Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

Cloud Computing

Assignment 2

Name: Nayab Khazin

Registration number: 2023-BSE-046

Submitted to: Sir Waqas Saleem

Semester: 5-B

Contents

1. Summary

In the assignment, focus is placed on the design of the multi-tier web environment in the Amazon Web Services (AWS) with the best practices of Infrastructure as Code (IaC) implemented through Terraform. The objective was to create a scalable secure highly available web

The system design involved an architecture using an Nginx reverse proxy which acted as a load balancer and multiple web servers in the back end.

The cloud infrastructure that has been deployed incorporates Virtual Private Cloud infrastructure, public subnets, security groups, EC2 instance configurations for both Nginx and Apache servers, as well as scripts to automatically configure the servers. Reusability and ease of maintenance are achieved through the implementation of modules from the Terraform tool in these setups.

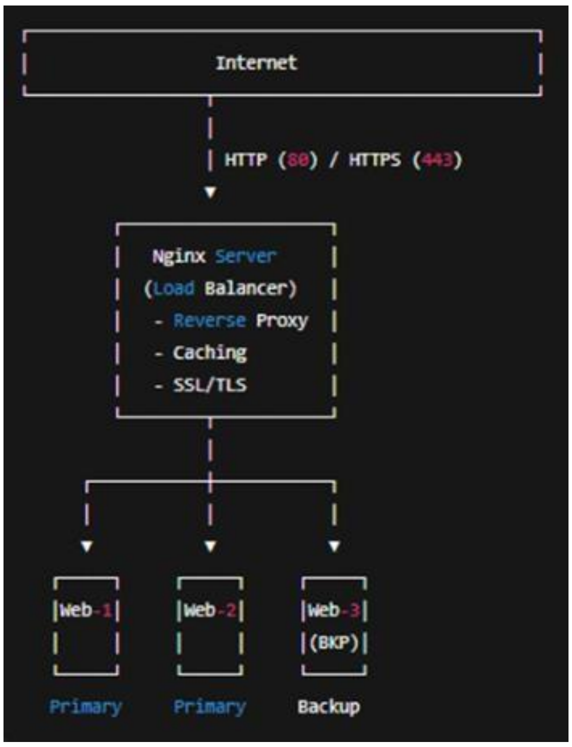
Some key achievements on this project include implementing load balancing, caching, availability with a backup server, secure access with security groups, automated provisioning with Terraform, and resource cleanup following testing.

2. Architecture Design

2.1 Architecture Overview

This system is developed on Amazon Web Services using Terraform. In this system, it uses a three-tier architecture. That is for providing security. The processing of requests from the clients on the internet takes place through an Nginx server that handles reverse proxying, load balancing, SSL/TLS terminating, and cache management.

The Nginx server receives the incoming traffic and handles the requests by forwarding the requests to several Apache backend servers. The two servers act as the master servers, while the remaining server is configured as the standby server. The method aids in optimizing the efficiency and preventing the backend servers from being routed to the internet.



Architecture Diagram

This is evident from the design diagram where traffic is coming from the Internet through HTTP (80) or HTTPS (443) to NginxLoadBalancer. NginxLoadBalancer will then split this traffic to several back-end Apache Web Servers.
Web-1 and Web-2 act as backup servers in this case, while Web-3 is the backup node in the system. There is an indication of redundancy in routing requests in the system.

Component Descriptions

Nginx Server: Serves as a reverse proxy and load balancer. It provides functionality to encrypt and decrypt data with SSL/TLS, handles cache, and supports load balancing.
Web-1 and Web-2 (Primary Servers): Apache servers managing and serving content from applications. A number can be provided to increase availability and scalability.
Web-3 (Backup Server): To be used in cases where primary servers are down or overloaded.
VPC & Subnets: Ensure an isolated networking environment in AWS for all resources.
Security Groups: Define and manage the flow of incoming and outgoing traffic among system components.

Network Topology

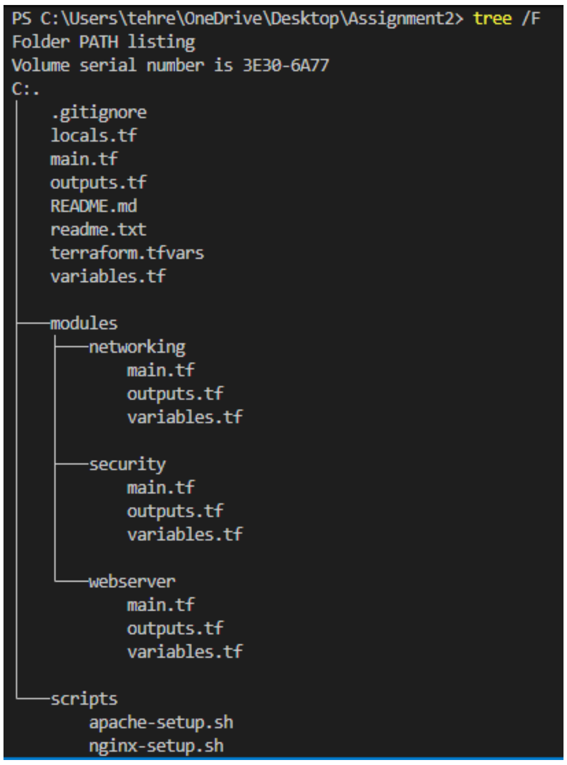
EC2 instances run on a dedicated customized VPC. The running server of Nginx is located in a public subnet and receives traffic from the internet. The Backend Apache servers are secure; the servers only receive traffic from the Nginx servers as defined in the security groups. Such architecture patterns improve the security of the networks.
Designing Security
Security implements the principle of least privilege principle. The Nginx web server enables both HTTPS as well as HTTP connections from the outside environment, but the remaining servers are not reachable from the outside environment. SSH connectivity is provided just by the authorized IP addresses. Security Groups determine the
Their design encompasses internal communication between the modules, as well as encryption in SSL/TLS to safeguard client information.

Implementation Details

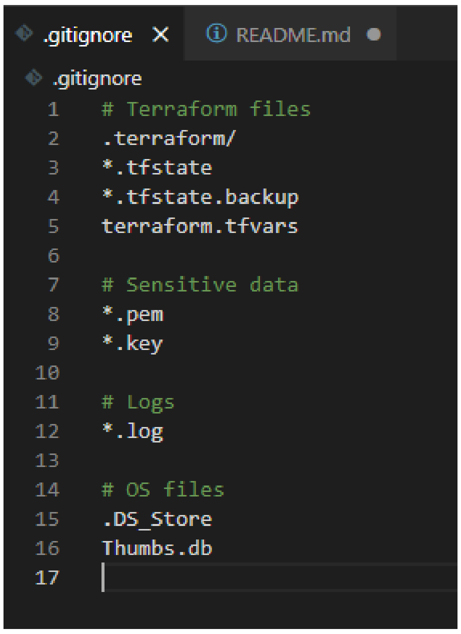
Part 1: Infrastructure Setup

Terraform was used to define the complete AWS infrastructure. The project follows a modular structure with separate files for networking, compute resources, variables, outputs, and locals. Variables allow flexible configuration without modifying core code.

1.1

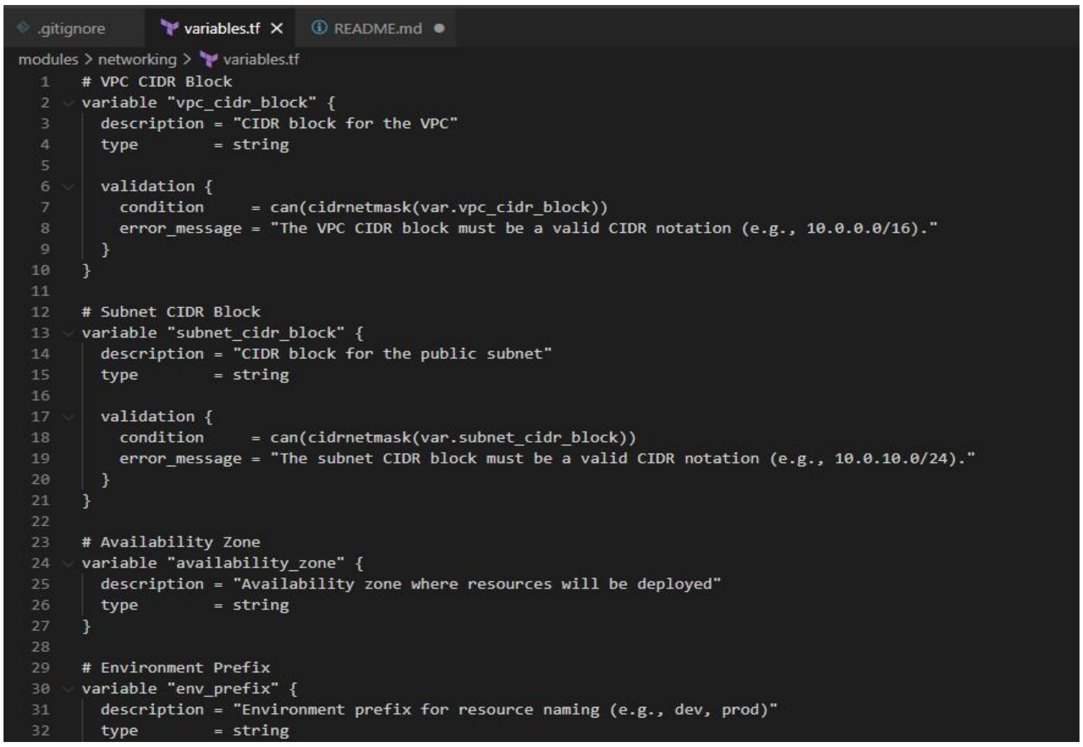


```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> tree /F
Folder PATH listing
Volume serial number is 3E30-6A77
C:.
│   .gitignore
│   locals.tf
│   main.tf
│   outputs.tf
│   README.md
│   readme.txt
│   terraform.tfvars
│   variables.tf
│
├───modules
│   ├───networking
│   │       main.tf
│   │       outputs.tf
│   │       variables.tf
│   │
│   ├───security
│   │       main.tf
│   │       outputs.tf
│   │       variables.tf
│   │
│   └───webserver
│           main.tf
│           outputs.tf
│           variables.tf
│
└───scripts
        apache-setup.sh
        nginx-setup.sh
```

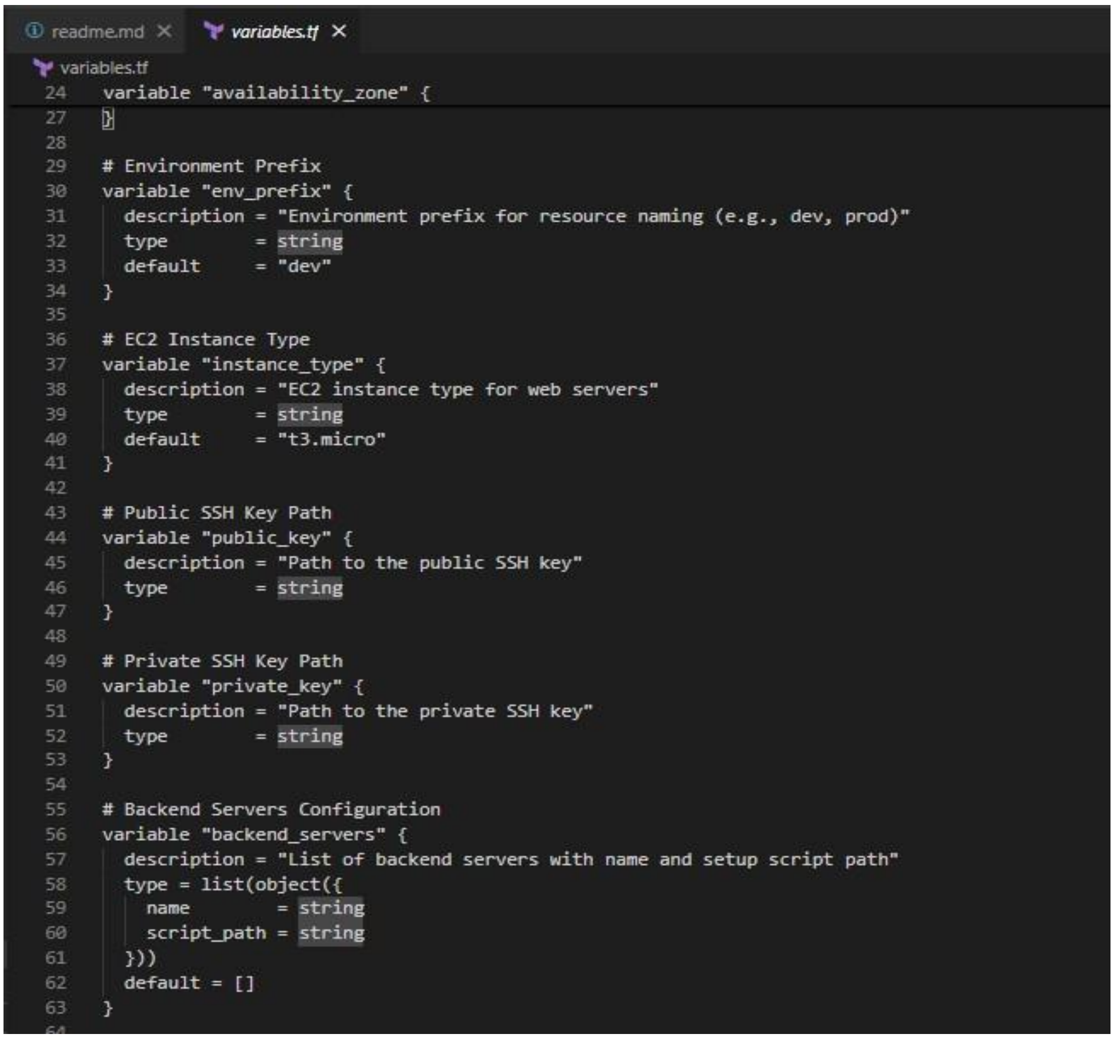


```
# Terraform files
.terraform/
*.tfstate
*.tfstate.backup
terraform.tfvars

# Sensitive data
*.pem
*.key

# Logs
*.log

# OS files
.DS_Store
Thumbs.db

```

1.2



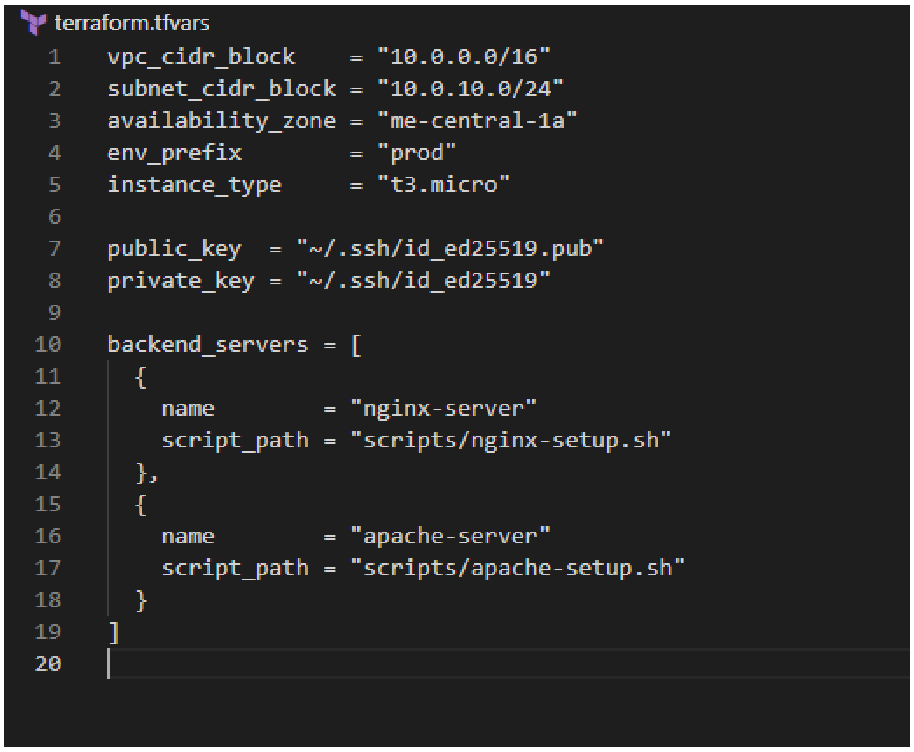
```
# VPC CIDR Block
variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
  type        = string

  validation {
    condition     = can(cidrnetmask(var.vpc_cidr_block))
    error_message = "The VPC CIDR block must be a valid CIDR notation (e.g., 10.0.0.0/16)."
  }
}

# Subnet CIDR Block
variable "subnet_cidr_block" {
  description = "CIDR block for the public subnet"
  type        = string

  validation {
    condition     = can(cidrnetmask(var.subnet_cidr_block))
    error_message = "The subnet CIDR block must be a valid CIDR notation (e.g., 10.0.10.0/24)."
  }
}

# Availability Zone
variable "availability_zone" {
  description = "Availability zone where resources will be deployed"
  type        = string
}

# Environment Prefix
variable "env_prefix" {
  description = "Environment prefix for resource naming (e.g., dev, prod)"
  type        = string
```



readme.md × variables.tf ×

variables.tf

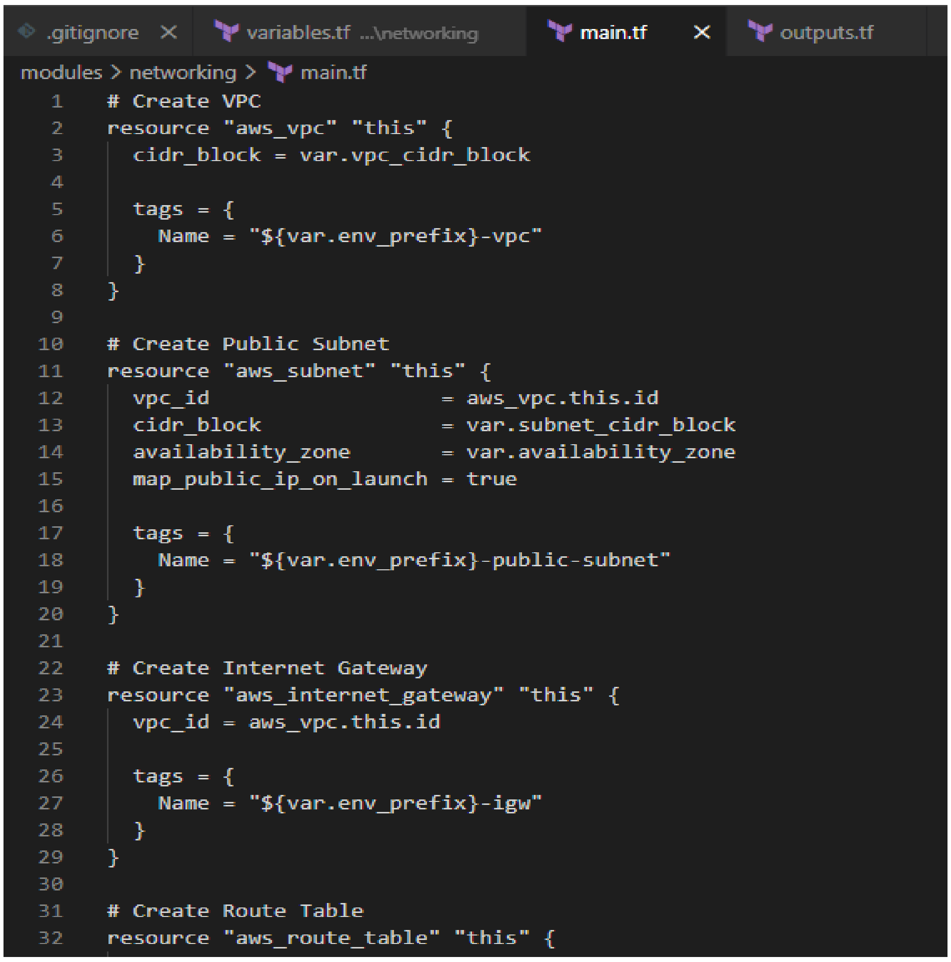
```
variable "availability_zone" {
}

# Environment Prefix
variable "env_prefix" {
  description = "Environment prefix for resource naming (e.g., dev, prod)"
  type        = string
  default     = "dev"
}

# EC2 Instance Type
variable "instance_type" {
  description = "EC2 instance type for web servers"
  type        = string
  default     = "t3.micro"
}

# Public SSH Key Path
variable "public_key" {
  description = "Path to the public SSH key"
  type        = string
}

# Private SSH Key Path
variable "private_key" {
  description = "Path to the private SSH key"
  type        = string
}

# Backend Servers Configuration
variable "backend_servers" {
  description = "List of backend servers with name and setup script path"
  type = list(object({
    name        = string
    script_path = string
  }))
  default = []
}
```



terraform.tfvars

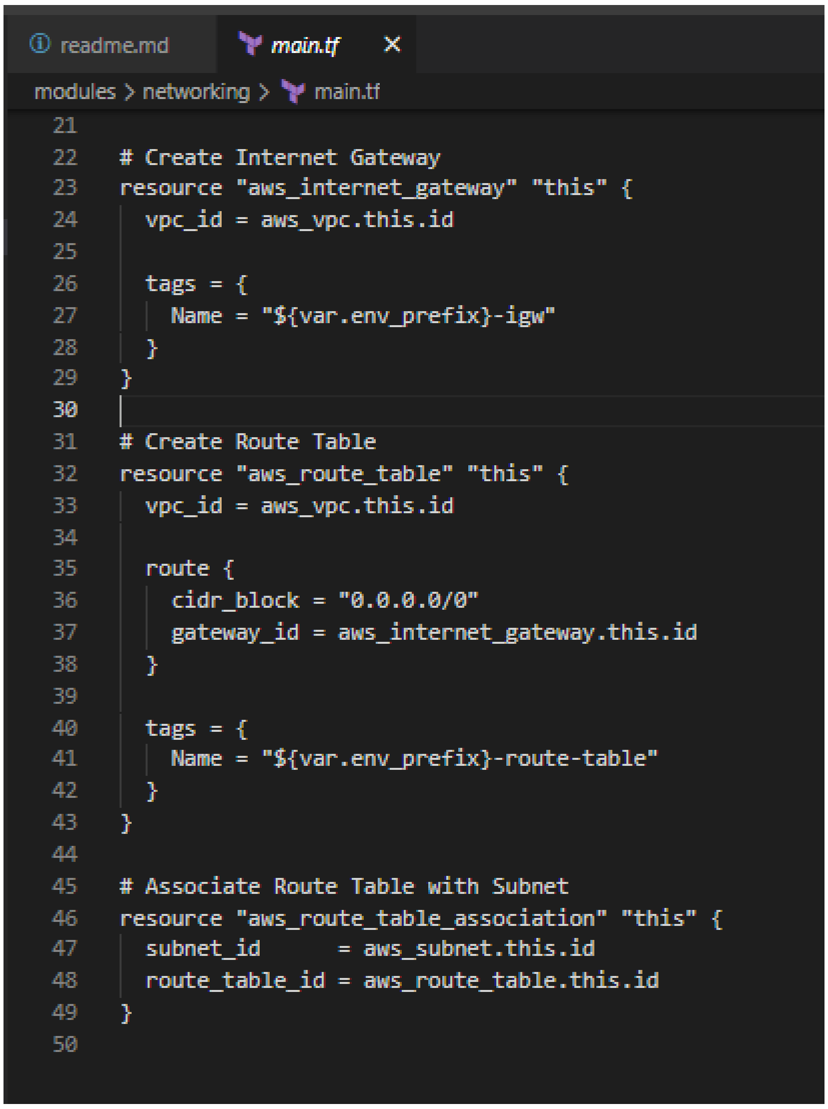
```
vpc_cidr_block    = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix        = "prod"
instance_type     = "t3.micro"

public_key  = "~/.ssh/id_ed25519.pub"
private_key = "~/.ssh/id_ed25519"

backend_servers = [
  {
    name        = "nginx-server"
    script_path = "scripts/nginx-setup.sh"
  },
  {
    name        = "apache-server"
    script_path = "scripts/apache-setup.sh"
  }
]

```

1.3


.gitignore | variables.tf ...\networking | main.tf | outputs.tf

modules > networking > main.tf

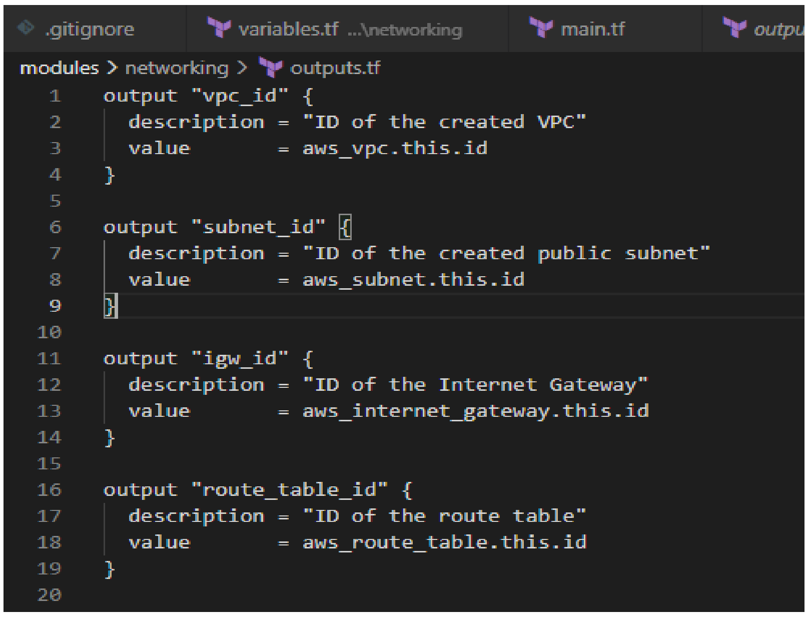
```
# Create VPC
resource "aws_vpc" "this" {
  cidr_block = var.vpc_cidr_block

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

# Create Public Subnet
resource "aws_subnet" "this" {
  vpc_id                  = aws_vpc.this.id
  cidr_block              = var.subnet_cidr_block
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true

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

# Create Internet Gateway
resource "aws_internet_gateway" "this" {
  vpc_id = aws_vpc.this.id

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

# Create Route Table
resource "aws_route_table" "this" {
```



readme.md   main.tf ×

modules > networking > main.tf

```hcl

# Create Internet Gateway
resource "aws_internet_gateway" "this" {
  vpc_id = aws_vpc.this.id

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

# Create Route Table
resource "aws_route_table" "this" {
  vpc_id = aws_vpc.this.id

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

  tags = {
    Name = "${var.env_prefix}-route-table"
  }
}

# Associate Route Table with Subnet
resource "aws_route_table_association" "this" {
  subnet_id      = aws_subnet.this.id
  route_table_id = aws_route_table.this.id
}

```

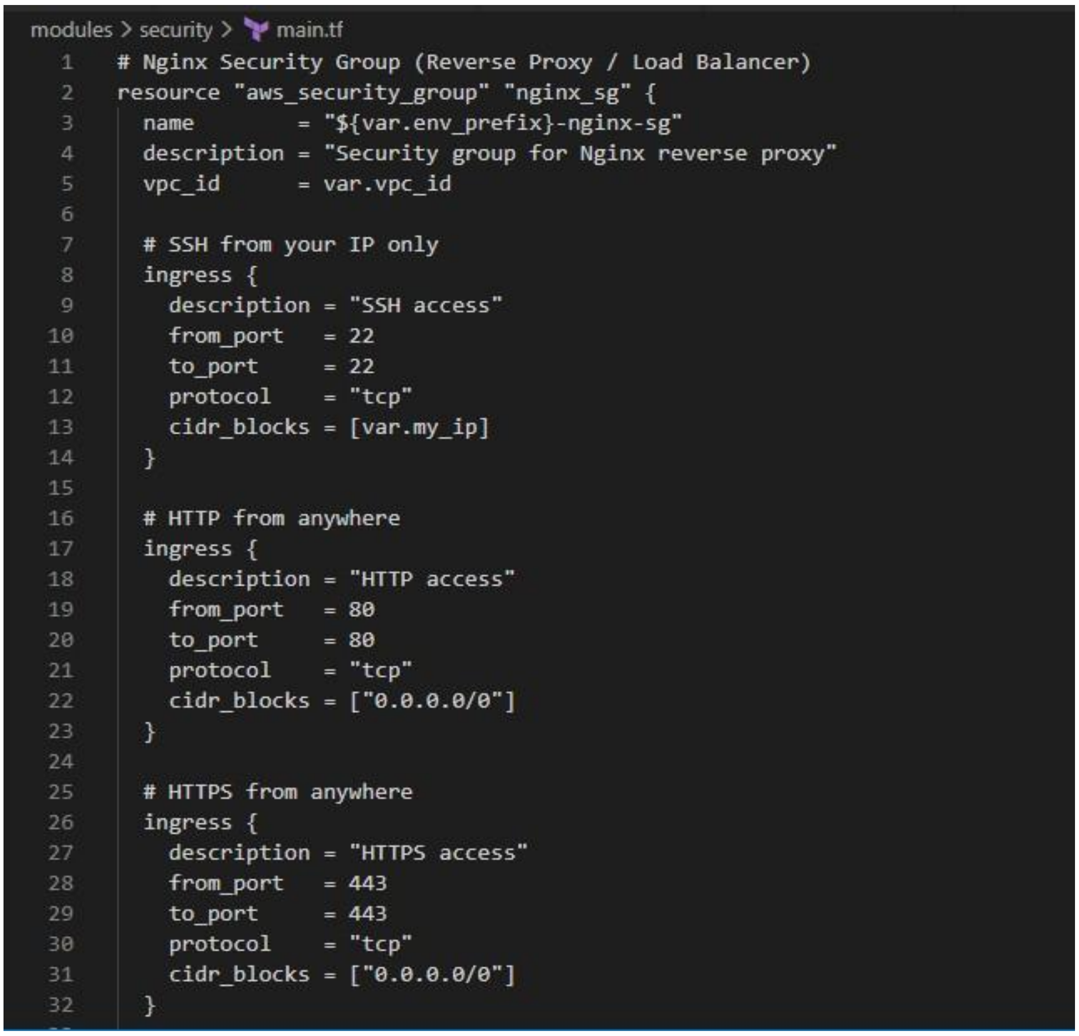


.gitignore   variables.tf ...\networking   main.tf   outpu

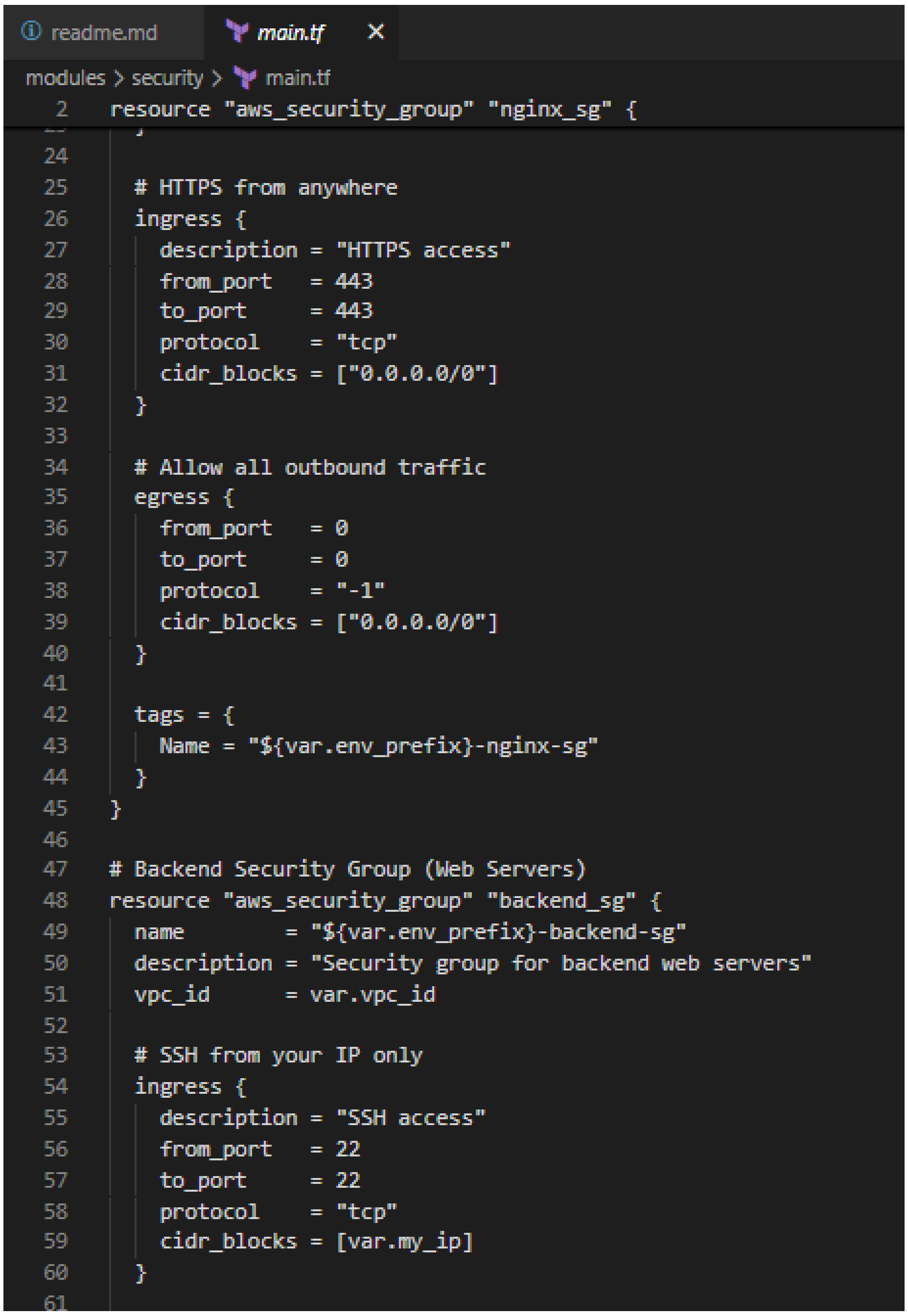
modules > networking > outputs.tf

```hcl
output "vpc_id" {
  description = "ID of the created VPC"
  value       = aws_vpc.this.id
}

output "subnet_id" {
  description = "ID of the created public subnet"
  value       = aws_subnet.this.id
}

output "igw_id" {
  description = "ID of the Internet Gateway"
  value       = aws_internet_gateway.this.id
}

output "route_table_id" {
  description = "ID of the route table"
  value       = aws_route_table.this.id
}

