Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

Assignment 2

Advanced Terraform & Nginx Multi-Tier Architecture

Name : Shumaila Munsib

Roll No: 2023-BSE-062

Course Title: Cloud Computing

Submitted to : Sir Waqas

TABLE OF CONTENTS

1. SUMMARY

This report documents the design, implementation, and testing of a high-availability multi-tier web infrastructure deployed on Amazon Web Services (AWS) using Terraform and Nginx.

Project Overview:
The assignment required deploying a production-ready web infrastructure consisting of one Nginx reverse proxy/load balancer and three backend Apache web servers, all managed as Infrastructure as Code (IaC) using Terraform modules.

Infrastructure Deployed:
- 1 Nginx server: Configured as reverse proxy and load balancer with HTTPS, SSL/TLS encryption, caching, and security headers
- 3 Backend web servers: Two primary servers (web-1, web-2) for active load balancing and one backup server (web-3) for high availability
- Custom VPC: Isolated network environment with public subnet
- Security Groups: Firewall rules restricting access appropriately
- Modular Terraform Code: Reusable modules for networking, security, and compute resources

Key Achievements:
Successfully deployed all infrastructure components using Terraform
Implemented round-robin load balancing between primary servers
Configured automatic failover to backup server when primaries fail
Enabled HTTPS with self-signed SSL certificates and security headers
Implemented Nginx caching for improved performance
Verified all functionality through comprehensive testing
Created reusable, modular Terraform code following best practices

Technical Stack:
- Cloud Provider: AWS (EC2, VPC, Security Groups)
- IaC Tool: Terraform v1.x
- Load Balancer: Nginx 1.28.0
- Web Server: Apache HTTP Server 2.4

- Operating System: Amazon Linux 2
- SSL/TLS: OpenSSL with self-signed certificates

The project successfully demonstrates cloud infrastructure automation, load balancing, high availability patterns, and security best practices. All testing confirmed that the infrastructure operates as designed with proper load distribution, caching, failover capabilities, and secure communication.

2. ARCHITECTURE DESIGN

2.1 Architecture Diagram

```
                 Internet
                    ↓
            [Internet Gateway]
                    ↓
              VPC (10.0.0.0/16)
                    ↓
           Public Subnet (10.0.10.0/24)
                    ↓
        ┌─────────────────────────────────┐
        │   Nginx Load Balancer      │
        │   - HTTPS (443)          │
        │   - HTTP Redirect (80)     │
        │   - SSL/TLS             │
        │   - Caching             │
        │   - Security Headers       │
        └───────────────┬─────────────────┘
                  │
     ┌────────────────────┼────────────────────┐
     ↓            ↓           ↓
 ┌──────────┐   ┌──────────┐   ┌──────────┐
 │  web-1  │   │ web-2  │   │ web-3  │
 │ Primary │   │ Primary │   │ Backup  │
 │ Apache  │   │ Apache  │   │ Apache  │
 │  :80   │   │  :80   │   │  :80   │
 └──────────┘   └──────────┘   └──────────┘
```

Network Flow:
1. User requests reach Internet Gateway
2. Traffic routed to Nginx Load Balancer (HTTPS)
3. Nginx distributes requests to backend servers
4. Round-robin between web-1 and web-2 (primary)
5. web-3 activates only when both primaries fail

2.2 Component Descriptions

Nginx Load Balancer:
- Role: Reverse proxy and load balancer
- Functions:
  - SSL/TLS termination with self-signed certificate
  - Load balancing using round-robin algorithm
  - HTTP to HTTPS redirect for security
  - Response caching for performance
  - Security header injection (HSTS, XSS protection)
  - Access and error logging
- Ports: 443 (HTTPS), 80 (HTTP redirect)
- Instance Type: t3.micro

Backend Web Servers:
- web-1 (Primary): Active load balancing, serves 50% of traffic
- web-2 (Primary): Active load balancing, serves 50% of traffic
- web-3 (Backup): Passive standby, activated on primary failure
- Technology: Apache HTTP Server
- Content: Custom HTML pages with server metadata
- Port: 80 (HTTP)
- Instance Type: t3.micro

2.3 Network Topology

VPC Configuration:
- CIDR Block: 10.0.0.0/16
- Region: us-east-1
- DNS Hostnames: Enabled
- DNS Resolution: Enabled

Subnet Configuration:
- Type: Public subnet
- CIDR Block: 10.0.10.0/24
- Availability Zone: us-east-1a
- Auto-assign Public IP: Enabled
- Route Table: Routes to Internet Gateway

Connectivity:
- Internet Gateway: Provides internet access
- Route Table: Default route (0.0.0.0/0) → IGW

- All instances have public IPs for management

2.4 Security Design

Security Groups:

1. Nginx Security Group:
   Inbound Rules:
   • SSH (22): Your IP only - for management
   • HTTP (80): 0.0.0.0/0 - redirects to HTTPS
   • HTTPS (443): 0.0.0.0/0 - main application access

   Outbound Rules:
   • All traffic: 0.0.0.0/0 - for backend communication

2. Backend Security Group:
   Inbound Rules:
   • SSH (22): Your IP only - for management
   • HTTP (80): Nginx Security Group only - isolated backends

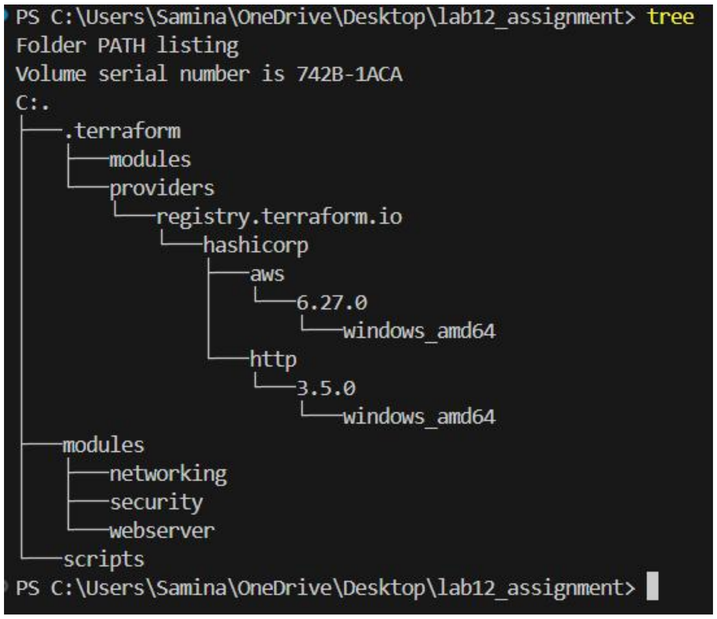
   Outbound Rules:
   • All traffic: 0.0.0.0/0 - for updates and dependencies

Security Features:
✓ SSL/TLS encryption for all client traffic
✓ Security headers (HSTS, X-Frame-Options, X-XSS-Protection)
✓ Backend isolation (only accessible via Nginx)
✓ Restricted SSH access
✓ Principle of least privilege applied

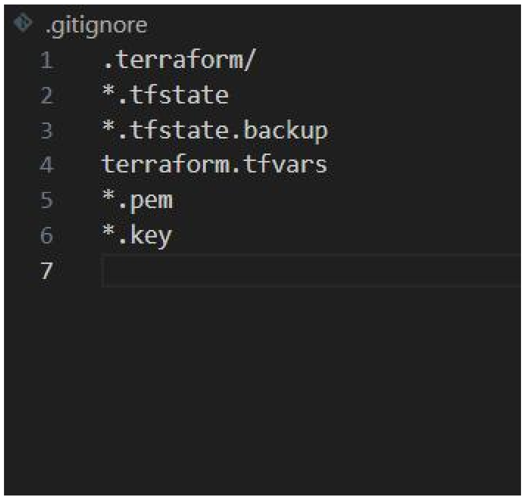
3. Implementation Details

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.

Task 1:

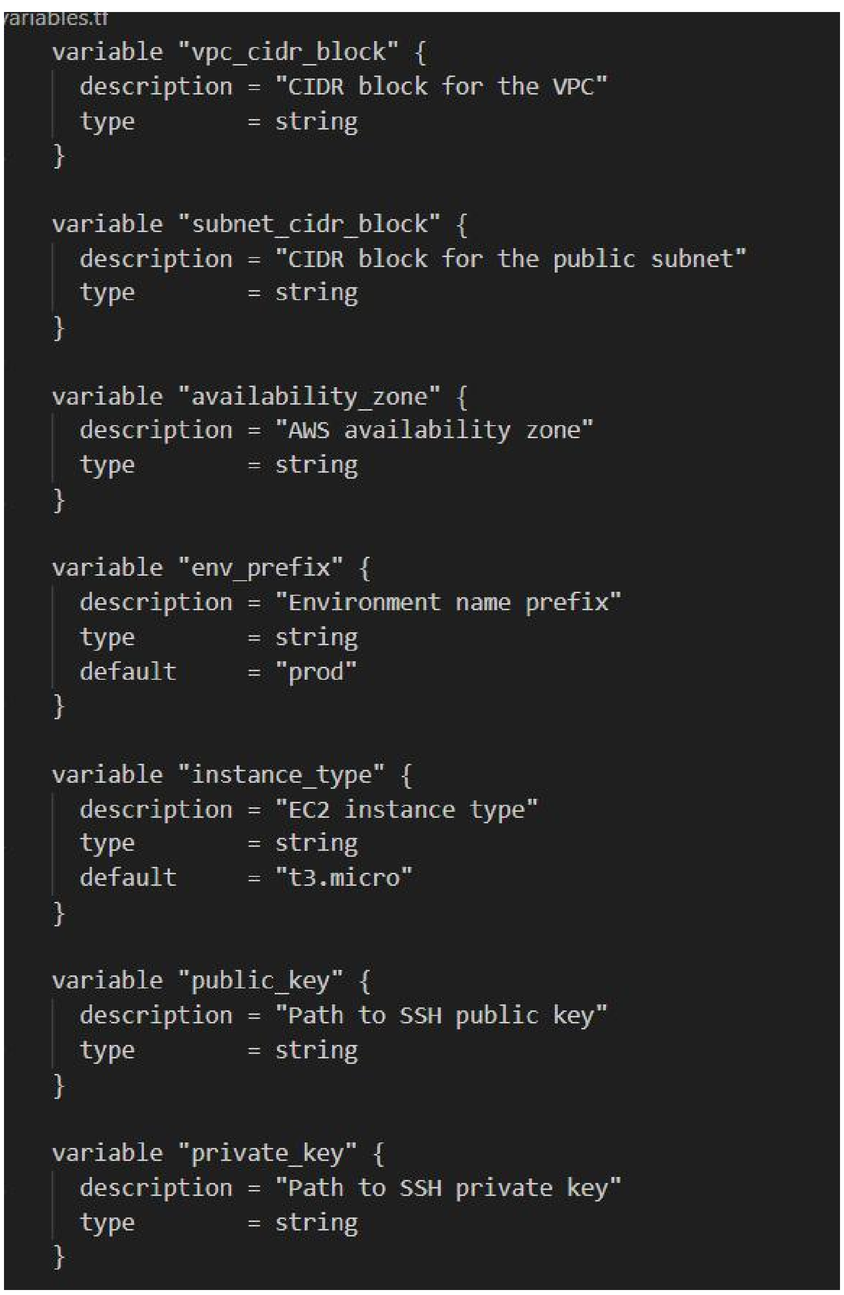


```
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> tree
Folder PATH listing
Volume serial number is 742B-1ACA
C:.
├───.terraform
│   ├───modules
│   └───providers
│       └───registry.terraform.io
│           └───hashicorp
│               ├───aws
│               │   └───6.27.0
│               │       └───windows_amd64
│               └───http
│                   └───3.5.0
│                       └───windows_amd64
├───modules
│   ├───networking
│   ├───security
│   └───webserver
└───scripts
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment>
```

