**NAME: SANA TARIQ**
**CLASS: BSE 5B**
**REG.NO: 058**
**SUBJECT: CLOUD COMPUTING**

# LAB#12

## TASK 1: Organize Terraform Files

```
@SanaTariq205 /workspaces/cc_sanatariq_058 (main) $ mkdir -p ~/Lab12
@SanaTariq205 /workspaces/cc_sanatariq_058 (main) $ cd ~/Lab12
@SanaTariq205 ~/Lab12 $ touch main.tf variables.tf outputs.tf locals.tf terraform.tfvars entry-script.sh
@SanaTariq205 ~/Lab12 $ ls -la
total 12
drwxrwxr-x 2 codespace codespace 4096 Jan  4 14:37 .
drwxr-x--- 1 codespace codespace 4096 Jan  4 14:37 ..
-rw-rw-r-- 1 codespace codespace    0 Jan  4 14:37 entry-script.sh
-rw-rw-r-- 1 codespace codespace    0 Jan  4 14:37 locals.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  4 14:37 main.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  4 14:37 outputs.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  4 14:37 terraform.tfvars
-rw-rw-r-- 1 codespace codespace    0 Jan  4 14:37 variables.tf
```

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

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

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

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

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

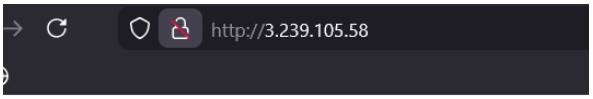
```
 GNU nano 7.2                                          main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

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

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

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

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

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

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

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

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

```
@SanaTariq205 /workspaces/Lab12 $ @SanaTariq205 /workspaces/Lab12 $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
/home/codespace/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
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:
SHA256:1dyrD5yIZCWYC7gDJ91CdNq8pDZ3fmiEewfmnAZ2rmE codespace@codespaces-787f62
The key's randomart image is:
+--[ED25519 256]--+
|.o .             |
|.+=  o   . o     |
|*.++o . . + .    |
|.=o.o. o .   .   |
|o+ *.*o S    .   |
|..+ %o+. o o     |
|   E @.o. =      |
|  . * o    o     |
|   .         .   |
+----[SHA256]-----+
```

```
@SanaTariq205 ~/Lab12 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (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!
```

```
Changes to Outputs:
  + aws_instance_public_ip = (known after apply)
aws_instance.myapp-server: Creating...
aws_instance.myapp-server: Still creating... [00m10s elapsed]
aws_instance.myapp-server: Creation complete after 15s [id=i-0713da65da774dd7c]

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




# ðŸš€ Lab12 Nginx Server

**Public IP:** 3.239.105.58

**Hostname:** $(hostname)

**Deployed:** $(date)

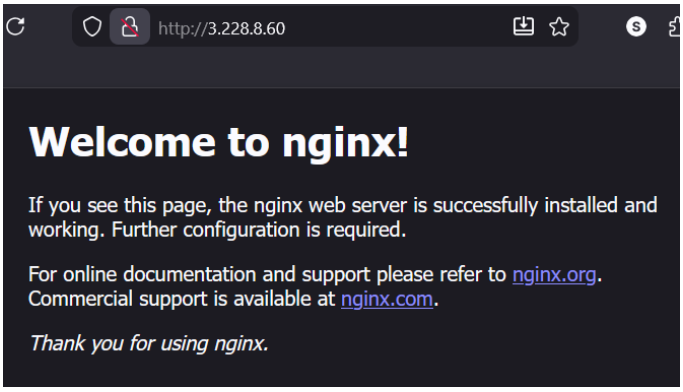
**Status:** âœ… Running

```
Destroy complete! Resources: 7 destroyed.
@SanaTariq205 ~/Lab12 $
```

## TASK 2: Remote-Exec Provisioner

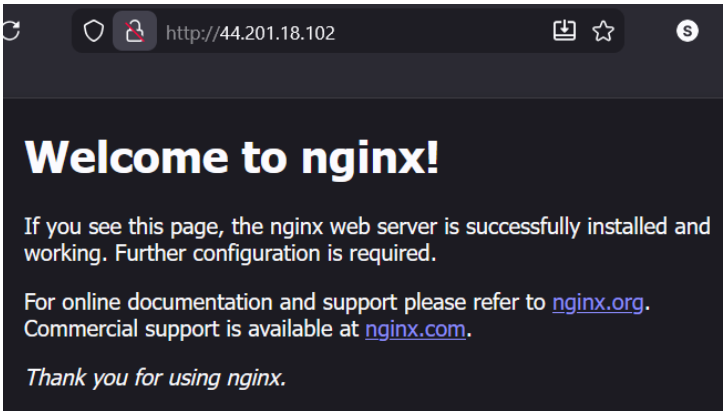
```
resource "aws_instance" "myapp-server" {
  ami                         = "ami-03f9680ef0c07a3d1"
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  security_groups             = [aws_default_security_group.default_sg.id]
  availability_zone           = 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"
```