```

1.4



```
# Nginx Security Group (Reverse Proxy / Load Balancer)
resource "aws_security_group" "nginx_sg" {
  name        = "${var.env_prefix}-nginx-sg"
  description = "Security group for Nginx reverse proxy"
  vpc_id      = var.vpc_id

  # SSH from your IP only
  ingress {
    description = "SSH access"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
  }

  # HTTP from anywhere
  ingress {
    description = "HTTP access"
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # HTTPS from anywhere
  ingress {
    description = "HTTPS access"
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

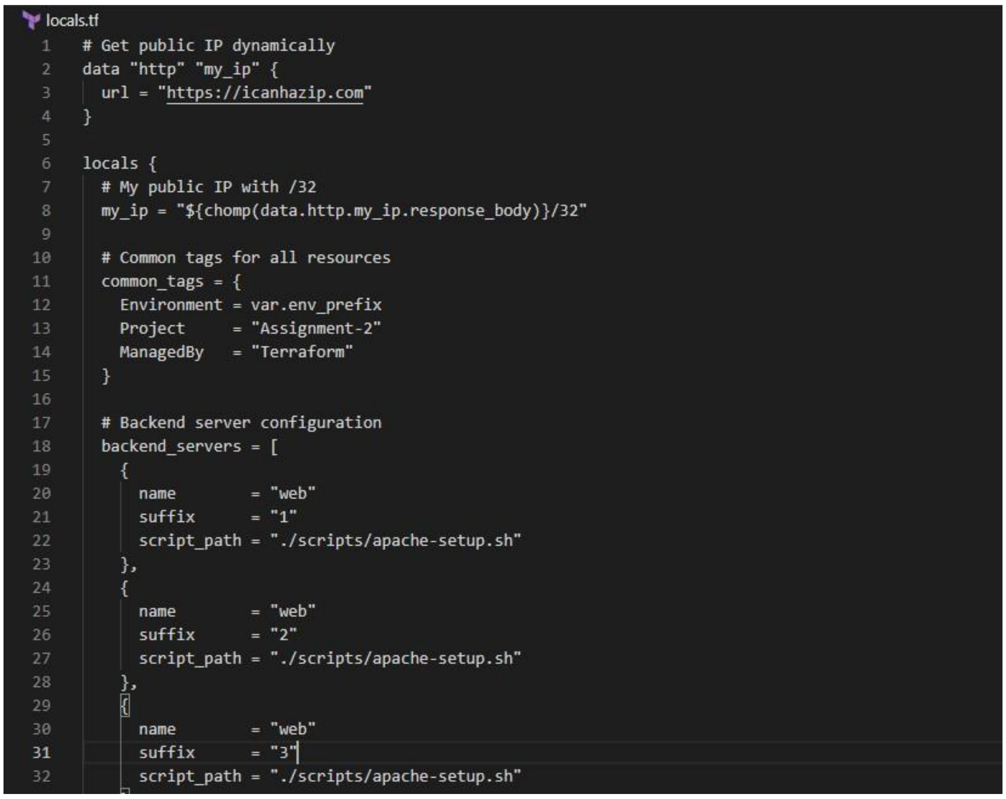


readme.md   main.tf

modules > security > main.tf

```
resource "aws_security_group" "nginx_sg" {

  # HTTPS from anywhere
  ingress {
    description = "HTTPS access"
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # Allow all outbound traffic
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

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

# Backend Security Group (Web Servers)
resource "aws_security_group" "backend_sg" {
  name        = "${var.env_prefix}-backend-sg"
  description = "Security group for backend web servers"
  vpc_id      = var.vpc_id

  # SSH from your IP only
  ingress {
    description = "SSH access"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [var.my_ip]
  }

```

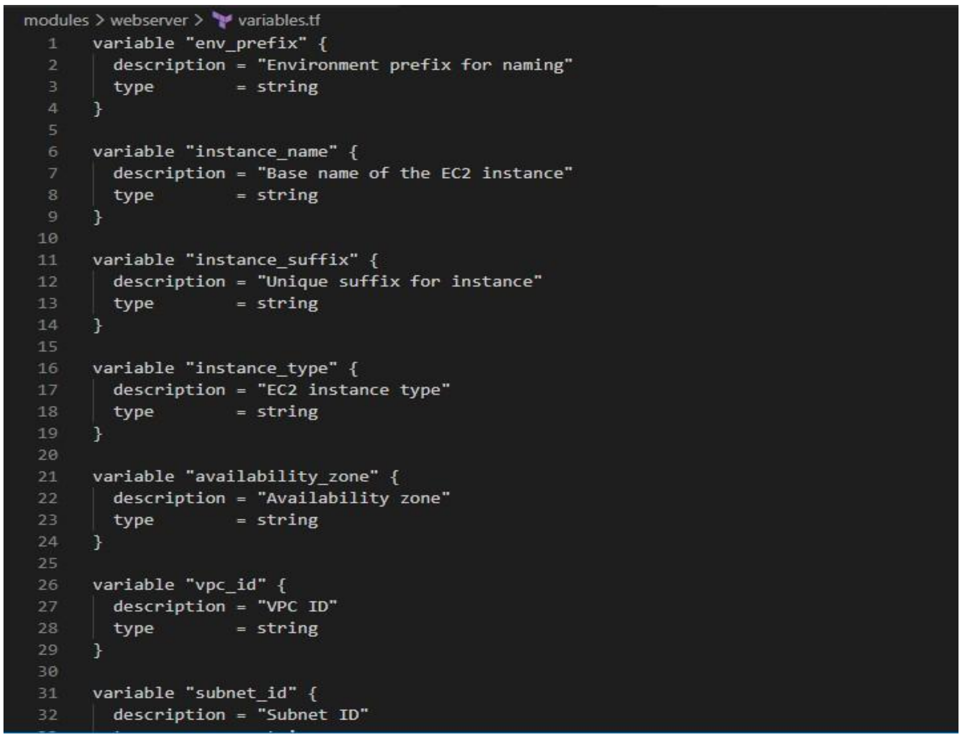
1.5

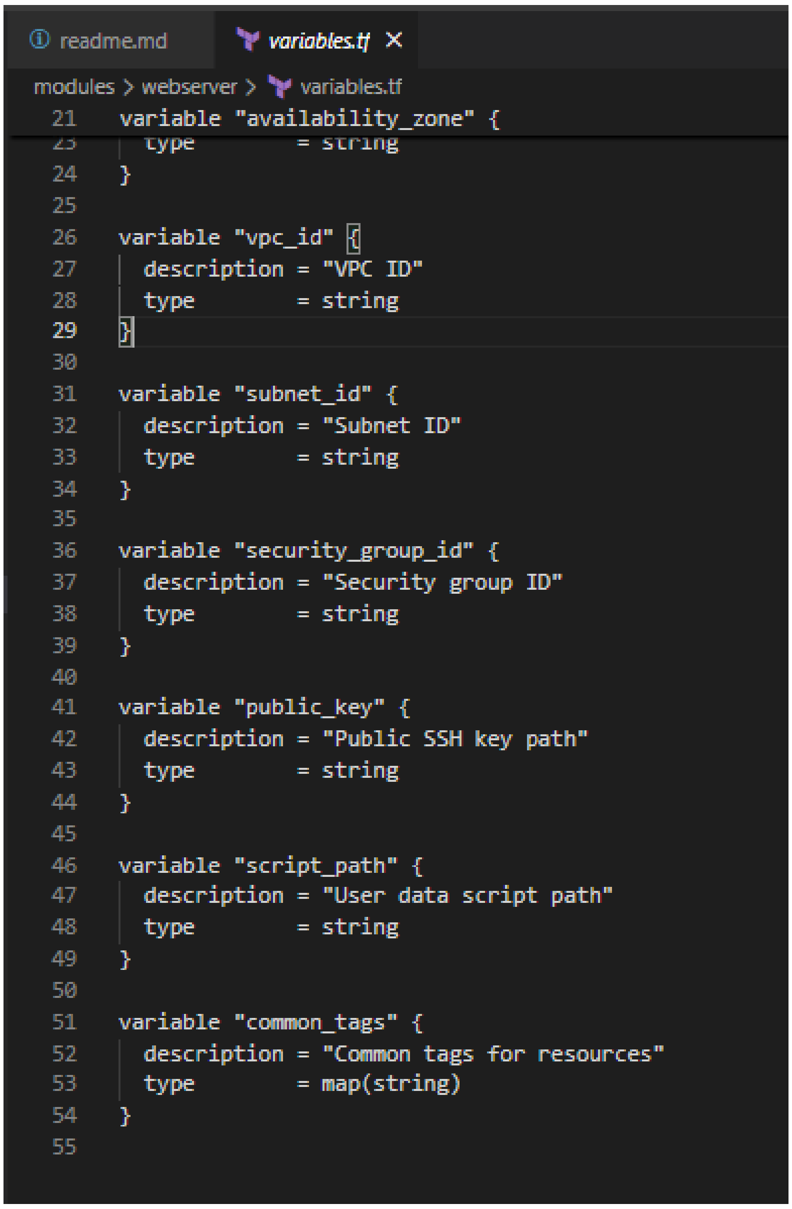

locals.tf
```
# Get public IP dynamically
data "http" "my_ip" {
  url = "https://icanhazip.com"
}

locals {
  # My public IP with /32
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"

  # Common tags for all resources
  common_tags = {
    Environment = var.env_prefix
    Project     = "Assignment-2"
    ManagedBy   = "Terraform"
  }

  # Backend server configuration
  backend_servers = [
    {
      name        = "web"
      suffix      = "1"
      script_path = "./scripts/apache-setup.sh"
    },
    {
      name        = "web"
      suffix      = "2"
      script_path = "./scripts/apache-setup.sh"
    },
    {
      name        = "web"
      suffix      = "3"
      script_path = "./scripts/apache-setup.sh"
```

Part 2: Webserver Module

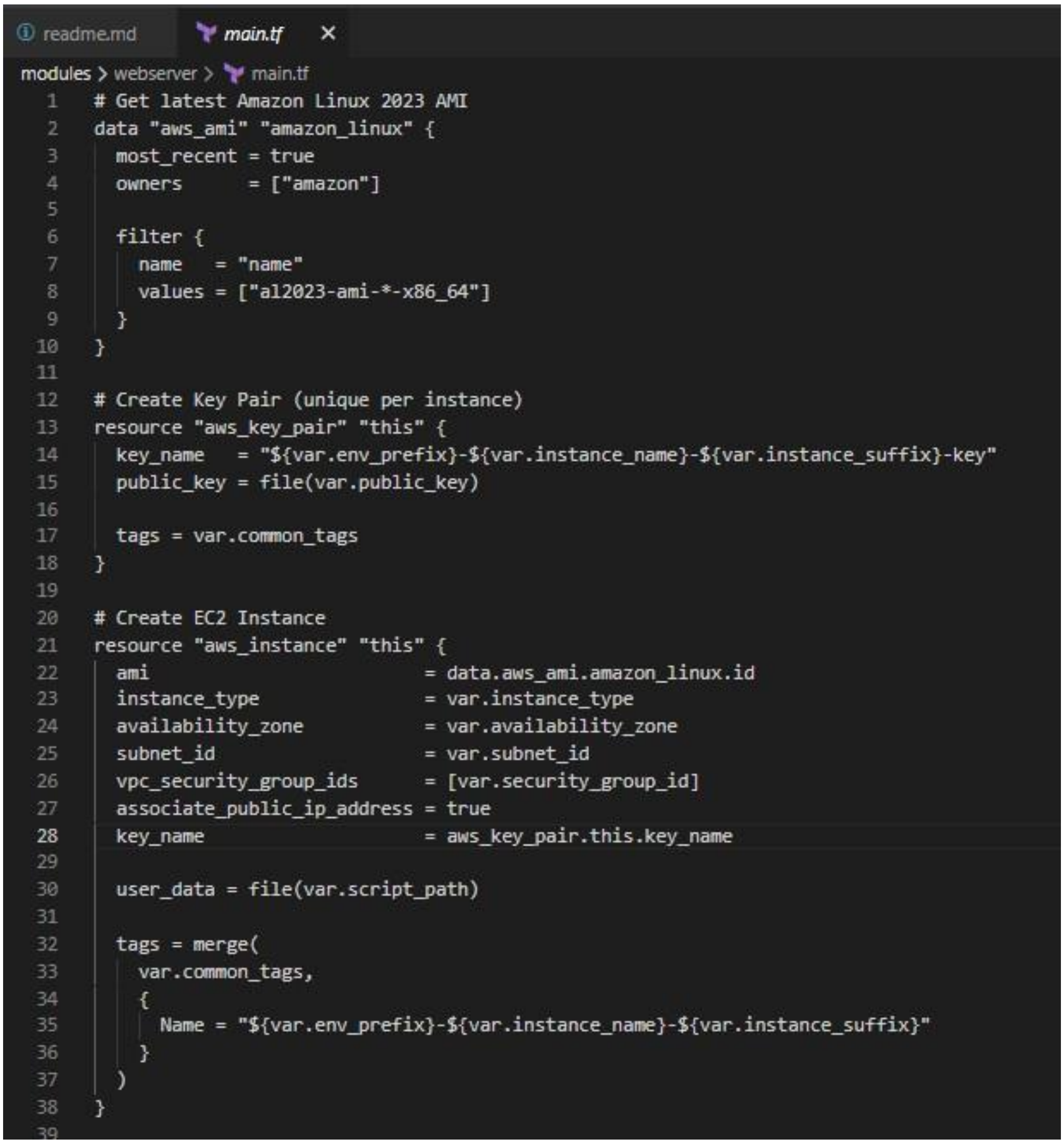
Reusable Terraform modules were created for web servers. Each module provisions an EC2 instance, attaches security groups, assigns key pairs, and executes user-data scripts for automatic configuration.

2.1

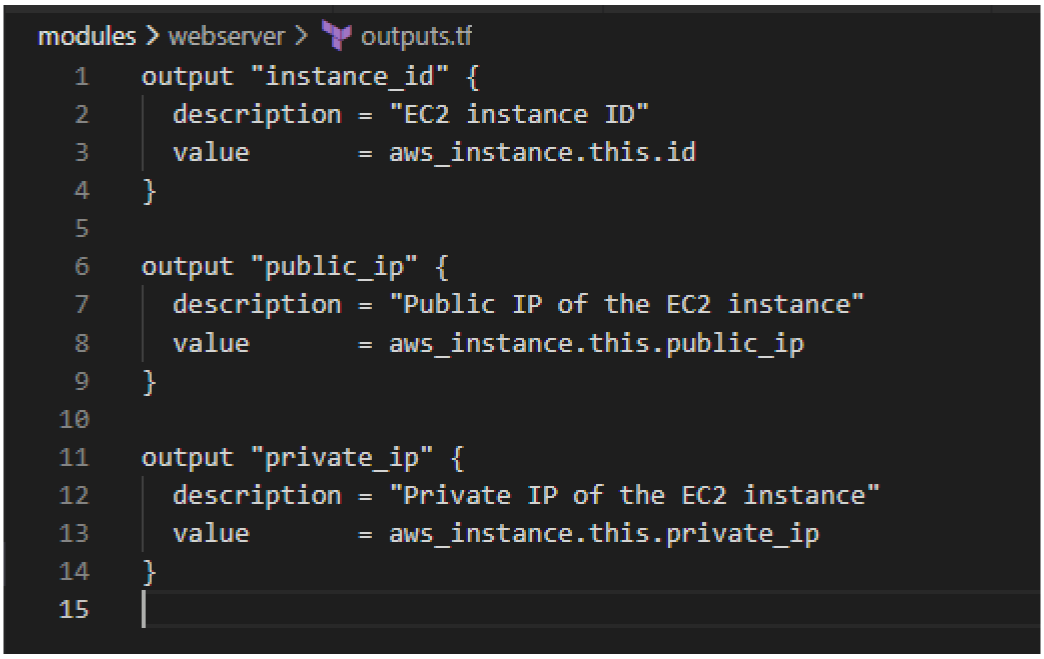

modules > webserver > variables.tf
```
variable "env_prefix" {
  description = "Environment prefix for naming"
  type        = string
}

variable "instance_name" {
  description = "Base name of the EC2 instance"
  type        = string
}

variable "instance_suffix" {
  description = "Unique suffix for instance"
  type        = string
}

variable "instance_type" {
  description = "EC2 instance type"
  type        = string
}

variable "availability_zone" {
  description = "Availability zone"
  type        = string
}

variable "vpc_id" {
  description = "VPC ID"
  type        = string
}

variable "subnet_id" {
  description = "Subnet ID"
```



```
variable "availability_zone" {
  type        = string
}

variable "vpc_id" {
  description = "VPC ID"
  type        = string
}

variable "subnet_id" {
  description = "Subnet ID"
  type        = string
}

variable "security_group_id" {
  description = "Security group ID"
  type        = string
}

variable "public_key" {
  description = "Public SSH key path"
  type        = string
}

variable "script_path" {
  description = "User data script path"
  type        = string
}

variable "common_tags" {
  description = "Common tags for resources"
  type        = map(string)
}
```



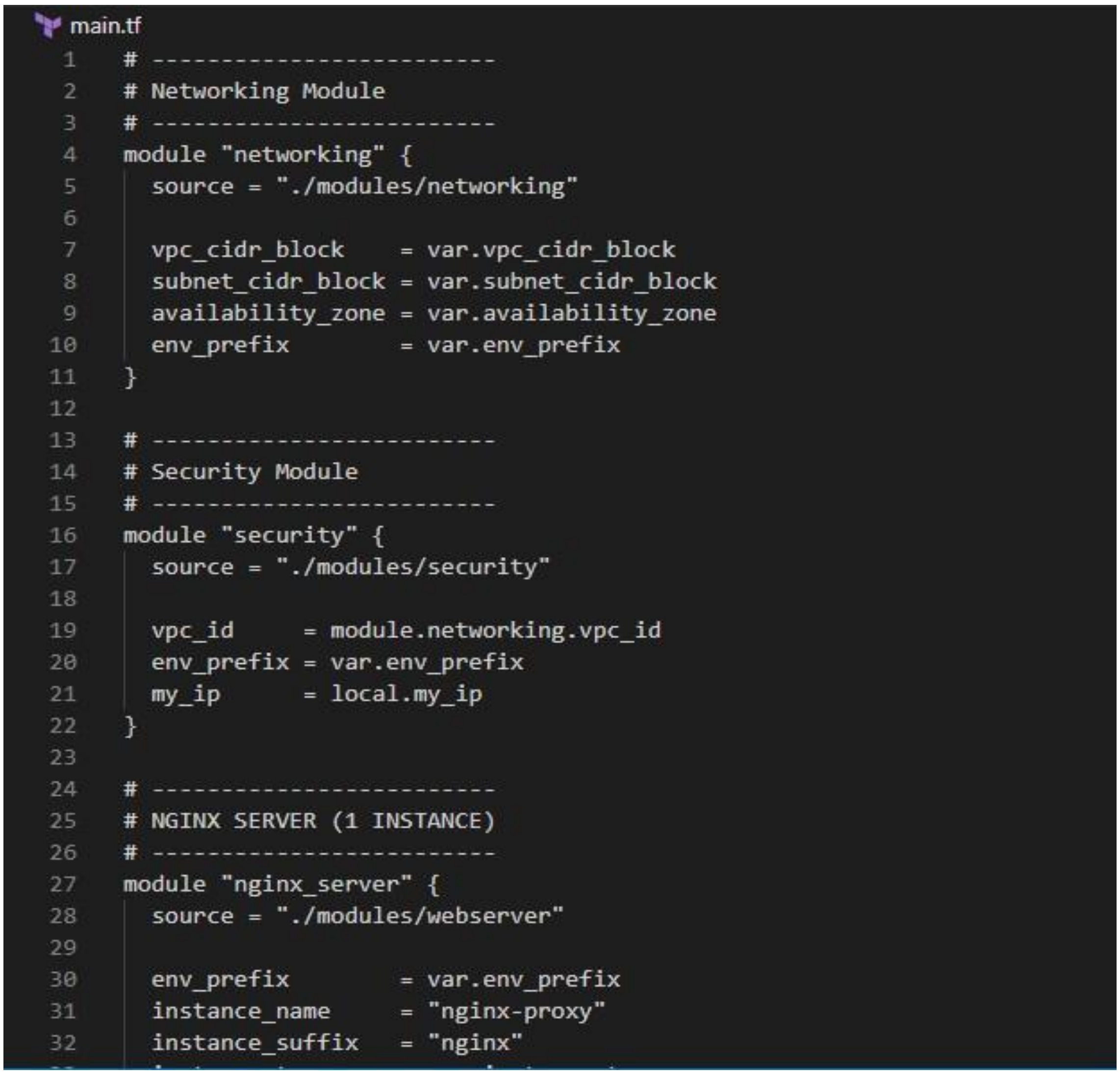
```hcl
# Get latest Amazon Linux 2023 AMI
data "aws_ami" "amazon_linux" {
  most_recent = true
  owners      = ["amazon"]

  filter {
    name   = "name"
    values = ["al2023-ami-*-x86_64"]
  }
}

# Create Key Pair (unique per instance)
resource "aws_key_pair" "this" {
  key_name   = "${var.env_prefix}-${var.instance_name}-${var.instance_suffix}-key"
  public_key = file(var.public_key)

  tags = var.common_tags
}

# Create EC2 Instance
resource "aws_instance" "this" {
  ami                         = data.aws_ami.amazon_linux.id
  instance_type               = var.instance_type
  availability_zone           = var.availability_zone
  subnet_id                   = var.subnet_id
  vpc_security_group_ids      = [var.security_group_id]
  associate_public_ip_address = true
  key_name                    = aws_key_pair.this.key_name

  user_data = file(var.script_path)

  tags = merge(
    var.common_tags,
    {
      Name = "${var.env_prefix}-${var.instance_name}-${var.instance_suffix}"
    }
  )
}
```



modules > webserver > outputs.tf

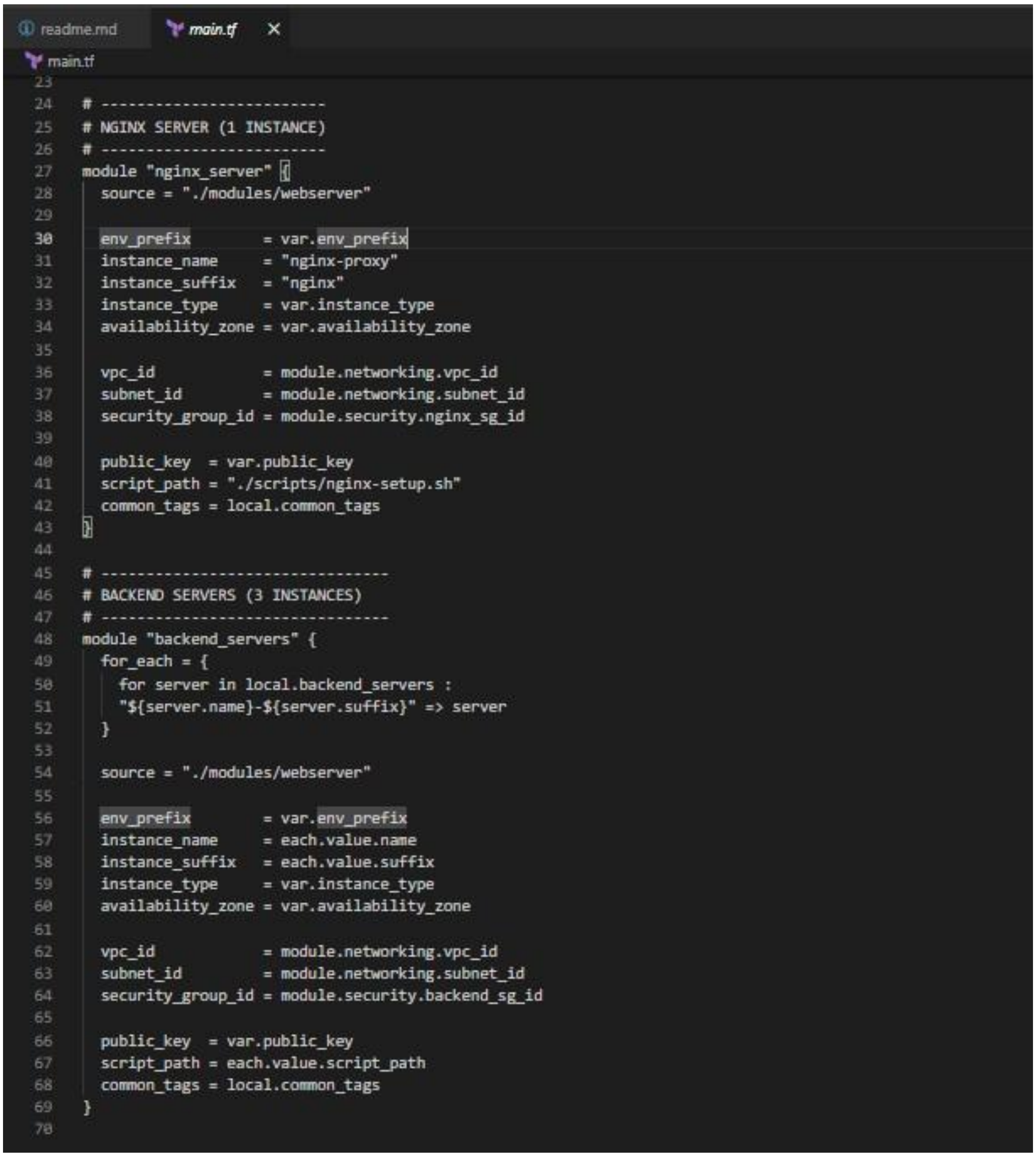
```
output "instance_id" {
  description = "EC2 instance ID"
  value       = aws_instance.this.id
}

output "public_ip" {
  description = "Public IP of the EC2 instance"
  value       = aws_instance.this.public_ip
}

output "private_ip" {
  description = "Private IP of the EC2 instance"
  value       = aws_instance.this.private_ip
}

```

2.2



main.tf

```
# -------------------------
# Networking Module
# -------------------------
module "networking" {
  source = "./modules/networking"

  vpc_cidr_block    = var.vpc_cidr_block
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix        = var.env_prefix
}

# -------------------------
# Security Module
# -------------------------
module "security" {
  source = "./modules/security"

  vpc_id     = module.networking.vpc_id
  env_prefix = var.env_prefix
  my_ip      = local.my_ip
}

# -------------------------
# NGINX SERVER (1 INSTANCE)
# -------------------------
module "nginx_server" {
  source = "./modules/webserver"

  env_prefix        = var.env_prefix
  instance_name     = "nginx-proxy"
  instance_suffix   = "nginx"
```



```
# -------------------------
# NGINX SERVER (1 INSTANCE)
# -------------------------
module "nginx_server" {
  source = "./modules/webserver"

  env_prefix        = var.env_prefix
  instance_name     = "nginx-proxy"
  instance_suffix   = "nginx"
  instance_type     = var.instance_type
  availability_zone = var.availability_zone

  vpc_id            = module.networking.vpc_id
  subnet_id         = module.networking.subnet_id
  security_group_id = module.security.nginx_sg_id

  public_key  = var.public_key
  script_path = "./scripts/nginx-setup.sh"
  common_tags = local.common_tags
}