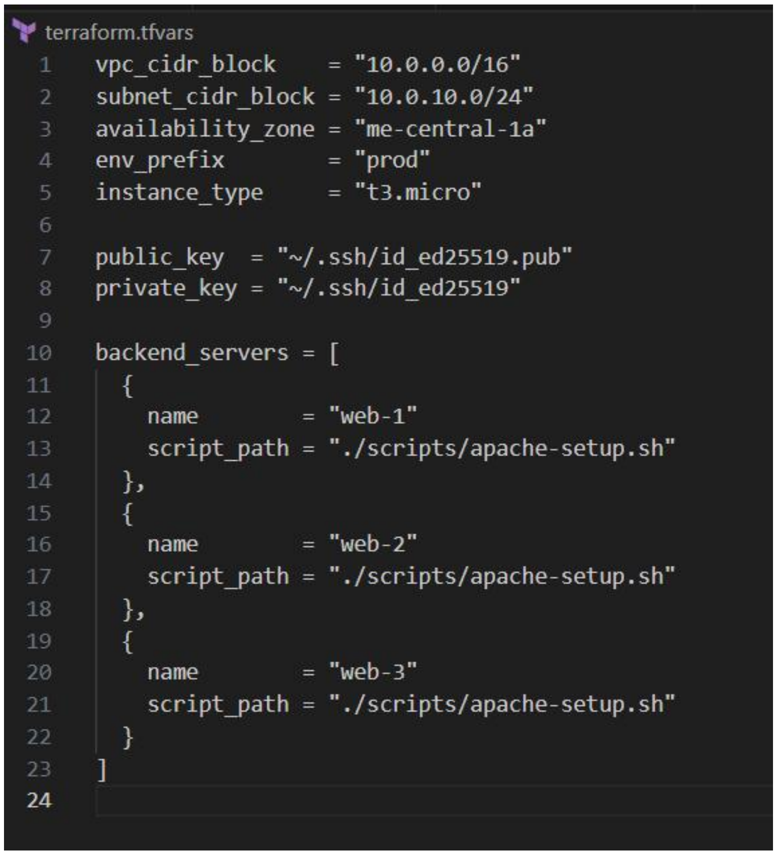


.gitignore

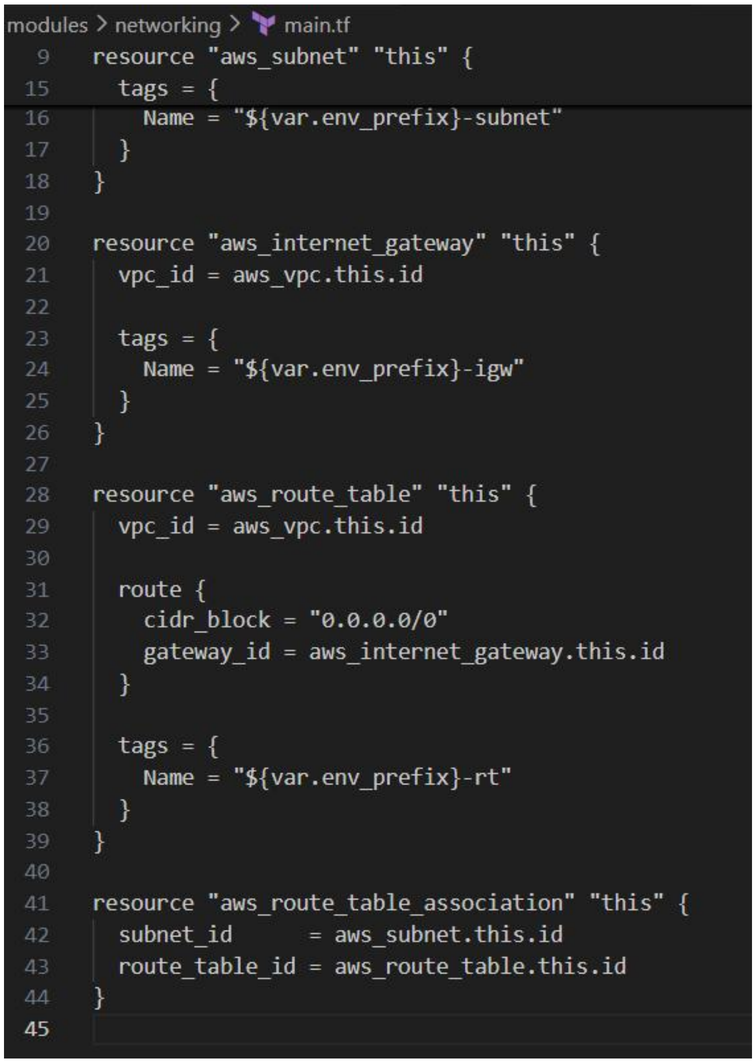
```
.terraform/
*.tfstate
*.tfstate.backup
terraform.tfvars
*.pem
*.key
```


ariables.tf

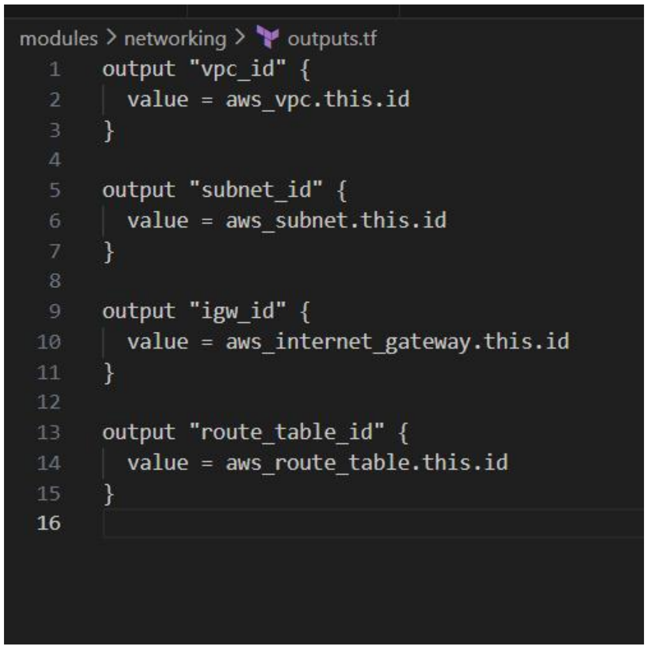
```
variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
  type        = string
}

variable "subnet_cidr_block" {
  description = "CIDR block for the public subnet"
  type        = string
}

variable "availability_zone" {
  description = "AWS availability zone"
  type        = string
}

variable "env_prefix" {
  description = "Environment name prefix"
  type        = string
  default     = "prod"
}

variable "instance_type" {
  description = "EC2 instance type"
  type        = string
  default     = "t3.micro"
}

variable "public_key" {
  description = "Path to SSH public key"
  type        = string
}

variable "private_key" {
  description = "Path to SSH private key"
  type        = string
}
```


terraform.tfvars

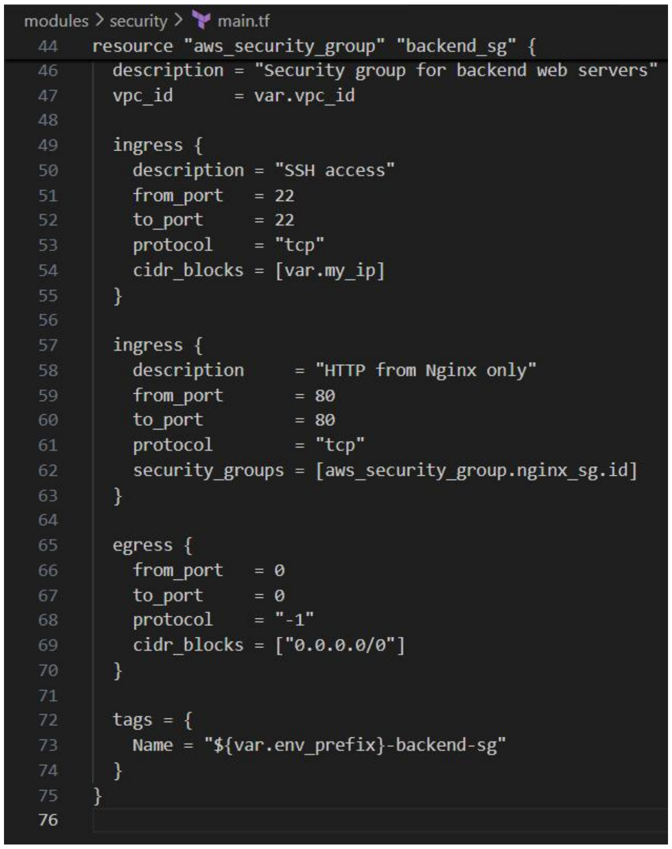
```
vpc_cidr_block    = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix        = "prod"
instance_type     = "t3.micro"

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

backend_servers = [
  {
    name        = "web-1"
    script_path = "./scripts/apache-setup.sh"
  },
  {
    name        = "web-2"
    script_path = "./scripts/apache-setup.sh"
  },
  {
    name        = "web-3"
    script_path = "./scripts/apache-setup.sh"
  }
]

```


modules > networking > main.tf

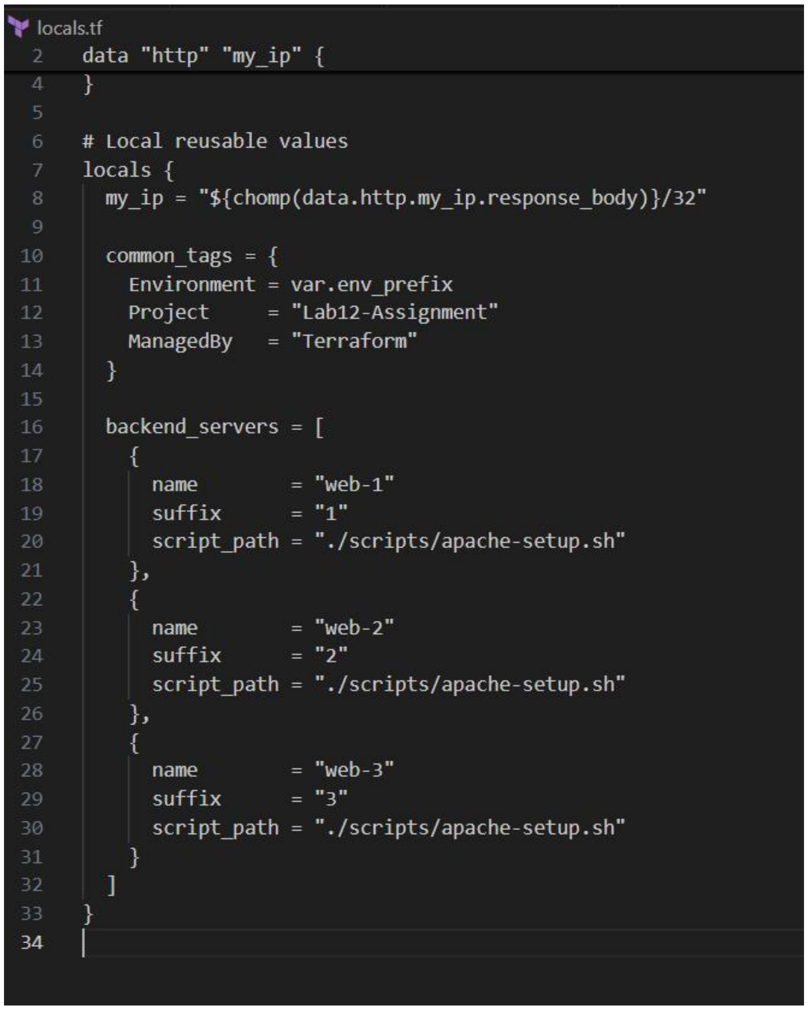
```
resource "aws_subnet" "this" {
  tags = {
    Name = "${var.env_prefix}-subnet"
  }
}

resource "aws_internet_gateway" "this" {
  vpc_id = aws_vpc.this.id

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

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}-rt"
  }
}

resource "aws_route_table_association" "this" {
  subnet_id      = aws_subnet.this.id
  route_table_id = aws_route_table.this.id
}

```


modules > networking > outputs.tf

```
output "vpc_id" {
  value = aws_vpc.this.id
}

output "subnet_id" {
  value = aws_subnet.this.id
}

output "igw_id" {
  value = aws_internet_gateway.this.id
}

output "route_table_id" {
  value = aws_route_table.this.id
}

```


modules > security > main.tf

```
resource "aws_security_group" "backend_sg" {
  description = "Security group for backend web servers"
  vpc_id      = var.vpc_id

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

  ingress {
    description     = "HTTP from Nginx only"
    from_port       = 80
    to_port         = 80
    protocol        = "tcp"
    security_groups = [aws_security_group.nginx_sg.id]
  }

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

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



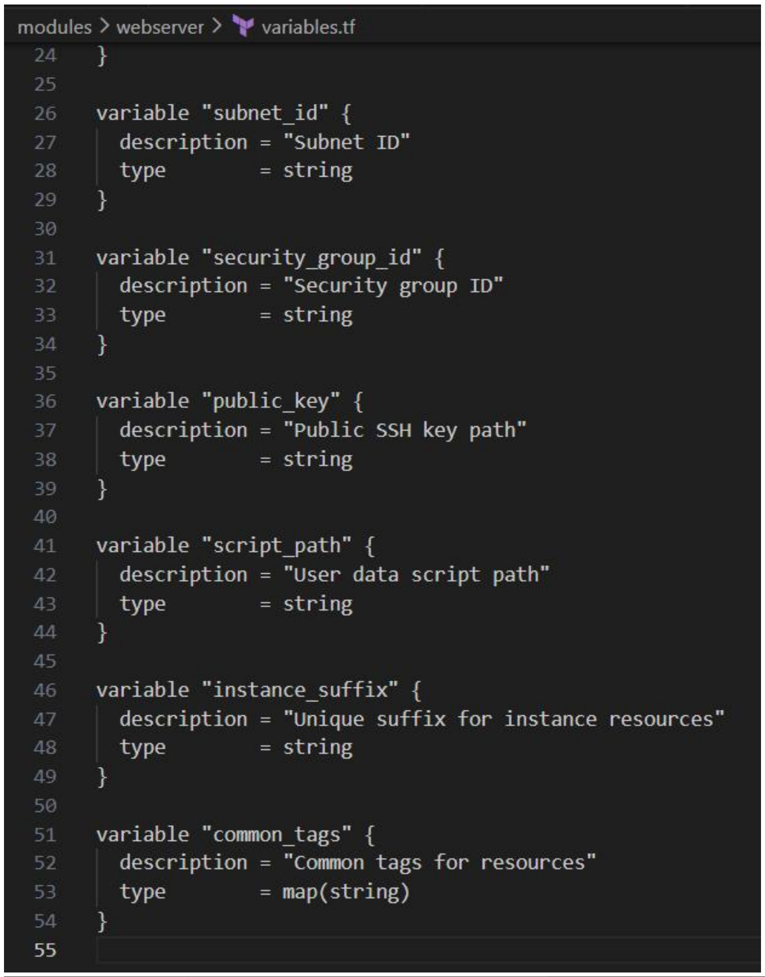
locals.tf