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

Outputs:

aws_instance_public_ip = "3.228.8.60"
@SanaTariq205 ~/Lab12 $
@SanaTariq205 ~/Lab12 $
@SanaTariq205 ~/Lab12 $ terraform output
aws_instance_public_ip = "3.228.8.60"
@SanaTariq205 ~/Lab12 $
```




## TASK 3: File & Local-Exec Provisioners

```
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"
```




```
resource "aws_instance" "myapp-server" {
  ami                         = "ami-03f9680ef0c07a3d1"
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  security_groups             = [aws_default_security_group.default_sg.id]
  availability_zone           = var.availability_zone
  associate_public_ip_address = true
  key_name                    = aws_key_pair.ssh-key.key_name
  user_data = file("./entry-script.sh")
  tags = {
    Name = "${var.env_prefix}-ec2-instance"
  }
}
```

## TASK 4: Create Subnet Module

```
@SanaTariq205 ~/Lab12 $ mkdir -p modules/subnet
@SanaTariq205 ~/Lab12 $ touch modules/subnet/main.tf
@SanaTariq205 ~/Lab12 $ touch modules/subnet/variables.tf
@SanaTariq205 ~/Lab12 $ touch modules/subnet/outputs.tf
@SanaTariq205 ~/Lab12 $ ls -R modules/
modules/:
subnet

modules/subnet:
main.tf  outputs.tf  variables.tf
```

```
  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 *

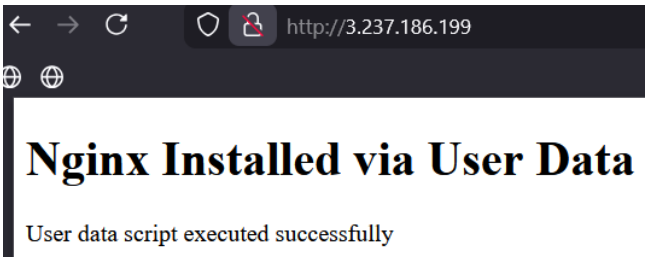
```
resource "aws_subnet" "myapp_subnet_1" {
  vpc_id                  = var.vpc_id
  cidr_block              = var.subnet_cidr_block
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true
  tags = {
    Name = "${var.env_prefix}-subnet-1"
  }
}

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

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

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

GNU nano 7.2 modules/subnet/outputs.tf *

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

```
resource "aws_instance" "myapp-server" {
  ami                         = "ami-03f9680ef0c07a3d1"
  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"
  }
}
```

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

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




## TASK 5: Create Webserver Module

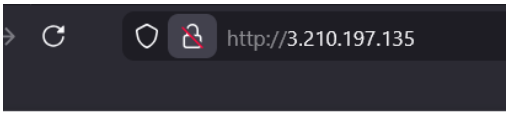
```
  GNU nano 7.2                                  modules/webserver/variables.tf *
variable "env_prefix" {}
variable "instance_type" {}
variable "availability_zone" {}
variable "public_key" {}
variable "my_ip" {}
variable "vpc_id" {}
variable "subnet_id" {}
variable "script_path" {}
variable "instance_suffix"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

```
  GNU nano 7.2                                  modules/webserver/main.tf *
resource "aws_security_group" "web_sg" {
  vpc_id      = var.vpc_id
  name        = "${var.env_prefix}-web-sg-${var.instance_suffix}"
  description = "Security group for web server allowing HTTP, HTTPS and SSH"

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

  ingress {
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

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

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