# --------------------------------
# BACKEND SERVERS (3 INSTANCES)
# --------------------------------
module "backend_servers" {
  for_each = {
    for server in local.backend_servers :
    "${server.name}-${server.suffix}" => server
  }

  source = "./modules/webserver"

  env_prefix        = var.env_prefix
  instance_name     = each.value.name
  instance_suffix   = each.value.suffix
  instance_type     = var.instance_type
  availability_zone = var.availability_zone

  vpc_id            = module.networking.vpc_id
  subnet_id         = module.networking.subnet_id
  security_group_id = module.security.backend_sg_id

  public_key  = var.public_key
  script_path = each.value.script_path
  common_tags = local.common_tags
}
```

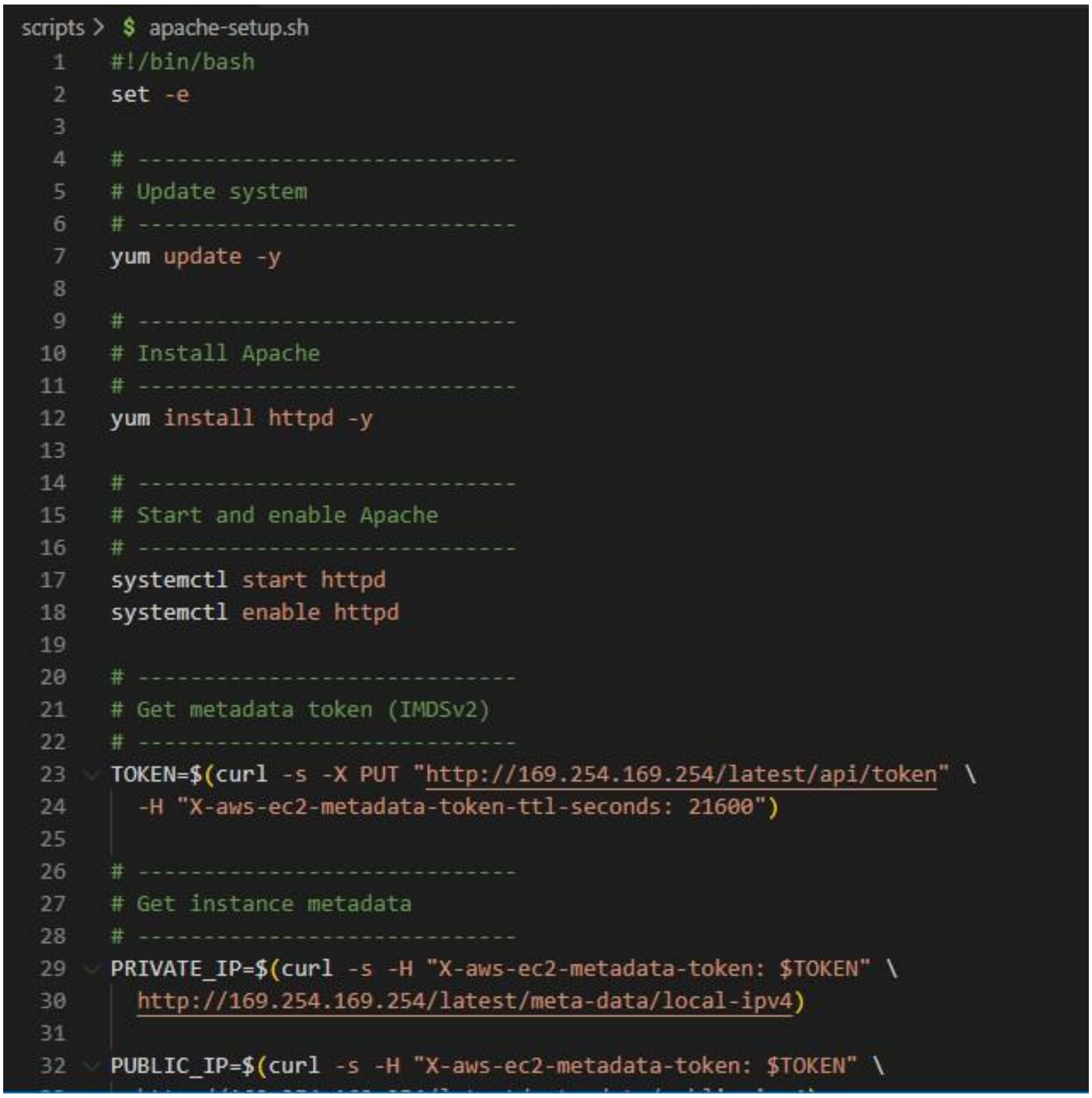
Part 3: Server Scripts

Shell scripts were used to configure servers automatically:

- **Apache Setup Script** installs Apache, starts the service, and deploys a sample web page.
- **Nginx Setup Script** configures reverse proxy rules, caching, and load balancing across backend servers.

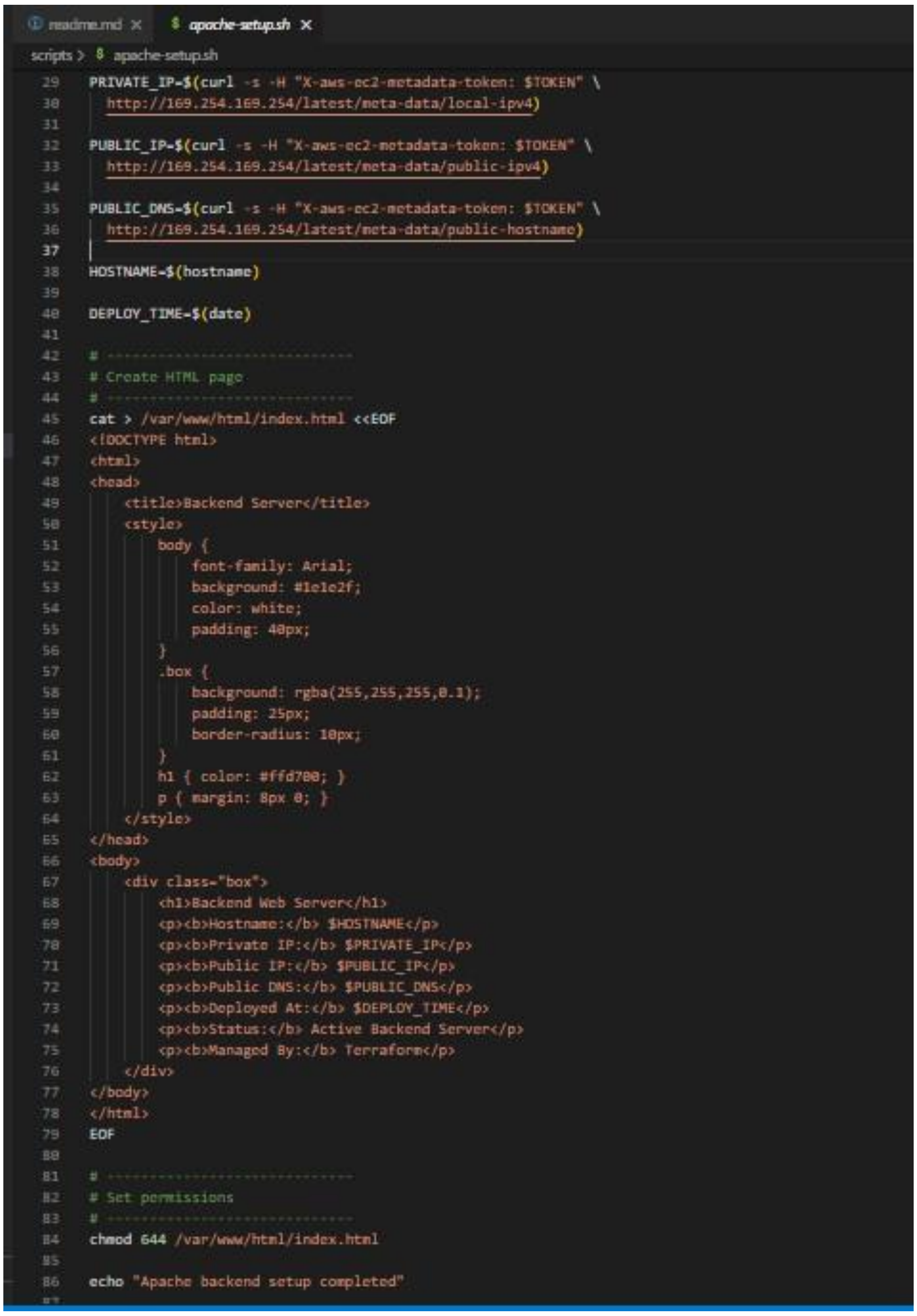
Code snippets were embedded in the scripts directory and executed during instance initialization.

3.1



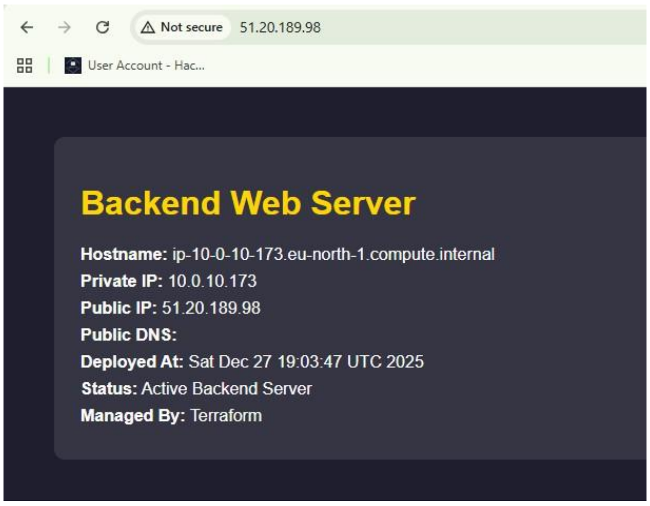
scripts > $ apache-setup.sh

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

# ------------------------------
# Update system
# ------------------------------
yum update -y

# ------------------------------
# Install Apache
# ------------------------------
yum install httpd -y

# ------------------------------
# Start and enable Apache
# ------------------------------
systemctl start httpd
systemctl enable httpd

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

# ------------------------------
# Get instance metadata
# ------------------------------
PRIVATE_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
  http://169.254.169.254/latest/meta-data/local-ipv4)

PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
```


readme.md × apache-setup.sh ×

scripts > apache-setup.sh

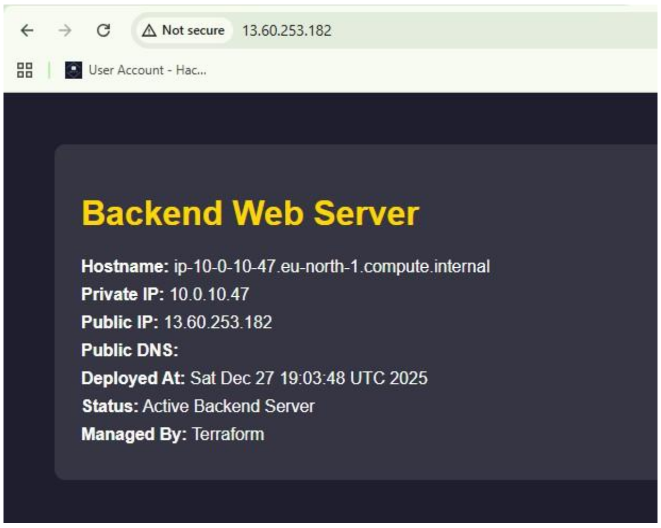
```bash
PRIVATE_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
  http://169.254.169.254/latest/meta-data/local-ipv4)

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

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

HOSTNAME=$(hostname)

DEPLOY_TIME=$(date)

# ------------------------------
# Create HTML page
# ------------------------------
cat > /var/www/html/index.html <<EOF
<!DOCTYPE html>
<html>
<head>
    <title>Backend Server</title>
    <style>
        body {
            font-family: Arial;
            background: #1e1e2f;
            color: white;
            padding: 40px;
        }
        .box {
            background: rgba(255,255,255,0.1);
            padding: 25px;
            border-radius: 10px;
        }
        h1 { color: #ffd700; }
        p { margin: 8px 0; }
    </style>
</head>
<body>
    <div class="box">
        <h1>Backend Web Server</h1>
        <p><b>Hostname:</b> $HOSTNAME</p>
        <p><b>Private IP:</b> $PRIVATE_IP</p>
        <p><b>Public IP:</b> $PUBLIC_IP</p>
        <p><b>Public DNS:</b> $PUBLIC_DNS</p>
        <p><b>Deployed At:</b> $DEPLOY_TIME</p>
        <p><b>Status:</b> Active Backend Server</p>
        <p><b>Managed By:</b> Terraform</p>
    </div>
</body>
</html>
EOF

# ------------------------------
# Set permissions
# ------------------------------
chmod 644 /var/www/html/index.html

echo "Apache backend setup completed"
```


Not secure 51.20.189.98

User Account - Hac...

Backend Web Server

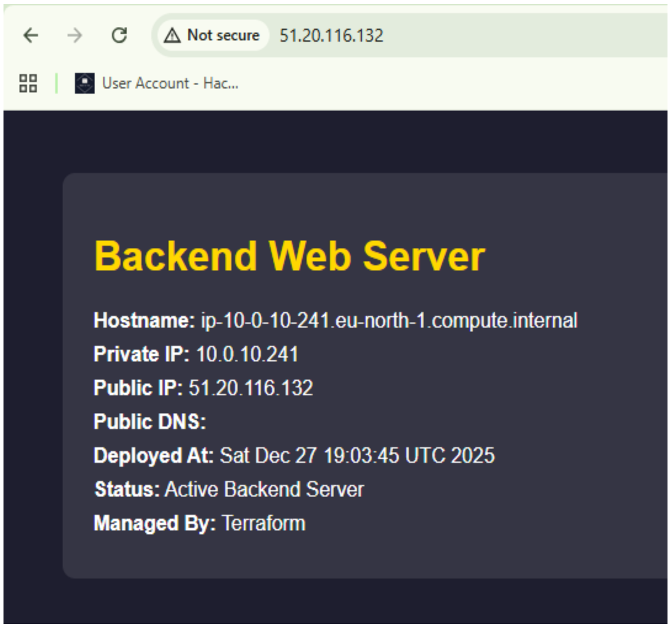
Hostname: ip-10-0-10-173.eu-north-1.compute.internal
Private IP: 10.0.10.173
Public IP: 51.20.189.98
Public DNS:
Deployed At: Sat Dec 27 19:03:47 UTC 2025
Status: Active Backend Server
Managed By: Terraform


Not secure 13.60.253.182

User Account - Hac...

Backend Web Server

Hostname: ip-10-0-10-47.eu-north-1.compute.internal
Private IP: 10.0.10.47
Public IP: 13.60.253.182
Public DNS:
Deployed At: Sat Dec 27 19:03:48 UTC 2025
Status: Active Backend Server
Managed By: Terraform

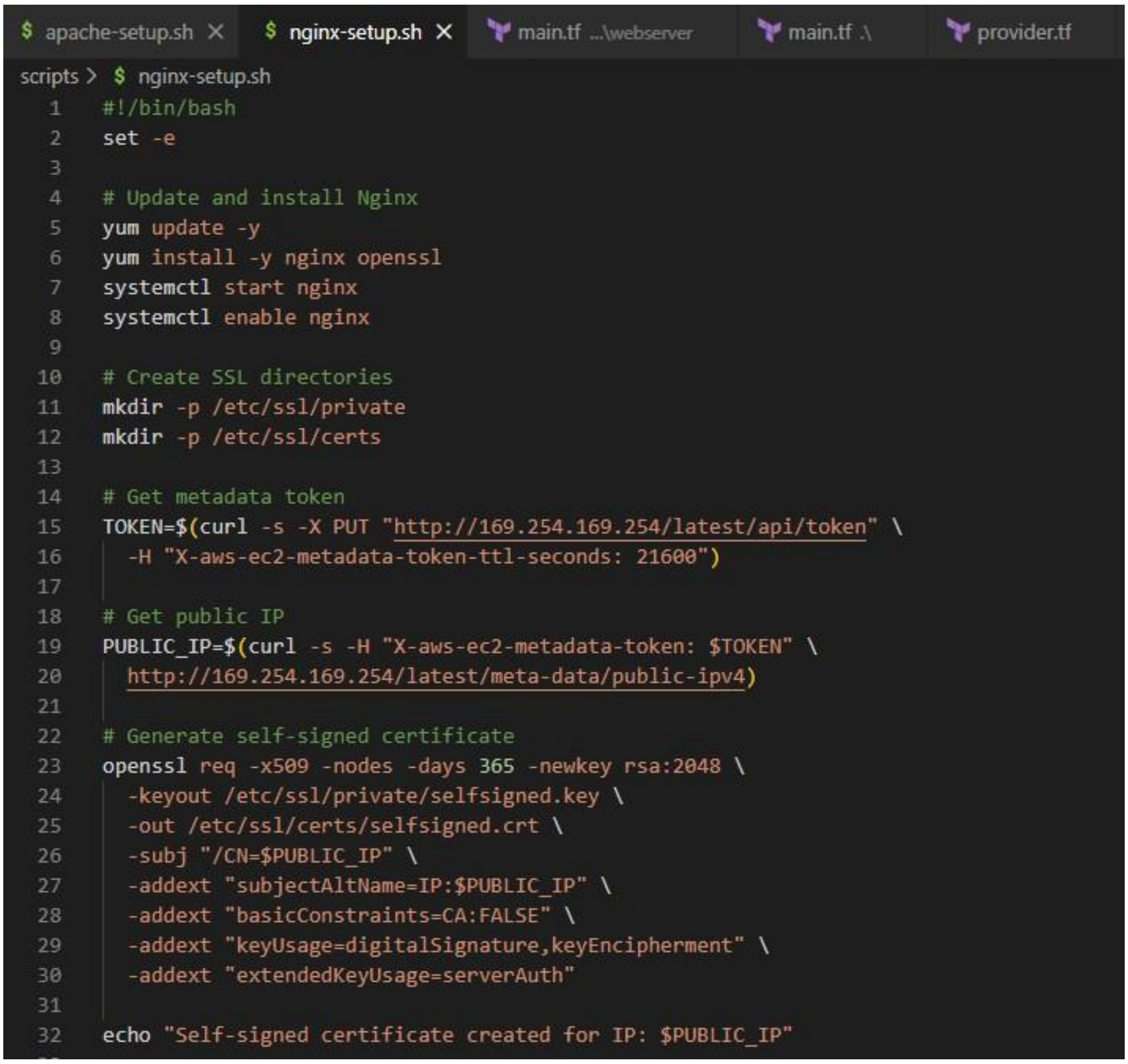

Not secure 51.20.116.132

User Account - Hac...

Backend Web Server

Hostname: ip-10-0-10-241.eu-north-1.compute.internal
Private IP: 10.0.10.241
Public IP: 51.20.116.132
Public DNS:
Deployed At: Sat Dec 27 19:03:45 UTC 2025
Status: Active Backend Server
Managed By: Terraform

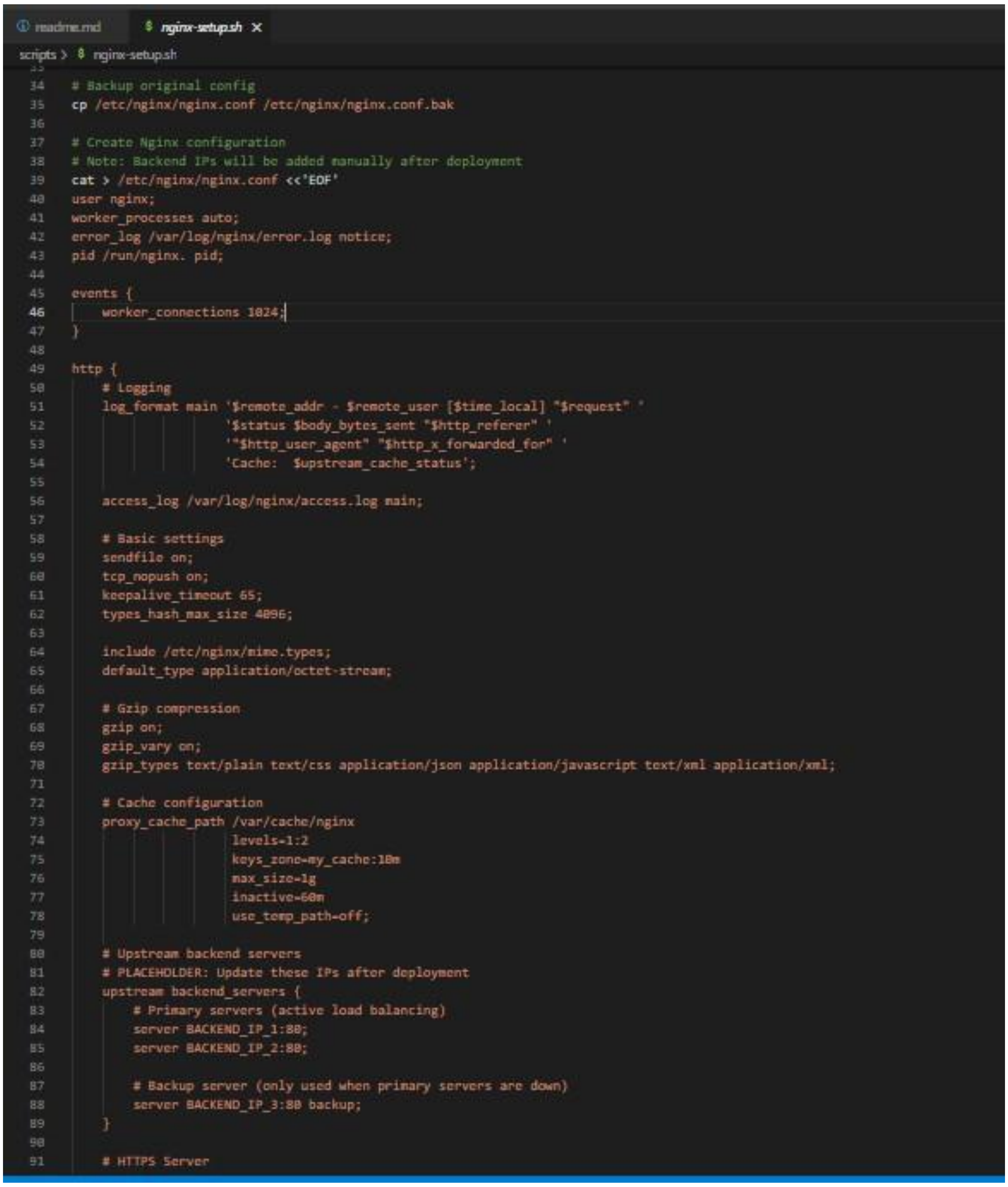
3.2



apache-setup.sh | nginx-setup.sh | main.tf ...\webserver | main.tf .\ | provider.tf

scripts > nginx-setup.sh

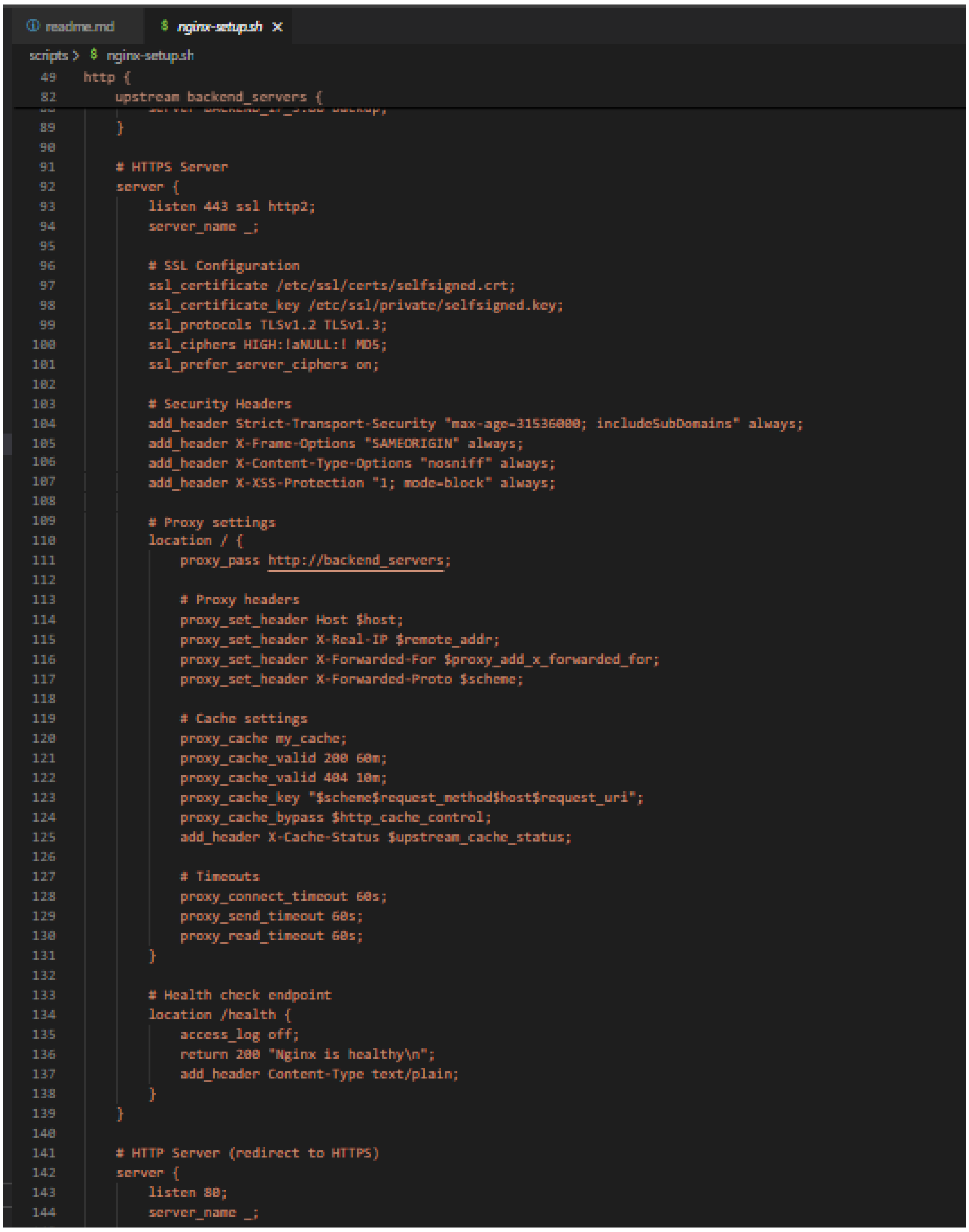
```bash
#!/bin/bash
set -e

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

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

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

# Get public IP
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"
```



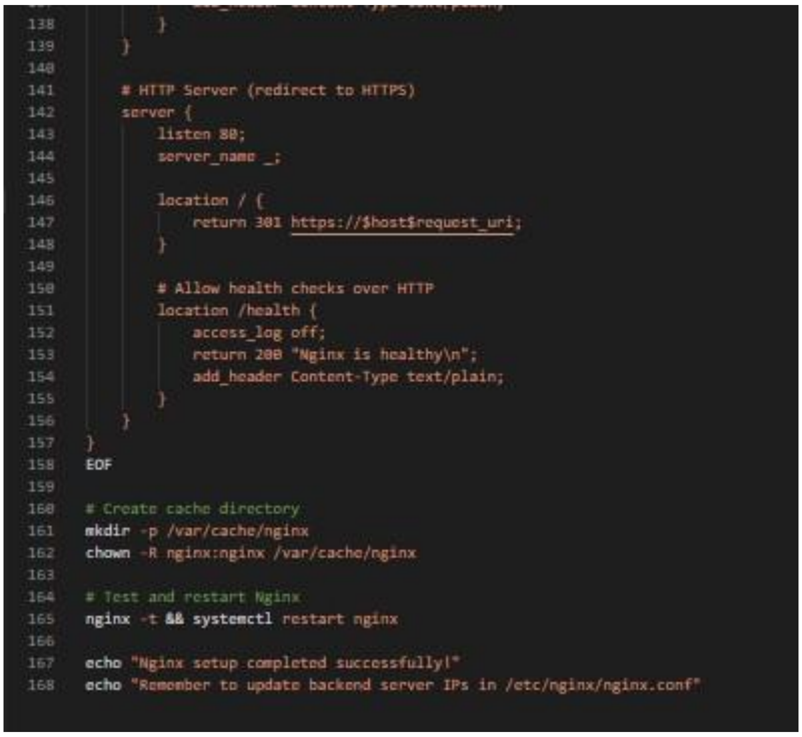
readme.md   nginx-setup.sh

scripts > nginx-setup.sh

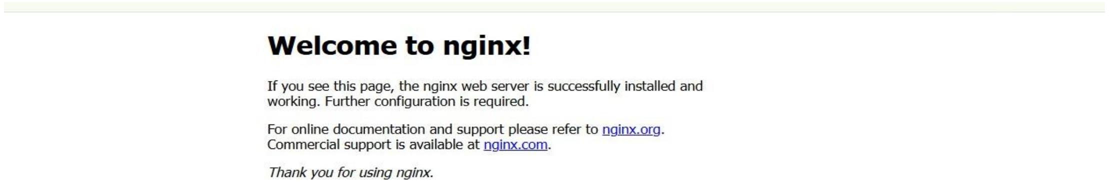
```
# Backup original config
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak

# Create Nginx configuration
# Note: Backend IPs will be added manually after deployment
cat > /etc/nginx/nginx.conf <<'EOF'
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx. pid;

events {
    worker_connections 1024;
}

http {
    # Logging
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for" '
                    'Cache:  $upstream_cache_status';

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

    # Basic settings
    sendfile on;
    tcp_nopush on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

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

    # Gzip compression
    gzip on;
    gzip_vary on;
    gzip_types text/plain text/css application/json application/javascript text/xml application/xml;

    # Cache configuration
    proxy_cache_path /var/cache/nginx
                     levels=1:2
                     keys_zone=my_cache:10m
                     max_size=1g
                     inactive=60m
                     use_temp_path=off;

    # Upstream backend servers
    # PLACEHOLDER: Update these IPs after deployment
    upstream backend_servers {
        # Primary servers (active load balancing)
        server BACKEND_IP_1:80;
        server BACKEND_IP_2:80;

        # Backup server (only used when primary servers are down)
        server BACKEND_IP_3:80 backup;
    }

    # HTTPS Server
```



```
http {
    upstream backend_servers {
    }

    # HTTPS Server
    server {
        listen 443 ssl http2;
        server_name _;

        # SSL Configuration
        ssl_certificate /etc/ssl/certs/selfsigned.crt;
        ssl_certificate_key /etc/ssl/private/selfsigned.key;
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;

        # Security Headers
        add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
        add_header X-Frame-Options "SAMEORIGIN" always;
        add_header X-Content-Type-Options "nosniff" always;
        add_header X-XSS-Protection "1; mode=block" always;

        # Proxy settings
        location / {
            proxy_pass http://backend_servers;

            # 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;

            # Cache settings
            proxy_cache my_cache;
            proxy_cache_valid 200 60m;
            proxy_cache_valid 404 10m;
            proxy_cache_key "$scheme$request_method$host$request_uri";
            proxy_cache_bypass $http_cache_control;
            add_header X-Cache-Status $upstream_cache_status;

            # Timeouts
            proxy_connect_timeout 60s;
            proxy_send_timeout 60s;
            proxy_read_timeout 60s;
        }

        # Health check endpoint
        location /health {
            access_log off;
            return 200 "Nginx is healthy\n";
            add_header Content-Type text/plain;
        }
    }

    # HTTP Server (redirect to HTTPS)
    server {
        listen 80;
        server_name _;
```



```
        }
    }

    # HTTP Server (redirect to HTTPS)
    server {
        listen 80;
        server_name _;

        location / {
            return 301 https://$host$request_uri;
        }

        # Allow health checks over HTTP
        location /health {
            access_log off;
            return 200 "Nginx is healthy\n";
            add_header Content-Type text/plain;
        }
    }
}
EOF

# Create cache directory
mkdir -p /var/cache/nginx
chown -R nginx:nginx /var/cache/nginx

# Test and restart Nginx
nginx -t && systemctl restart nginx

echo "Nginx setup completed successfully!"
echo "Remember to update backend server IPs in /etc/nginx/nginx.conf"
```



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.