```
data "http" "my_ip" {
}

# Local reusable values
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"

  common_tags = {
    Environment = var.env_prefix
    Project     = "Lab12-Assignment"
    ManagedBy   = "Terraform"
  }

  backend_servers = [
    {
      name        = "web-1"
      suffix      = "1"
      script_path = "./scripts/apache-setup.sh"
    },
    {
      name        = "web-2"
      suffix      = "2"
      script_path = "./scripts/apache-setup.sh"
    },
    {
      name        = "web-3"
      suffix      = "3"
      script_path = "./scripts/apache-setup.sh"
    }
  ]
}

```

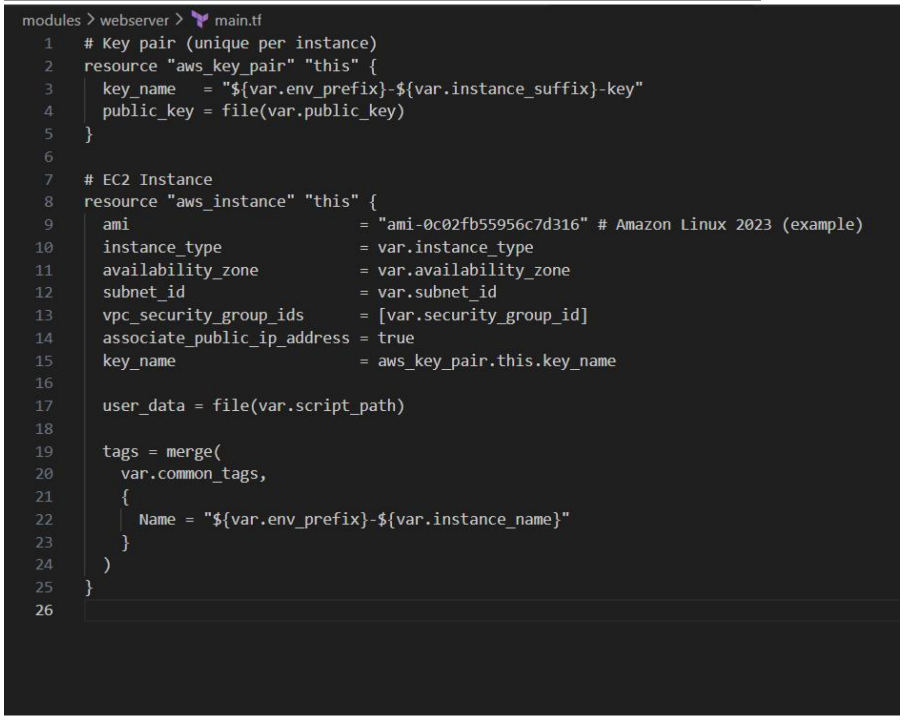
Part 2:

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.

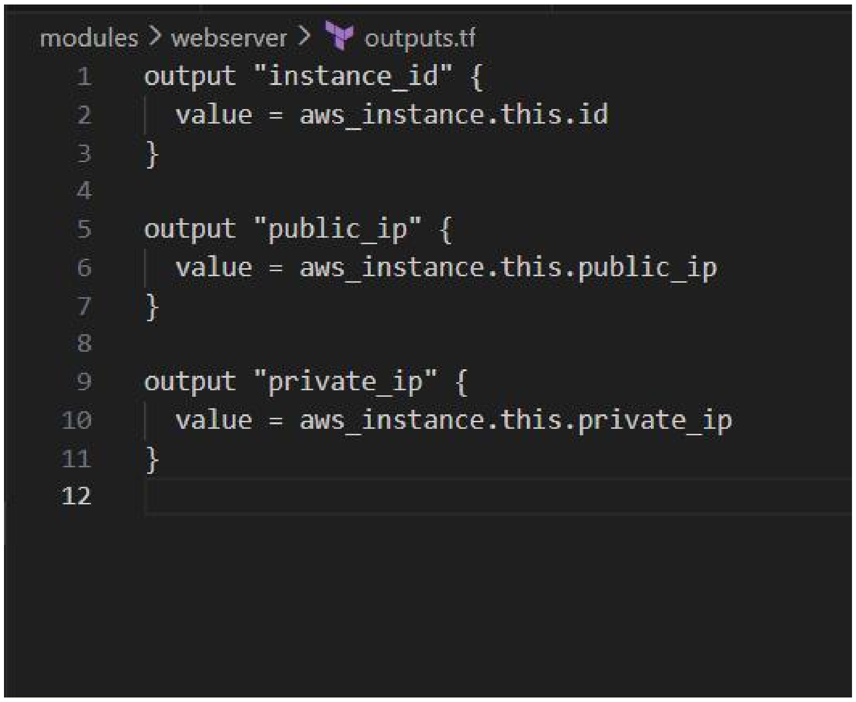
Task 2:


modules > webserver > variables.tf

```
}

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

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

```

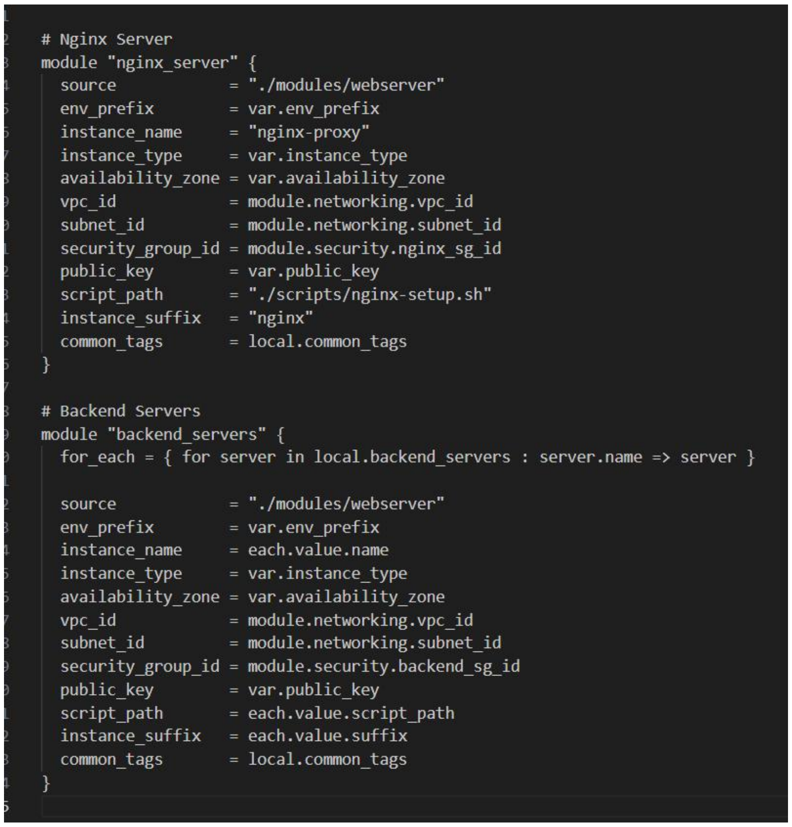

modules > webserver > main.tf

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

# EC2 Instance
resource "aws_instance" "this" {
  ami                         = "ami-0c02fb55956c7d316" # Amazon Linux 2023 (example)
  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}"
    }
  )
}

```



modules > webserver > outputs.tf

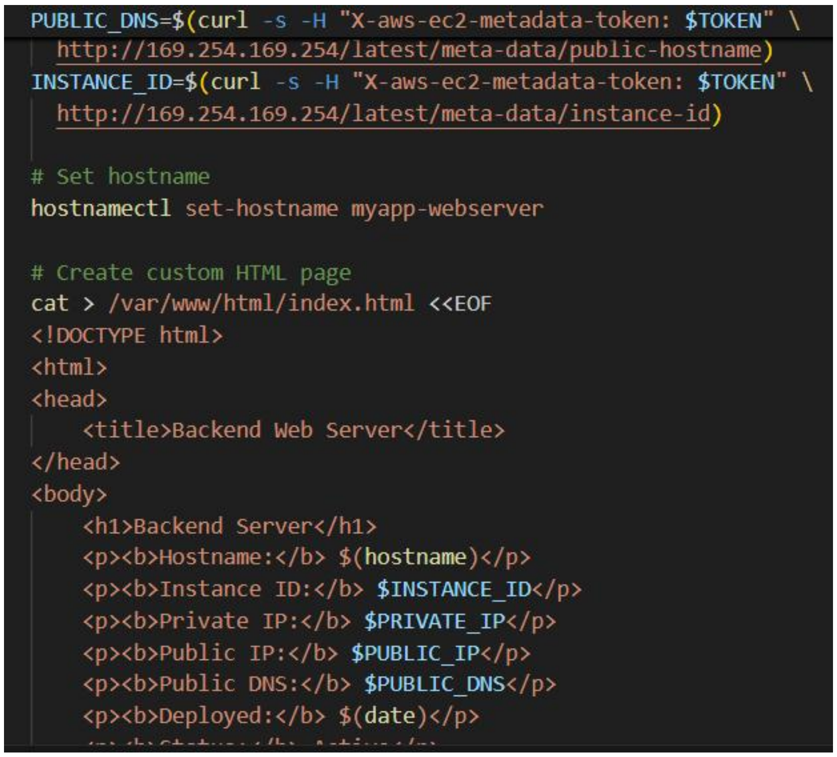
```
output "instance_id" {
  value = aws_instance.this.id
}

output "public_ip" {
  value = aws_instance.this.public_ip
}

output "private_ip" {
  value = aws_instance.this.private_ip
}

```



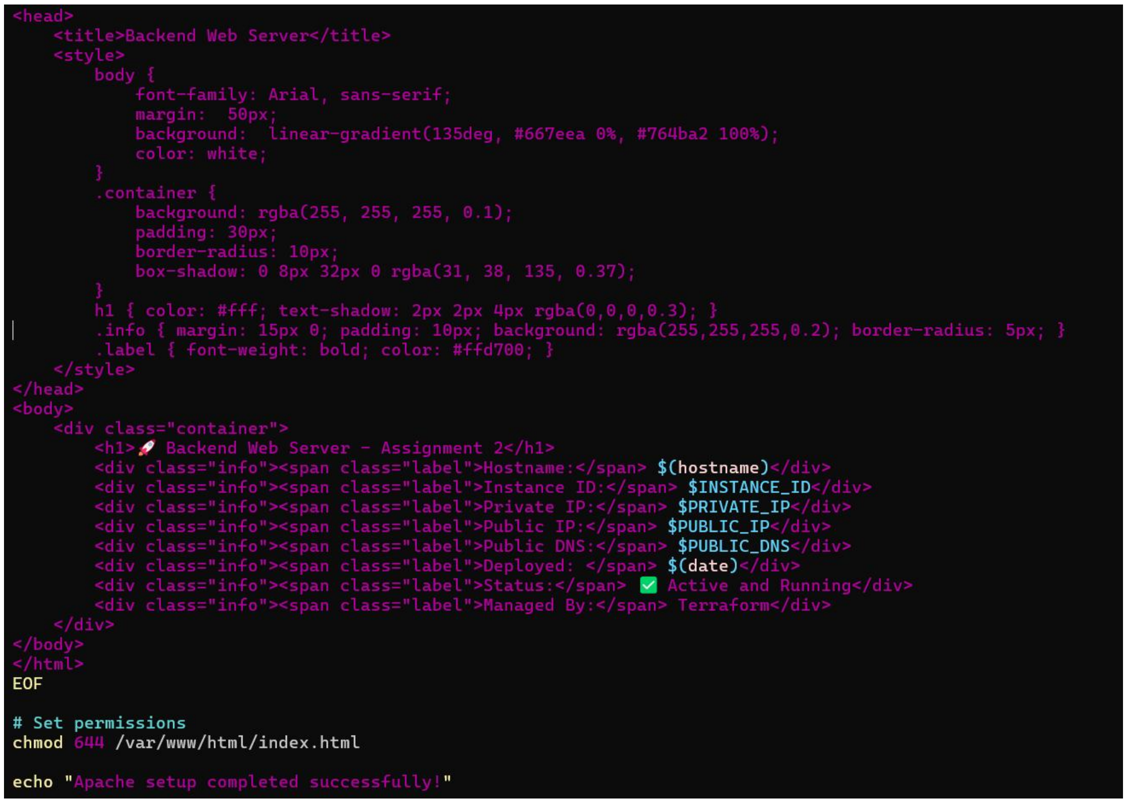
```
# Nginx Server
module "nginx_server" {
  source            = "./modules/webserver"
  env_prefix        = var.env_prefix
  instance_name     = "nginx-proxy"
  instance_type     = var.instance_type
  availability_zone = var.availability_zone
  vpc_id            = module.networking.vpc_id
  subnet_id         = module.networking.subnet_id
  security_group_id = module.security.nginx_sg_id
  public_key        = var.public_key
  script_path       = "./scripts/nginx-setup.sh"
  instance_suffix   = "nginx"
  common_tags       = local.common_tags
}

# Backend Servers
module "backend_servers" {
  for_each = { for server in local.backend_servers : server.name => server }

  source            = "./modules/webserver"
  env_prefix        = var.env_prefix
  instance_name     = each.value.name
  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
  instance_suffix   = each.value.suffix
  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

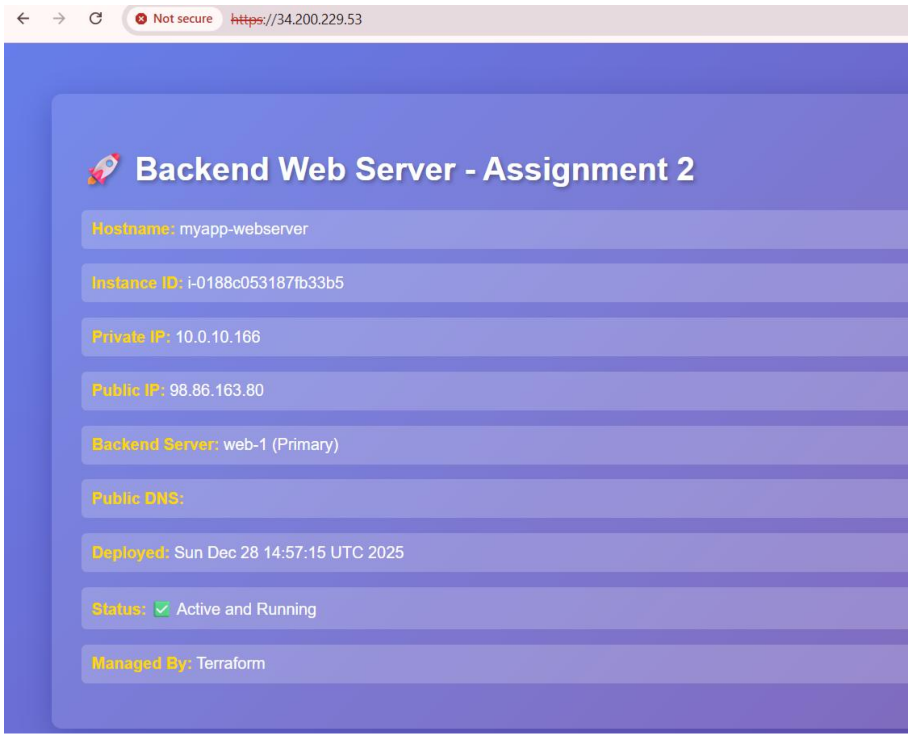
configures reverse proxy rules, caching, and load balancing across backend servers. Code snippets were embedded in the scripts directory and executed during instance initialization.
Task 3:



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

# Set hostname
hostnamectl set-hostname myapp-webserver

# Create custom HTML page
cat > /var/www/html/index.html <<EOF
<!DOCTYPE html>
<html>
<head>
    <title>Backend Web Server</title>
</head>
<body>
    <h1>Backend Server</h1>
    <p><b>Hostname:</b> $(hostname)</p>
    <p><b>Instance ID:</b> $INSTANCE_ID</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:</b> $(date)</p>
```



```
<head>
    <title>Backend Web Server</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin:  50px;
            background:  linear-gradient(135deg, #667eea 0%, #764ba2 100%);
            color: white;
        }
        .container {
            background: rgba(255, 255, 255, 0.1);
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 8px 32px 0 rgba(31, 38, 135, 0.37);
        }
        h1 { color: #fff; text-shadow: 2px 2px 4px rgba(0,0,0,0.3); }
        .info { margin: 15px 0; padding: 10px; background: rgba(255,255,255,0.2); border-radius: 5px; }
        .label { font-weight: bold; color: #ffd700; }
    </style>
</head>
<body>
    <div class="container">
        <h1>🚀 Backend Web Server - Assignment 2</h1>
        <div class="info"><span class="label">Hostname:</span> $(hostname)</div>
        <div class="info"><span class="label">Instance ID:</span> $INSTANCE_ID</div>
        <div class="info"><span class="label">Private IP:</span> $PRIVATE_IP</div>
        <div class="info"><span class="label">Public IP:</span> $PUBLIC_IP</div>
        <div class="info"><span class="label">Public DNS:</span> $PUBLIC_DNS</div>
        <div class="info"><span class="label">Deployed: </span> $(date)</div>
        <div class="info"><span class="label">Status:</span> ✅ Active and Running</div>
        <div class="info"><span class="label">Managed By:</span> Terraform</div>
    </div>
</body>
</html>
EOF

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

echo "Apache setup completed successfully!"
```