```
  GNU nano 7.2                                          main.tf *
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"
}
```

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




# Lab12 Nginx Server

**Successfully deployed with Terraform**

**IP Address:** 3.210.197.135

**Hostname:** $(hostname)

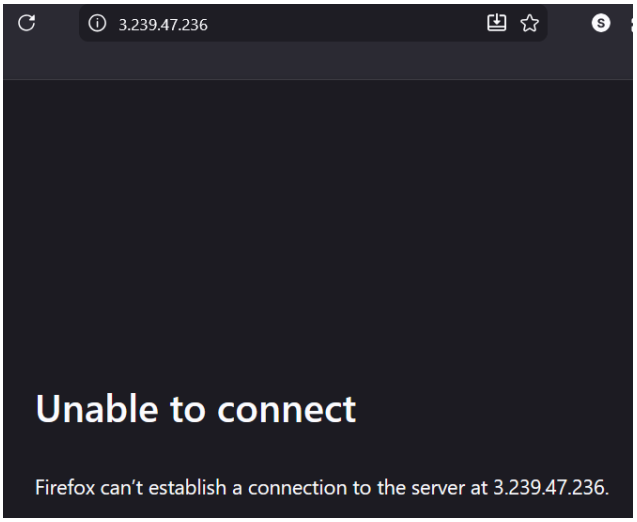
**Deployed:** $(date)

```
aws_vpc.myapp_vpc: Destroying... [id=vpc-0e65973cda
aws_vpc.myapp_vpc: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
```

## TASK 6: Configure HTTPS

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

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

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

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

# Generate self-signed certificate
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/private/selfsigned.key \
  -out /etc/ssl/certs/selfsigned.crt \
  -subj "/CN=$PUBLIC_IP" \
  -addext "subjectAltName=IP:$PUBLIC_IP" \
  -addext "basicConstraints=CA:FALSE" \
  -addext "keyUsage=digitalSignature,keyEncipherment" \
  -addext "extendedKeyUsage=serverAuth"

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

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

Outputs:
```




## TASK 7: Configure Reverse Proxy

```
 GNU nano 7.2                                  apache.sh *
#!/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/>
echo "<h2>Public IP: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/p>
echo "<h2>Public DNS: $(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/>
echo "<h2>Deployed via Terraform</h2>" >> /var/www/html/index.html
```

```
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"
}
```

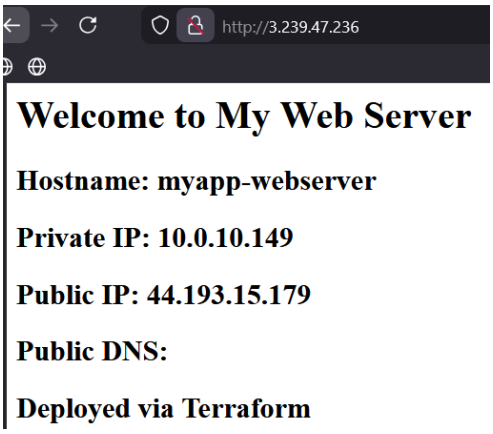
```
 GNU nano 7.2                                        outputs.tf
output "webserver_public_ip" {
  value = module.myapp-webserver.aws_instance.public_ip
}
output "aws_web-1_public_ip" {
  value = module.myapp-web-1.aws_instance.public_ip
}
```

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

Outputs:

aws_web-1_public_ip = "44.193.15.179"
webserver_public_ip = "3.239.47.236"
```

```
@SanaTariq205 🗄 ~/Lab12 $ ssh ec2-user@3.239.47.236
The authenticity of host '3.239.47.236 (3.239.47.236)' can't be established.
ED25519 key fingerprint is SHA256:PxiVRsksfaQk7pbvAalFz3vXTC1TCMDDttwZR9nmJfY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.239.47.236' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-10-113 ~]$
```

```
[ec2-user@ip-10-0-10-113 ~]$ sudo cat /var/log/nginx/error.log
[ec2-user@ip-10-0-10-113 ~]$ cat /var/log/nginx/access.log
84.33.241.191 - - [05/Jan/2026:03:42:25 +0000] "GET /.env HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; >
4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.140 Safari/537.36" "-"
84.33.241.191 - - [05/Jan/2026:03:42:26 +0000] "POST / HTTP/1.1" 200 190 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.140 Safari/537.36" "-"
216.180.246.218 - - [05/Jan/2026:03:45:12 +0000] "\x16\x03\x01\x00\xFA\x01\x00\x00\xF6\x03\x030\xCB\x8F\xCF\x90\x82F\xA
GA\xAA\xB8\xCB5\xAC\x7F\xA5l]\xB3r\x9F\xF5\x8F<\xE3\x8F\xBBS\xD7\x02\xC4 A\xBCX\xBB\x1F\xB8\x93J-\xF5\xB8S\xF3\x1A\xDA|
(7\x91s\xE3\xB5\xB1Q\xFEE\xE5`\x1CfB\x002\xC0/\xC0+\xC0\x11\xC0\x07\xC0\x13\xC0\x09\xC0\x14\xC0" 400 157 "-" "-" "-"
216.180.246.218 - - [05/Jan/2026:03:45:28 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "-" "'Mozilla/5.0 (compatible; Ger
meCrawlerd/1.0; +https://www.nokia.com/genomecrawler)'" "-"
93.174.93.12 - - [05/Jan/2026:04:09:39 +0000] "GET / HTTP/1.0" 200 190 "-" "Mozilla/5.0 (X11; Linux x86_64; en-US; rv:2
0b2pre) Gecko/20100712 Minefield/4.0b2pre" "-"
204.76.203.219 - - [05/Jan/2026:04:11:13 +0000] "GET / HTTP/1.1" 200 190 "-" "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" "-"
204.76.203.219 - - [05/Jan/2026:05:03:43 +0000] "GET / HTTP/1.1" 200 190 "-" "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" "-"
```