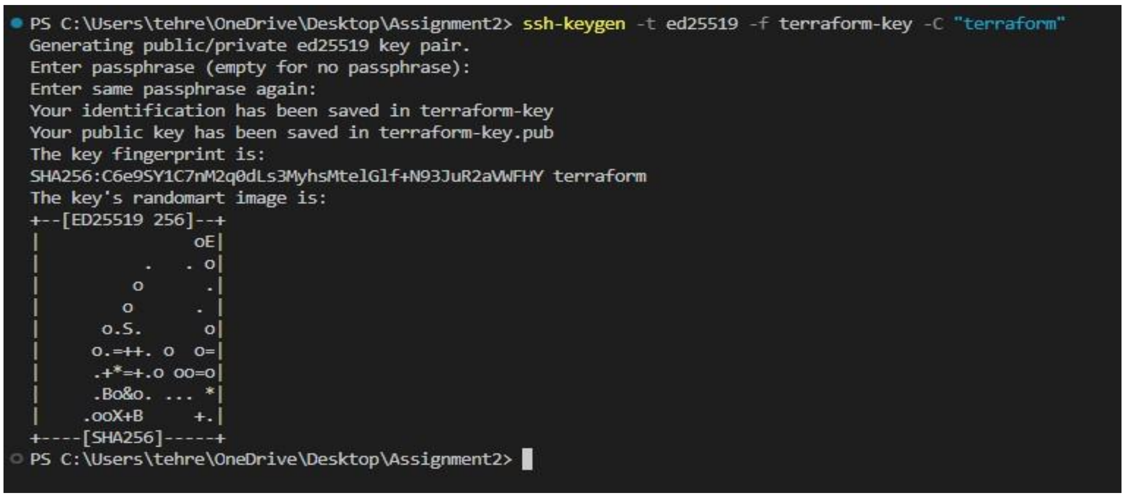
Part 4: Deployment

Infrastructure deployment was performed using standard Terraform commands:

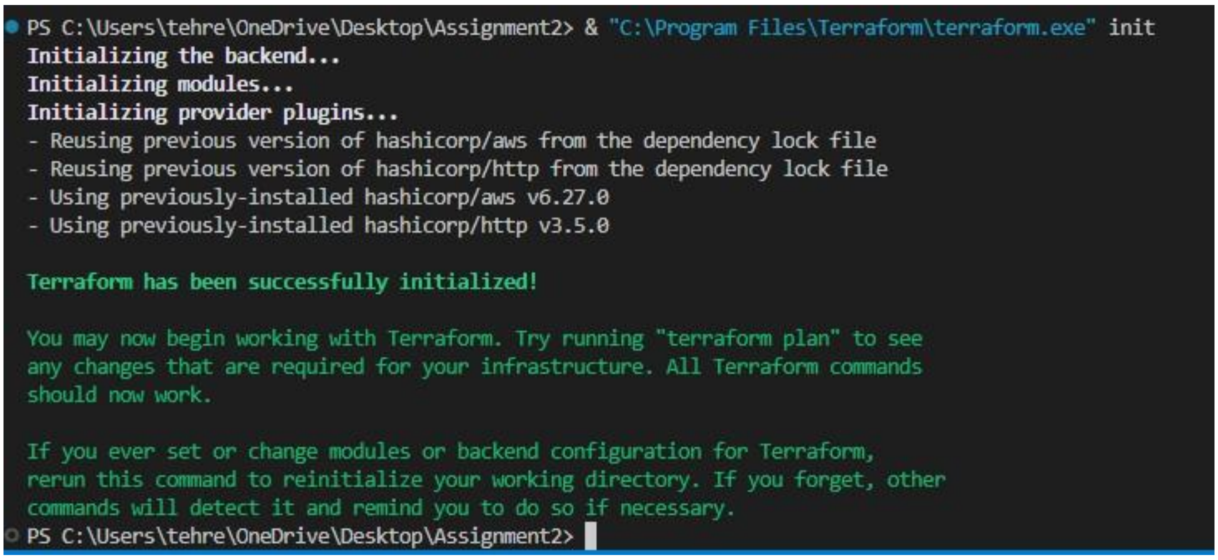
- terraform init
- terraform validate
- terraform plan
- terraform apply

The deployment process successfully created all AWS resources, which were verified through the AWS Management Console.

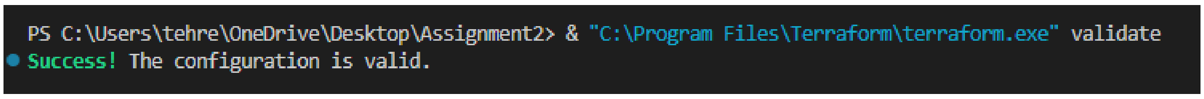
<u>4.1</u>



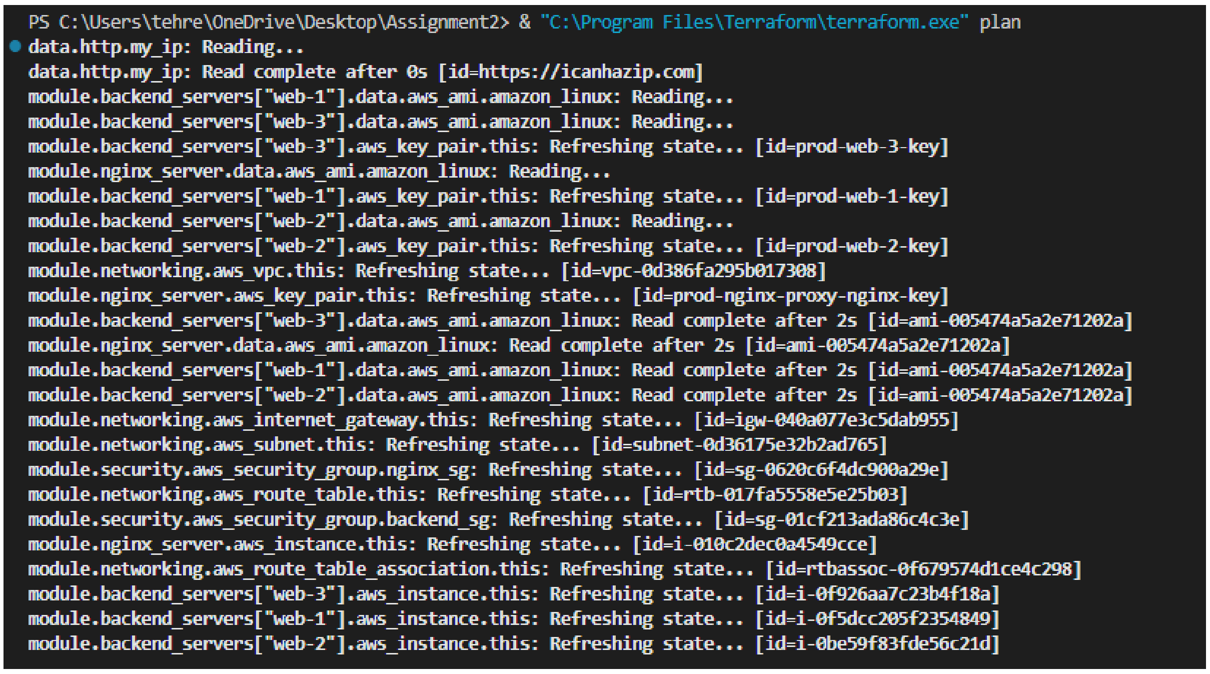
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh-keygen -t ed25519 -f terraform-key -C "terraform"
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in terraform-key
Your public key has been saved in terraform-key.pub
The key fingerprint is:
SHA256:C6e9SY1C7nM2q0dLs3MyhsMtelGlf+N93JuR2aVWFHY terraform
The key's randomart image is:
+--[ED25519 256]--+
|               oE|
|          .   . o|
|        o       .|
|       o       . |
|      o.S.      o|
|     o.=++. o  o=|
|     .+*=+.o oo=o|
|     .Bo&o. ... *|
|    .ooX+B     +.|
+----[SHA256]-----+
PS C:\Users\tehre\OneDrive\Desktop\Assignment2>
```



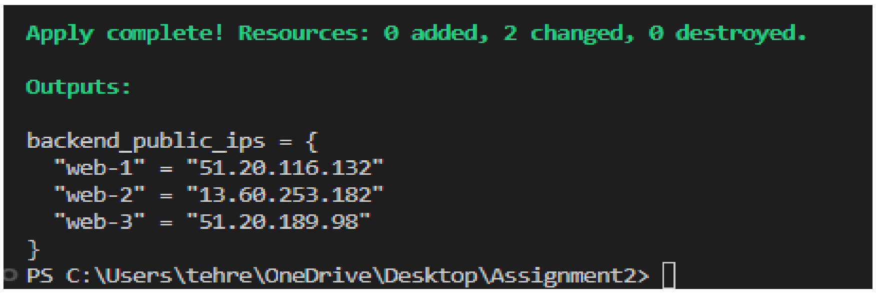
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> & "C:\Program Files\Terraform\terraform.exe" init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/http 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.
PS C:\Users\tehre\OneDrive\Desktop\Assignment2>
```



```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> & "C:\Program Files\Terraform\terraform.exe" validate
Success! The configuration is valid.
```

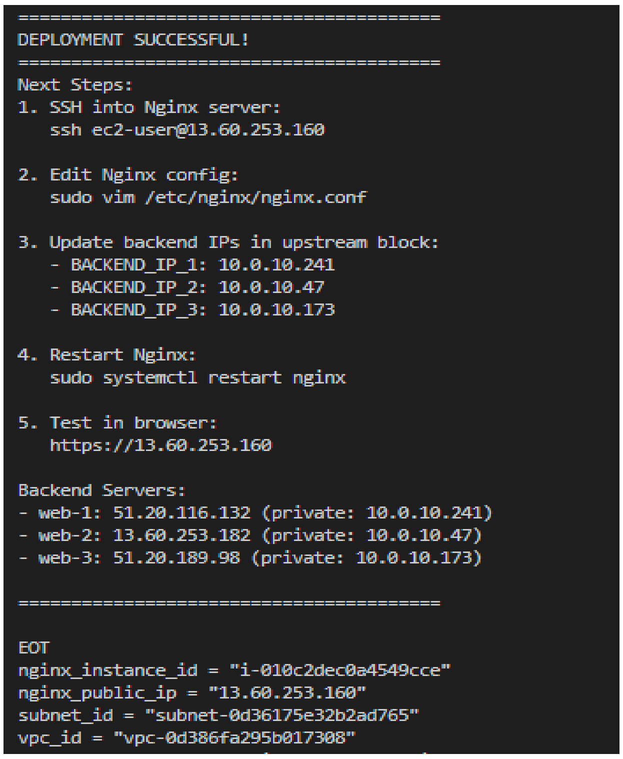


```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> & "C:\Program Files\Terraform\terraform.exe" plan
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.backend_servers["web-1"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-3"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-3"].aws_key_pair.this: Refreshing state... [id=prod-web-3-key]
module.nginx_server.data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-1"].aws_key_pair.this: Refreshing state... [id=prod-web-1-key]
module.backend_servers["web-2"].data.aws_ami.amazon_linux: Reading...
module.backend_servers["web-2"].aws_key_pair.this: Refreshing state... [id=prod-web-2-key]
module.networking.aws_vpc.this: Refreshing state... [id=vpc-0d386fa295b017308]
module.nginx_server.aws_key_pair.this: Refreshing state... [id=prod-nginx-proxy-nginx-key]
module.backend_servers["web-3"].data.aws_ami.amazon_linux: Read complete after 2s [id=ami-005474a5a2e71202a]
module.nginx_server.data.aws_ami.amazon_linux: Read complete after 2s [id=ami-005474a5a2e71202a]
module.backend_servers["web-1"].data.aws_ami.amazon_linux: Read complete after 2s [id=ami-005474a5a2e71202a]
module.backend_servers["web-2"].data.aws_ami.amazon_linux: Read complete after 2s [id=ami-005474a5a2e71202a]
module.networking.aws_internet_gateway.this: Refreshing state... [id=igw-040a077e3c5dab955]
module.networking.aws_subnet.this: Refreshing state... [id=subnet-0d36175e32b2ad765]
module.security.aws_security_group.nginx_sg: Refreshing state... [id=sg-0620c6f4dc900a29e]
module.networking.aws_route_table.this: Refreshing state... [id=rtb-017fa5558e5e25b03]
module.security.aws_security_group.backend_sg: Refreshing state... [id=sg-01cf213ada86c4c3e]
module.nginx_server.aws_instance.this: Refreshing state... [id=i-010c2dec0a4549cce]
module.networking.aws_route_table_association.this: Refreshing state... [id=rtbassoc-0f679574d1ce4c298]
module.backend_servers["web-3"].aws_instance.this: Refreshing state... [id=i-0f926aa7c23b4f18a]
module.backend_servers["web-1"].aws_instance.this: Refreshing state... [id=i-0f5dcc205f2354849]
module.backend_servers["web-2"].aws_instance.this: Refreshing state... [id=i-0be59f83fde56c21d]
```

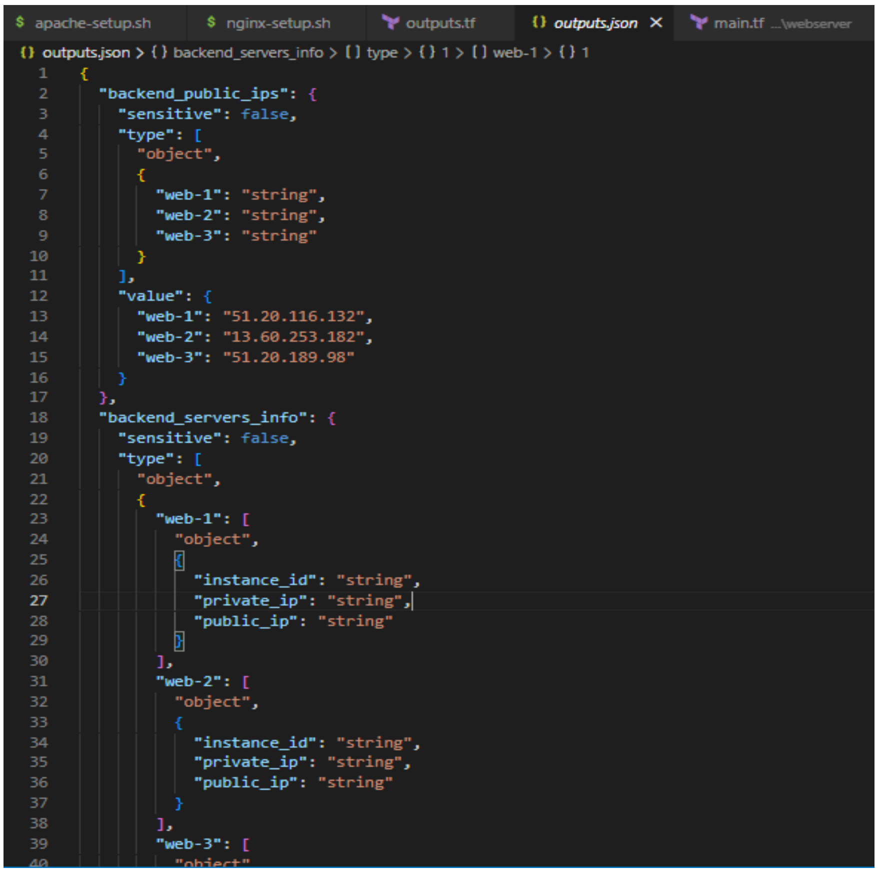


```
Apply complete! Resources: 0 added, 2 changed, 0 destroyed.

Outputs:

backend_public_ips = {
  "web-1" = "51.20.116.132"
  "web-2" = "13.60.253.182"
  "web-3" = "51.20.189.98"
}
PS C:\Users\tehre\OneDrive\Desktop\Assignment2>
```

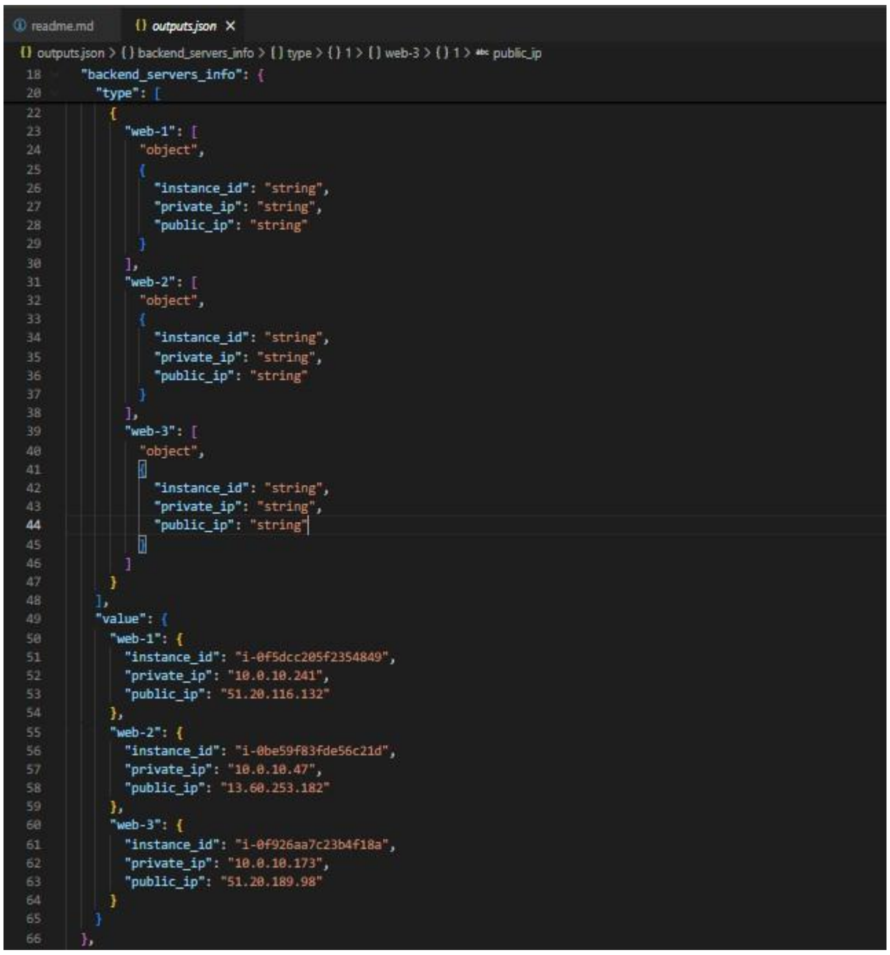
4.2



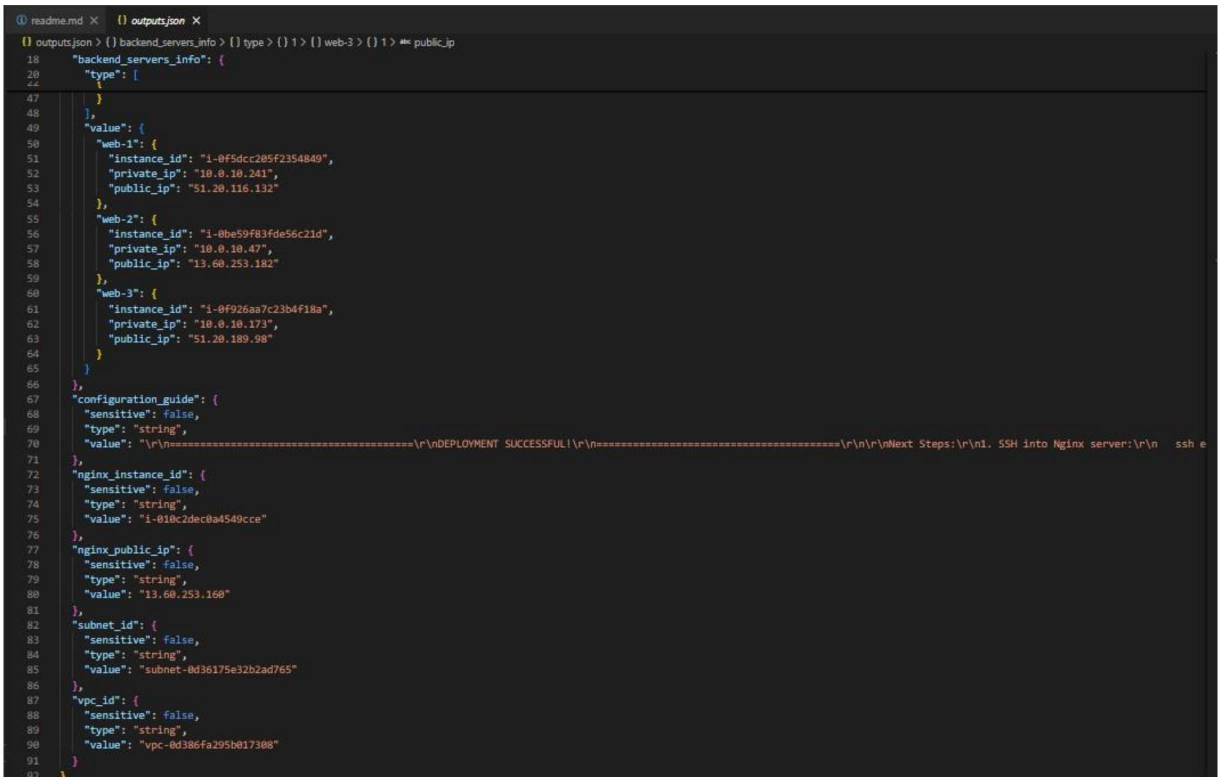
```
========================================
DEPLOYMENT SUCCESSFUL!
========================================
Next Steps:
1. SSH into Nginx server:
   ssh ec2-user@13.60.253.160

2. Edit Nginx config:
   sudo vim /etc/nginx/nginx.conf

3. Update backend IPs in upstream block:
   - BACKEND_IP_1: 10.0.10.241
   - BACKEND_IP_2: 10.0.10.47
   - BACKEND_IP_3: 10.0.10.173

4. Restart Nginx:
   sudo systemctl restart nginx

5. Test in browser:
   https://13.60.253.160

Backend Servers:
- web-1: 51.20.116.132 (private: 10.0.10.241)
- web-2: 13.60.253.182 (private: 10.0.10.47)
- web-3: 51.20.189.98 (private: 10.0.10.173)

========================================

EOT
nginx_instance_id = "i-010c2dec0a4549cce"
nginx_public_ip = "13.60.253.160"
subnet_id = "subnet-0d36175e32b2ad765"
vpc_id = "vpc-0d386fa295b017308"
```


apache-setup.sh
nginx-setup.sh
outputs.tf
outputs.json
main.tf ...\webserver
outputs.json > { } backend_servers_info > [ ] type > { } 1 > [ ] web-1 > { } 1

```
{
  "backend_public_ips": {
    "sensitive": false,
    "type": [
      "object",
      {
        "web-1": "string",
        "web-2": "string",
        "web-3": "string"
      }
    ],
    "value": {
      "web-1": "51.20.116.132",
      "web-2": "13.60.253.182",
      "web-3": "51.20.189.98"
    }
  },
  "backend_servers_info": {
    "sensitive": false,
    "type": [
      "object",
      {
        "web-1": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-2": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-3": [
```

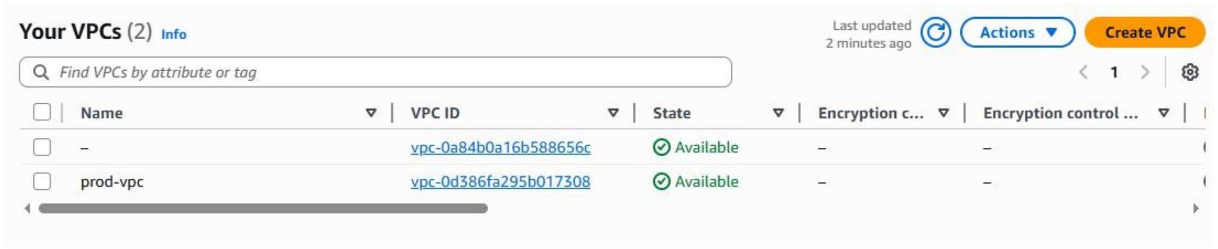

readme.md
outputs.json
outputs.json > { } backend_servers_info > [ ] type > { } 1 > [ ] web-3 > { } 1 > public_ip

```
  "backend_servers_info": {
    "type": [
      {
        "web-1": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-2": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-3": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ]
      }
    ],
    "value": {
      "web-1": {
        "instance_id": "i-0f5dcc205f2354849",
        "private_ip": "10.0.10.241",
        "public_ip": "51.20.116.132"
      },
      "web-2": {
        "instance_id": "i-0be59f83fde56c21d",
        "private_ip": "10.0.10.47",
        "public_ip": "13.60.253.182"
      },
      "web-3": {
        "instance_id": "i-0f926aa7c23b4f18a",
        "private_ip": "10.0.10.173",
        "public_ip": "51.20.189.98"
      }
    }
  },
```

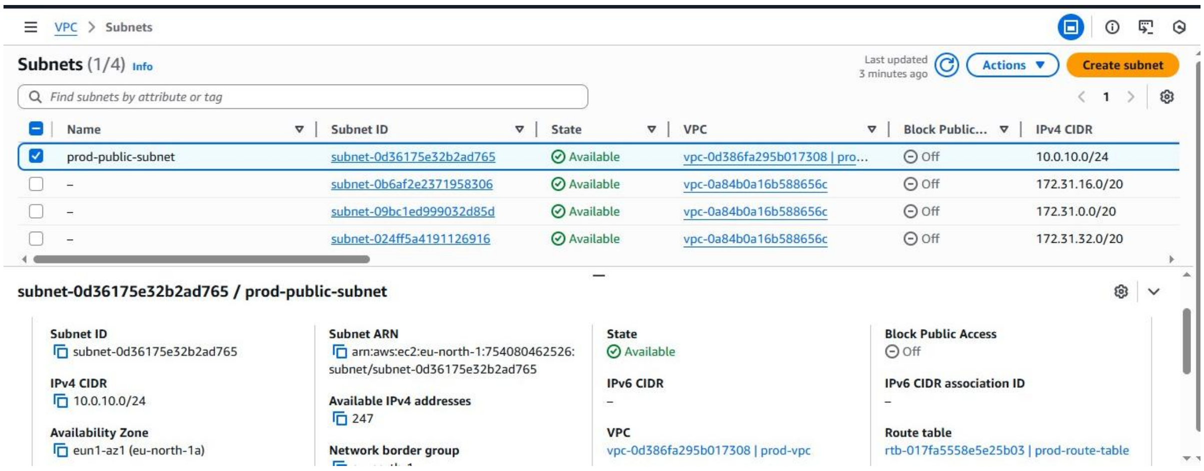

readme.md
outputs.json
outputs.json > { } backend_servers_info > [ ] type > { } 1 > [ ] web-3 > { } 1 > public_ip

```
    "backend_servers_info": {
      "type": [
        }
      ],
      "value": {
        "web-1": {
          "instance_id": "i-0f5dcc205f2354849",
          "private_ip": "10.0.10.241",
          "public_ip": "51.20.116.132"
        },
        "web-2": {
          "instance_id": "i-0be59f83fde56c21d",
          "private_ip": "10.0.10.47",
          "public_ip": "13.60.253.182"
        },
        "web-3": {
          "instance_id": "i-0f926aa7c23b4f18a",
          "private_ip": "10.0.10.173",
          "public_ip": "51.20.189.98"
        }
      }
    },
    "configuration_guide": {
      "sensitive": false,
      "type": "string",
      "value": "\r\n========================================\r\nDEPLOYMENT SUCCESSFUL!\r\n========================================\r\n\r\nNext Steps:\r\n1. SSH into Nginx server:\r\n   ssh e
    },
    "nginx_instance_id": {
      "sensitive": false,
      "type": "string",
      "value": "i-010c2dec0a4549cce"
    },
    "nginx_public_ip": {
      "sensitive": false,
      "type": "string",
      "value": "13.60.253.160"
    },
    "subnet_id": {
      "sensitive": false,
      "type": "string",
      "value": "subnet-0d36175e32b2ad765"
    },
    "vpc_id": {
      "sensitive": false,
      "type": "string",
      "value": "vpc-0d386fa295b017308"
    }
```

4.3



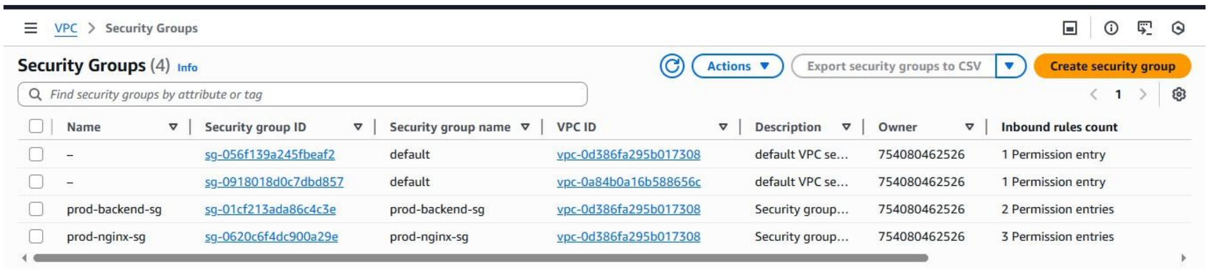
Name	VPC ID	State	Encryption c...	Encryption control ...
–	vpc-0a84b0a16b588656c	Available	–	–
prod-vpc	vpc-0d386fa295b017308	Available	–	–



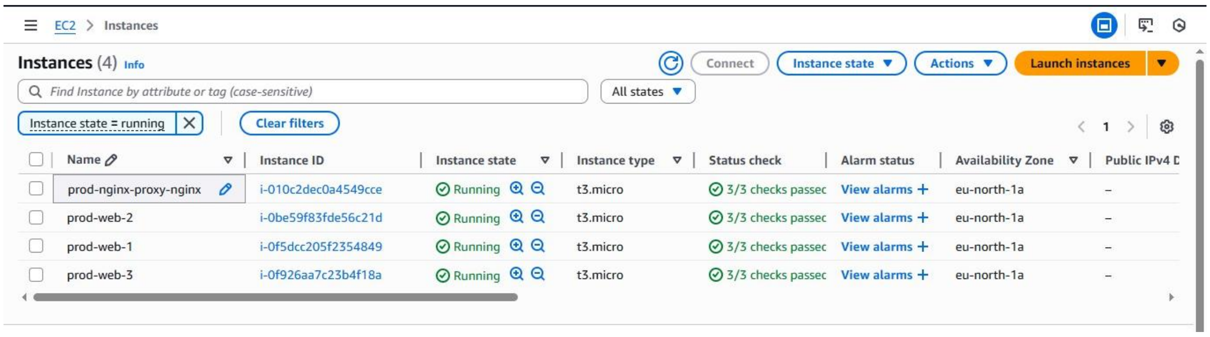
Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
prod-public-subnet	subnet-0d36175e32b2ad765	Available	vpc-0d386fa295b017308 | pro...	Off	10.0.10.0/24
–	subnet-0b6af2e2371958306	Available	vpc-0a84b0a16b588656c	Off	172.31.16.0/20
–	subnet-09bc1ed999032d85d	Available	vpc-0a84b0a16b588656c	Off	172.31.0.0/20
–	subnet-024ff5a4191126916	Available	vpc-0a84b0a16b588656c	Off	172.31.32.0/20

subnet-0d36175e32b2ad765 / prod-public-subnet

Subnet ID subnet-0d36175e32b2ad765	**Subnet ARN** arn:aws:ec2:eu-north-1:754080462526:subnet/subnet-0d36175e32b2ad765	**State** Available	**Block Public Access** Off
IPv4 CIDR 10.0.10.0/24	**Available IPv4 addresses** 247	**IPv6 CIDR** –	**IPv6 CIDR association ID** –
Availability Zone eun1-az1 (eu-north-1a)	**Network border group**	**VPC** vpc-0d386fa295b017308 | prod-vpc	**Route table** rtb-017fa5558e5e25b03 | prod-route-table



Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count
–	sg-056f139a245fbeaf2	default	vpc-0d386fa295b017308	default VPC se...	754080462526	1 Permission entry
–	sg-0918018d0c7dbd857	default	vpc-0a84b0a16b588656c	default VPC se...	754080462526	1 Permission entry
prod-backend-sg	sg-01cf213ada86c4c3e	prod-backend-sg	vpc-0d386fa295b017308	Security group...	754080462526	2 Permission entries
prod-nginx-sg	sg-0620c6f4dc900a29e	prod-nginx-sg	vpc-0d386fa295b017308	Security group...	754080462526	3 Permission entries

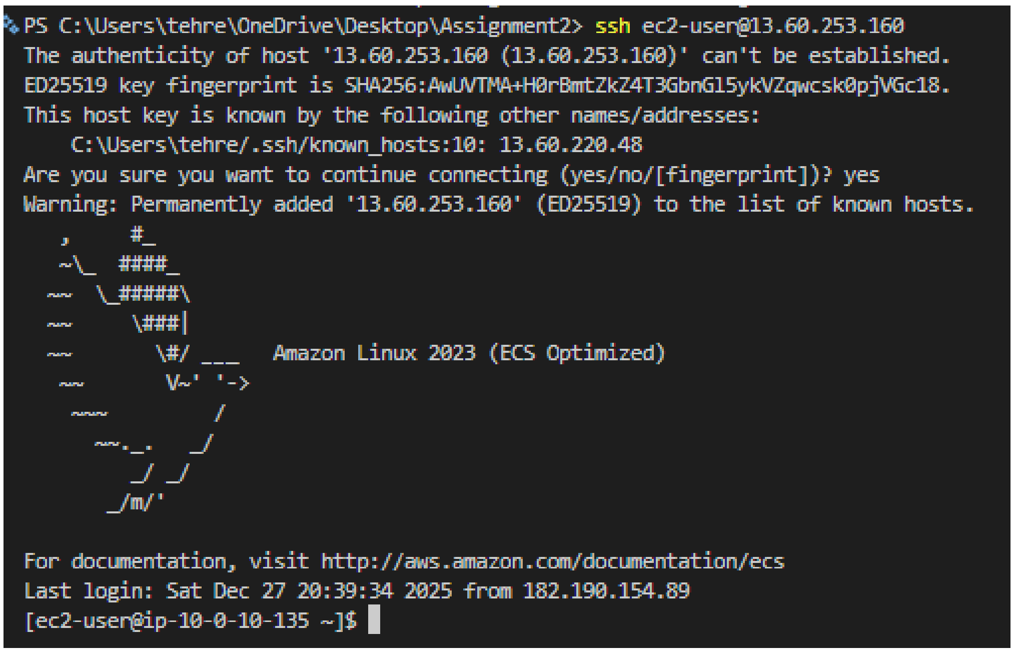


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
prod-nginx-proxy-nginx	i-010c2dec0a4549cce	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	–
prod-web-2	i-0be59f83fde56c21d	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	–
prod-web-1	i-0f5dcc205f2354849	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	–
prod-web-3	i-0f926aa7c23b4f18a	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	–

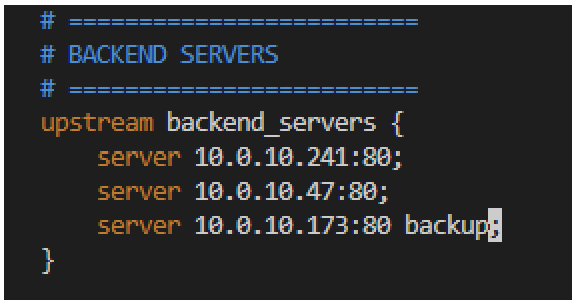
Part 5: Testing and Verification

Testing included verifying load balancing behaviour, cache functionality, backup server availability, and security rules. Browser developer tools were used to inspect HTTP headers and confirm caching behaviour.

