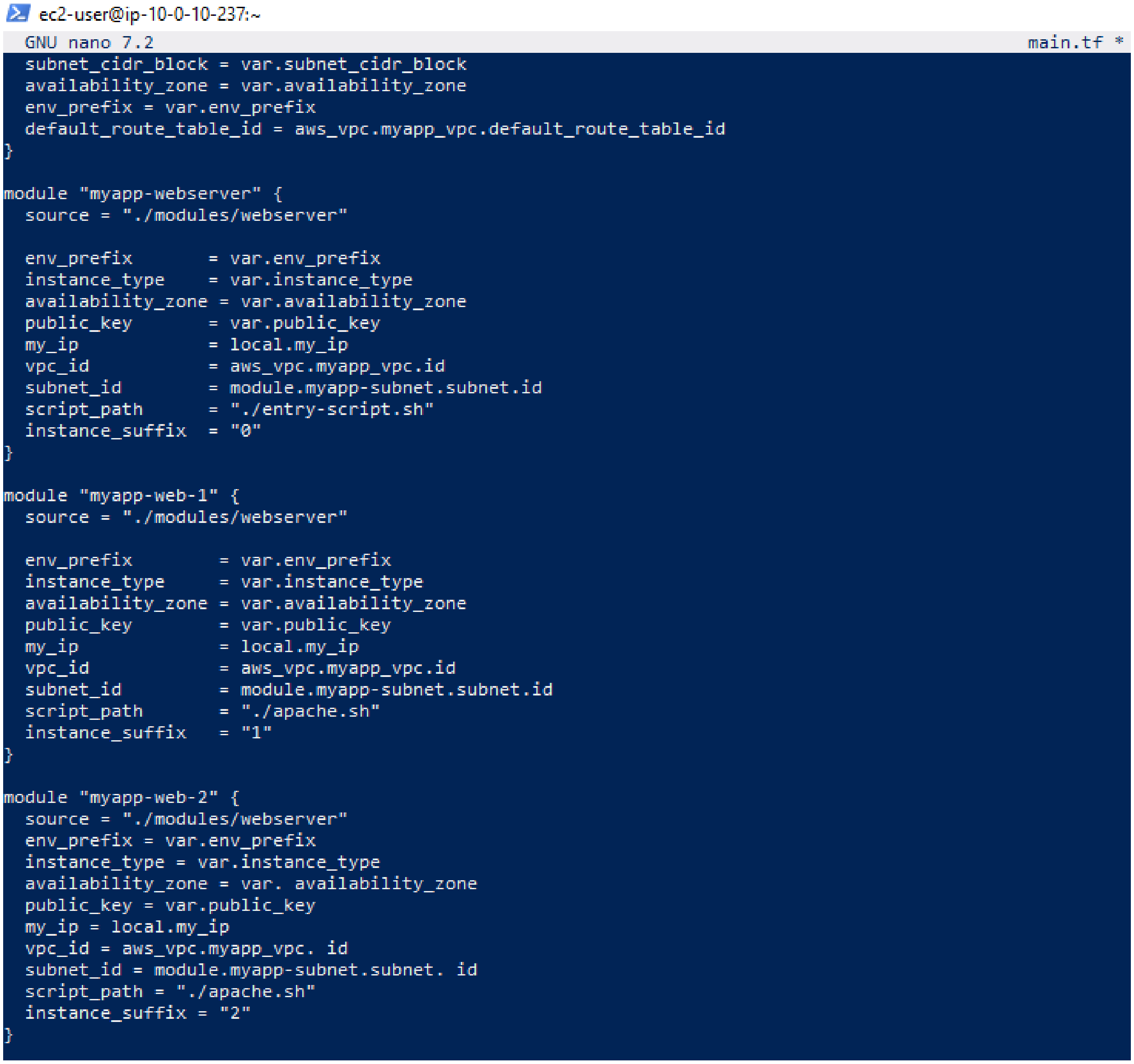
Private IP: 10.0.10.158

Public IP: 3.29.63.153

Public DNS:

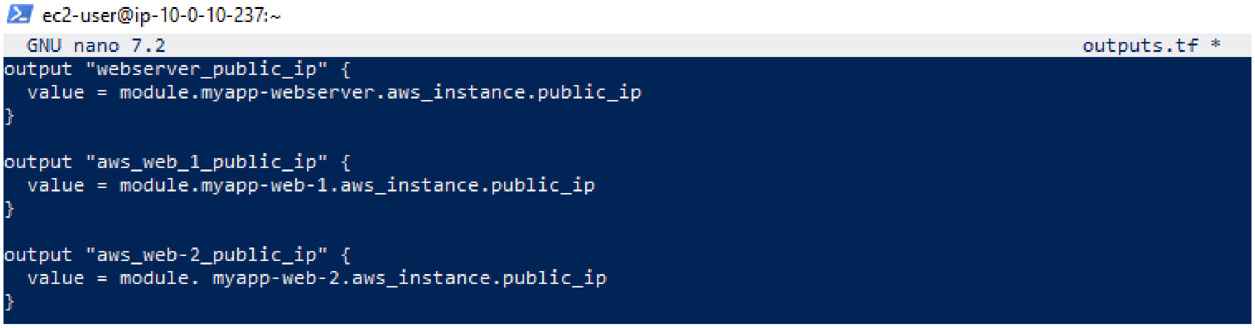
Deployed via Terraform

Task#08: Configure Nginx as load balancer



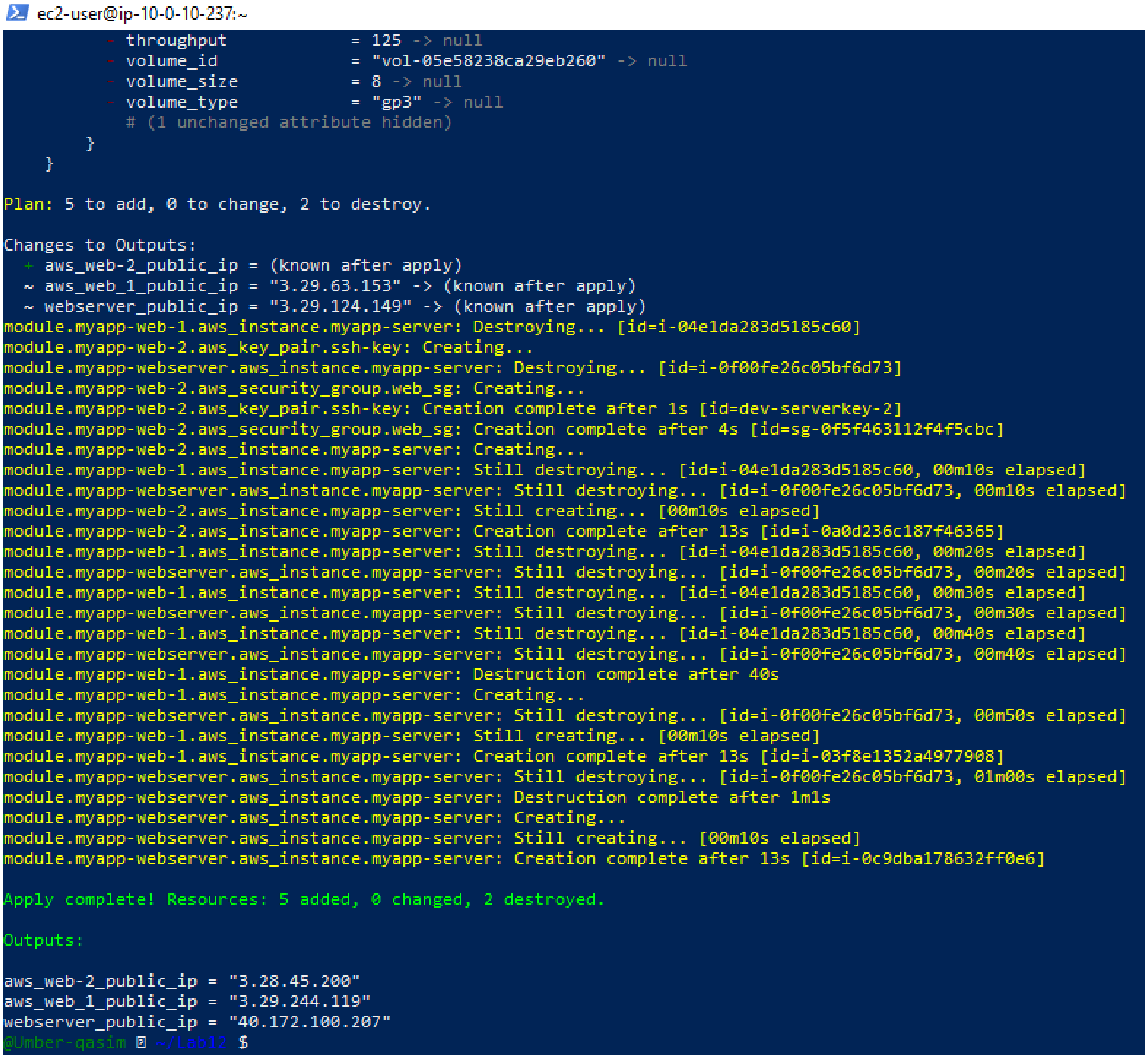
ec2-user@ip-10-0-10-237:~

```
  GNU nano 7.2                                                          main.tf *
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix = var.env_prefix
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id
}

module "myapp-webserver" {
  source = "./modules/webserver"

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

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"
}

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"
}
```

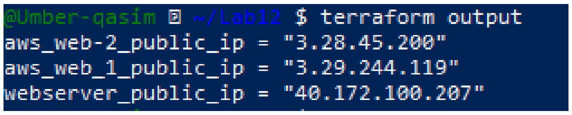


ec2-user@ip-10-0-10-237:~

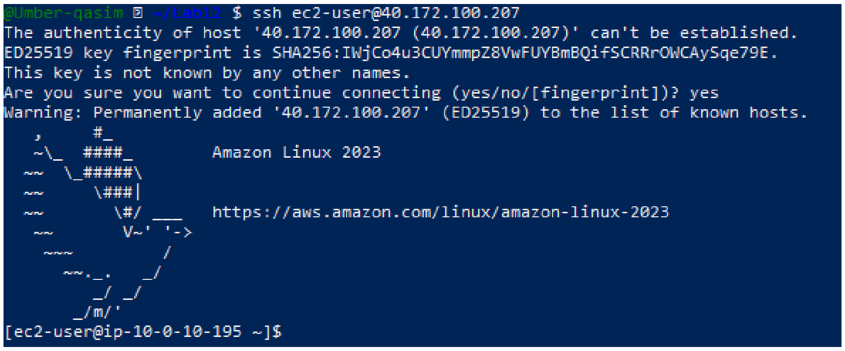
```
  GNU nano 7.2                                                       outputs.tf *
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}

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
}
```