Not secure ~~https~~://34.200.229.53

🚀 Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-0188c053187fb33b5

Private IP: 10.0.10.166

Public IP: 98.86.163.80

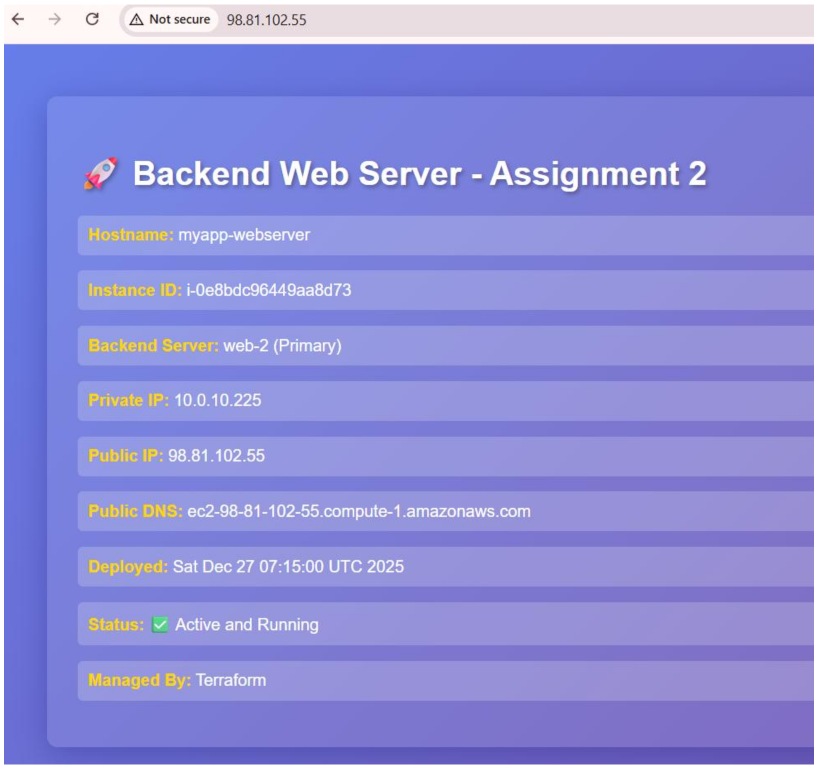
Backend Server: web-1 (Primary)

Public DNS:

Deployed: Sun Dec 28 14:57:15 UTC 2025

Status: ✅ Active and Running

Managed By: Terraform


Not secure 98.81.102.55

🚀 Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-0e8bdc96449aa8d73

Backend Server: web-2 (Primary)

Private IP: 10.0.10.225

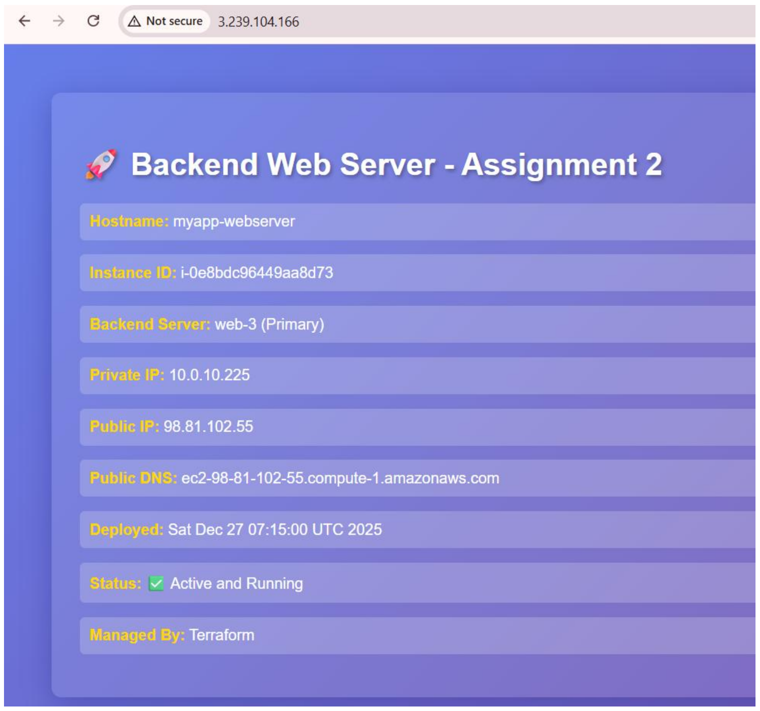
Public IP: 98.81.102.55

Public DNS: ec2-98-81-102-55.compute-1.amazonaws.com

Deployed: Sat Dec 27 07:15:00 UTC 2025

Status: ✅ Active and Running

Managed By: Terraform



Not secure 3.239.104.166

🚀 Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-0e8bdc96449aa8d73

Backend Server: web-3 (Primary)

Private IP: 10.0.10.225

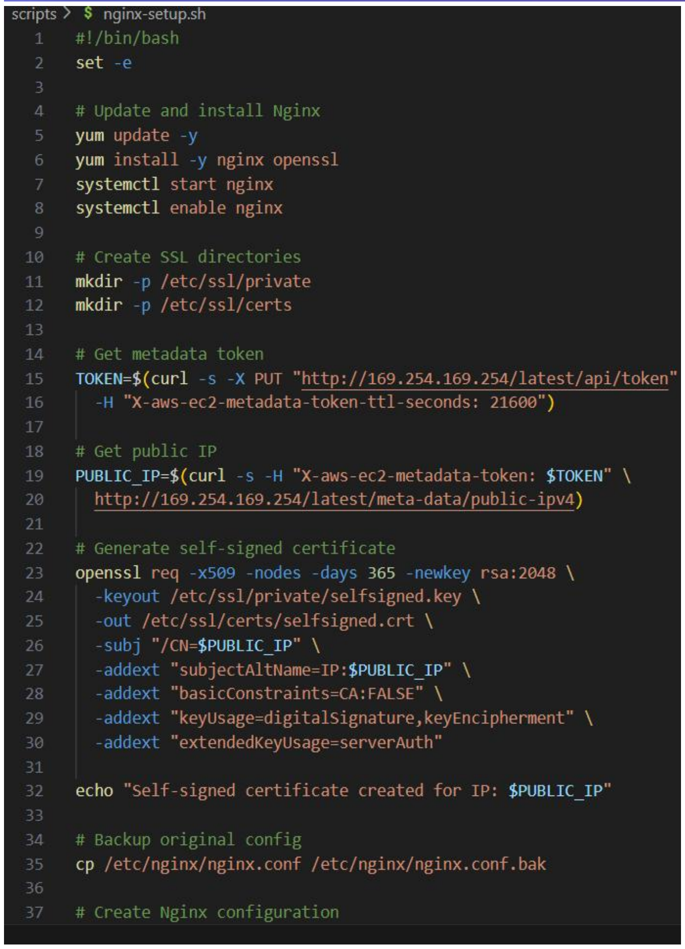
Public IP: 98.81.102.55

Public DNS: ec2-98-81-102-55.compute-1.amazonaws.com

Deployed: Sat Dec 27 07:15:00 UTC 2025

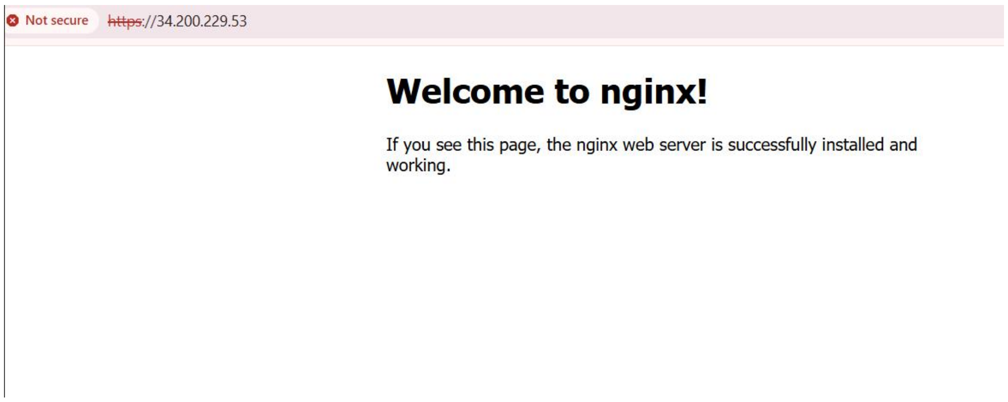
Status: ✅ Active and Running

Managed By: Terraform



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"