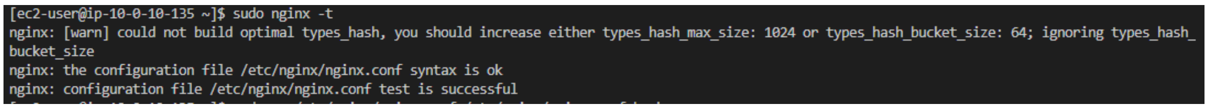
5.1



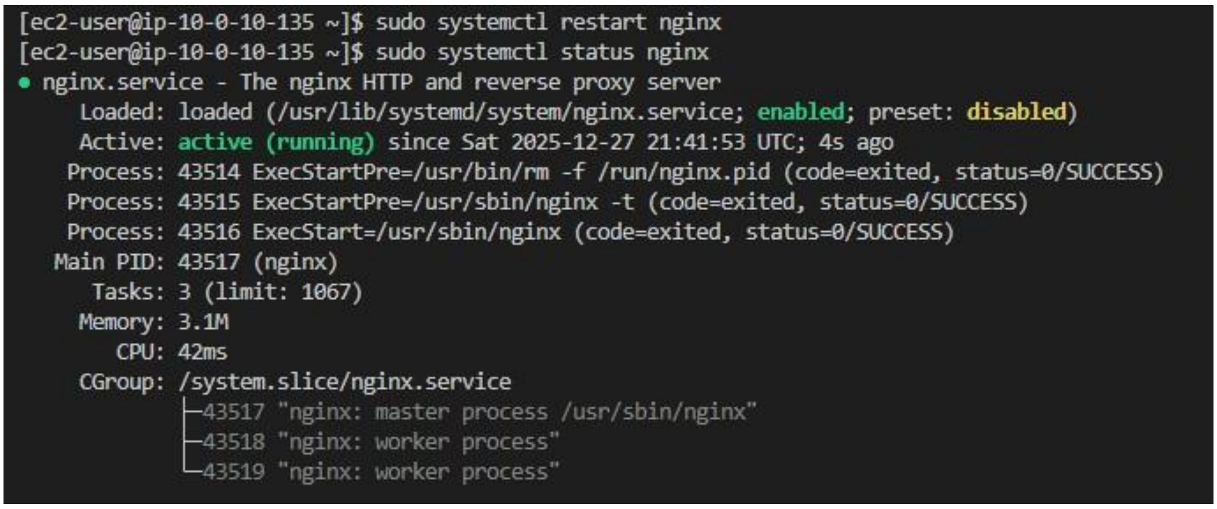
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh ec2-user@13.60.253.160
The authenticity of host '13.60.253.160 (13.60.253.160)' can't be established.
ED25519 key fingerprint is SHA256:AwUVTMA+H0rBmtZkZ4T3GbnGl5ykVZqwcsk0pjVGc18.
This host key is known by the following other names/addresses:
    C:\Users\tehre/.ssh/known_hosts:10: 13.60.220.48
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '13.60.253.160' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sat Dec 27 20:39:34 2025 from 182.190.154.89
[ec2-user@ip-10-0-10-135 ~]$ 
```



```
# ==========================
# BACKEND SERVERS
# ==========================
upstream backend_servers {
    server 10.0.10.241:80;
    server 10.0.10.47:80;
    server 10.0.10.173:80 backup;
}
```

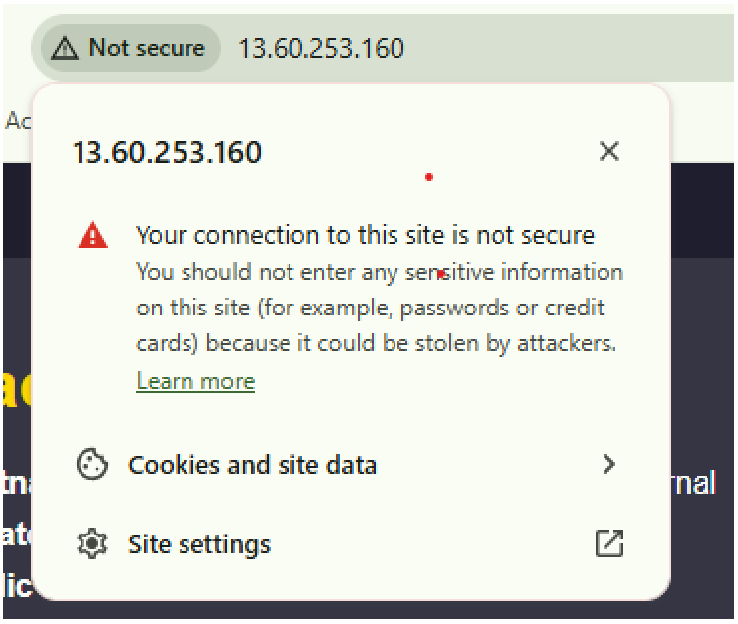


```
[ec2-user@ip-10-0-10-135 ~]$ sudo nginx -t
nginx: [warn] could not build optimal types_hash, you should increase either types_hash_max_size: 1024 or types_hash_bucket_size: 64; ignoring types_hash_
bucket_size
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```



```
[ec2-user@ip-10-0-10-135 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-135 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sat 2025-12-27 21:41:53 UTC; 4s ago
    Process: 43514 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 43515 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 43516 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 43517 (nginx)
      Tasks: 3 (limit: 1067)
     Memory: 3.1M
        CPU: 42ms
     CGroup: /system.slice/nginx.service
             ├─43517 "nginx: master process /usr/sbin/nginx"
             ├─43518 "nginx: worker process"
             └─43519 "nginx: worker process"
```

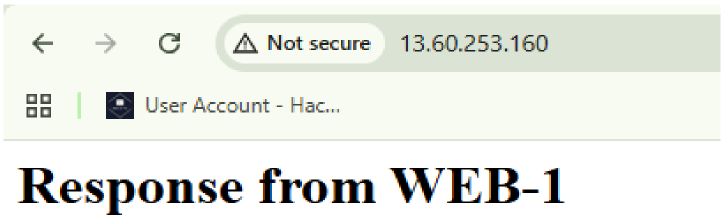
5.2



13.60.253.160

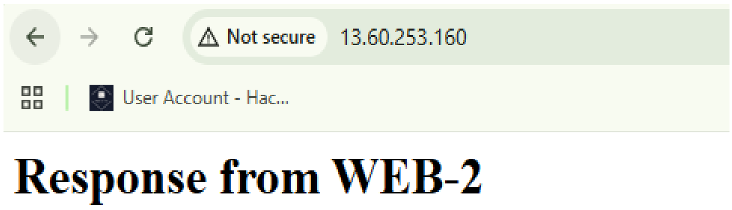
Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards) because it could be stolen by attackers. Learn more


Not secure 13.60.253.160

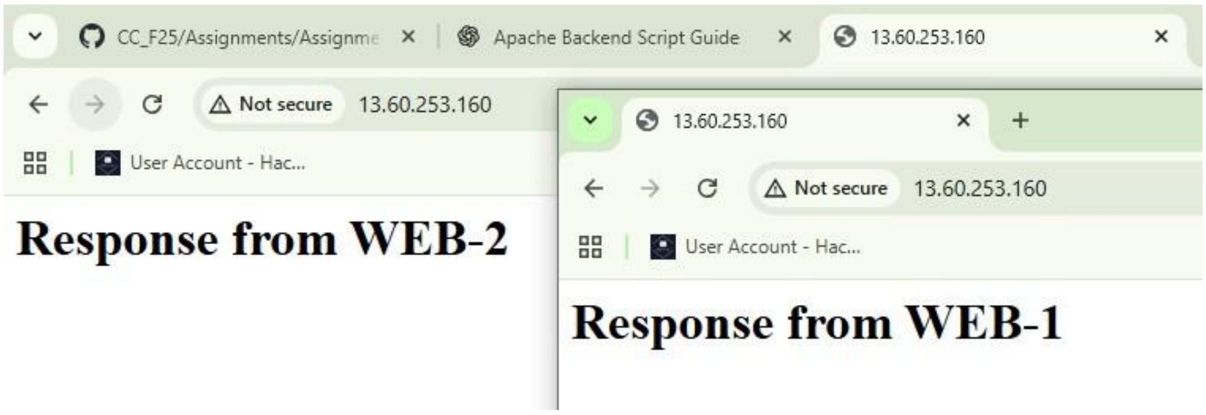
User Account - Hac...

Response from WEB-1


Not secure 13.60.253.160

User Account - Hac...

Response from WEB-2


CC_F25/Assignments/Assignme | Apache Backend Script Guide | 13.60.253.160

Not secure 13.60.253.160

User Account - Hac...

Response from WEB-2

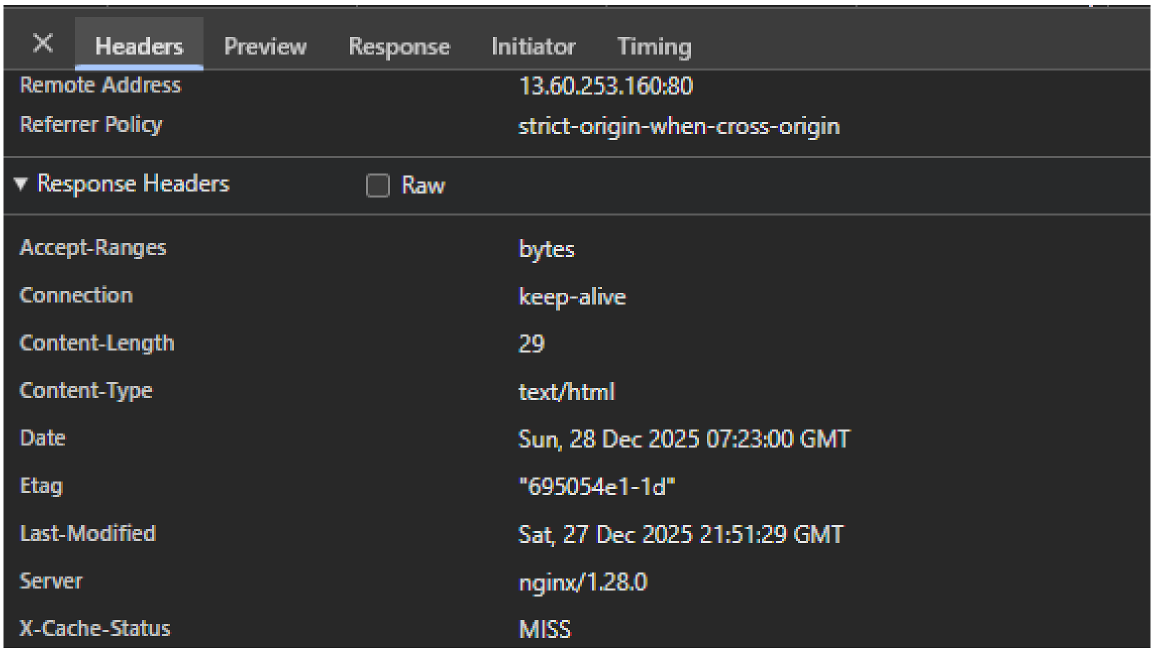
13.60.253.160

Not secure 13.60.253.160

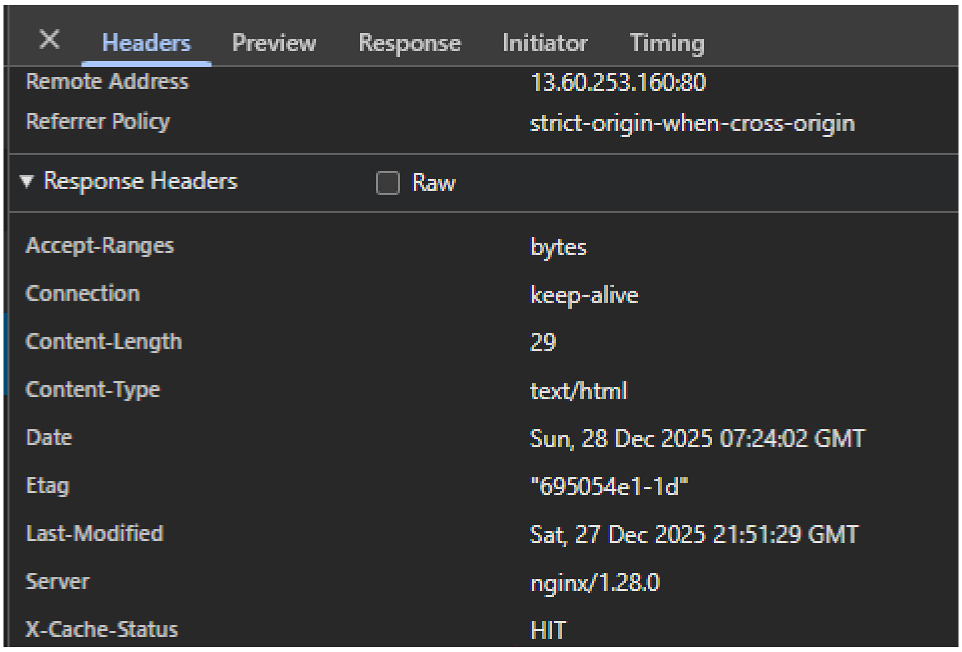
User Account - Hac...

Response from WEB-1

5.3


Headers Preview Response Initiator Timing

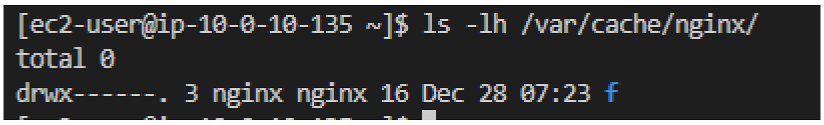
```
Remote Address          13.60.253.160:80
Referrer Policy         strict-origin-when-cross-origin
▼ Response Headers      ☐ Raw
Accept-Ranges           bytes
Connection              keep-alive
Content-Length          29
Content-Type            text/html
Date                    Sun, 28 Dec 2025 07:23:00 GMT
Etag                    "695054e1-1d"
Last-Modified           Sat, 27 Dec 2025 21:51:29 GMT
Server                  nginx/1.28.0
X-Cache-Status          MISS
```



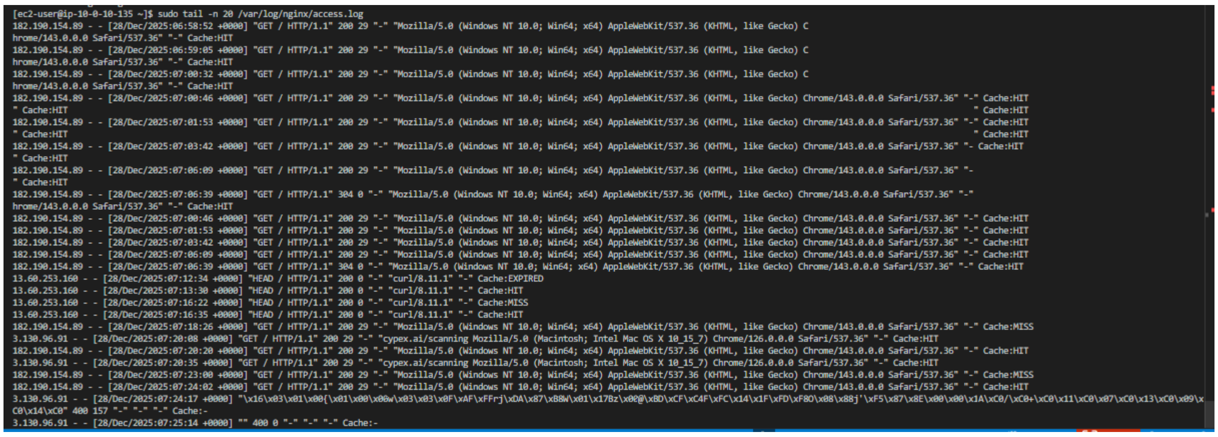
Remote Address	13.60.253.160:80
Referrer Policy	strict-origin-when-cross-origin

▼ Response Headers ☐ Raw

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	29
Content-Type	text/html
Date	Sun, 28 Dec 2025 07:24:02 GMT
Etag	"695054e1-1d"
Last-Modified	Sat, 27 Dec 2025 21:51:29 GMT
Server	nginx/1.28.0
X-Cache-Status	HIT

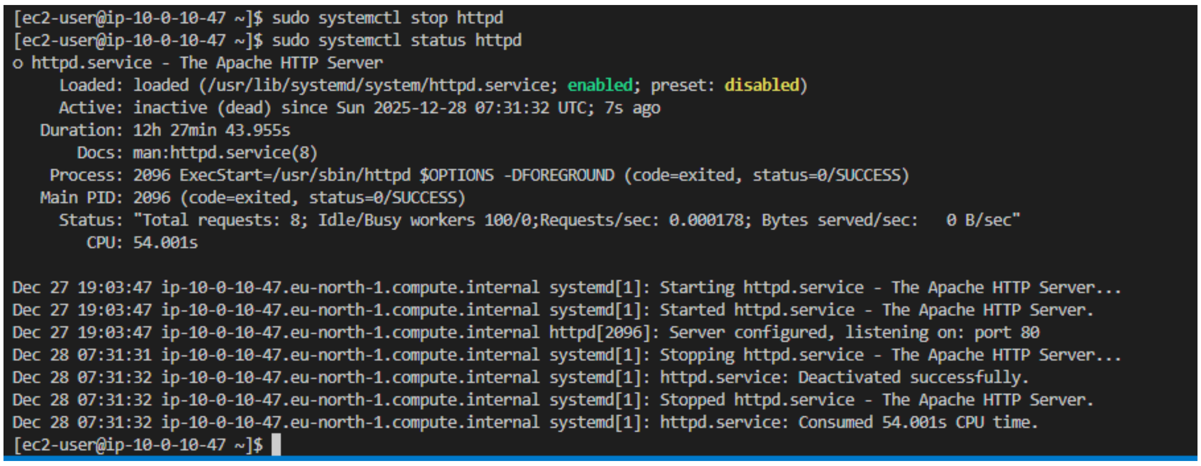


```
[ec2-user@ip-10-0-10-135 ~]$ ls -lh /var/cache/nginx/
total 0
drwx------. 3 nginx nginx 16 Dec 28 07:23 f
```

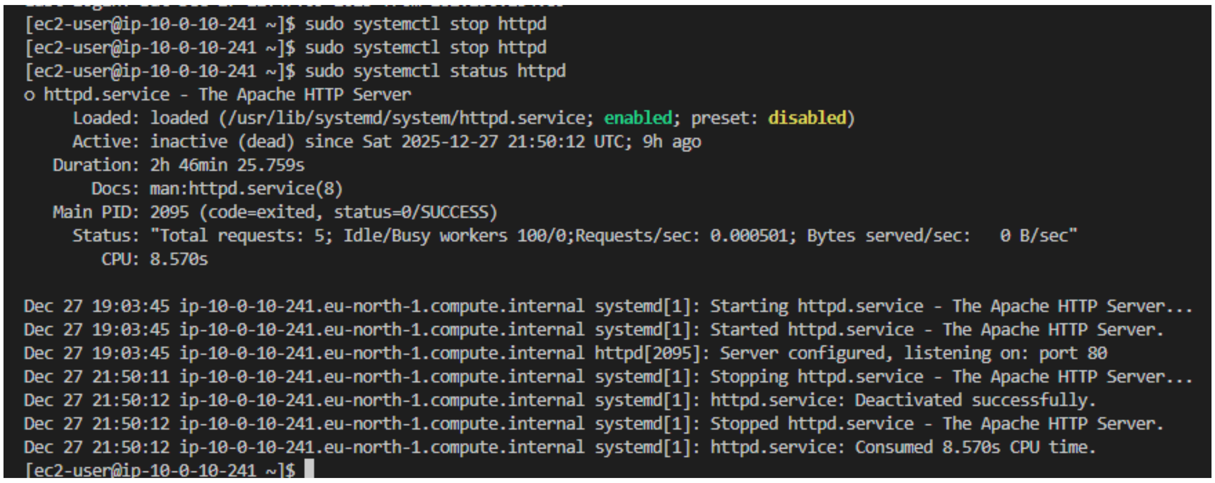


```
[ec2-user@ip-10-0-10-135 ~]$ sudo tail -n 20 /var/log/nginx/access.log
182.190.154.89 - - [28/Dec/2025:06:58:52 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:06:59:05 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:00:32 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:00:46 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
" Cache:HIT                                                                                                                                                 " Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:01:53 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
" Cache:HIT                                                                                                                                                 " Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:03:42 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "- Cache:HIT
" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:06:09 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-
" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:06:39 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
hrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:00:46 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:01:53 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:03:42 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:06:09 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:06:39 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
13.60.253.160 - - [28/Dec/2025:07:12:34 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.11.1" "-" Cache:EXPIRED
13.60.253.160 - - [28/Dec/2025:07:13:30 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
13.60.253.160 - - [28/Dec/2025:07:16:22 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.11.1" "-" Cache:MISS
13.60.253.160 - - [28/Dec/2025:07:16:35 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:18:26 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:MISS
3.130.96.91 - - [28/Dec/2025:07:20:08 +0000] "GET / HTTP/1.1" 200 29 "-" "cypex.ai/scanning Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Chrome/126.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:20:20 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
3.130.96.91 - - [28/Dec/2025:07:20:35 +0000] "GET / HTTP/1.1" 200 29 "-" "cypex.ai/scanning Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Chrome/126.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:23:00 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:MISS
182.190.154.89 - - [28/Dec/2025:07:24:02 +0000] "GET / HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
3.130.96.91 - - [28/Dec/2025:07:24:17 +0000] "\x16\x03\x01\x00{\x01\x00\x00w\x03\x03\x0F\xAF\xFFrj\xDA\x87\xB8W\x01\x17Bz\x00@\xBD\xCF\xC4F\xFC\x14\x1F\xFD\xF8O\x08\x88j'\xF5\x87\x8E\x00\x00\x1A\xC0/\xC0+\xC0\x11\xC0\x07\xC0\x13\xC0\x09\x
C0\x14\xC0" 400 157 "-" "-" "-" Cache:-
3.130.96.91 - - [28/Dec/2025:07:25:14 +0000] "" 400 0 "-" "-" "-" Cache:-
```

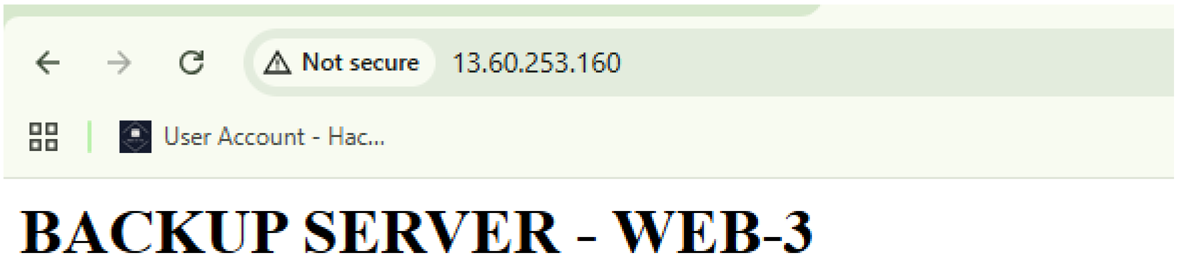
<u>5.4</u>



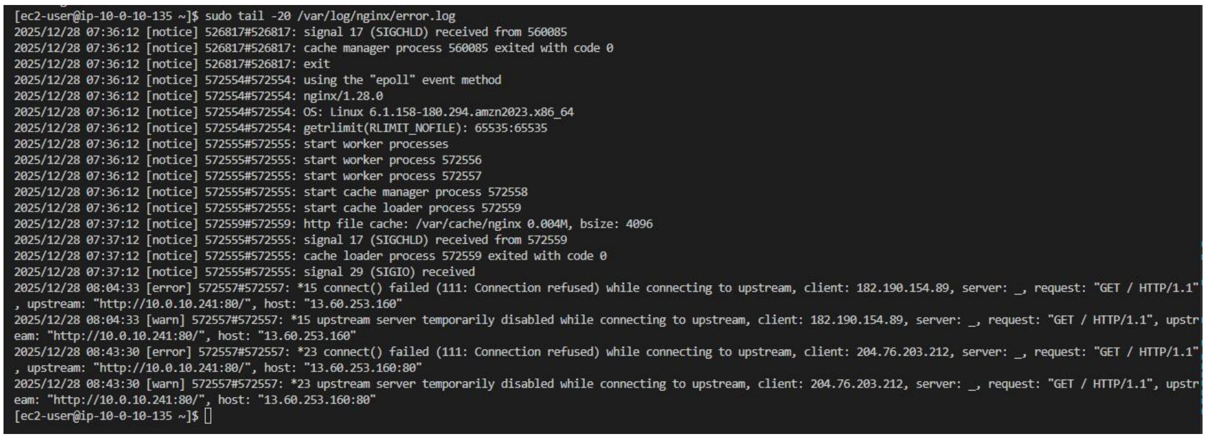
```
[ec2-user@ip-10-0-10-47 ~]$ sudo systemctl stop httpd
[ec2-user@ip-10-0-10-47 ~]$ sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: inactive (dead) since Sun 2025-12-28 07:31:32 UTC; 7s ago
   Duration: 12h 27min 43.955s
       Docs: man:httpd.service(8)
    Process: 2096 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
   Main PID: 2096 (code=exited, status=0/SUCCESS)
     Status: "Total requests: 8; Idle/Busy workers 100/0;Requests/sec: 0.000178; Bytes served/sec:   0 B/sec"
        CPU: 54.001s

Dec 27 19:03:47 ip-10-0-10-47.eu-north-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Dec 27 19:03:47 ip-10-0-10-47.eu-north-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Dec 27 19:03:47 ip-10-0-10-47.eu-north-1.compute.internal httpd[2096]: Server configured, listening on: port 80
Dec 28 07:31:31 ip-10-0-10-47.eu-north-1.compute.internal systemd[1]: Stopping httpd.service - The Apache HTTP Server...
Dec 28 07:31:32 ip-10-0-10-47.eu-north-1.compute.internal systemd[1]: httpd.service: Deactivated successfully.
Dec 28 07:31:32 ip-10-0-10-47.eu-north-1.compute.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Dec 28 07:31:32 ip-10-0-10-47.eu-north-1.compute.internal systemd[1]: httpd.service: Consumed 54.001s CPU time.
[ec2-user@ip-10-0-10-47 ~]$ 
```



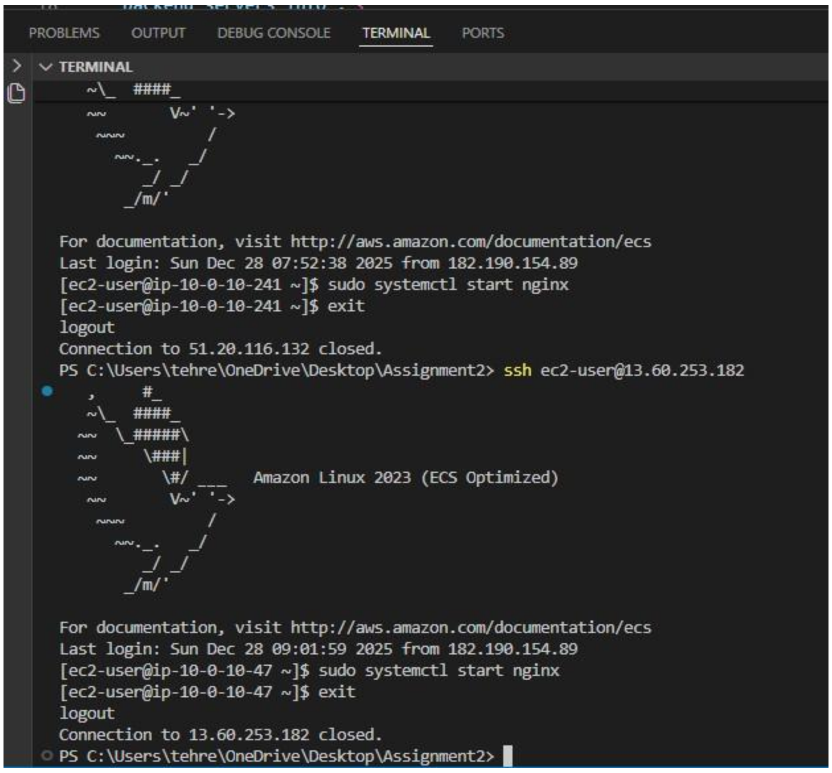
```
[ec2-user@ip-10-0-10-241 ~]$ sudo systemctl stop httpd
[ec2-user@ip-10-0-10-241 ~]$ sudo systemctl stop httpd
[ec2-user@ip-10-0-10-241 ~]$ sudo systemctl status httpd
o httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: inactive (dead) since Sat 2025-12-27 21:50:12 UTC; 9h ago
   Duration: 2h 46min 25.759s
       Docs: man:httpd.service(8)
   Main PID: 2095 (code=exited, status=0/SUCCESS)
     Status: "Total requests: 5; Idle/Busy workers 100/0;Requests/sec: 0.000501; Bytes served/sec:   0 B/sec"
        CPU: 8.570s

Dec 27 19:03:45 ip-10-0-10-241.eu-north-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Dec 27 19:03:45 ip-10-0-10-241.eu-north-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Dec 27 19:03:45 ip-10-0-10-241.eu-north-1.compute.internal httpd[2095]: Server configured, listening on: port 80
Dec 27 21:50:11 ip-10-0-10-241.eu-north-1.compute.internal systemd[1]: Stopping httpd.service - The Apache HTTP Server...
Dec 27 21:50:12 ip-10-0-10-241.eu-north-1.compute.internal systemd[1]: httpd.service: Deactivated successfully.
Dec 27 21:50:12 ip-10-0-10-241.eu-north-1.compute.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Dec 27 21:50:12 ip-10-0-10-241.eu-north-1.compute.internal systemd[1]: httpd.service: Consumed 8.570s CPU time.
[ec2-user@ip-10-0-10-241 ~]$
```