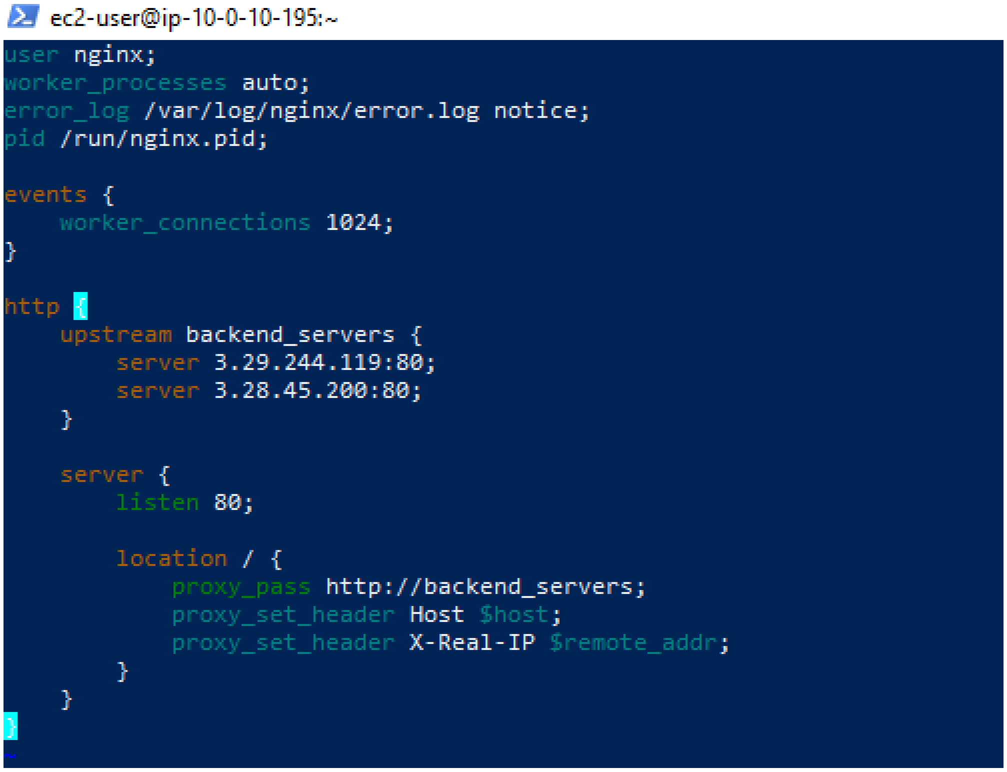
```
          - throughput                = 125 -> null
          - volume_id                 = "vol-05e58238ca29eb260" -> null
          - volume_size               = 8 -> null
          - volume_type               = "gp3" -> null
            # (1 unchanged attribute hidden)
        }
    }

Plan: 5 to add, 0 to change, 2 to destroy.

Changes to Outputs:
  + aws_web-2_public_ip = (known after apply)
  ~ aws_web_1_public_ip = "3.29.63.153" -> (known after apply)
  ~ webserver_public_ip = "3.29.124.149" -> (known after apply)
module.myapp-web-1.aws_instance.myapp-server: Destroying... [id=i-04e1da283d5185c60]
module.myapp-web-2.aws_key_pair.ssh-key: Creating...
module.myapp-webserver.aws_instance.myapp-server: Destroying... [id=i-0f00fe26c05bf6d73]
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]
module.myapp-web-2.aws_security_group.web_sg: Creation complete after 4s [id=sg-0f5f463112f4f5cbc]
module.myapp-web-2.aws_instance.myapp-server: Creating...
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-04e1da283d5185c60, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0f00fe26c05bf6d73, 00m10s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Creation complete after 13s [id=i-0a0d236c187f46365]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-04e1da283d5185c60, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0f00fe26c05bf6d73, 00m20s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-04e1da283d5185c60, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0f00fe26c05bf6d73, 00m30s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-04e1da283d5185c60, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0f00fe26c05bf6d73, 00m40s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Destruction complete after 40s
module.myapp-web-1.aws_instance.myapp-server: Creating...
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0f00fe26c05bf6d73, 00m50s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still creating... [00m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Creation complete after 13s [id=i-03f8e1352a4977908]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0f00fe26c05bf6d73, 01m00s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 1m1s
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 13s [id=i-0c9dba178632ff0e6]

Apply complete! Resources: 5 added, 0 changed, 2 destroyed.

Outputs:

aws_web-2_public_ip = "3.28.45.200"
aws_web_1_public_ip = "3.29.244.119"
webserver_public_ip = "40.172.100.207"
@Umber-qasim ⊡ ~/Lab12 $
```



```
@Umber-qasim ⊡ ~/Lab12 $ terraform output
aws_web-2_public_ip = "3.28.45.200"
aws_web_1_public_ip = "3.29.244.119"
webserver_public_ip = "40.172.100.207"
```

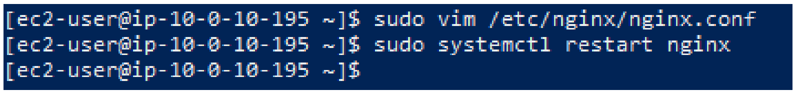


```
@Umber-qasim ⊡ ~/Lab12 $ ssh ec2-user@40.172.100.207
The authenticity of host '40.172.100.207 (40.172.100.207)' can't be established.
ED25519 key fingerprint is SHA256:IWjCo4u3CUYmmpZ8VwFUYBmBQifSCRRrOWCAySqe79E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '40.172.100.207' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-10-195 ~]$
```

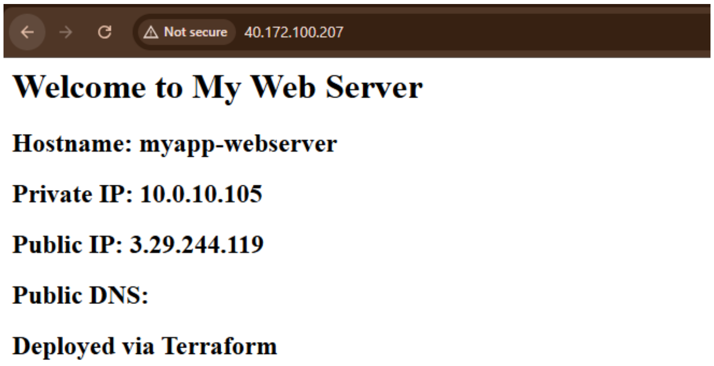


ec2-user@ip-10-0-10-195:~

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    upstream backend_servers {
        server 3.29.244.119:80;
        server 3.28.45.200:80;
    }

    server {
        listen 80;