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

# Create Nginx configuration
```



Not secure ~~https~~://34.200.229.53

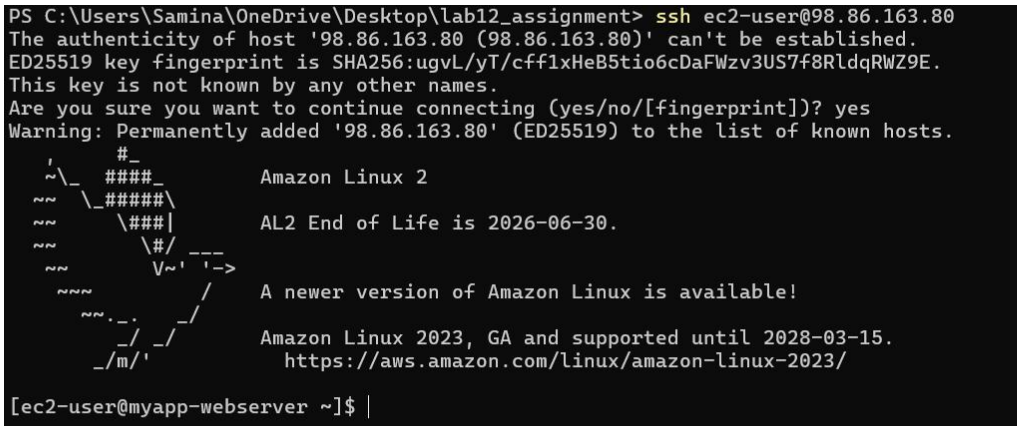
Welcome to nginx!

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

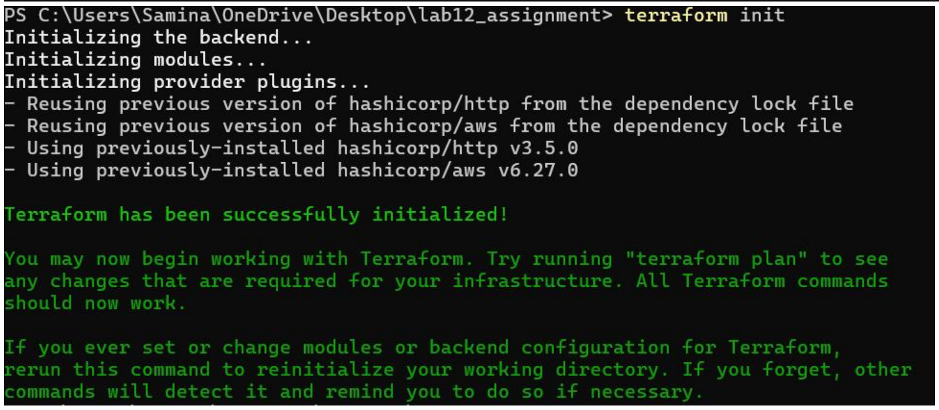
Part 4:

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

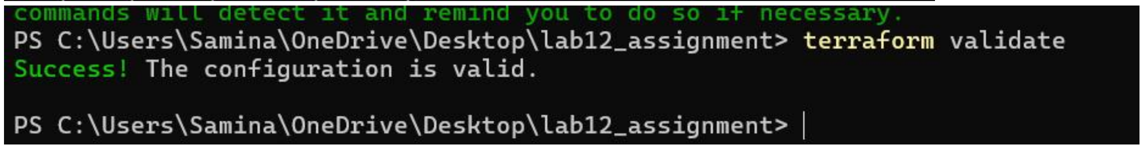
Task 4:



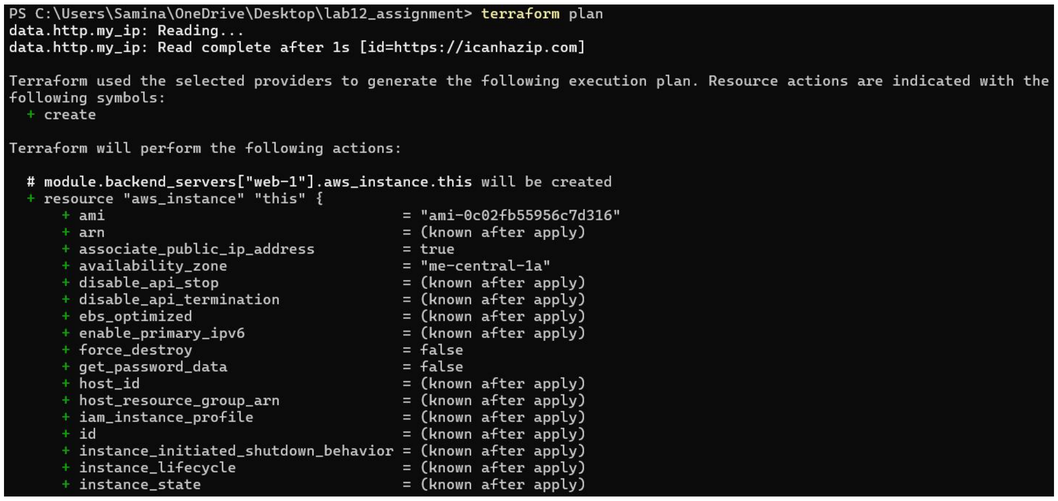
```
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> ssh ec2-user@98.86.163.80
The authenticity of host '98.86.163.80 (98.86.163.80)' can't be established.
ED25519 key fingerprint is SHA256:ugvL/yT/cff1xHeB5tio6cDaFWzv3US7f8RldqRWZ9E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.86.163.80' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@myapp-webserver ~]$ |
```



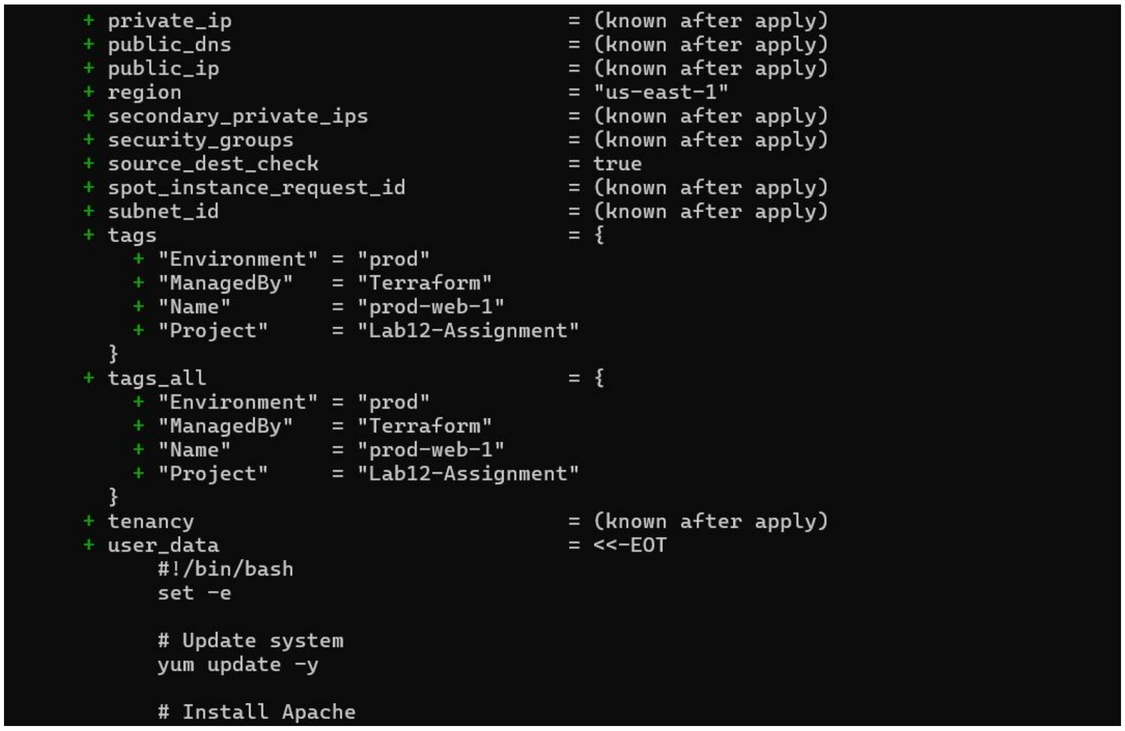
```
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Reusing previous version of hashicorp/http from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/http v3.5.0
- Using previously-installed hashicorp/aws v6.27.0

Terraform has been successfully initialized!

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

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



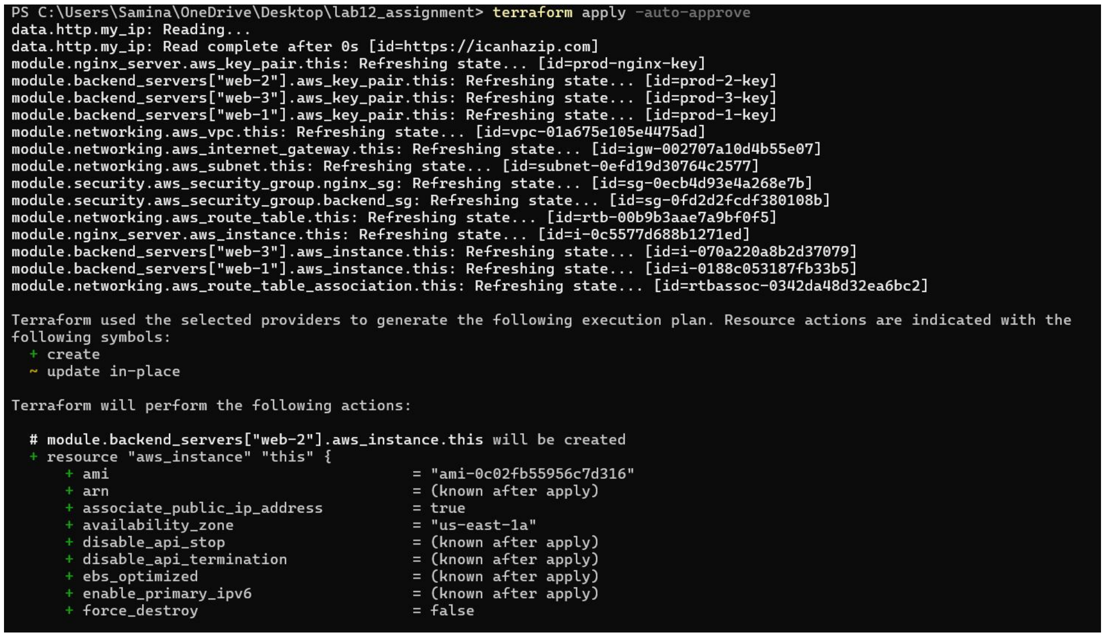
```
commands will detect it and remind you to do so if necessary.
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform validate
Success! The configuration is valid.

PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> |
```

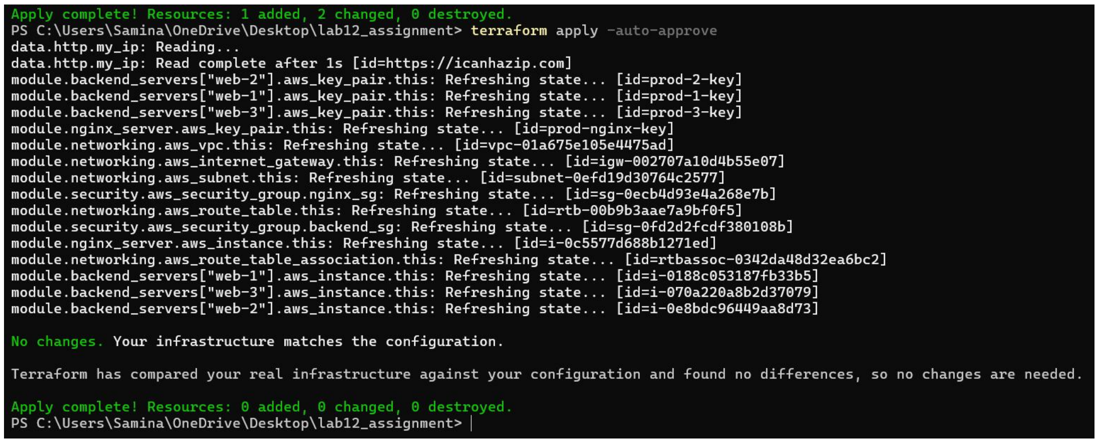


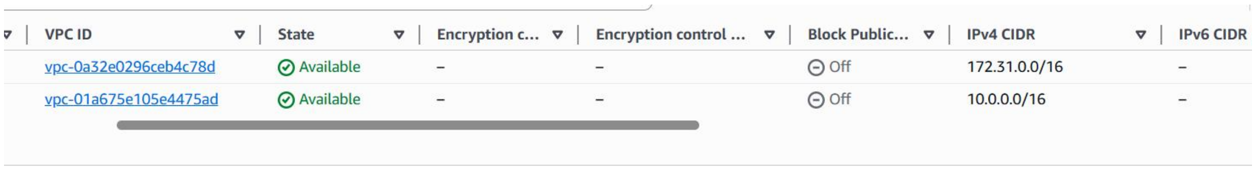
```
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform plan
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 1s [id=https://icanhazip.com]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # module.backend_servers["web-1"].aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0c02fb55956c7d316"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = "me-central-1a"
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
```



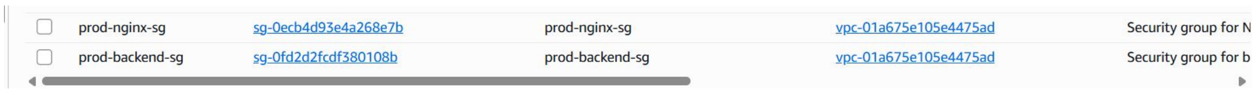
```
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + region                               = "us-east-1"
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Environment" = "prod"
          + "ManagedBy"   = "Terraform"
          + "Name"        = "prod-web-1"
          + "Project"     = "Lab12-Assignment"
        }
      + tags_all                             = {
          + "Environment" = "prod"
          + "ManagedBy"   = "Terraform"
          + "Name"        = "prod-web-1"
          + "Project"     = "Lab12-Assignment"
        }
      + tenancy                              = (known after apply)
      + user_data                            = <<-EOT
            #!/bin/bash
            set -e

            # Update system
            yum update -y

            # Install Apache
```


```
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform apply -auto-approve
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.nginx_server.aws_key_pair.this: Refreshing state... [id=prod-nginx-key]
module.backend_servers["web-2"].aws_key_pair.this: Refreshing state... [id=prod-2-key]
module.backend_servers["web-3"].aws_key_pair.this: Refreshing state... [id=prod-3-key]
module.backend_servers["web-1"].aws_key_pair.this: Refreshing state... [id=prod-1-key]
module.networking.aws_vpc.this: Refreshing state... [id=vpc-01a675e105e4475ad]
module.networking.aws_internet_gateway.this: Refreshing state... [id=igw-002707a10d4b55e07]
module.networking.aws_subnet.this: Refreshing state... [id=subnet-0efd19d30764c2577]
module.security.aws_security_group.nginx_sg: Refreshing state... [id=sg-0ecb4d93e4a268e7b]
module.security.aws_security_group.backend_sg: Refreshing state... [id=sg-0fd2d2fcdf380108b]
module.networking.aws_route_table.this: Refreshing state... [id=rtb-00b9b3aae7a9bf0f5]
module.nginx_server.aws_instance.this: Refreshing state... [id=i-0c5577d688b1271ed]
module.backend_servers["web-3"].aws_instance.this: Refreshing state... [id=i-070a220a8b2d37079]
module.backend_servers["web-1"].aws_instance.this: Refreshing state... [id=i-0188c053187fb33b5]
module.networking.aws_route_table_association.this: Refreshing state... [id=rtbassoc-0342da48d32ea6bc2]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create
  ~ update in-place

Terraform will perform the following actions:

  # module.backend_servers["web-2"].aws_instance.this will be created
  + resource "aws_instance" "this" {
      + ami                                  = "ami-0c02fb55956c7d316"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = "us-east-1a"
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
```