```
[ec2-user@ip-10-0-10-113 ~]$ cat /etc/nginx/mime.types

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

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

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



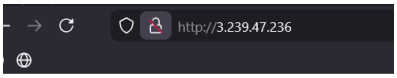

```
module "myapp-web-2" {
  source            = "./modules/webserver"
  env_prefix        = var.env_prefix
  instance_type     = var.instance_type
  availability_zone = var.availability_zone
  public_key        = var.public_key
  my_ip             = local.my_ip
  vpc_id            = aws_vpc.myapp_vpc.id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

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

```
Apply complete! Resources: 2 added, 0 changed, 1 destroyed.

Outputs:

aws_web-1_public_ip = "44.193.15.179"
aws_web-2_public_ip = "3.222.205.24"
webserver_public_ip = "3.239.47.236"
@SanaTariq205 ~/Lab12 $
```

## TASK 8: Configure Load Balancer

```
[ec2-user@ip-10-0-10-113 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-113 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2026-01-05 14:10:33 UTC; 7s ago
  Process: 3412 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 3409 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 3407 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 3414 (nginx)
   CGroup: /system.slice/nginx.service
           ├─3414 nginx: master process /usr/sbin/nginx
           ├─3415 nginx: worker process
           └─3416 nginx: worker process

Jan 05 14:10:33 ip-10-0-10-113.ec2.internal systemd[1]: Starting The nginx HTTP and reverse proxy server.
Jan 05 14:10:33 ip-10-0-10-113.ec2.internal nginx[3409]: nginx: the configuration file /etc/nginx/nginx.co
Jan 05 14:10:33 ip-10-0-10-113.ec2.internal nginx[3409]: nginx: configuration file /etc/nginx/nginx.conf t
Jan 05 14:10:33 ip-10-0-10-113.ec2.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-10-0-10-113 ~]$
```




# Welcome to My Web Server

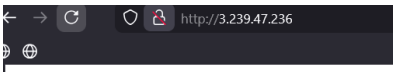
**Hostname: myapp-webserver**

**Private IP: 10.0.10.129**

**Public IP: 3.222.205.24**

**Public DNS:**

**Deployed via Terraform**




# Welcome to My Web Server

**Hostname: myapp-webserver**

**Private IP: 10.0.10.149**

**Public IP: 44.193.15.179**

**Public DNS:**

**Deployed via Terraform**

## TASK 9: High Availability with Backup

```
[ec2-user@ip-10-0-10-113 ~]$ sudo tee /etc/nginx/nginx.conf << 'EOF'
> user nginx;
> worker_processes auto;
> error_log /var/log/nginx/error.log;
> pid /run/nginx.pid;
>
> events {
>     worker_connections 1024;
> }
>
> http {
>     include       /etc/nginx/mime.types;
>     default_type  application/octet-stream;
>
>     log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
>                       '$status $body_bytes_sent "$http_referer" "$http_user_agent" "$http_x_forwarded_for"';
>
>     access_log  /var/log/nginx/access.log  main;
>
>     sendfile        on;
>     tcp_nopush      on;
>     tcp_nodelay     on;
>     keepalive_timeout  65;
ackup se>     types_hash_max_size 4096;
>
>     # Load balancing upstream with backup server
>     upstream backend_servers {
>         server 44.193.15.179:80;
>         server 3.222.205.24:80 backup;
>     }
>
>     include /etc/nginx/conf.d/*.conf;
> }
> EOF
```

---

# Welcome to My Web Server

### Hostname: myapp-webserver

### Private IP: 10.0.10.149

### Public IP: 44.193.15.179

### Public DNS:

### Deployed via Terraform

## TASK 10: Enable Caching