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



```
[ec2-user@ip-10-0-10-195 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-0-10-195 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-195 ~]$
```



Welcome to My Web Server

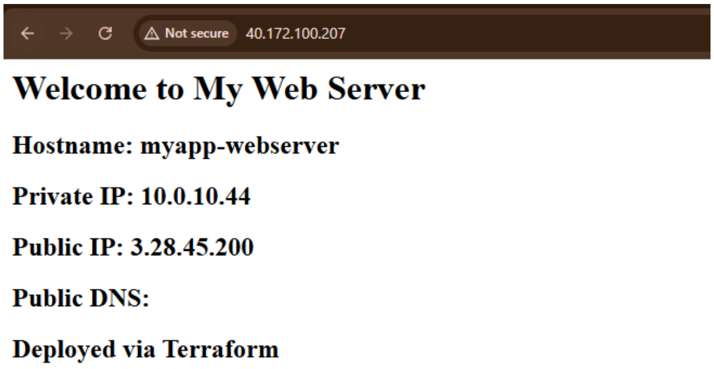
Hostname: myapp-webserver

Private IP: 10.0.10.105

Public IP: 3.29.244.119

Public DNS:

Deployed via Terraform



Welcome to My Web Server

Hostname: myapp-webserver

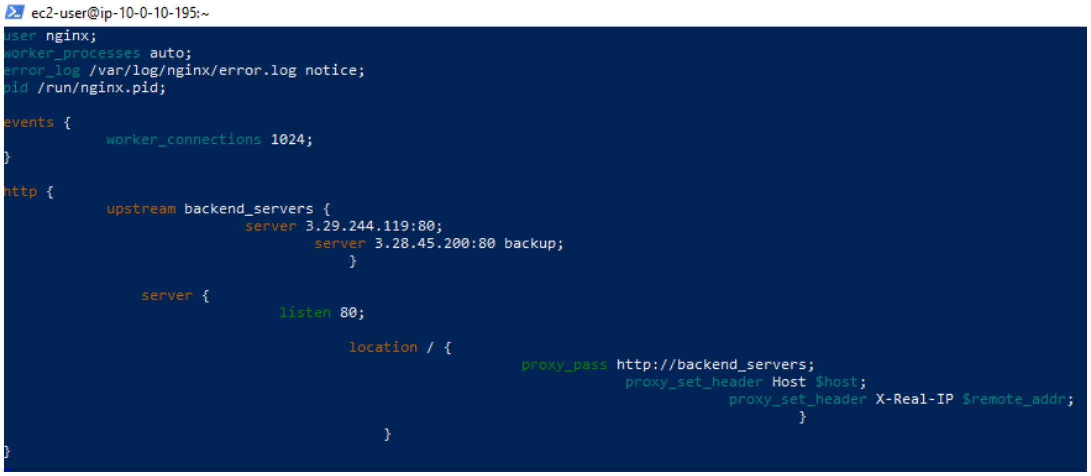
Private IP: 10.0.10.44

Public IP: 3.28.45.200

Public DNS:

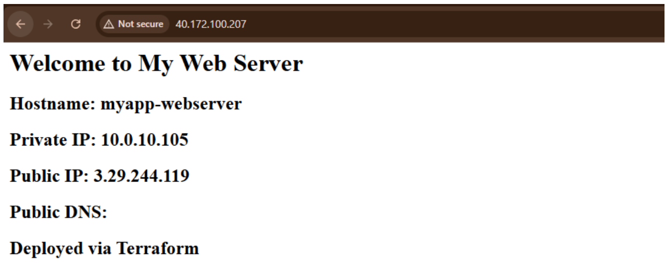
Deployed via Terraform

Task#09: Configure high availability with backup servers



ec2-user@ip-10-0-10-195:~

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
            worker_connections 1024;
}

http {
            upstream backend_servers {
                            server 3.29.244.119:80;
                                    server 3.28.45.200:80 backup;
                                        }

                server {
                                listen 80;

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

                                            }
}
```



⚠ Not secure 40.172.100.207

Welcome to My Web Server

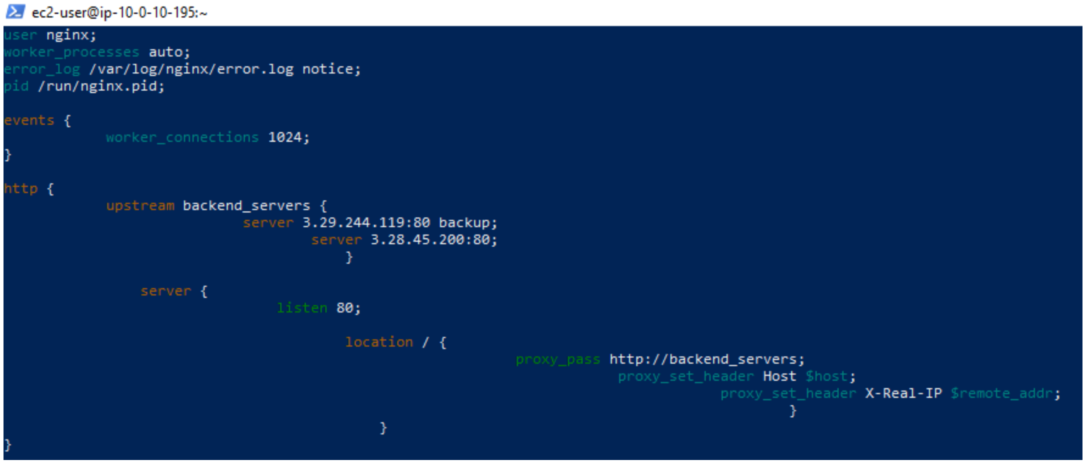
Hostname: myapp-webserver

Private IP: 10.0.10.105

Public IP: 3.29.244.119

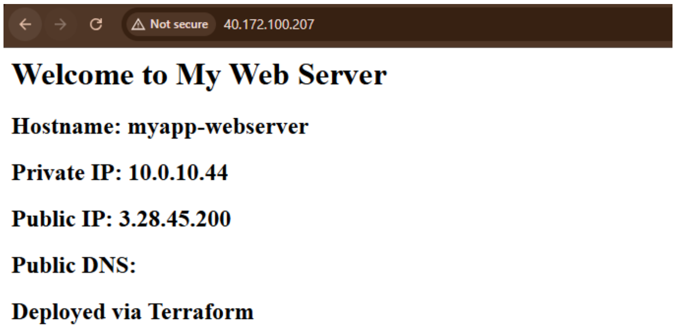
Public DNS:

Deployed via Terraform



ec2-user@ip-10-0-10-195:~

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
            worker_connections 1024;
}

http {
            upstream backend_servers {
                            server 3.29.244.119:80 backup;
                                    server 3.28.45.200:80;
                                        }

                server {
                                listen 80;

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

                                            }
}
```