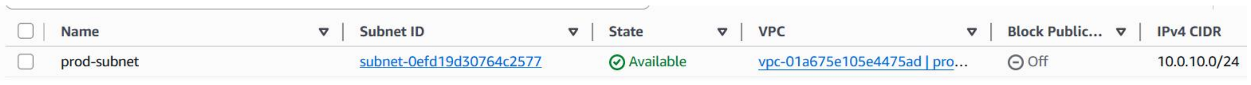

```
Apply complete! Resources: 1 added, 2 changed, 0 destroyed.
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform apply -auto-approve
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 1s [id=https://icanhazip.com]
module.backend_servers["web-2"].aws_key_pair.this: Refreshing state... [id=prod-2-key]
module.backend_servers["web-1"].aws_key_pair.this: Refreshing state... [id=prod-1-key]
module.backend_servers["web-3"].aws_key_pair.this: Refreshing state... [id=prod-3-key]
module.nginx_server.aws_key_pair.this: Refreshing state... [id=prod-nginx-key]
module.networking.aws_vpc.this: Refreshing state... [id=vpc-01a675e105e4475ad]
module.networking.aws_internet_gateway.this: Refreshing state... [id=igw-002707a10d4b55e07]
module.networking.aws_subnet.this: Refreshing state... [id=subnet-0efd19d30764c2577]
module.security.aws_security_group.nginx_sg: Refreshing state... [id=sg-0ecb4d93e4a268e7b]
module.networking.aws_route_table.this: Refreshing state... [id=rtb-00b9b3aae7a9bf0f5]
module.security.aws_security_group.backend_sg: Refreshing state... [id=sg-0fd2d2fcdf380108b]
module.nginx_server.aws_instance.this: Refreshing state... [id=i-0c5577d688b1271ed]
module.networking.aws_route_table_association.this: Refreshing state... [id=rtbassoc-0342da48d32ea6bc2]
module.backend_servers["web-1"].aws_instance.this: Refreshing state... [id=i-0188c053187fb33b5]
module.backend_servers["web-3"].aws_instance.this: Refreshing state... [id=i-070a220a8b2d37079]
module.backend_servers["web-2"].aws_instance.this: Refreshing state... [id=i-0e8bdc96449aa8d73]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment>
```



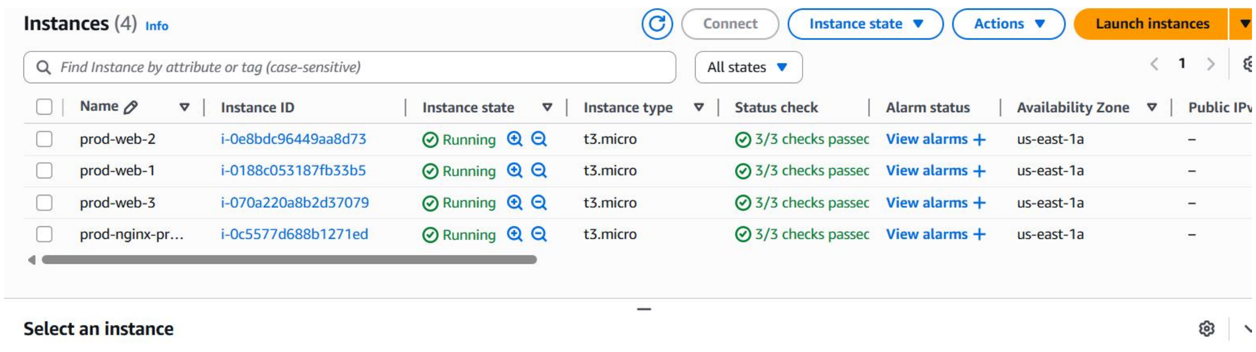
VPC ID	State	Encryption c…	Encryption control …	Block Public…	IPv4 CIDR	IPv6 CIDR
vpc-0a32e0296ceb4c78d	✓ Available	–	–	Off	172.31.0.0/16	–
vpc-01a675e105e4475ad	✓ Available	–	–	Off	10.0.0.0/16	–



☐	prod-nginx-sg	sg-0ecb4d93e4a268e7b	prod-nginx-sg	vpc-01a675e105e4475ad	Security group for N
☐	prod-backend-sg	sg-0fd2d2fcdf380108b	prod-backend-sg	vpc-01a675e105e4475ad	Security group for b



☐	Name	Subnet ID	State	VPC	Block Public…	IPv4 CIDR
☐	prod-subnet	subnet-0efd19d30764c2577	✓ Available	vpc-01a675e105e4475ad | pro…	Off	10.0.10.0/24



Instances (4) Info

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

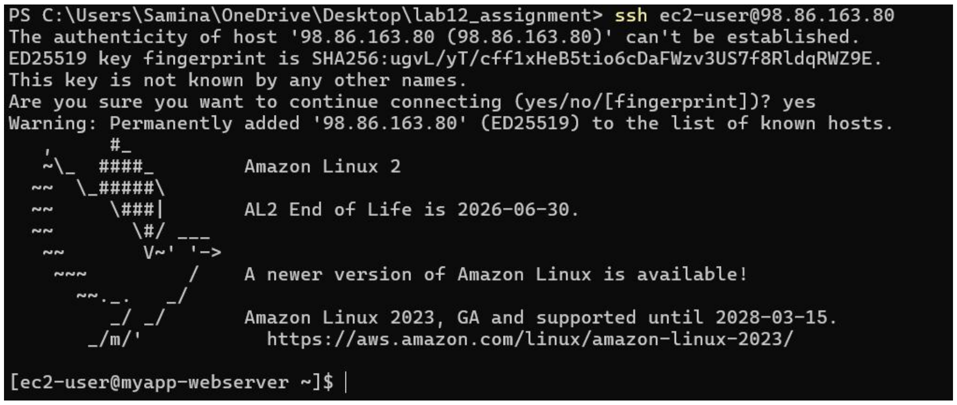
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
prod-web-2	i-0e8bdc96449aa8d73	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	–
prod-web-1	i-0188c053187fb33b5	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	–
prod-web-3	i-070a220a8b2d37079	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	–
prod-nginx-pr…	i-0c5577d688b1271ed	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1a	–

Select an instance

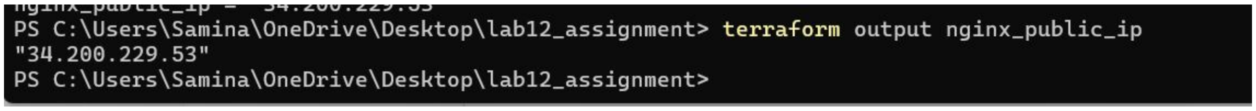
Part 5:

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.

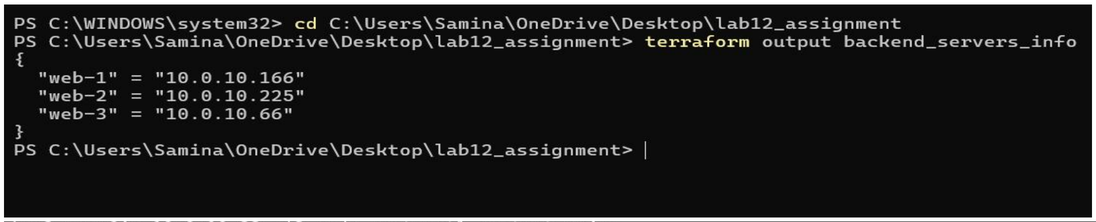
Task 5:



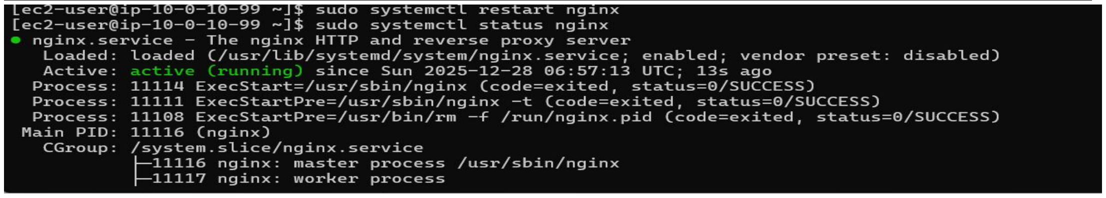
```
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> ssh ec2-user@98.86.163.80
The authenticity of host '98.86.163.80 (98.86.163.80)' can't be established.
ED25519 key fingerprint is SHA256:ugvL/yT/cff1xHeB5tio6cDaFWzv3US7f8RldqRWZ9E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.86.163.80' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@myapp-webserver ~]$ 
```



```
nginx_public_ip = "34.200.229.53"
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform output nginx_public_ip
"34.200.229.53"
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment>
```

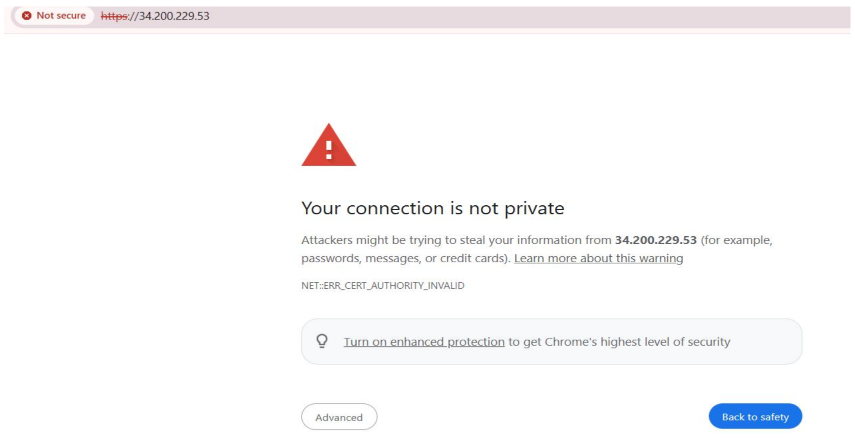


```
PS C:\WINDOWS\system32> cd C:\Users\Samina\OneDrive\Desktop\lab12_assignment
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> terraform output backend_servers_info
{
  "web-1" = "10.0.10.166"
  "web-2" = "10.0.10.225"
  "web-3" = "10.0.10.66"
}
PS C:\Users\Samina\OneDrive\Desktop\lab12_assignment> 
```



```
[ec2-user@ip-10-0-10-99 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-99 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2025-12-28 06:57:13 UTC; 13s ago
  Process: 11114 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 11111 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 11108 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 11116 (nginx)
   CGroup: /system.slice/nginx.service
           ├─11116 nginx: master process /usr/sbin/nginx
           └─11117 nginx: worker process
```

5.2



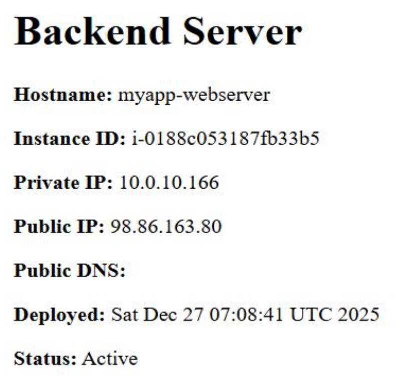
Not secure ~~https~~://34.200.229.53

Your connection is not private

Attackers might be trying to steal your information from **34.200.229.53** (for example, passwords, messages, or credit cards). Learn more about this warning

NET::ERR_CERT_AUTHORITY_INVALID

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



Backend Server

Hostname: myapp-webserver

Instance ID: i-0188c053187fb33b5

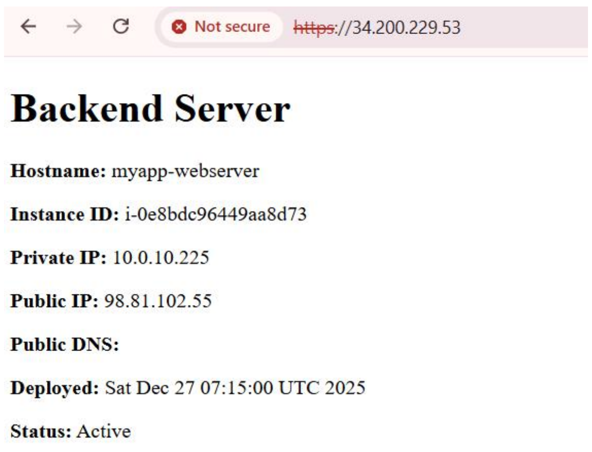
Private IP: 10.0.10.166

Public IP: 98.86.163.80

Public DNS:

Deployed: Sat Dec 27 07:08:41 UTC 2025

Status: Active



Not secure ~~https~~://34.200.229.53

Backend Server

Hostname: myapp-webserver

Instance ID: i-0e8bdc96449aa8d73

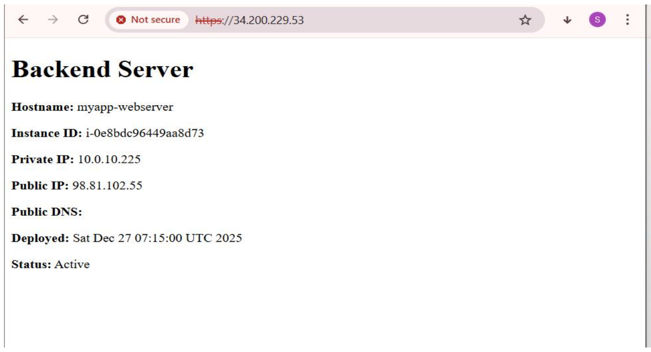
Private IP: 10.0.10.225

Public IP: 98.81.102.55

Public DNS:

Deployed: Sat Dec 27 07:15:00 UTC 2025

Status: Active


Not secure ~~https~~://34.200.229.53

Backend Server

Hostname: myapp-webserver

Instance ID: i-0e8bdc96449aa8d73

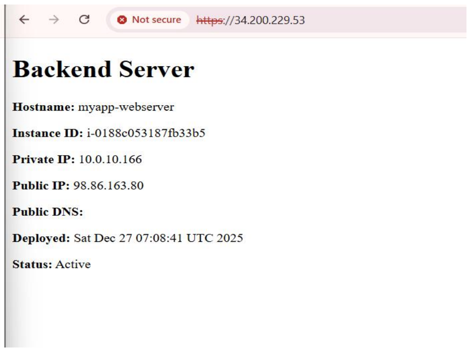
Private IP: 10.0.10.225

Public IP: 98.81.102.55

Public DNS:

Deployed: Sat Dec 27 07:15:00 UTC 2025

Status: Active


Not secure ~~https~~://34.200.229.53

Backend Server

Hostname: myapp-webserver

Instance ID: i-0188c053187fb33b5

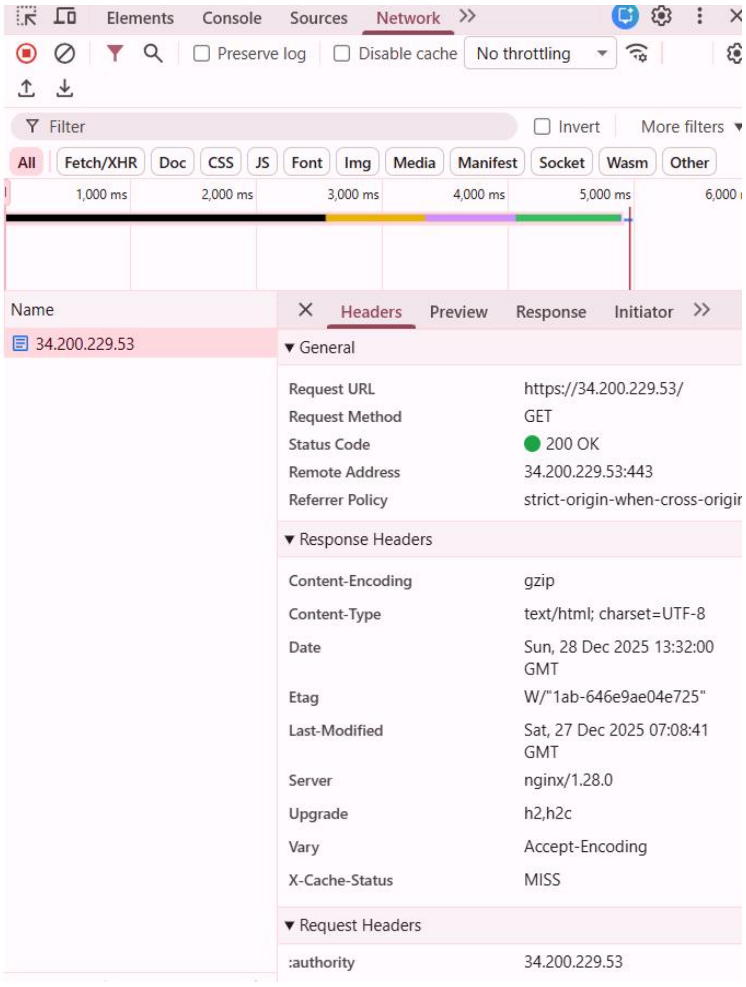
Private IP: 10.0.10.166

Public IP: 98.86.163.80

Public DNS:

Deployed: Sat Dec 27 07:08:41 UTC 2025

Status: Active



34.200.229.53

▼ General

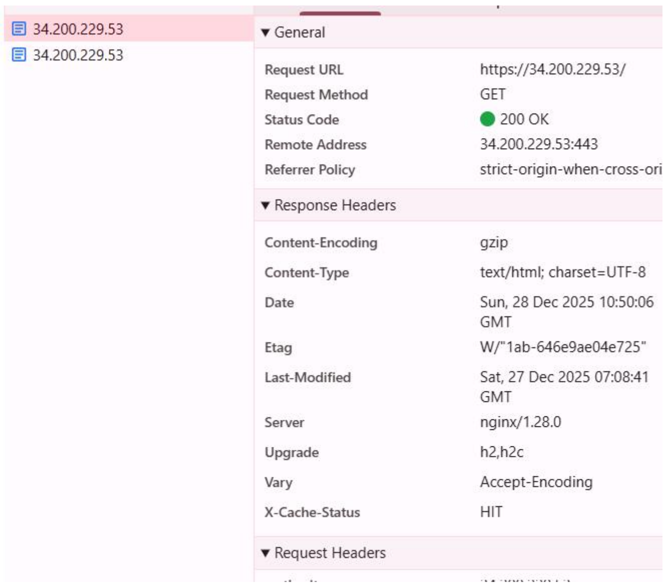
Request URL	https://34.200.229.53/
Request Method	GET
Status Code	● 200 OK
Remote Address	34.200.229.53:443
Referrer Policy	strict-origin-when-cross-origir

▼ Response Headers

Content-Encoding	gzip
Content-Type	text/html; charset=UTF-8
Date	Sun, 28 Dec 2025 13:32:00 GMT
Etag	W/"1ab-646e9ae04e725"
Last-Modified	Sat, 27 Dec 2025 07:08:41 GMT
Server	nginx/1.28.0
Upgrade	h2,h2c
Vary	Accept-Encoding
X-Cache-Status	MISS

▼ Request Headers

:authority	34.200.229.53



34.200.229.53
34.200.229.53

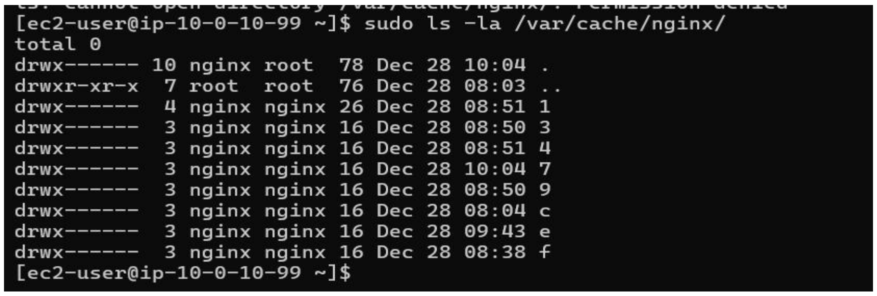
▼ General

Request URL	https://34.200.229.53/
Request Method	GET
Status Code	🟢 200 OK
Remote Address	34.200.229.53:443
Referrer Policy	strict-origin-when-cross-ori

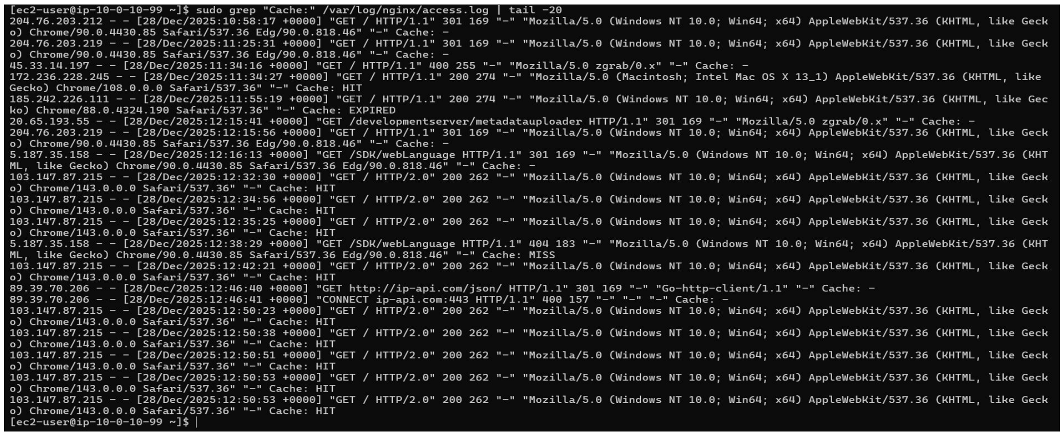
▼ Response Headers

Content-Encoding	gzip
Content-Type	text/html; charset=UTF-8
Date	Sun, 28 Dec 2025 10:50:06 GMT
Etag	W/"1ab-646e9ae04e725"
Last-Modified	Sat, 27 Dec 2025 07:08:41 GMT
Server	nginx/1.28.0
Upgrade	h2,h2c
Vary	Accept-Encoding
X-Cache-Status	HIT

▼ Request Headers

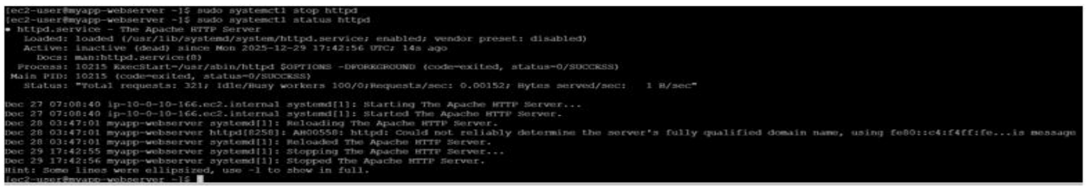


```
[ec2-user@ip-10-0-10-99 ~]$ sudo ls -la /var/cache/nginx/
total 0
drwx------ 10 nginx root  78 Dec 28 10:04 .
drwxr-xr-x  7 root  root  76 Dec 28 08:03 ..
drwx------  4 nginx nginx 26 Dec 28 08:51 1
drwx------  3 nginx nginx 16 Dec 28 08:50 3
drwx------  3 nginx nginx 16 Dec 28 08:51 4
drwx------  3 nginx nginx 16 Dec 28 10:04 7
drwx------  3 nginx nginx 16 Dec 28 08:50 9
drwx------  3 nginx nginx 16 Dec 28 08:04 c
drwx------  3 nginx nginx 16 Dec 28 09:43 e
drwx------  3 nginx nginx 16 Dec 28 08:38 f
[ec2-user@ip-10-0-10-99 ~]$
```

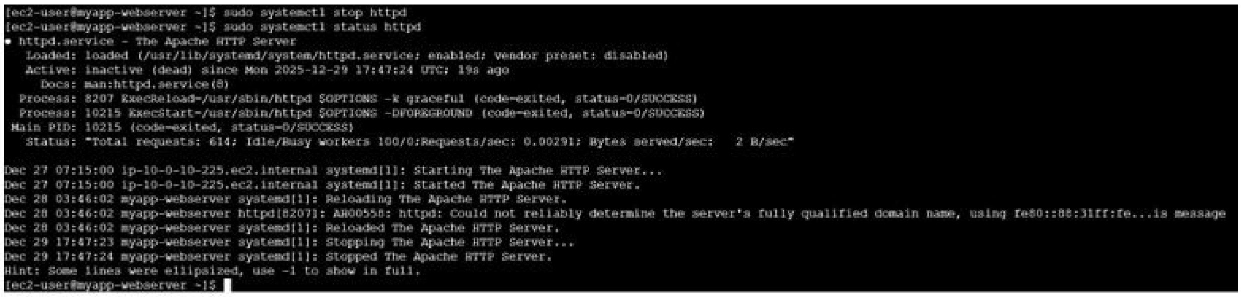


```
[ec2-user@ip-10-0-10-99 ~]$ sudo grep "Cache:" /var/log/nginx/access.log | tail -20
204.76.203.212 - - [28/Dec/2025:10:58:17 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
204.76.203.219 - - [28/Dec/2025:11:25:31 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
45.33.14.197 - - [28/Dec/2025:11:34:16 +0000] "GET / HTTP/1.1" 400 255 "-" "Mozilla/5.0 zgrab/0.x" "-" Cache: -
172.236.228.245 - - [28/Dec/2025:11:34:27 +0000] "GET / HTTP/1.1" 200 274 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 13_1) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/108.0.0.0 Safari/537.36" "-" Cache: HIT
185.242.226.111 - - [28/Dec/2025:11:55:19 +0000] "GET / HTTP/1.1" 200 274 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
ko) Chrome/88.0.4324.190 Safari/537.36" "-" Cache: EXPIRED
20.65.193.55 - - [28/Dec/2025:12:15:41 +0000] "GET /developmentserver/metadatauploader HTTP/1.1" 301 169 "-" "Mozilla/5.0 zgrab/0.x" "-" Cache: -
204.76.203.219 - - [28/Dec/2025:12:15:56 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
5.187.35.158 - - [28/Dec/2025:12:16:13 +0000] "GET /SDK/webLanguage HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
103.147.87.215 - - [28/Dec/2025:12:32:30 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
103.147.87.215 - - [28/Dec/2025:12:34:56 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
103.147.87.215 - - [28/Dec/2025:12:35:25 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
5.187.35.158 - - [28/Dec/2025:12:38:29 +0000] "GET /SDK/webLanguage HTTP/1.1" 404 183 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: MISS
103.147.87.215 - - [28/Dec/2025:12:42:21 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
89.39.70.206 - - [28/Dec/2025:12:46:40 +0000] "GET http://ip-api.com/json/ HTTP/1.1" 301 169 "-" "Go-http-client/1.1" "-" Cache: -
89.39.70.206 - - [28/Dec/2025:12:46:41 +0000] "CONNECT ip-api.com:443 HTTP/1.1" 400 157 "-" "-" "-" Cache: -
103.147.87.215 - - [28/Dec/2025:12:50:23 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
103.147.87.215 - - [28/Dec/2025:12:50:38 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
103.147.87.215 - - [28/Dec/2025:12:50:51 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
103.147.87.215 - - [28/Dec/2025:12:50:53 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
103.147.87.215 - - [28/Dec/2025:12:50:53 +0000] "GET / HTTP/2.0" 200 262 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/143.0.0.0 Safari/537.36" "-" Cache: HIT
[ec2-user@ip-10-0-10-99 ~]$
```