```
[ec2-user@ip-10-0-10-113 ~]$ sudo tee /etc/nginx/nginx.conf << 'EOF'
> user nginx;
> worker_processes auto;
> error_log /var/log/nginx/error.log;
> pid /run/nginx.pid;
>
> events {
>     worker_connections 1024;
> }
>
> http {
>     include       /etc/nginx/mime.types;
>     default_type  application/octet-stream;
>
>     log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
>                       '$status $body_bytes_sent "$http_referer" "$http_user_agent" "$http_x_forwarded_for"';
>
>     access_log  /var/log/nginx/access.log  main;
>
>     sendfile        on;
>     tcp_nopush      on;
>     tcp_nodelay     on;
>     keepalive_timeout  65;
>     types_hash_max_size 4096;
>
>     # CACHE SETTINGS
>     proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_cache:10m inactive=60m max_size=1g;
>
>     # Load balancing upstream with backup server
>     upstream backend_servers {
>         server 44.193.15.179:80;
>         server 3.222.205.24:80 backup;
>     }
>
>     include /etc/nginx/conf.d/*.conf;
> }
> EOF
```

```
[ec2-user@ip-10-0-10-113 ~]$ sudo tee /etc/nginx/conf.d/default.conf << 'EOF'
> server {
>     listen 80;
>     server_name _;
>
>     location / {
>         proxy_pass http://backend_servers;
>
>         # Caching configuration
>         proxy_cache my_cache;
>         proxy_cache_valid 200 60m;
>         proxy_cache_key "$scheme$request_uri";
>         add_header X-Cache-Status $upstream_cache_status;
>
>         # Standard proxy headers
>         proxy_set_header Host $host;
>         proxy_set_header X-Real-IP $remote_addr;
>         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
>         proxy_set_header X-Forwarded-Proto $scheme;
>     }
> }
> EOF
```

## Welcome to My Web Server

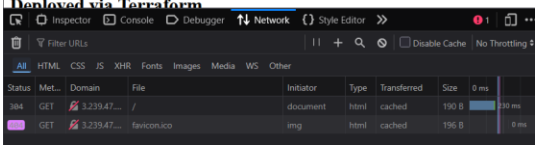
**Hostname: myapp-webserver**

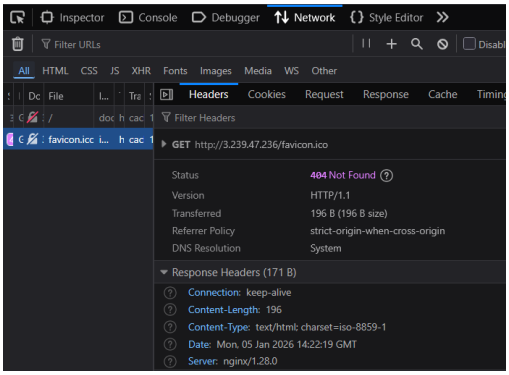
**Private IP: 10.0.10.149**

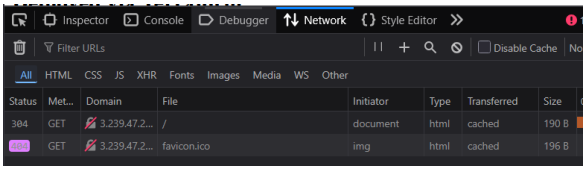
**Public IP: 44.193.15.179**

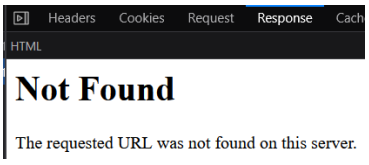
**Public DNS:**

**Deployed via Terraform**




```
[ec2-user@ip-10-0-10-113 ~]$ ls -la /var/cache/nginx/
total 0
drwxr-xr-x 3 nginx nginx 15 Jan  5 14:21 .
drwxr-xr-x 7 root  root  76 Jan  5 14:20 ..
drwx------ 3 nginx nginx 16 Jan  5 14:21 9
[ec2-user@ip-10-0-10-113 ~]$
```

```
Destroy complete! Resources: 13 destroyed.
@SanaTariq205 ~/Lab12 $
```

```
@SanaTariq205 ~/Lab12 $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 145,
  "lineage": "acc83830-78d8-bbf0-28bb-2f6e6bf018a2",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@SanaTariq205 ~/Lab12 $
```