⚠ Not secure 40.172.100.207

Welcome to My Web Server

Hostname: myapp-webserver

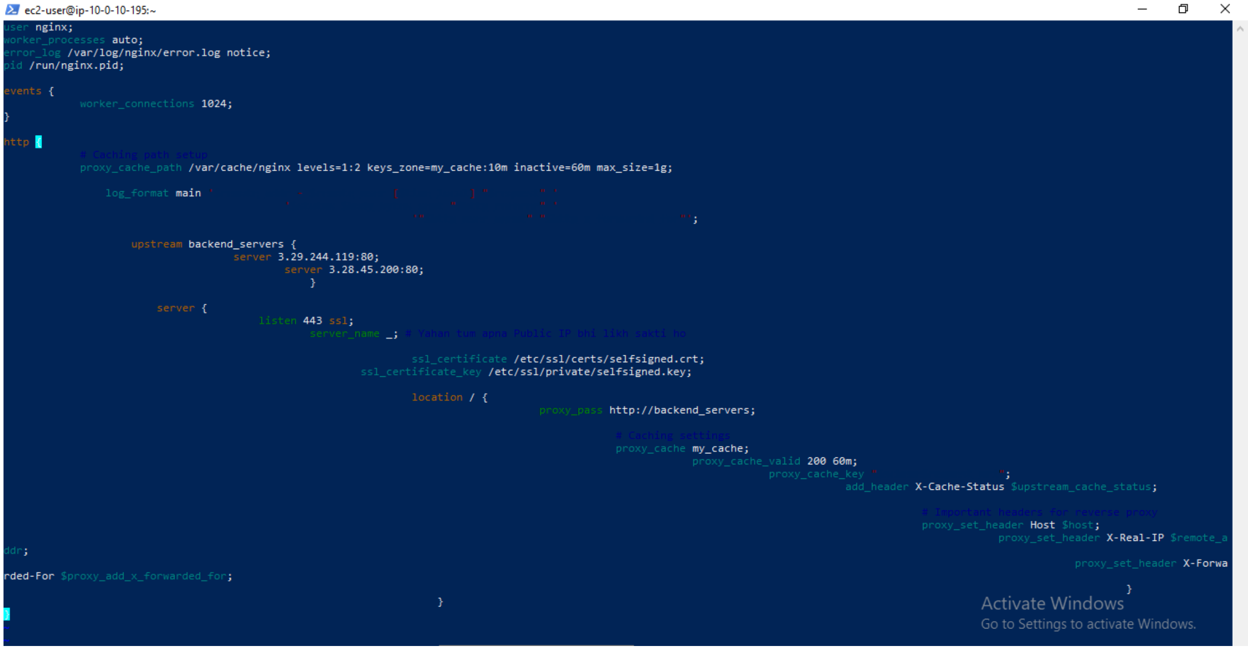
Private IP: 10.0.10.44

Public IP: 3.28.45.200

Public DNS:

Deployed via Terraform

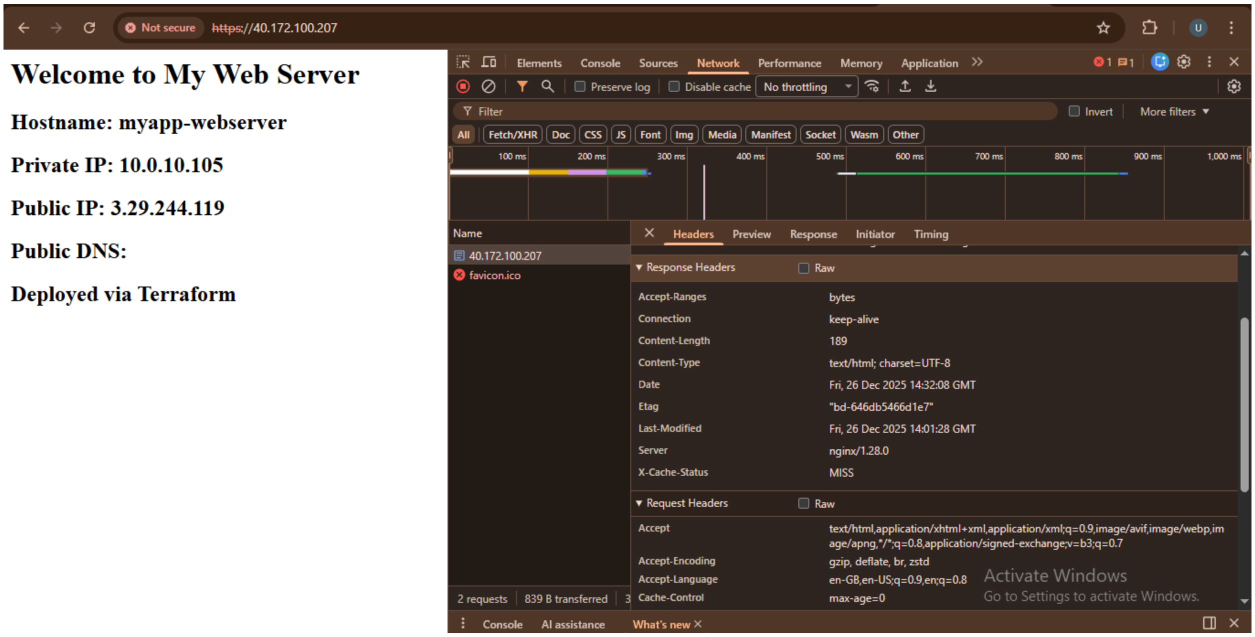
<u>Task#10:</u> Enable Nginx caching



ec2-user@ip-10-0-10-195:~

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
        worker_connections 1024;
}

http {
        # Caching path setup
        proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_cache:10m inactive=60m max_size=1g;

            log_format main '          -              [          ] "        " '
                          '              "               " '
                                         '"               " "            "';

                upstream backend_servers {
                                server 3.29.244.119:80;
                                        server 3.28.45.200:80;
                                              }

                    server {
                                listen 443 ssl;
                                        server_name _; # Yahan tum apna Public IP bhi likh sakti ho

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

                                                location / {
                                                          proxy_pass http://backend_servers;

                                                                # Caching settings
                                                                proxy_cache my_cache;
                                                                      proxy_cache_valid 200 60m;
                                                                            proxy_cache_key "                ";
                                                                                  add_header X-Cache-Status $upstream_cache_status;

                                                                                        # Important headers for reverse proxy
                                                                                        proxy_set_header Host $host;
                                                                                              proxy_set_header X-Real-IP $remote_a
ddr;
                                                                                                    proxy_set_header X-Forwa
rded-For $proxy_add_x_forwarded_for;
                                                                                                    }
                                  }
}
```