BACKUP SERVER - WEB-3

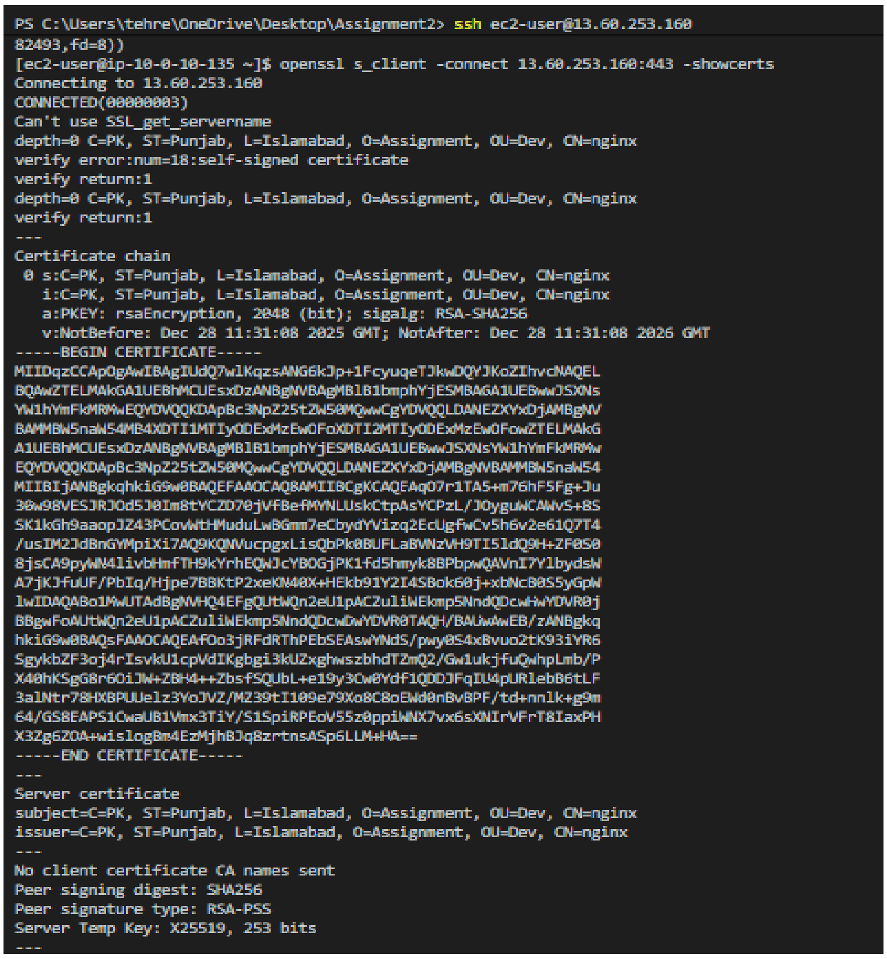


```
[ec2-user@ip-10-0-10-135 ~]$ sudo tail -20 /var/log/nginx/error.log
2025/12/28 07:36:12 [notice] 526817#526817: signal 17 (SIGCHLD) received from 560085
2025/12/28 07:36:12 [notice] 526817#526817: cache manager process 560085 exited with code 0
2025/12/28 07:36:12 [notice] 526817#526817: exit
2025/12/28 07:36:12 [notice] 572554#572554: using the "epoll" event method
2025/12/28 07:36:12 [notice] 572554#572554: nginx/1.28.0
2025/12/28 07:36:12 [notice] 572554#572554: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/28 07:36:12 [notice] 572554#572554: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/28 07:36:12 [notice] 572555#572555: start worker processes
2025/12/28 07:36:12 [notice] 572555#572555: start worker process 572556
2025/12/28 07:36:12 [notice] 572555#572555: start worker process 572557
2025/12/28 07:36:12 [notice] 572555#572555: start cache manager process 572558
2025/12/28 07:36:12 [notice] 572555#572555: start cache loader process 572559
2025/12/28 07:37:12 [notice] 572559#572559: http file cache: /var/cache/nginx 0.004M, bsize: 4096
2025/12/28 07:37:12 [notice] 572555#572555: signal 17 (SIGCHLD) received from 572559
2025/12/28 07:37:12 [notice] 572555#572555: cache loader process 572559 exited with code 0
2025/12/28 07:37:12 [notice] 572555#572555: signal 29 (SIGIO) received
2025/12/28 08:04:33 [error] 572557#572557: *15 connect() failed (111: Connection refused) while connecting to upstream, client: 182.190.154.89, server: _, request: "GET / HTTP/1.1"
, upstream: "http://10.0.10.241:80/", host: "13.60.253.160"
2025/12/28 08:04:33 [warn] 572557#572557: *15 upstream server temporarily disabled while connecting to upstream, client: 182.190.154.89, server: _, request: "GET / HTTP/1.1", upstr
eam: "http://10.0.10.241:80/", host: "13.60.253.160"
2025/12/28 08:43:30 [error] 572557#572557: *23 connect() failed (111: Connection refused) while connecting to upstream, client: 204.76.203.212, server: _, request: "GET / HTTP/1.1"
, upstream: "http://10.0.10.241:80/", host: "13.60.253.160:80"
2025/12/28 08:43:30 [warn] 572557#572557: *23 upstream server temporarily disabled while connecting to upstream, client: 204.76.203.212, server: _, request: "GET / HTTP/1.1", upstr
eam: "http://10.0.10.241:80/", host: "13.60.253.160:80"
[ec2-user@ip-10-0-10-135 ~]$
```

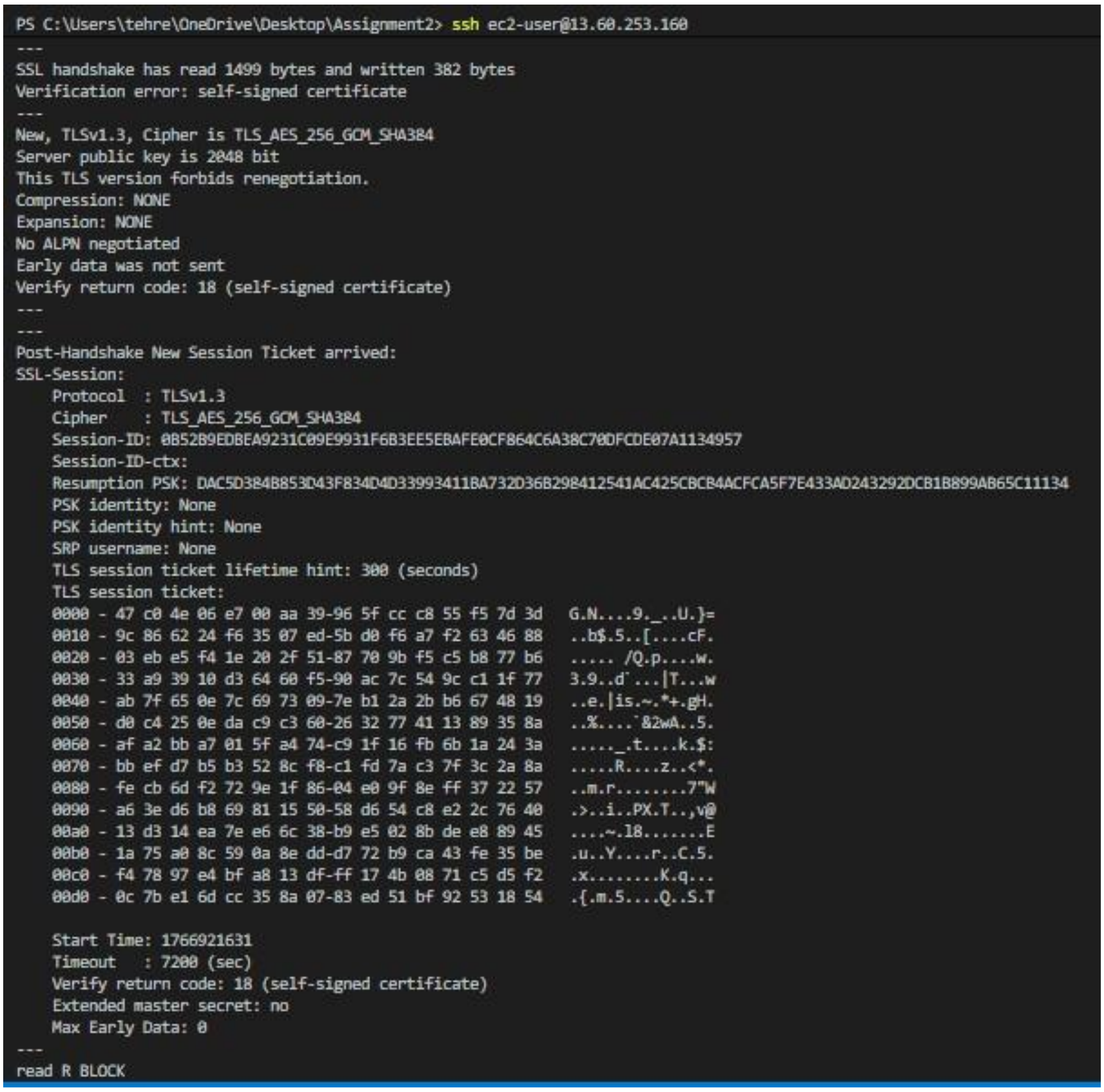


```
      ~\_  ####_
      ~~       V~' '->
       ~~~        /
         ~~._.   _/
            _/ _/
          _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 07:52:38 2025 from 182.190.154.89
[ec2-user@ip-10-0-10-241 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-10-241 ~]$ exit
logout
Connection to 51.20.116.132 closed.
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh ec2-user@13.60.253.182
      ,     #_
      ~\_  ####_
     ~~  \_#####\
     ~~     \###|
     ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
      ~~       V~' '->
       ~~~        /
         ~~._.   _/
            _/ _/
          _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 09:01:59 2025 from 182.190.154.89
[ec2-user@ip-10-0-10-47 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-10-47 ~]$ exit
logout
Connection to 13.60.253.182 closed.
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> 
```

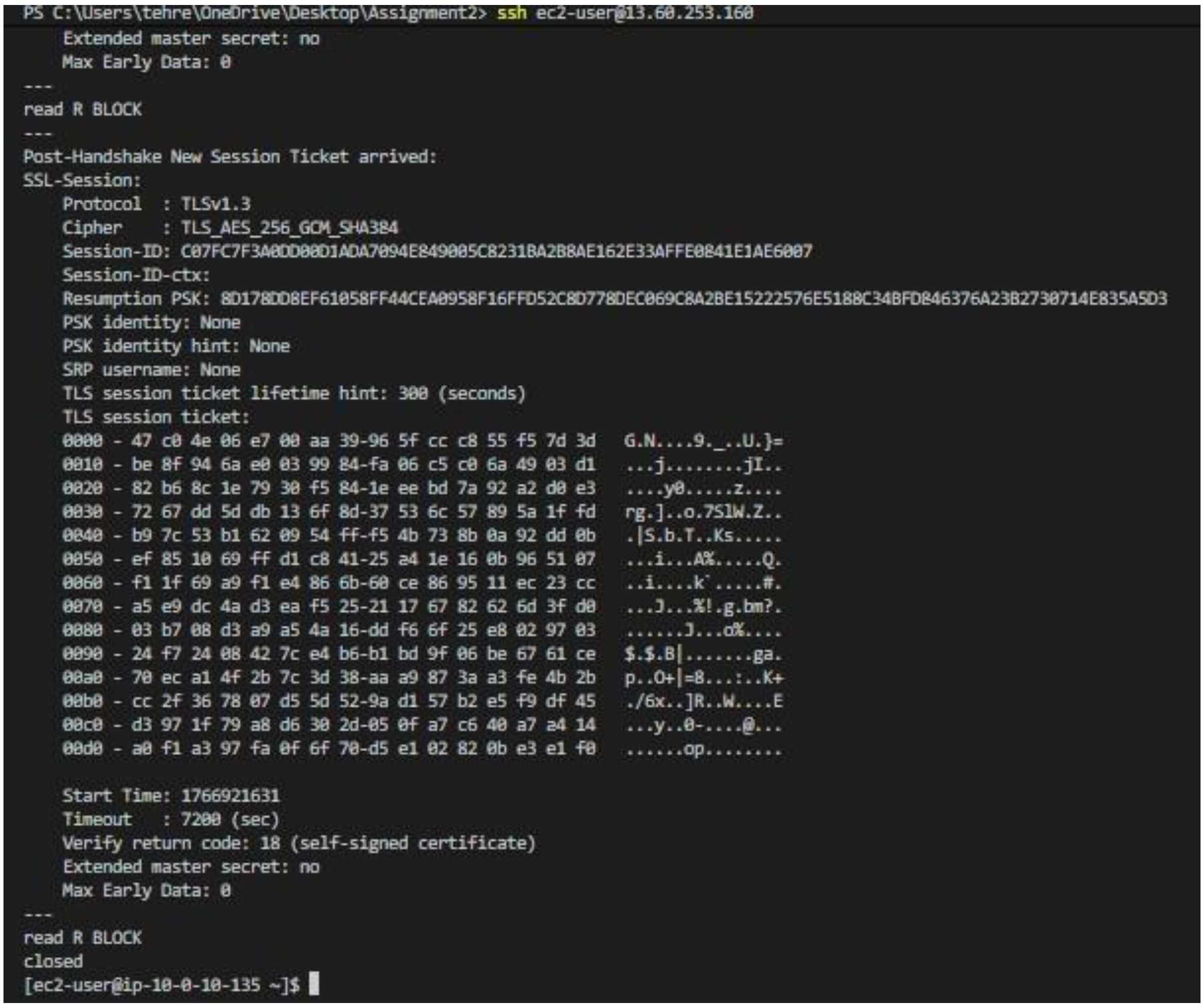
5.5



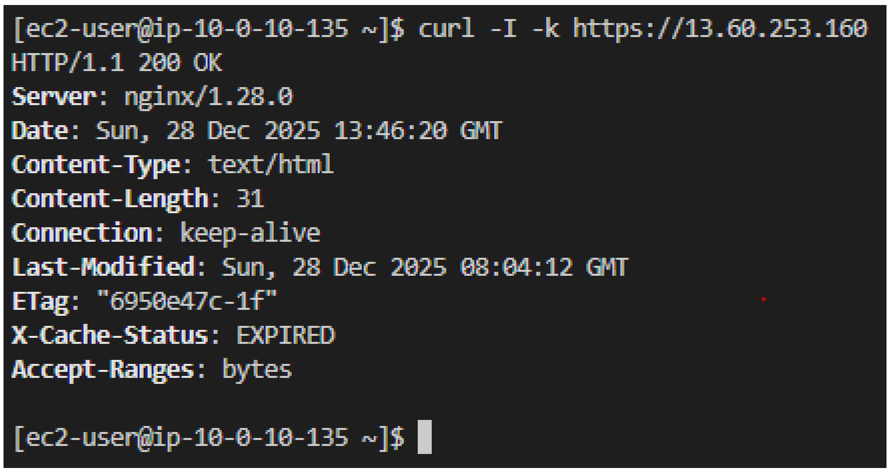
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh ec2-user@13.60.253.160
82493,fd=8))
[ec2-user@ip-10-0-10-135 ~]$ openssl s_client -connect 13.60.253.160:443 -showcerts
Connecting to 13.60.253.160
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C=PK, ST=Punjab, L=Islamabad, O=Assignment, OU=Dev, CN=nginx
verify error:num=18:self-signed certificate
verify return:1
depth=0 C=PK, ST=Punjab, L=Islamabad, O=Assignment, OU=Dev, CN=nginx
verify return:1
---
Certificate chain
 0 s:C=PK, ST=Punjab, L=Islamabad, O=Assignment, OU=Dev, CN=nginx
   i:C=PK, ST=Punjab, L=Islamabad, O=Assignment, OU=Dev, CN=nginx
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Dec 28 11:31:08 2025 GMT; NotAfter: Dec 28 11:31:08 2026 GMT
-----BEGIN CERTIFICATE-----
MIIDqzCCApOgAwIBAgIUdQ7wlKqzsANG6kJp+1FcyuqeTJkwDQYJKoZIhvcNAQEL
BQAwZTELMAkGA1UEBhMCUEsxDzANBgNVBAgMBlB1bmphYjESMBAGA1UEBwwJSXNs
YW1hYmFkMRMwEQYDVQQKDApBc3NpZ25tZW50MQwwCgYDVQQLDANEZXYxDjAMBgNV
BAMMBW5naW54MB4XDTI1MTIyODExMzEwOFoXDTI2MTIyODExMzEwOFowZTELMAkG
A1UEBhMCUEsxDzANBgNVBAgMBlB1bmphYjESMBAGA1UEBwwJSXNsYW1hYmFkMRMw
EQYDVQQKDApBc3NpZ25tZW50MQwwCgYDVQQLDANEZXYxDjAMBgNVBAMMBW5naW54
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqO7r1TA5+m76hF5Fg+Ju
30w98VESJRJOd5J0Im8tYCZD70jVfBefMYNLUskCtpAsYCPzL/JOyguWCAWvS+8S
SK1kGh9aaopJZ43PCovWtHMuduLwBGmm7eCbydYVizq2EcUgfwCv5h6v2e61Q7T4
/usIM2JdBnGYMpiXi7AQ9KQNVucpgxLisQbPk0BUFLaBVNzVH9TI5ldQ9H+ZF0S0
8jsCA9pyWN4livbHmfTH9kYrhEQWJcYBOGjPK1fd5hmyk8BPbpwQAVnI7YlbydsW
A7jKJfuUF/PbIq/Hjpe7BBKtP2xeKN40X+HEkb91Y2I4SBok60j+xbNcB0S5yGpW
lwIDAQABo1MwUTAdBgNVHQ4EFgQUtWQn2eU1pACZuliWEkmp5NndQDcwHwYDVR0j
BBgwFoAUtWQn2eU1pACZuliWEkmp5NndQDcwDwYDVR0TAQH/BAUwAwEB/zANBgkq
hkiG9w0BAQsFAAOCAQEAfOo3jRFdRThPEbSEAswYNdS/pwy0S4xBvuo2tK93iYR6
SgykbZF3oj4rIsvkU1cpVdIKgbgi3kUZxghwszbhdTZmQ2/Gw1ukjfuQwhpLmb/P
X40hKSgG8r6OiJW+ZBH4++ZbsfSQUbL+e19y3Cw0YdF1QDDJFqIU4pURlebB6tLF
3alNtr78HXBPUUelz3YoJVZ/MZ39tI109e79Xo8C8oEWd0nBvBPF/td+nnlk+g9m
64/GS8EAPS1CwaUB1Vmx3TiY/S1SpiRPEoV55z0ppiWNX7vx6sXNIrVFrT8IaxPH
X3Zg6ZOA+wislogBm4EzMjhBJq8zrtnsASp6LLM+HA==
-----END CERTIFICATE-----
---
Server certificate
subject=C=PK, ST=Punjab, L=Islamabad, O=Assignment, OU=Dev, CN=nginx
issuer=C=PK, ST=Punjab, L=Islamabad, O=Assignment, OU=Dev, CN=nginx
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
```



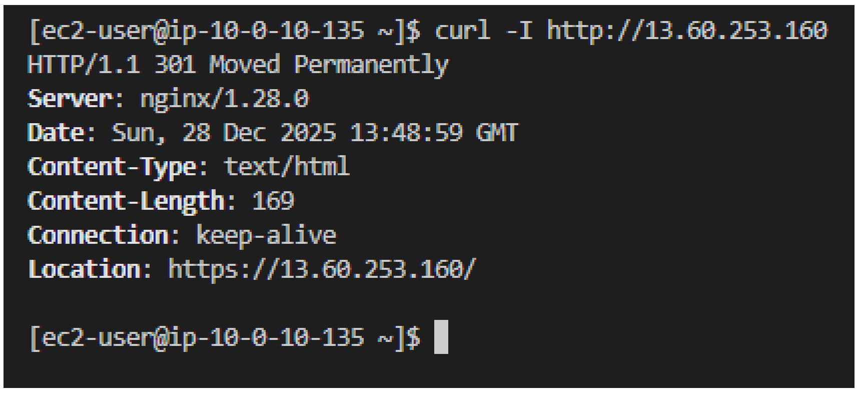
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh ec2-user@13.60.253.160
---
SSL handshake has read 1499 bytes and written 382 bytes
Verification error: self-signed certificate
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
This TLS version forbids renegotiation.
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self-signed certificate)
---
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 0B52B9EDBEA9231C09E9931F6B3EE5EBAFE0CF864C6A38C70DFCDE07A1134957
    Session-ID-ctx:
    Resumption PSK: DAC5D384B853D43F834D4D33993411BA732D36B298412541AC425CBCB4ACFCA5F7E433AD243292DCB1B899AB65C11134
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 47 c0 4e 06 e7 00 aa 39-96 5f cc c8 55 f5 7d 3d   G.N....9._..U.}=
    0010 - 9c 86 62 24 f6 35 07 ed-5b d0 f6 a7 f2 63 46 88   ..b$.5..[....cF.
    0020 - 03 eb e5 f4 1e 20 2f 51-87 70 9b f5 c5 b8 77 b6   ..... /Q.p....w.
    0030 - 33 a9 39 10 d3 64 60 f5-90 ac 7c 54 9c c1 1f 77   3.9..d`...|T...w
    0040 - ab 7f 65 0e 7c 69 73 09-7e b1 2a 2b b6 67 48 19   ..e.|is.~.*+.gH.
    0050 - d0 c4 25 0e da c9 c3 60-26 32 77 41 13 89 35 8a   ..%....`&2wA..5.
    0060 - af a2 bb a7 01 5f a4 74-c9 1f 16 fb 6b 1a 24 3a   ....._.t....k.$:
    0070 - bb ef d7 b5 b3 52 8c f8-c1 fd 7a c3 7f 3c 2a 8a   .....R....z..<*.
    0080 - fe cb 6d f2 72 9e 1f 86-04 e0 9f 8e ff 37 22 57   ..m.r........7"W
    0090 - a6 3e d6 b8 69 81 15 50-58 d6 54 c8 e2 2c 76 40   .>..i..PX.T..,v@
    00a0 - 13 d3 14 ea 7e e6 6c 38-b9 e5 02 8b de e8 89 45   ....~.l8.......E
    00b0 - 1a 75 a0 8c 59 0a 8e dd-d7 72 b9 ca 43 fe 35 be   .u..Y....r..C.5.
    00c0 - f4 78 97 e4 bf a8 13 df-ff 17 4b 08 71 c5 d5 f2   .x........K.q...
    00d0 - 0c 7b e1 6d cc 35 8a 07-83 ed 51 bf 92 53 18 54   .{.m.5....Q..S.T

    Start Time: 1766921631
    Timeout   : 7200 (sec)
    Verify return code: 18 (self-signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```



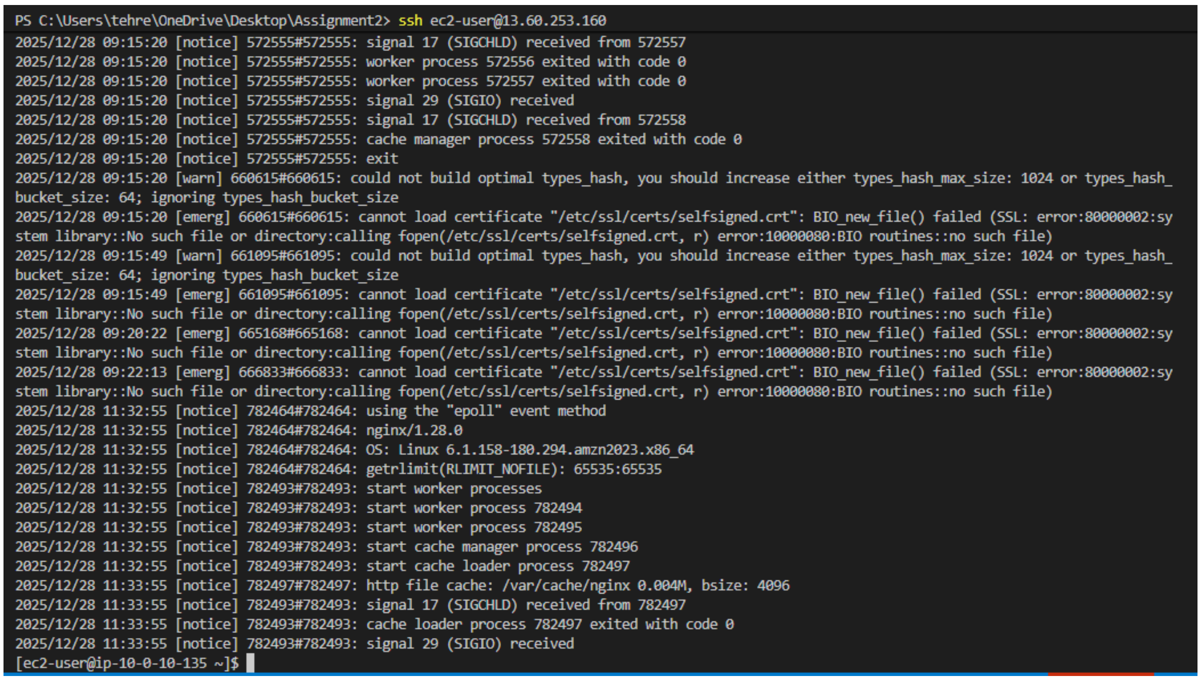
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh ec2-user@13.60.253.160
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: C07FC7F3A0DD00D1ADA7094E849005C8231BA2B8AE162E33AFFE0841E1AE6007
    Session-ID-ctx:
    Resumption PSK: 8D178DD8EF61058FF44CEA0958F16FFD52C8D778DEC069C8A2BE15222576E5188C34BFD846376A23B2730714E835A5D3
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 47 c0 4e 06 e7 00 aa 39-96 5f cc c8 55 f5 7d 3d   G.N....9._..U.}=
    0010 - be 8f 94 6a e0 03 99 84-fa 06 c5 c0 6a 49 03 d1   ...j........jI..
    0020 - 82 b6 8c 1e 79 30 f5 84-1e ee bd 7a 92 a2 d0 e3   ....y0.....z....
    0030 - 72 67 dd 5d db 13 6f 8d-37 53 6c 57 89 5a 1f fd   rg.]..o.7SlW.Z..
    0040 - b9 7c 53 b1 62 09 54 ff-f5 4b 73 8b 0a 92 dd 0b   .|S.b.T..Ks.....
    0050 - ef 85 10 69 ff d1 c8 41-25 a4 1e 16 0b 96 51 07   ...i...A%.....Q.
    0060 - f1 1f 69 a9 f1 e4 86 6b-60 ce 86 95 11 ec 23 cc   ..i....k`.....#.
    0070 - a5 e9 dc 4a d3 ea f5 25-21 17 67 82 62 6d 3f d0   ...J...%!.g.bm?.
    0080 - 03 b7 08 d3 a9 a5 4a 16-dd f6 6f 25 e8 02 97 03   ......J...o%....
    0090 - 24 f7 24 08 42 7c e4 b6-b1 bd 9f 06 be 67 61 ce   $.$.B|.......ga.
    00a0 - 70 ec a1 4f 2b 7c 3d 38-aa a9 87 3a a3 fe 4b 2b   p..O+|=8...:..K+
    00b0 - cc 2f 36 78 07 d5 5d 52-9a d1 57 b2 e5 f9 df 45   ./6x..]R..W....E
    00c0 - d3 97 1f 79 a8 d6 30 2d-05 0f a7 c6 40 a7 a4 14   ...y..0-....@...
    00d0 - a0 f1 a3 97 fa 0f 6f 70-d5 e1 02 82 0b e3 e1 f0   ......op........

    Start Time: 1766921631
    Timeout   : 7200 (sec)
    Verify return code: 18 (self-signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
closed
[ec2-user@ip-10-0-10-135 ~]$
```



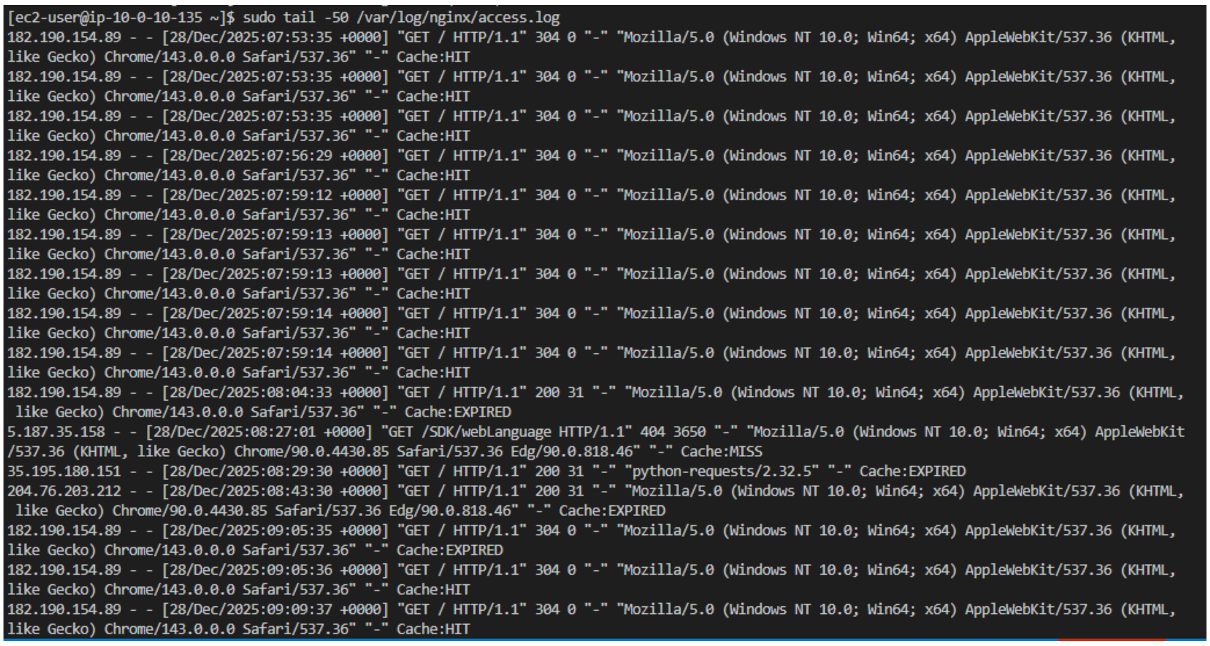
```
[ec2-user@ip-10-0-10-135 ~]$ curl -I -k https://13.60.253.160
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 13:46:20 GMT
Content-Type: text/html
Content-Length: 31
Connection: keep-alive
Last-Modified: Sun, 28 Dec 2025 08:04:12 GMT
ETag: "6950e47c-1f"
X-Cache-Status: EXPIRED
Accept-Ranges: bytes

[ec2-user@ip-10-0-10-135 ~]$
```



```
[ec2-user@ip-10-0-10-135 ~]$ curl -I http://13.60.253.160
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 13:48:59 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://13.60.253.160/

[ec2-user@ip-10-0-10-135 ~]$
```