```
[ec2-user@ip-10-0-10-195 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-195 ~]$
```



https://40.172.100.207

Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.105

Public IP: 3.29.244.119

Public DNS:

Deployed via Terraform

Name
40.172.100.207
favicon.ico

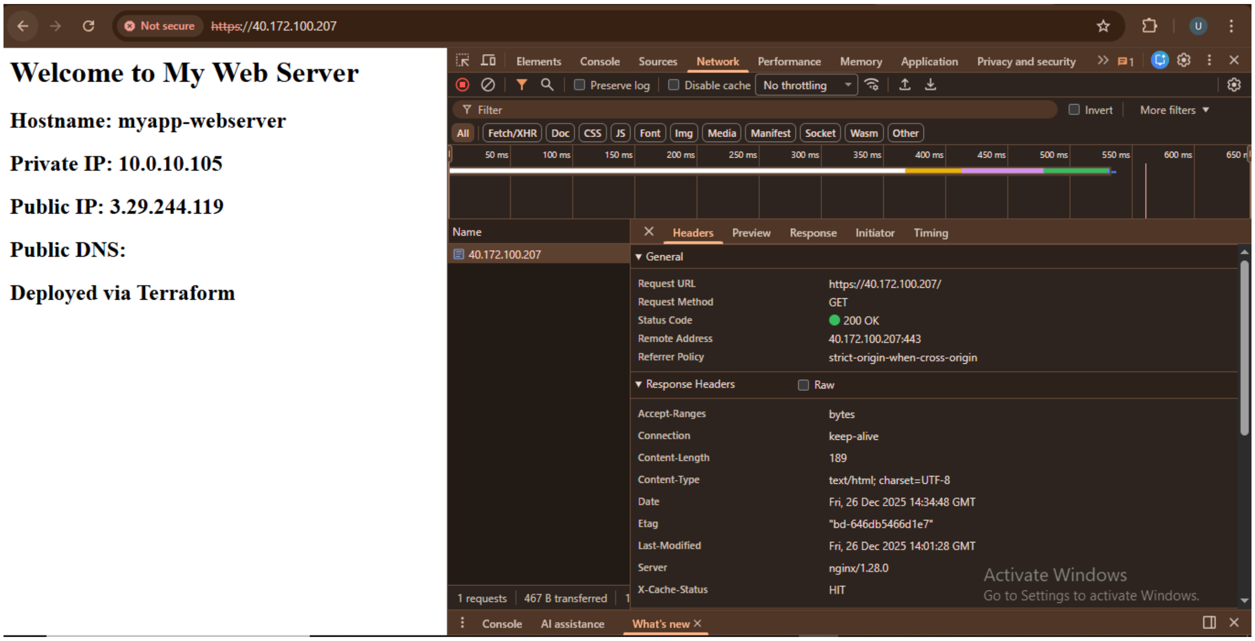
2 requests | 839 B transferred | 3

▾ Response Headers □ Raw

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	189
Content-Type	text/html; charset=UTF-8
Date	Fri, 26 Dec 2025 14:32:08 GMT
Etag	"bd-646db5466d1e7"
Last-Modified	Fri, 26 Dec 2025 14:01:28 GMT
Server	nginx/1.28.0
X-Cache-Status	MISS

▾ Request Headers □ Raw

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	en-GB,en-US;q=0.9,en;q=0.8
Cache-Control	max-age=0



Welcome to My Web Server

Hostname: myapp-webserver

Private IP: 10.0.10.105

Public IP: 3.29.244.119

Public DNS:

Deployed via Terraform

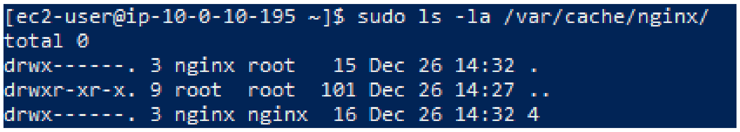
40.172.100.207

▾ General

Request URL	https://40.172.100.207/
Request Method	GET
Status Code	● 200 OK
Remote Address	40.172.100.207:443
Referrer Policy	strict-origin-when-cross-origin

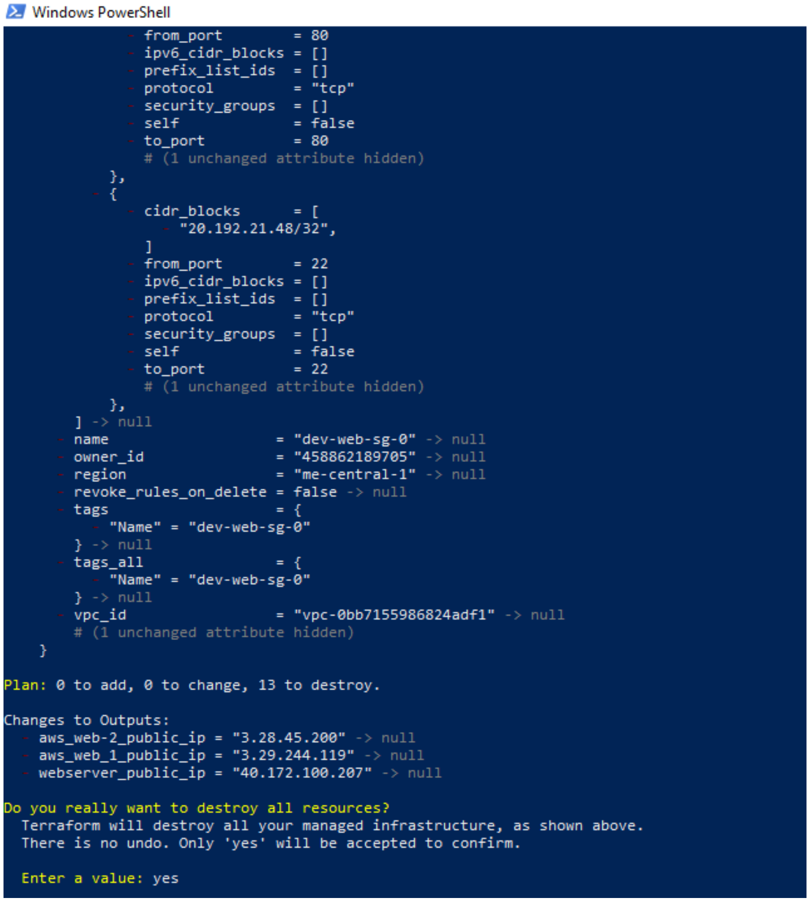
▾ Response Headers □ Raw

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	189
Content-Type	text/html; charset=UTF-8
Date	Fri, 26 Dec 2025 14:34:48 GMT
Etag	"bd-646db5466d1e7"
Last-Modified	Fri, 26 Dec 2025 14:01:28 GMT
Server	nginx/1.28.0
X-Cache-Status	HIT

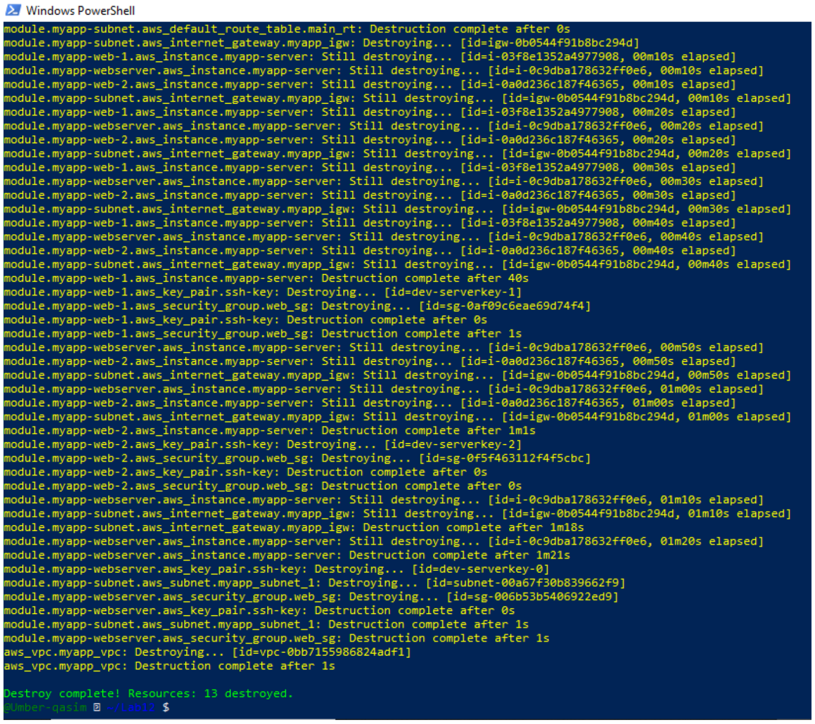


```
[ec2-user@ip-10-0-10-195 ~]$ sudo ls -la /var/cache/nginx/
total 0
drwx------. 3 nginx root    15 Dec 26 14:32 .
drwxr-xr-x. 9 root  root   101 Dec 26 14:27 ..
drwx------. 3 nginx nginx  16 Dec 26 14:32 4
```

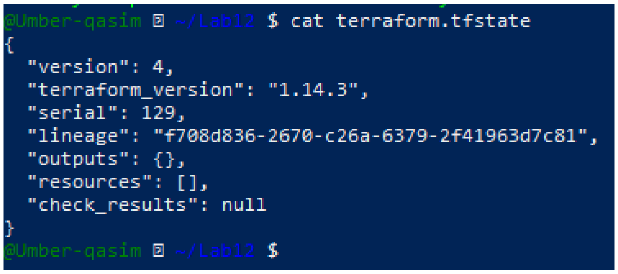
Destroying all resources:



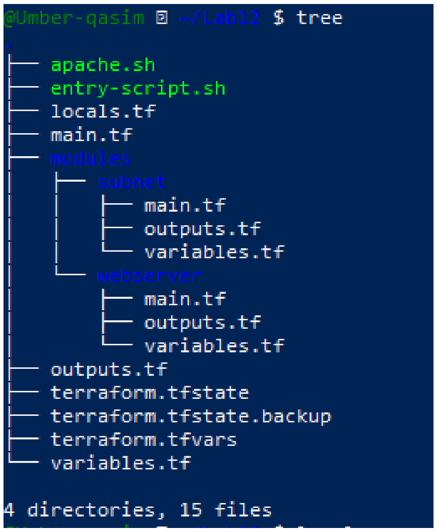
```
              - from_port        = 80
              - ipv6_cidr_blocks = []
              - prefix_list_ids  = []
              - protocol         = "tcp"
              - security_groups  = []
              - self             = false
              - to_port          = 80
                # (1 unchanged attribute hidden)
            },
          - {
              - cidr_blocks      = [
                  - "20.192.21.48/32",
                ]
              - from_port        = 22
              - ipv6_cidr_blocks = []
              - prefix_list_ids  = []
              - protocol         = "tcp"
              - security_groups  = []
              - self             = false
              - to_port          = 22
                # (1 unchanged attribute hidden)
            },
        ] -> null
      - name                   = "dev-web-sg-0" -> null
      - owner_id               = "458862189705" -> null
      - region                 = "me-central-1" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {
          - "Name" = "dev-web-sg-0"
        } -> null
      - tags_all               = {
          - "Name" = "dev-web-sg-0"
        } -> null
      - vpc_id                 = "vpc-0bb7155986824adf1" -> null
        # (1 unchanged attribute hidden)
    }

Plan: 0 to add, 0 to change, 13 to destroy.

Changes to Outputs:
  - aws_web-2_public_ip = "3.28.45.200" -> null
  - aws_web_1_public_ip = "3.29.244.119" -> null
  - webserver_public_ip = "40.172.100.207" -> 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
```



```
module.myapp-subnet.aws_default_route_table.main_rt: Destruction complete after 0s
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destroying... [id=igw-0b0544f91b8bc294d]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-03f8e1352a4977908, 00m10s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 00m10s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-0a0d236c187f46365, 00m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 00m10s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-03f8e1352a4977908, 00m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 00m20s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-0a0d236c187f46365, 00m20s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 00m20s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-03f8e1352a4977908, 00m30s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 00m30s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-0a0d236c187f46365, 00m30s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 00m30s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Still destroying... [id=i-03f8e1352a4977908, 00m40s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 00m40s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-0a0d236c187f46365, 00m40s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 00m40s elapsed]
module.myapp-web-1.aws_instance.myapp-server: Destruction complete after 40s
module.myapp-web-1.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-1]
module.myapp-web-1.aws_security_group.web_sg: Destroying... [id=sg-0af09c6eae69d74f4]
module.myapp-web-1.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-web-1.aws_security_group.web_sg: Destruction complete after 1s
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 00m50s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-0a0d236c187f46365, 00m50s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 00m50s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 01m00s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Still destroying... [id=i-0a0d236c187f46365, 01m00s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 01m00s elapsed]
module.myapp-web-2.aws_instance.myapp-server: Destruction complete after 1m1s
module.myapp-web-2.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-2]
module.myapp-web-2.aws_security_group.web_sg: Destroying... [id=sg-0f5f463112f4f5cbc]
module.myapp-web-2.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-web-2.aws_security_group.web_sg: Destruction complete after 0s
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 01m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Still destroying... [id=igw-0b0544f91b8bc294d, 01m10s elapsed]
module.myapp-subnet.aws_internet_gateway.myapp_igw: Destruction complete after 1m18s
module.myapp-webserver.aws_instance.myapp-server: Still destroying... [id=i-0c9dba178632ff0e6, 01m20s elapsed]
module.myapp-webserver.aws_instance.myapp-server: Destruction complete after 1m21s
module.myapp-webserver.aws_key_pair.ssh-key: Destroying... [id=dev-serverkey-0]
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destroying... [id=subnet-00a67f30b839662f9]
module.myapp-webserver.aws_security_group.web_sg: Destroying... [id=sg-006b53b5406922ed9]
module.myapp-webserver.aws_key_pair.ssh-key: Destruction complete after 0s
module.myapp-subnet.aws_subnet.myapp_subnet_1: Destruction complete after 1s
module.myapp-webserver.aws_security_group.web_sg: Destruction complete after 1s
aws_vpc.myapp_vpc: Destroying... [id=vpc-0bb7155986824adf1]
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 13 destroyed.
@Umber-qasim ~/Lab12 $
```



```
@Umber-qasim ➜ ~/Lab12 $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 129,
  "lineage": "f708d836-2670-c26a-6379-2f41963d7c81",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Umber-qasim ➜ ~/Lab12 $
```



```
@Umber-qasim ➜ ~/Lab12 $ tree
.
├── apache.sh
├── entry-script.sh
├── locals.tf
├── main.tf
├── modules
│   ├── subnet
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   └── webserver
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
└── variables.tf

4 directories, 15 files
```