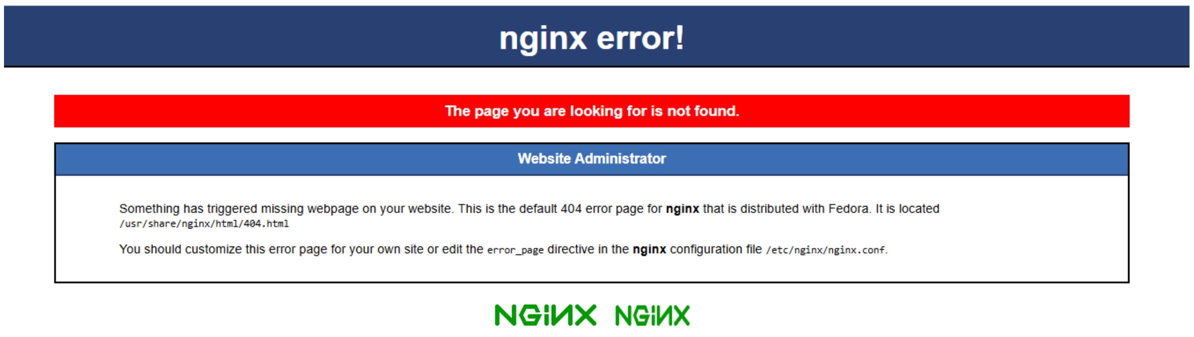
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> ssh ec2-user@13.60.253.160
2025/12/28 09:15:20 [notice] 572555#572555: signal 17 (SIGCHLD) received from 572557
2025/12/28 09:15:20 [notice] 572555#572555: worker process 572556 exited with code 0
2025/12/28 09:15:20 [notice] 572555#572555: worker process 572557 exited with code 0
2025/12/28 09:15:20 [notice] 572555#572555: signal 29 (SIGIO) received
2025/12/28 09:15:20 [notice] 572555#572555: signal 17 (SIGCHLD) received from 572558
2025/12/28 09:15:20 [notice] 572555#572555: cache manager process 572558 exited with code 0
2025/12/28 09:15:20 [notice] 572555#572555: exit
2025/12/28 09:15:20 [warn] 660615#660615: could not build optimal types_hash, you should increase either types_hash_max_size: 1024 or types_hash_
bucket_size: 64; ignoring types_hash_bucket_size
2025/12/28 09:15:20 [emerg] 660615#660615: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file() failed (SSL: error:80000002:sy
stem library::No such file or directory:calling fopen(/etc/ssl/certs/selfsigned.crt, r) error:10000080:BIO routines::no such file)
2025/12/28 09:15:49 [warn] 661095#661095: could not build optimal types_hash, you should increase either types_hash_max_size: 1024 or types_hash_
bucket_size: 64; ignoring types_hash_bucket_size
2025/12/28 09:15:49 [emerg] 661095#661095: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file() failed (SSL: error:80000002:sy
stem library::No such file or directory:calling fopen(/etc/ssl/certs/selfsigned.crt, r) error:10000080:BIO routines::no such file)
2025/12/28 09:20:22 [emerg] 665168#665168: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file() failed (SSL: error:80000002:sy
stem library::No such file or directory:calling fopen(/etc/ssl/certs/selfsigned.crt, r) error:10000080:BIO routines::no such file)
2025/12/28 09:22:13 [emerg] 666833#666833: cannot load certificate "/etc/ssl/certs/selfsigned.crt": BIO_new_file() failed (SSL: error:80000002:sy
stem library::No such file or directory:calling fopen(/etc/ssl/certs/selfsigned.crt, r) error:10000080:BIO routines::no such file)
2025/12/28 11:32:55 [notice] 782464#782464: using the "epoll" event method
2025/12/28 11:32:55 [notice] 782464#782464: nginx/1.28.0
2025/12/28 11:32:55 [notice] 782464#782464: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/28 11:32:55 [notice] 782464#782464: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/28 11:32:55 [notice] 782493#782493: start worker processes
2025/12/28 11:32:55 [notice] 782493#782493: start worker process 782494
2025/12/28 11:32:55 [notice] 782493#782493: start worker process 782495
2025/12/28 11:32:55 [notice] 782493#782493: start cache manager process 782496
2025/12/28 11:32:55 [notice] 782493#782493: start cache loader process 782497
2025/12/28 11:33:55 [notice] 782497#782497: http file cache: /var/cache/nginx 0.004M, bsize: 4096
2025/12/28 11:33:55 [notice] 782493#782493: signal 17 (SIGCHLD) received from 782497
2025/12/28 11:33:55 [notice] 782493#782493: cache loader process 782497 exited with code 0
2025/12/28 11:33:55 [notice] 782493#782493: signal 29 (SIGIO) received
[ec2-user@ip-10-0-10-135 ~]$
```



```
[ec2-user@ip-10-0-10-135 ~]$ sudo tail -50 /var/log/nginx/access.log
182.190.154.89 - - [28/Dec/2025:07:53:35 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:53:35 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:53:35 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:56:29 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:59:12 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:59:13 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:59:13 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:59:14 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:07:59:14 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:08:04:33 +0000] "GET / HTTP/1.1" 200 31 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:EXPIRED
5.187.35.158 - - [28/Dec/2025:08:27:01 +0000] "GET /SDK/webLanguage HTTP/1.1" 404 3650 "-" "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" "-" Cache:MISS
35.195.180.151 - - [28/Dec/2025:08:29:30 +0000] "GET / HTTP/1.1" 200 31 "-" "python-requests/2.32.5" "-" Cache:EXPIRED
204.76.203.212 - - [28/Dec/2025:08:43:30 +0000] "GET / HTTP/1.1" 200 31 "-" "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" "-" Cache:EXPIRED
182.190.154.89 - - [28/Dec/2025:09:05:35 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:EXPIRED
182.190.154.89 - - [28/Dec/2025:09:05:36 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
182.190.154.89 - - [28/Dec/2025:09:09:37 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:HIT
```

Bonus tasks

Task 1



nginx error!

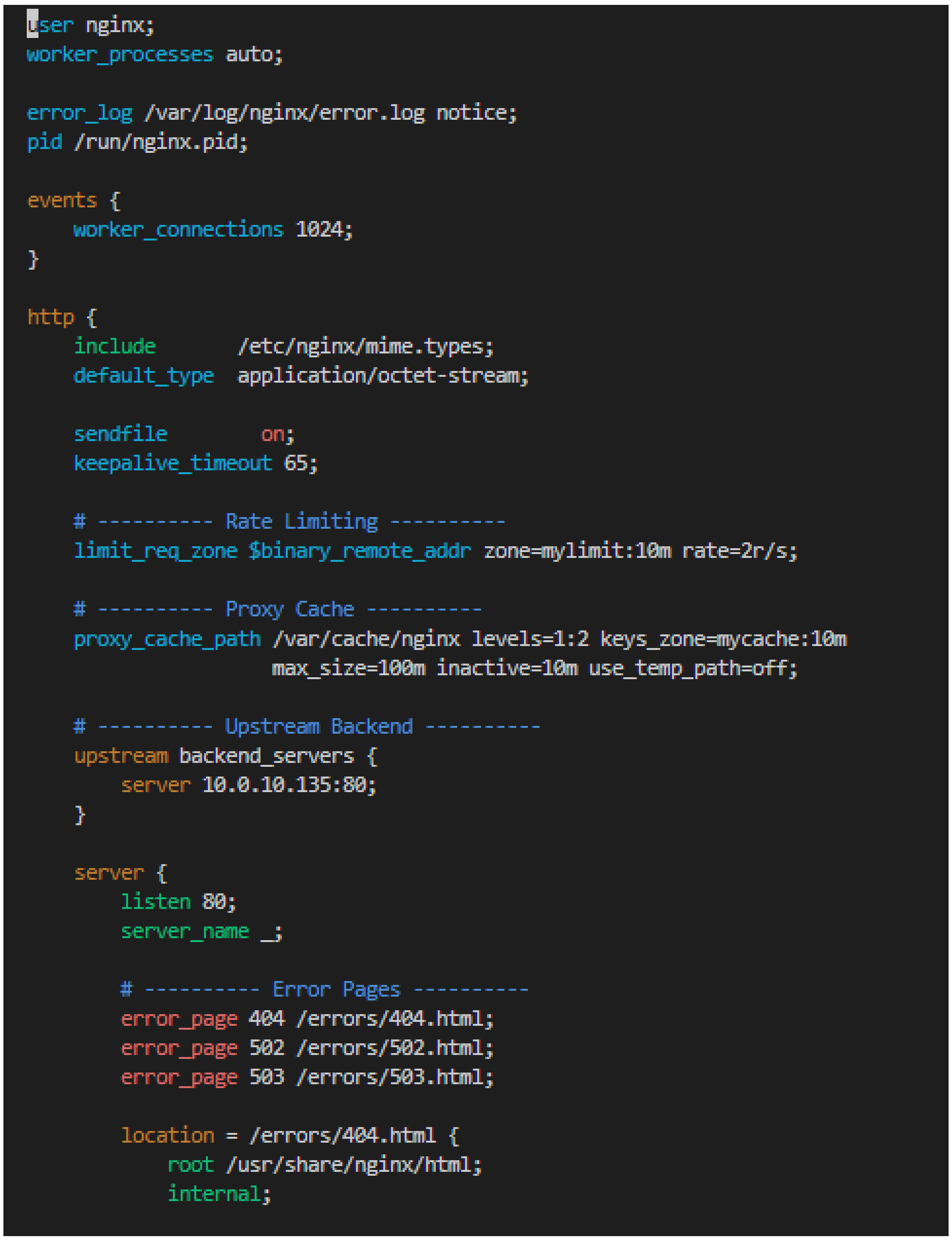
The page you are looking for is not found.

Website Administrator

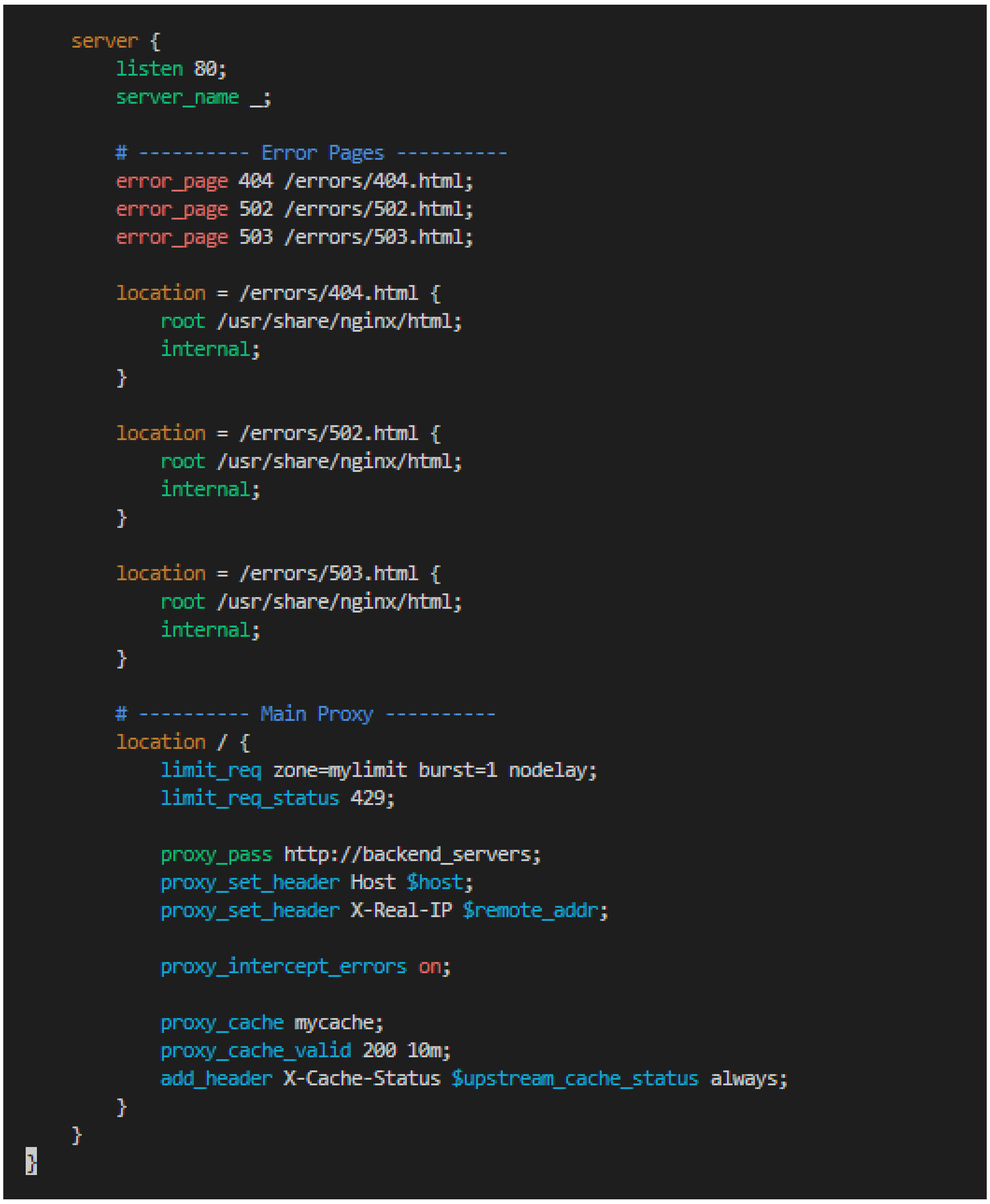
Something has triggered missing webpage on your website. This is the default 404 error page for **nginx** that is distributed with Fedora. It is located `/usr/share/nginx/html/404.html`

You should customize this error page for your own site or edit the `error_page` directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

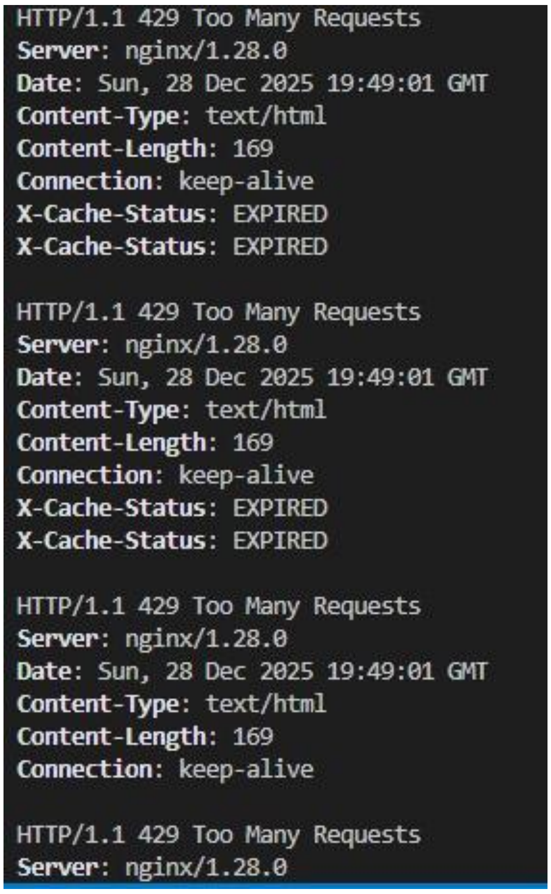
Task 2:



```
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;

    # ---------- Rate Limiting ----------
    limit_req_zone $binary_remote_addr zone=mylimit:10m rate=2r/s;

    # ---------- Proxy Cache ----------
    proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=mycache:10m
                     max_size=100m inactive=10m use_temp_path=off;

    # ---------- Upstream Backend ----------
    upstream backend_servers {
        server 10.0.10.135:80;
    }

    server {
        listen 80;
        server_name _;

        # ---------- Error Pages ----------
        error_page 404 /errors/404.html;
        error_page 502 /errors/502.html;
        error_page 503 /errors/503.html;

        location = /errors/404.html {
            root /usr/share/nginx/html;
            internal;
```

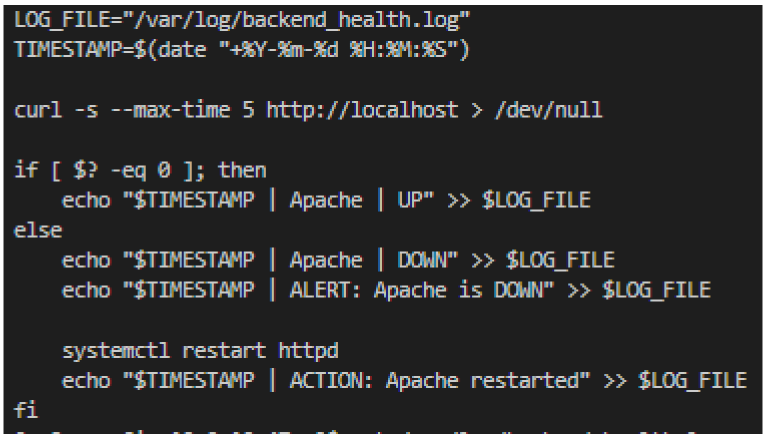


```
    server {
        listen 80;
        server_name _;

        # ---------- Error Pages ----------
        error_page 404 /errors/404.html;
        error_page 502 /errors/502.html;
        error_page 503 /errors/503.html;

        location = /errors/404.html {
            root /usr/share/nginx/html;
            internal;
        }

        location = /errors/502.html {
            root /usr/share/nginx/html;
            internal;
        }

        location = /errors/503.html {
            root /usr/share/nginx/html;
            internal;
        }

        # ---------- Main Proxy ----------
        location / {
            limit_req zone=mylimit burst=1 nodelay;
            limit_req_status 429;

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

            proxy_intercept_errors on;

            proxy_cache mycache;
            proxy_cache_valid 200 10m;
            add_header X-Cache-Status $upstream_cache_status always;
        }
    }
}
```

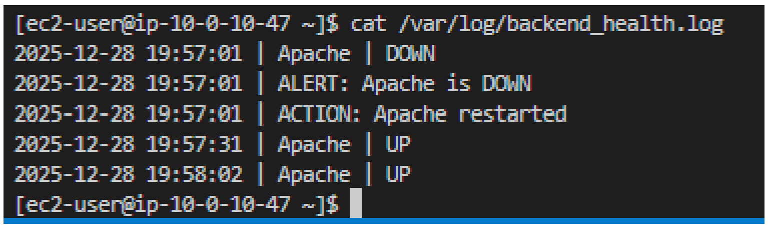


```
HTTP/1.1 429 Too Many Requests
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 19:49:01 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
X-Cache-Status: EXPIRED
X-Cache-Status: EXPIRED

HTTP/1.1 429 Too Many Requests
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 19:49:01 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
X-Cache-Status: EXPIRED
X-Cache-Status: EXPIRED

HTTP/1.1 429 Too Many Requests
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 19:49:01 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive

HTTP/1.1 429 Too Many Requests
Server: nginx/1.28.0
```

Task 3



```
LOG_FILE="/var/log/backend_health.log"
TIMESTAMP=$(date "+%Y-%m-%d %H:%M:%S")

curl -s --max-time 5 http://localhost > /dev/null

if [ $? -eq 0 ]; then
    echo "$TIMESTAMP | Apache | UP" >> $LOG_FILE
else
    echo "$TIMESTAMP | Apache | DOWN" >> $LOG_FILE
    echo "$TIMESTAMP | ALERT: Apache is DOWN" >> $LOG_FILE

    systemctl restart httpd
    echo "$TIMESTAMP | ACTION: Apache restarted" >> $LOG_FILE
fi
```



```
[ec2-user@ip-10-0-10-47 ~]$ cat /var/log/backend_health.log
2025-12-28 19:57:01 | Apache | DOWN
2025-12-28 19:57:01 | ALERT: Apache is DOWN
2025-12-28 19:57:01 | ACTION: Apache restarted
2025-12-28 19:57:31 | Apache | UP
2025-12-28 19:58:02 | Apache | UP
[ec2-user@ip-10-0-10-47 ~]$
```

Part 6: Cleanup

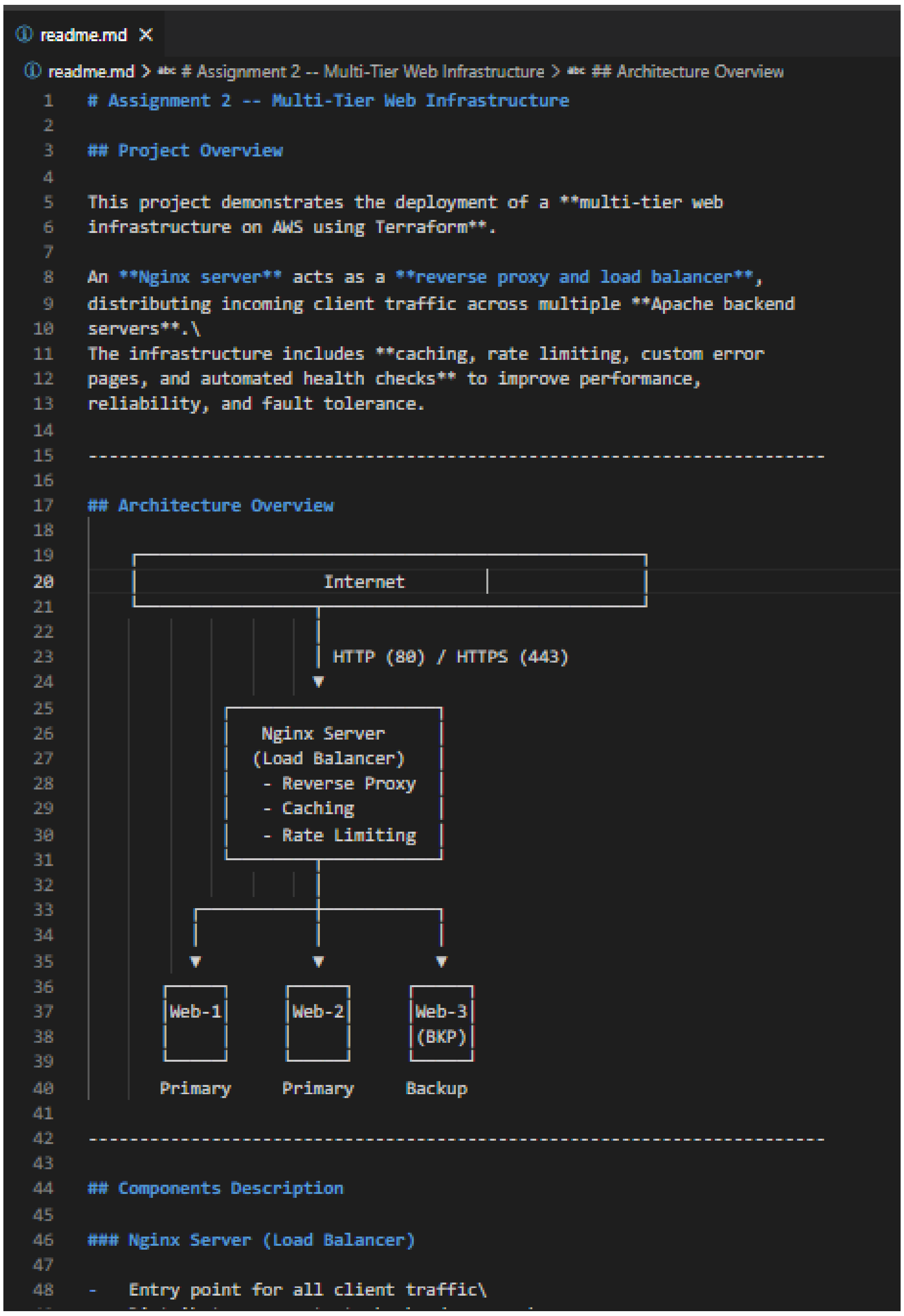
All resources were destroyed using Terraform to avoid unnecessary AWS charges. The cleanup process removed EC2 instances, networking components, and security groups.

Command used:

terraform destroy

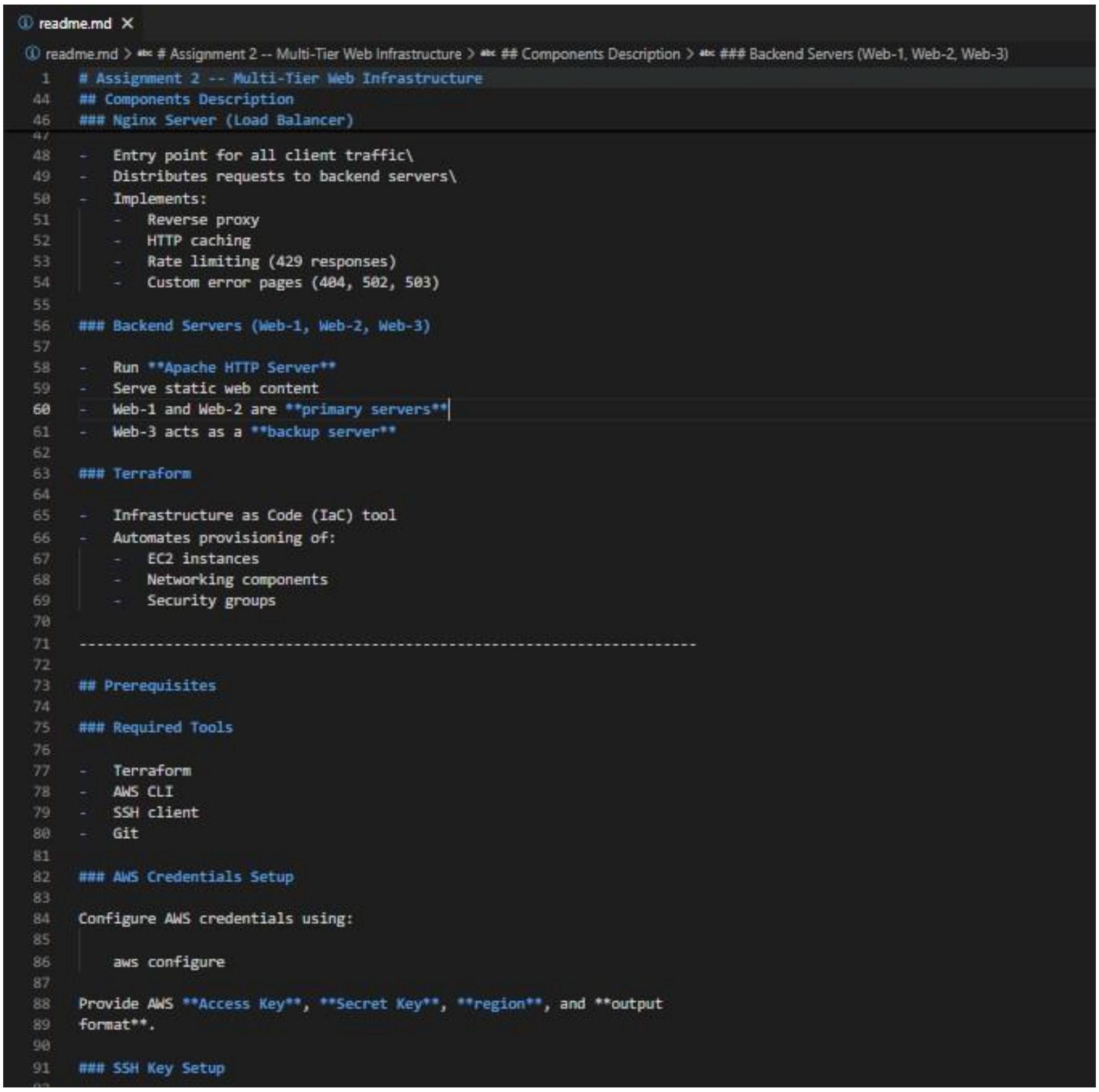
The Terraform state file confirmed that no resources remained.

6.1

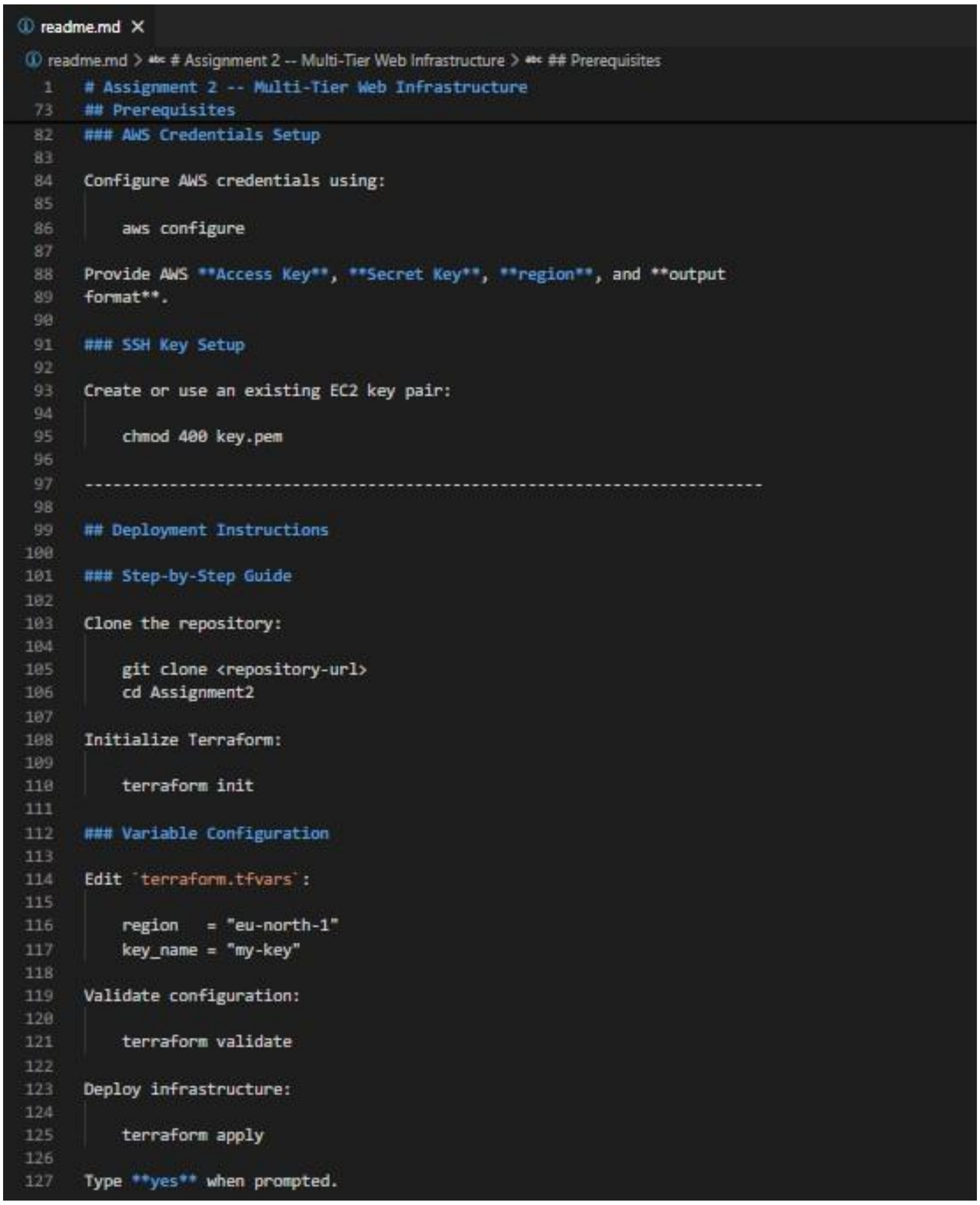


readme.md
readme.md > # Assignment 2 -- Multi-Tier Web Infrastructure > ## Architecture Overview

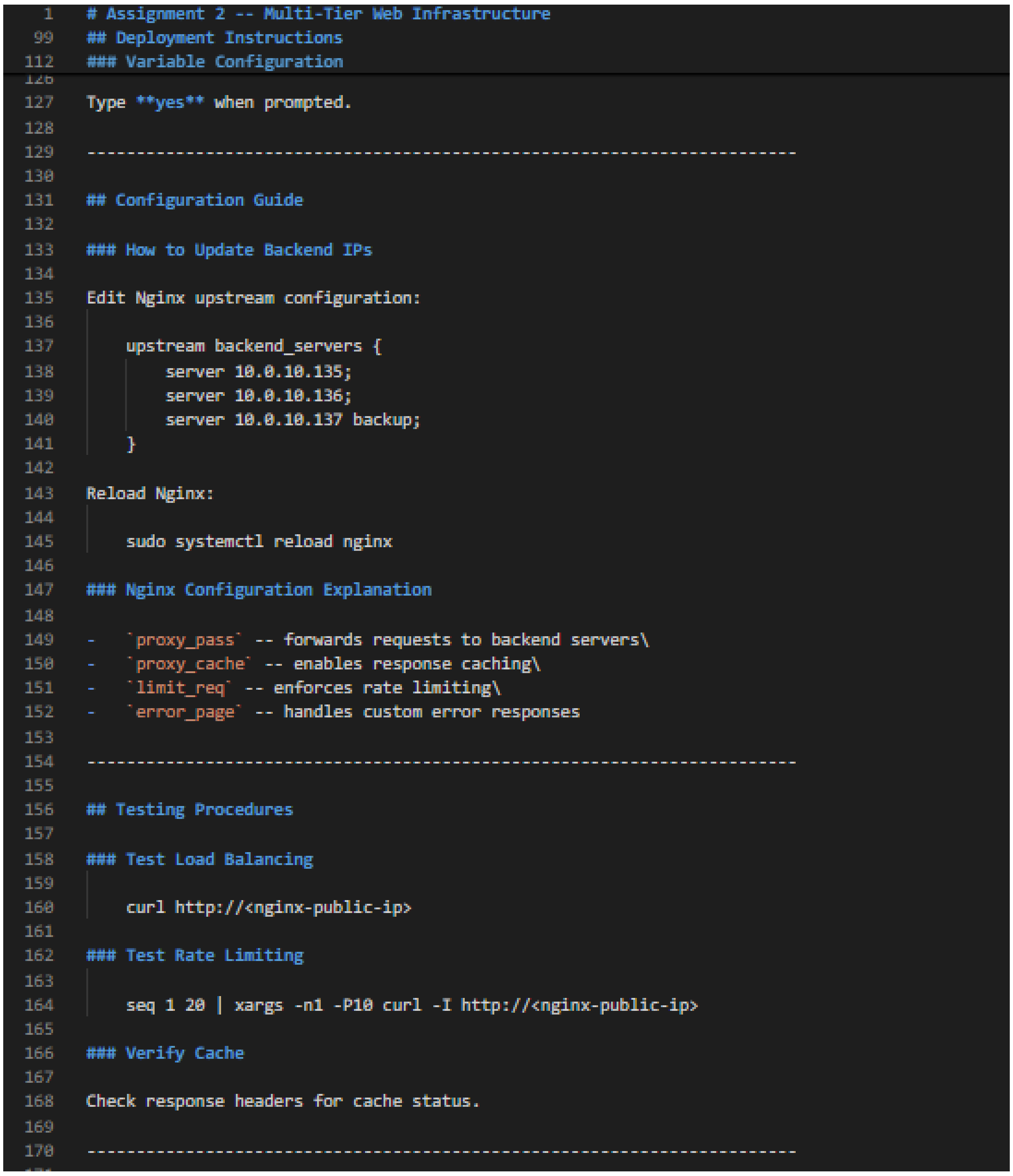
```
# Assignment 2 -- Multi-Tier Web Infrastructure

## Project Overview

This project demonstrates the deployment of a **multi-tier web
infrastructure on AWS using Terraform**.

An **Nginx server** acts as a **reverse proxy and load balancer**,
distributing incoming client traffic across multiple **Apache backend
servers**.\
The infrastructure includes **caching, rate limiting, custom error
pages, and automated health checks** to improve performance,
reliability, and fault tolerance.

------------------------------------------------------------------------

## Architecture Overview


    ┌──────────────────────────────────────────┐
    │                 Internet                 │
    └──────────────────────────────────────────┘
                          │
                          │ HTTP (80) / HTTPS (443)
                          ▼
               ┌─────────────────────┐
               │    Nginx Server     │
               │  (Load Balancer)    │
               │   - Reverse Proxy   │
               │   - Caching         │
               │   - Rate Limiting   │
               └─────────────────────┘
                          │
                ┌─────────┼─────────┐
                │         │         │
                ▼         ▼         ▼
             ┌─────┐   ┌─────┐   ┌─────┐
             │Web-1│   │Web-2│   │Web-3│
             │     │   │     │   │(BKP)│
             └─────┘   └─────┘   └─────┘
             Primary   Primary   Backup

------------------------------------------------------------------------

## Components Description

### Nginx Server (Load Balancer)

-   Entry point for all client traffic\
```


readme.md
readme.md > # Assignment 2 -- Multi-Tier Web Infrastructure > ## Components Description > ### Backend Servers (Web-1, Web-2, Web-3)

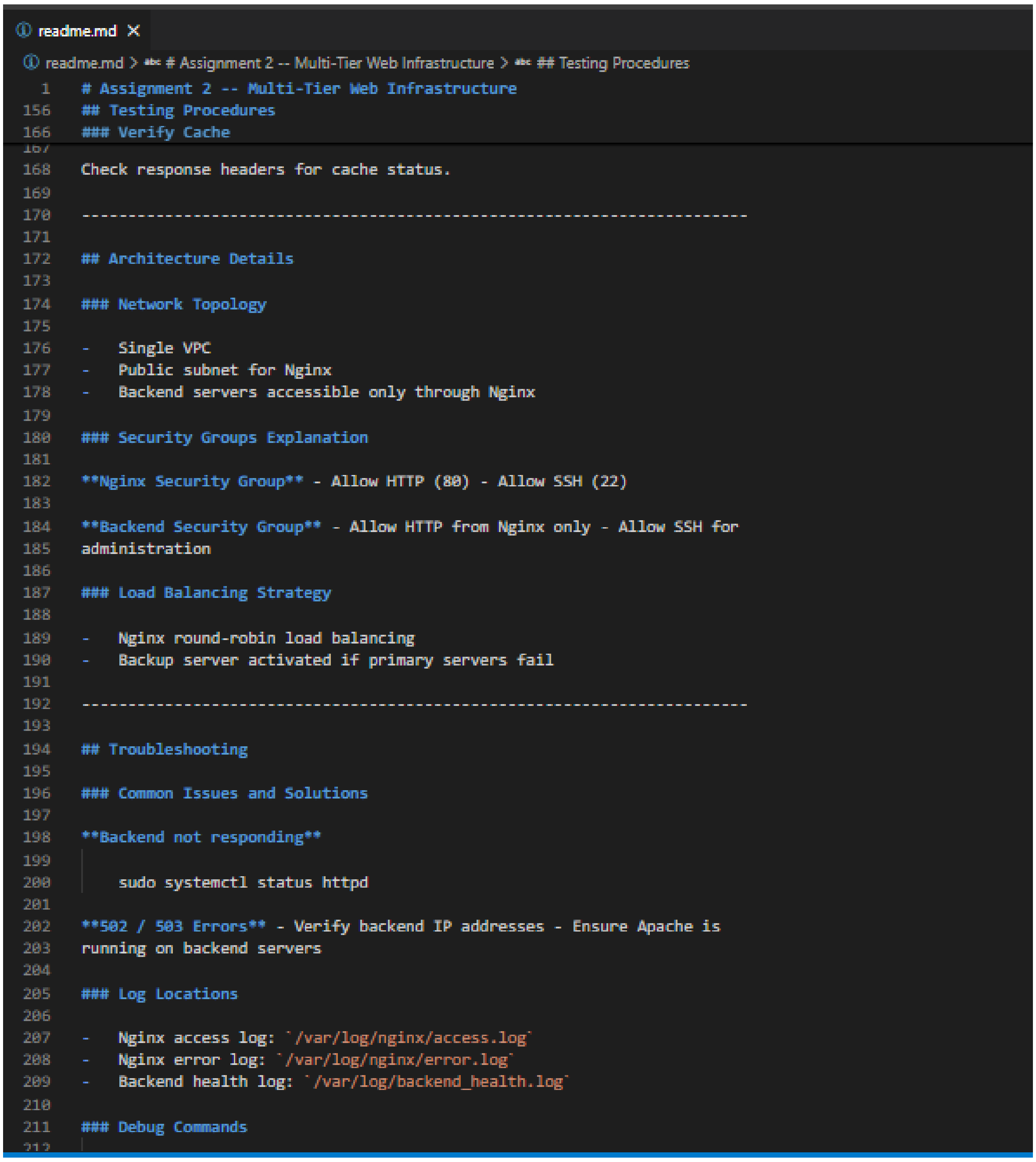
```
# Assignment 2 -- Multi-Tier Web Infrastructure
## Components Description
### Nginx Server (Load Balancer)

-   Entry point for all client traffic\
-   Distributes requests to backend servers\
-   Implements:
    -   Reverse proxy
    -   HTTP caching
    -   Rate limiting (429 responses)
    -   Custom error pages (404, 502, 503)

### Backend Servers (Web-1, Web-2, Web-3)

-   Run **Apache HTTP Server**
-   Serve static web content
-   Web-1 and Web-2 are **primary servers**
-   Web-3 acts as a **backup server**

### Terraform

-   Infrastructure as Code (IaC) tool
-   Automates provisioning of:
    -   EC2 instances
    -   Networking components
    -   Security groups

------------------------------------------------------------------------

## Prerequisites

### Required Tools

-   Terraform
-   AWS CLI
-   SSH client
-   Git

### AWS Credentials Setup

Configure AWS credentials using:

    aws configure

Provide AWS **Access Key**, **Secret Key**, **region**, and **output
format**.

### SSH Key Setup
```


readme.md ×
readme.md > # Assignment 2 -- Multi-Tier Web Infrastructure > ## Prerequisites

```
# Assignment 2 -- Multi-Tier Web Infrastructure
## Prerequisites
### AWS Credentials Setup

Configure AWS credentials using:

    aws configure

Provide AWS **Access Key**, **Secret Key**, **region**, and **output
format**.

### SSH Key Setup

Create or use an existing EC2 key pair:

    chmod 400 key.pem

-----------------------------------------------------------------------

## Deployment Instructions

### Step-by-Step Guide

Clone the repository:

    git clone <repository-url>
    cd Assignment2

Initialize Terraform:

    terraform init

### Variable Configuration

Edit `terraform.tfvars`:

    region   = "eu-north-1"
    key_name = "my-key"

Validate configuration:

    terraform validate

Deploy infrastructure:

    terraform apply

Type **yes** when prompted.
```



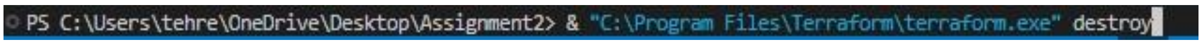
```
# Assignment 2 -- Multi-Tier Web Infrastructure
## Deployment Instructions
### Variable Configuration
Type **yes** when prompted.

-----------------------------------------------------------------------

## Configuration Guide

### How to Update Backend IPs

Edit Nginx upstream configuration:

    upstream backend_servers {
        server 10.0.10.135;
        server 10.0.10.136;
        server 10.0.10.137 backup;
    }

Reload Nginx:

    sudo systemctl reload nginx

### Nginx Configuration Explanation

-   `proxy_pass` -- forwards requests to backend servers\
-   `proxy_cache` -- enables response caching\
-   `limit_req` -- enforces rate limiting\
-   `error_page` -- handles custom error responses

-----------------------------------------------------------------------

## Testing Procedures

### Test Load Balancing

    curl http://<nginx-public-ip>

### Test Rate Limiting

    seq 1 20 | xargs -n1 -P10 curl -I http://<nginx-public-ip>

### Verify Cache

Check response headers for cache status.

-----------------------------------------------------------------------
```


readme.md ×

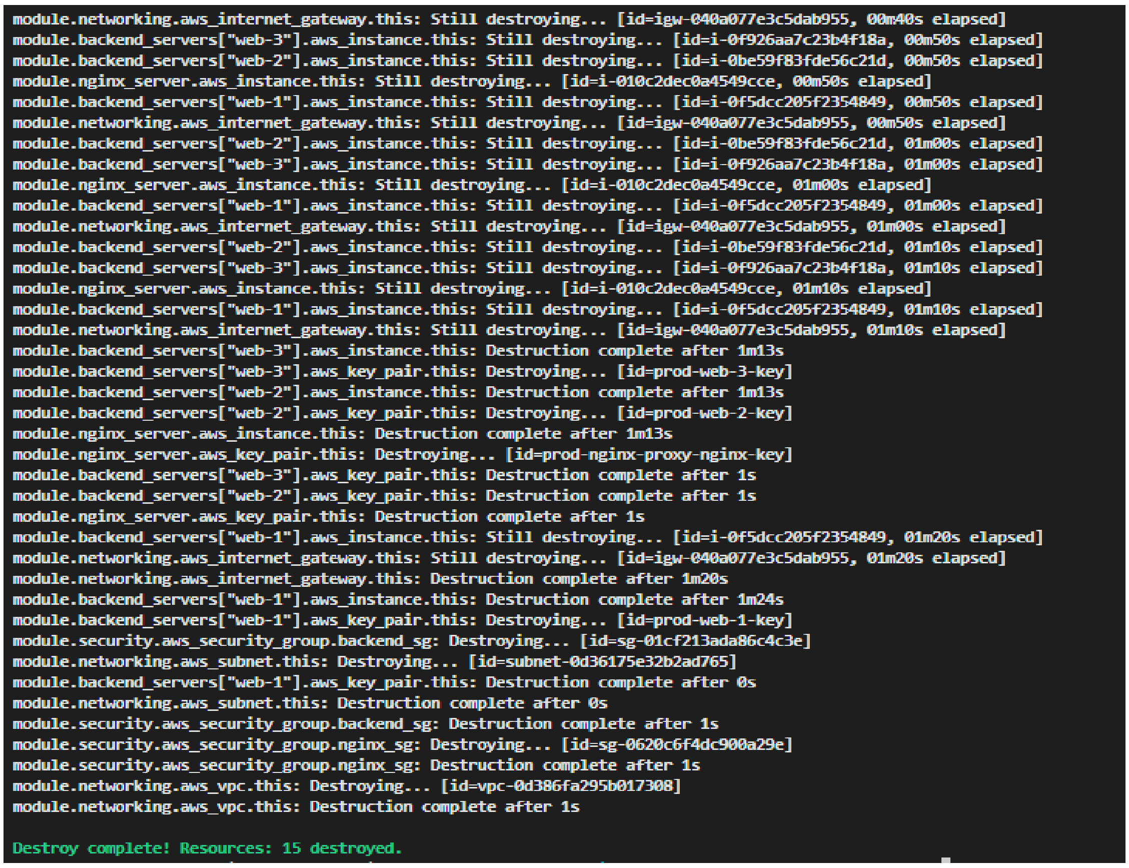
readme.md > # Assignment 2 -- Multi-Tier Web Infrastructure > ## Testing Procedures

```
# Assignment 2 -- Multi-Tier Web Infrastructure
## Testing Procedures
### Verify Cache

Check response headers for cache status.

-----------------------------------------------------------------------

## Architecture Details

### Network Topology

-   Single VPC
-   Public subnet for Nginx
-   Backend servers accessible only through Nginx

### Security Groups Explanation

**Nginx Security Group** - Allow HTTP (80) - Allow SSH (22)

**Backend Security Group** - Allow HTTP from Nginx only - Allow SSH for
administration

### Load Balancing Strategy

-   Nginx round-robin load balancing
-   Backup server activated if primary servers fail

-----------------------------------------------------------------------

## Troubleshooting

### Common Issues and Solutions

**Backend not responding**

    sudo systemctl status httpd

**502 / 503 Errors** - Verify backend IP addresses - Ensure Apache is
running on backend servers

### Log Locations

-   Nginx access log: `/var/log/nginx/access.log`
-   Nginx error log: `/var/log/nginx/error.log`
-   Backend health log: `/var/log/backend_health.log`

### Debug Commands
```

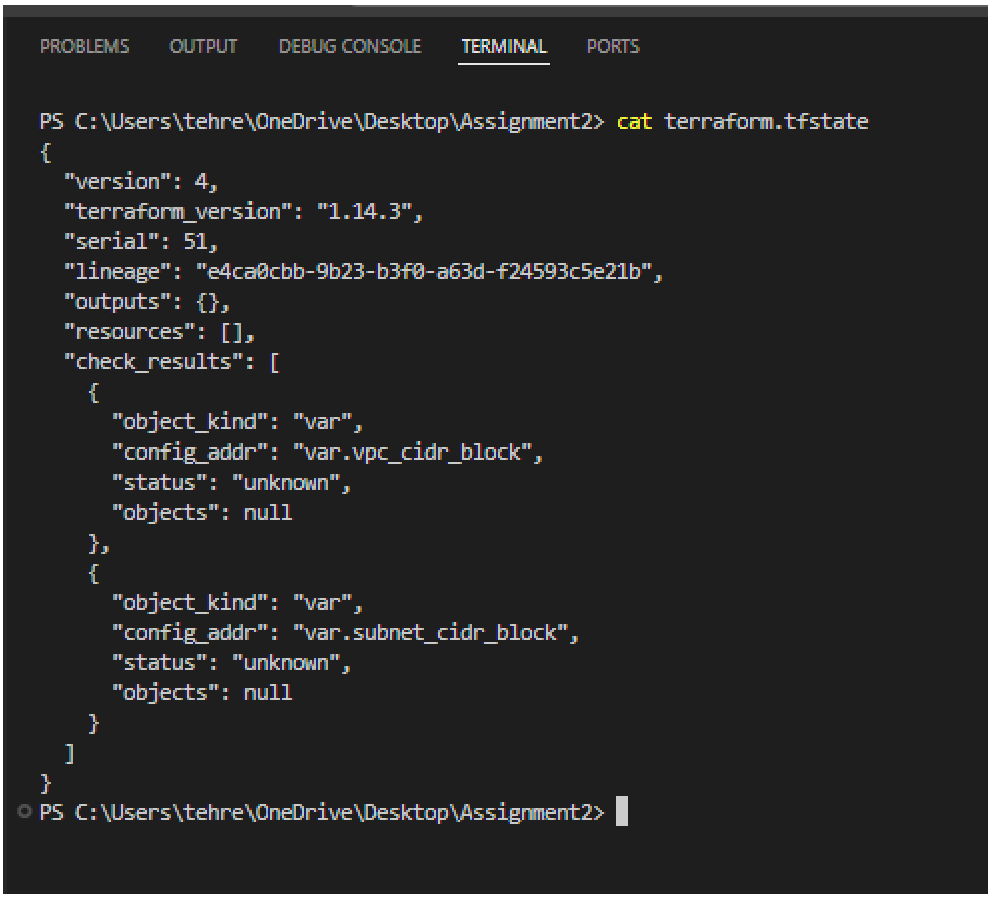
6.2



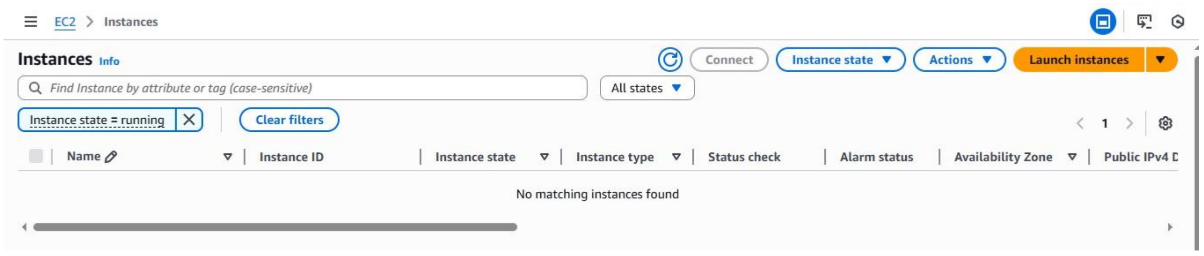
```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> & "C:\Program Files\Terraform\terraform.exe" destroy
```



```
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-040a077e3c5dab955, 00m40s elapsed]
module.backend_servers["web-3"].aws_instance.this: Still destroying... [id=i-0f926aa7c23b4f18a, 00m50s elapsed]
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-0be59f83fde56c21d, 00m50s elapsed]
module.nginx_server.aws_instance.this: Still destroying... [id=i-010c2dec0a4549cce, 00m50s elapsed]
module.backend_servers["web-1"].aws_instance.this: Still destroying... [id=i-0f5dcc205f2354849, 00m50s elapsed]
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-040a077e3c5dab955, 00m50s elapsed]
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-0be59f83fde56c21d, 01m00s elapsed]
module.backend_servers["web-3"].aws_instance.this: Still destroying... [id=i-0f926aa7c23b4f18a, 01m00s elapsed]
module.nginx_server.aws_instance.this: Still destroying... [id=i-010c2dec0a4549cce, 01m00s elapsed]
module.backend_servers["web-1"].aws_instance.this: Still destroying... [id=i-0f5dcc205f2354849, 01m00s elapsed]
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-040a077e3c5dab955, 01m00s elapsed]
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-0be59f83fde56c21d, 01m10s elapsed]
module.backend_servers["web-3"].aws_instance.this: Still destroying... [id=i-0f926aa7c23b4f18a, 01m10s elapsed]
module.nginx_server.aws_instance.this: Still destroying... [id=i-010c2dec0a4549cce, 01m10s elapsed]
module.backend_servers["web-1"].aws_instance.this: Still destroying... [id=i-0f5dcc205f2354849, 01m10s elapsed]
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-040a077e3c5dab955, 01m10s elapsed]
module.backend_servers["web-3"].aws_instance.this: Destruction complete after 1m13s
module.backend_servers["web-3"].aws_key_pair.this: Destroying... [id=prod-web-3-key]
module.backend_servers["web-2"].aws_instance.this: Destruction complete after 1m13s
module.backend_servers["web-2"].aws_key_pair.this: Destroying... [id=prod-web-2-key]
module.nginx_server.aws_instance.this: Destruction complete after 1m13s
module.nginx_server.aws_key_pair.this: Destroying... [id=prod-nginx-proxy-nginx-key]
module.backend_servers["web-3"].aws_key_pair.this: Destruction complete after 1s
module.backend_servers["web-2"].aws_key_pair.this: Destruction complete after 1s
module.nginx_server.aws_key_pair.this: Destruction complete after 1s
module.backend_servers["web-1"].aws_instance.this: Still destroying... [id=i-0f5dcc205f2354849, 01m20s elapsed]
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-040a077e3c5dab955, 01m20s elapsed]
module.networking.aws_internet_gateway.this: Destruction complete after 1m20s
module.backend_servers["web-1"].aws_instance.this: Destruction complete after 1m24s
module.backend_servers["web-1"].aws_key_pair.this: Destroying... [id=prod-web-1-key]
module.security.aws_security_group.backend_sg: Destroying... [id=sg-01cf213ada86c4c3e]
module.networking.aws_subnet.this: Destroying... [id=subnet-0d36175e32b2ad765]
module.backend_servers["web-1"].aws_key_pair.this: Destruction complete after 0s
module.networking.aws_subnet.this: Destruction complete after 0s
module.security.aws_security_group.backend_sg: Destruction complete after 1s
module.security.aws_security_group.nginx_sg: Destroying... [id=sg-0620c6f4dc900a29e]
module.security.aws_security_group.nginx_sg: Destruction complete after 1s
module.networking.aws_vpc.this: Destroying... [id=vpc-0d386fa295b017308]
module.networking.aws_vpc.this: Destruction complete after 1s

Destroy complete! Resources: 15 destroyed.
```



```
PS C:\Users\tehre\OneDrive\Desktop\Assignment2> cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 51,
  "lineage": "e4ca0cbb-9b23-b3f0-a63d-f24593c5e21b",
  "outputs": {},
  "resources": [],
  "check_results": [
    {
      "object_kind": "var",
      "config_addr": "var.vpc_cidr_block",
      "status": "unknown",
      "objects": null
    },
    {
      "object_kind": "var",
      "config_addr": "var.subnet_cidr_block",
      "status": "unknown",
      "objects": null
    }
  ]
}
PS C:\Users\tehre\OneDrive\Desktop\Assignment2>
```



No matching instances found

3. Testing Results

Testing was conducted to validate the correctness, reliability, performance, and security of the deployed cloud-based web architecture. The objective of this phase was to ensure that all core architectural goals—such as load balancing, caching efficiency, high availability, and secure access—were achieved successfully. Multiple functional and non-functional tests were performed on the deployed AWS infrastructure to evaluate real-world behavior under different conditions.

The testing process focused on five major areas: load balancing, cache performance, high availability, security, and performance metrics. Each test category is discussed in detail below, along with observations and outcomes.

3.1 Load Balancing Tests

The purpose of the load balancing tests was to verify that incoming client requests were distributed evenly across the available backend servers. The Nginx server was configured as a load balancer using a round-robin strategy, ensuring fair request distribution between Web-1 and Web-2.

To test this functionality, multiple HTTP requests were sent to the public IP address of the Nginx server using a web browser and command-line tools. Each backend Apache server was configured to display a unique identifier or hostname in the response message. This allowed easy identification of which backend server processed a particular request.

The results showed that consecutive requests were alternately handled by Web-1 and Web-2, confirming that the load balancer was functioning correctly. No single server was observed to receive a disproportionate number of requests under normal load conditions. This demonstrates that the system can efficiently distribute traffic and prevent overload on a single backend server.

Overall, the load balancing tests confirmed that the architecture supports horizontal scaling and improves fault tolerance by avoiding dependency on a single server.

3.2 Cache Performance Tests

Caching tests were performed to evaluate the effectiveness of the Nginx caching mechanism in improving response time and reducing backend server load. The cache was configured on the Nginx server to store frequently requested web content.

To verify caching behavior, browser developer tools were used to inspect HTTP response headers. The cache status was observed using the custom response header X-Cache-Status.

During the initial request to a web resource, the response header showed X-Cache-Status: MISS, indicating that the content was fetched from a backend Apache server and stored in the cache. Subsequent requests to the same resource returned X-Cache-Status: HIT, confirming that the response was served directly from the Nginx cache without contacting the backend servers.

This behavior demonstrates that caching was working as expected. The use of caching significantly reduced response times for repeated requests and decreased the processing load on backend servers. As a result, the system is more efficient and capable of handling higher traffic volumes.

3.3 High Availability Tests

High availability testing was conducted to ensure that the system remains operational even when one or more backend servers fail. This is a critical requirement for cloud-based systems where uninterrupted service is expected.

To test this scenario, one of the primary backend servers (Web-1 or Web-2) was manually stopped during active operation. While the server was ofline, client requests were continuously sent to the Nginx server.

The results showed that traffic was automatically routed to the remaining available servers. When both primary servers were unavailable, the backup server (Web-3) successfully handled incoming requests. This confirms that the backup mechanism was correctly configured and that the Nginx server could dynamically route traffic based on server availability.

These results demonstrate that the architecture provides fault tolerance and ensures continuous service availability, even in the event of server failure.

3.4 Security Tests

Security testing focused on validating access control rules and ensuring that backend servers were protected from direct public access. AWS Security Groups were used as virtual firewalls to enforce network security policies.

Several security tests were performed, including attempting to directly access backend Apache servers using their public IP addresses. These attempts failed, confirming that backend servers were not exposed to the internet and only accepted traffic from the Nginx server.

Additionally, SSH access was tested to verify that only authorized IP addresses could connect to EC2 instances. Unauthorized access attempts were blocked as expected. HTTP and HTTPS access were limited strictly to the Nginx server, reducing the overall attack surface.

The results confirm that the security configuration follows the principle of least privilege and effectively protects the infrastructure against unauthorized access.

3.5 Performance Metrics

Performance metrics were analyzed to measure the impact of load balancing and caching on system efficiency. Response time, server load, and request handling behavior were observed during testing.

With caching enabled, response times for repeated requests were noticeably faster compared to initial requests. Backend server load was reduced, as cached responses were served directly by the Nginx server. Load balancing further ensured that server resources were utilized evenly, preventing bottlenecks.

Although no formal benchmarking tools were used, manual observation confirmed improved responsiveness and stability under normal traffic conditions. These improvements demonstrate that the architectural design effectively enhances system performance and scalability.

3.6 Summary of Testing Results

The testing phase confirmed that all major architectural components functioned as intended. Load balancing ensured fair request distribution, caching improved performance, high availability mechanisms provided fault tolerance, and security controls protected backend resources. Performance metrics indicated reduced response times and balanced server utilization.

Overall, the testing results validate that the deployed AWS infrastructure meets the functional and non-functional requirements of a reliable, secure, and scalable cloud-based web application.

4. Challenges & Solutions

During the deployment and testing of the AWS-based web architecture using Terraform, several technical challenges were encountered. These challenges were resolved through configuration changes, debugging, and careful analysis. This section discusses the major challenges faced, the solutions applied, and the lessons learned during the project.

4.1 SSH Connectivity and Access Control Issues

Challenge

One of the initial challenges faced was the inability to establish SSH connections with the EC2 instances. Attempts to connect resulted in timeouts or permission errors. This issue was primarily caused by misconfigured **security group inbound rules** and incorrect handling of SSH key pairs.

Solution

To resolve this issue, the security group configuration was reviewed and updated. Port 22 (SSH) was explicitly allowed only from authorized IP addresses. The correct key pair associated with the EC2 instances was verified and used for authentication. After these changes, secure SSH access was successfully established.

4.2 Backend Server Communication Failure

Challenge

Another major challenge was that backend Apache servers were not responding to requests forwarded by the Nginx server. Although the backend instances were running, client requests resulted in connection timeouts or error messages. This indicated an issue in the reverse proxy or backend configuration.

Solution

The Nginx reverse proxy configuration was carefully examined and corrected. Backend server IP addresses and listening ports were verified, and the proxy settings were updated accordingly. Apache services on the backend servers were restarted to ensure proper operation. After applying these changes, requests were successfully routed to the backend servers.

4.3 Cache Header Visibility and Validation Issues

Challenge

During cache performance testing, the expected cache response headers (X-Cache-Status) were not appearing in browser developer tools. This made it difficult to confirm whether caching was working as intended.

Solution

To resolve this issue, browser cache was cleared, and the Nginx service was reloaded to apply the caching configuration. Response headers were then inspected again. After these steps, cache headers correctly displayed MISS for initial requests and HIT for subsequent requests, confirming proper cache functionality.

4.4 Lessons Learned

This project provided several important insights into cloud infrastructure deployment. One key lesson learned was the **importance of modular and well-structured Terraform code**, which simplifies debugging and future modifications. Additionally, effective use of logs, system status checks, and the AWS Management Console proved essential for identifying and resolving configuration issues.

The project also highlighted the **value of Infrastructure as Code (IaC)** for creating repeatable, consistent, and reliable deployments. Overall, these challenges strengthened practical understanding of real-world cloud architecture and troubleshooting techniques.

5. Conclusion

This project successfully demonstrated the design, deployment, and evaluation of a cloud-based web infrastructure using **Amazon Web Services (AWS)** and **Terraform**. A three-tier architecture was implemented in which an **Nginx server** functioned as a reverse proxy, load balancer, and cache layer, while multiple **Apache backend servers** served application content. The infrastructure was deployed within a custom **Virtual Private Cloud (VPC)** with carefully configured subnets and security groups to ensure controlled and secure communication between system components.

All major functional objectives were achieved. Load balancing was verified through even distribution of requests across backend servers, caching significantly improved response times, and high availability was ensured through the use of multiple backend servers and a backup node. Security testing confirmed that backend servers were protected from direct internet access and that communication was restricted according to the principle of least privilege. The testing results validated the reliability, performance, and security of the deployed architecture.

Throughout the project, several technical skills were developed and strengthened. Practical experience was gained in **Infrastructure as Code (IaC)** using Terraform, including automated provisioning of EC2 instances, networking resources, and security configurations. Hands-on experience with **Nginx configuration**, **Apache web servers**, **AWS networking**, and **cloud security concepts** was also acquired. Additionally, troubleshooting skills were enhanced through the use of system logs, browser developer tools, and the AWS Management Console.

Despite the successful implementation, there are several opportunities for future improvement. The architecture can be enhanced by introducing **Auto Scaling Groups** to automatically adjust the number of backend servers based on traffic demand. Using **AWS Elastic Load Balancer (ELB)** and **Route 53** could further improve scalability and availability. Monitoring tools such as **CloudWatch** can be integrated for real-time performance tracking, and advanced security features like **AWS WAF** can be added to protect against web-based attacks.

In conclusion, this project provided valuable hands-on experience with cloud infrastructure design and deployment. It demonstrated how modern cloud technologies and automation tools can be combined to build scalable, secure, and reliable web systems, laying a strong foundation for future work in cloud computing and DevOps practices.

6. Appendices

1. Terraform configuration files such as main.tf, variables.tf, and locals.tf are included. These files define the complete infrastructure setup.
2. Webserver modules are included to create the Nginx server and backend servers in a reusable way.
3. The apache-setup.sh script is included for configuring backend web servers.
4. The nginx-setup.sh script is included for configuring the Nginx reverse proxy server.