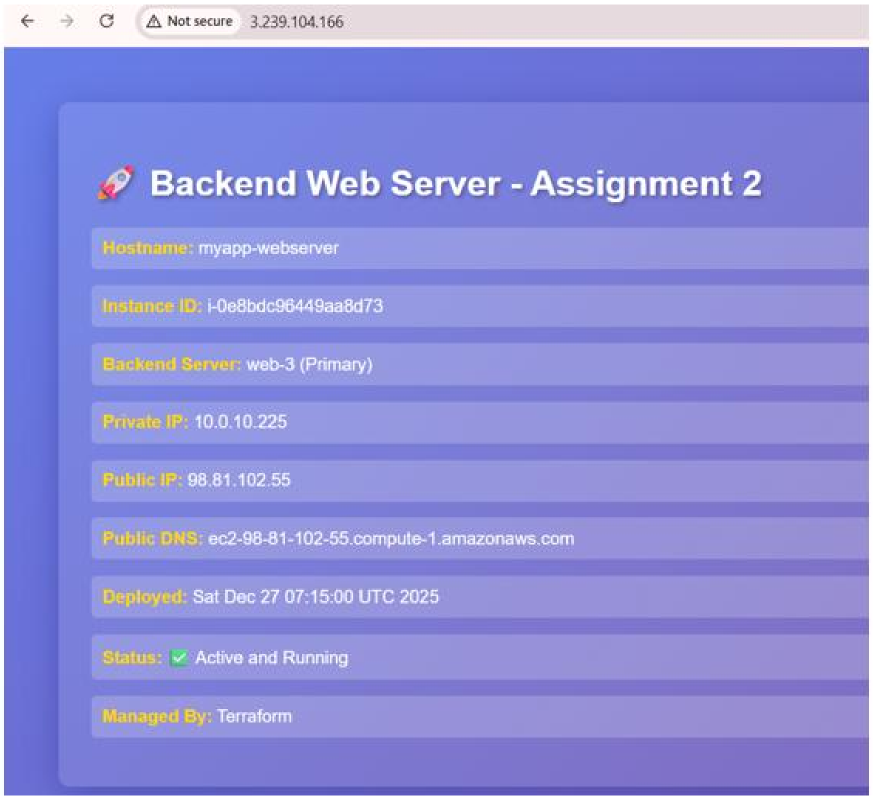
5.4



```
[ec2-user@myapp-webserver ~]$ sudo systemctl stop httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Mon 2025-12-29 17:42:56 UTC; 14s ago
     Docs: man:httpd.service(8)
  Process: 10215 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 10215 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 321; Idle/Busy workers 100/0;Requests/sec: 0.00152; Bytes served/sec:   1 B/sec"

Dec 27 07:08:40 ip-10-0-10-166.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Dec 27 07:08:40 ip-10-0-10-166.ec2.internal systemd[1]: Started The Apache HTTP Server.
Dec 28 03:47:01 myapp-webserver systemd[1]: Reloading The Apache HTTP Server.
Dec 28 03:47:01 myapp-webserver httpd[8258]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::c4:f4ff:fe...is message
Dec 28 03:47:01 myapp-webserver systemd[1]: Reloaded The Apache HTTP Server.
Dec 29 17:42:55 myapp-webserver systemd[1]: Stopping The Apache HTTP Server...
Dec 29 17:42:56 myapp-webserver systemd[1]: Stopped The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@myapp-webserver ~]$
```



```
[ec2-user@myapp-webserver ~]$ sudo systemctl stop httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: inactive (dead) since Mon 2025-12-29 17:47:24 UTC; 19s ago
     Docs: man:httpd.service(8)
  Process: 8207 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
  Process: 10215 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 10215 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 614; Idle/Busy workers 100/0;Requests/sec: 0.00291; Bytes served/sec:   2 B/sec"

Dec 27 07:15:00 ip-10-0-10-225.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Dec 27 07:15:00 ip-10-0-10-225.ec2.internal systemd[1]: Started The Apache HTTP Server.
Dec 28 03:46:02 myapp-webserver systemd[1]: Reloading The Apache HTTP Server.
Dec 28 03:46:02 myapp-webserver httpd[8207]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::88:31ff:fe...is message
Dec 28 03:46:02 myapp-webserver systemd[1]: Reloaded The Apache HTTP Server.
Dec 29 17:47:23 myapp-webserver systemd[1]: Stopping The Apache HTTP Server...
Dec 29 17:47:24 myapp-webserver systemd[1]: Stopped The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@myapp-webserver ~]$ 
```



Not secure 3.239.104.166

🚀 Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-0e8bdc96449aa8d73

Backend Server: web-3 (Primary)

Private IP: 10.0.10.225

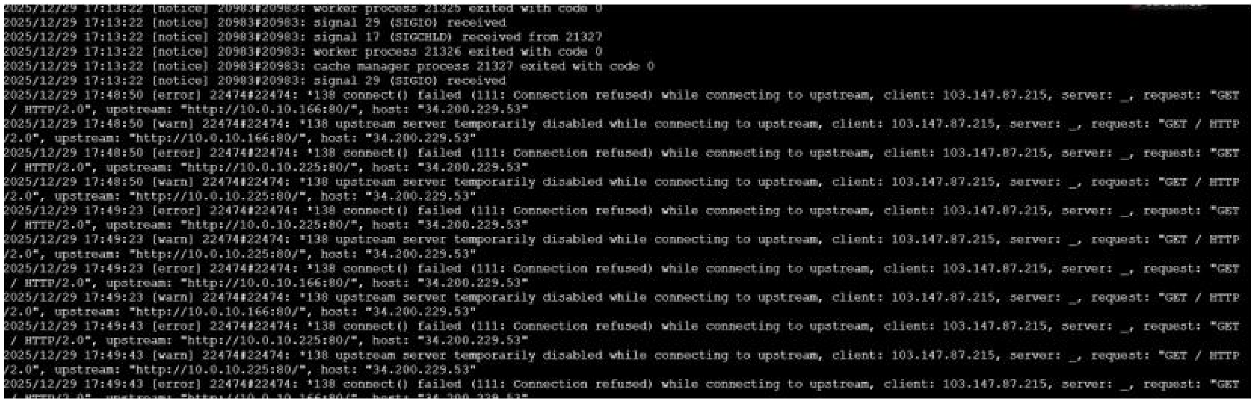
Public IP: 98.81.102.55

Public DNS: ec2-98-81-102-55.compute-1.amazonaws.com

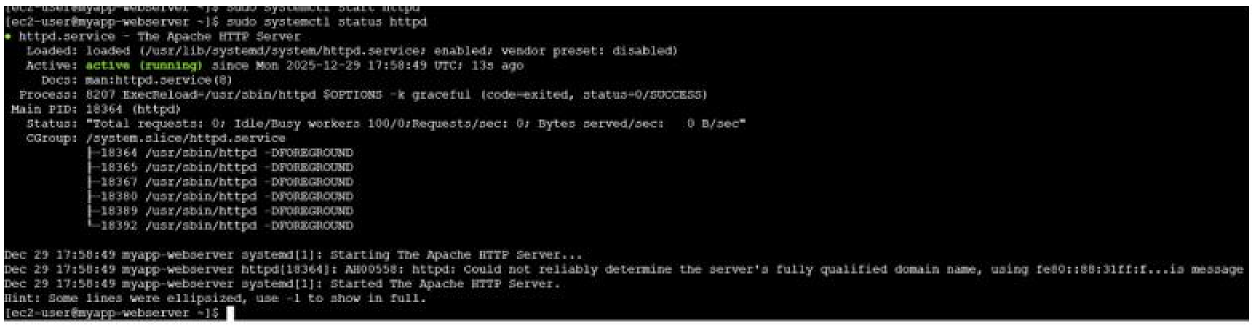
Deployed: Sat Dec 27 07:15:00 UTC 2025

Status: ✅ Active and Running

Managed By: Terraform

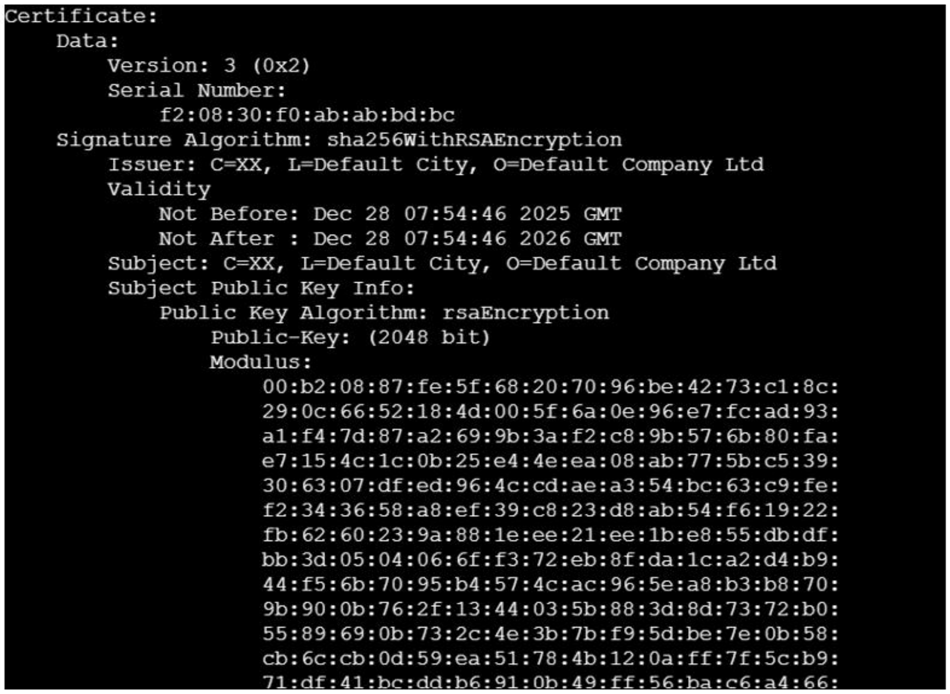


```
2025/12/29 17:13:22 [notice] 20983#20983: worker process 21325 exited with code 0
2025/12/29 17:13:22 [notice] 20983#20983: signal 29 (SIGIO) received
2025/12/29 17:13:22 [notice] 20983#20983: signal 17 (SIGCHLD) received from 21327
2025/12/29 17:13:22 [notice] 20983#20983: worker process 21326 exited with code 0
2025/12/29 17:13:22 [notice] 20983#20983: cache manager process 21327 exited with code 0
2025/12/29 17:13:22 [notice] 20983#20983: signal 29 (SIGIO) received
2025/12/29 17:48:50 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:48:50 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:48:50 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:48:50 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
```

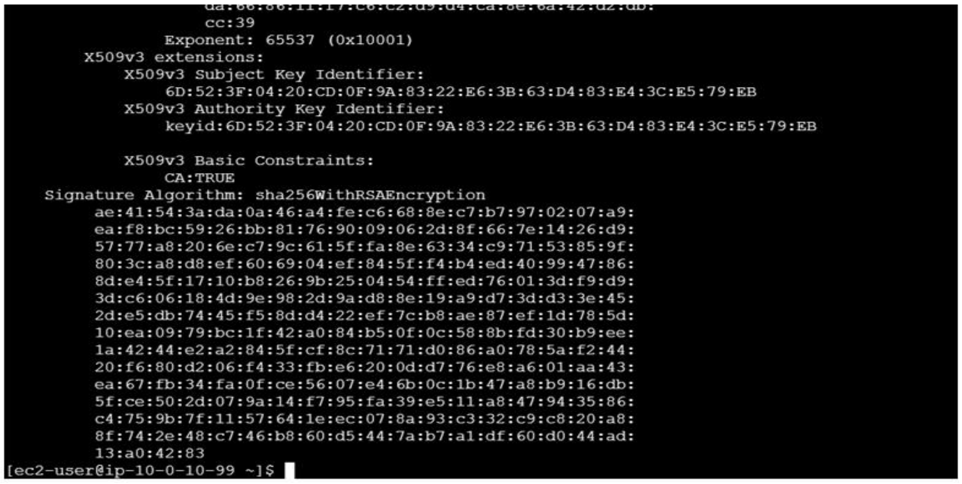


```
[ec2-user@myapp-webserver ~]$ sudo systemctl start httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2025-12-29 17:58:49 UTC; 13s ago
     Docs: man:httpd.service(8)
  Process: 8207 ExecReload=/usr/sbin/httpd $OPTIONS -k graceful (code=exited, status=0/SUCCESS)
 Main PID: 18364 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─18364 /usr/sbin/httpd -DFOREGROUND
           ├─18365 /usr/sbin/httpd -DFOREGROUND
           ├─18367 /usr/sbin/httpd -DFOREGROUND
           ├─18380 /usr/sbin/httpd -DFOREGROUND
           ├─18389 /usr/sbin/httpd -DFOREGROUND
           └─18392 /usr/sbin/httpd -DFOREGROUND

Dec 29 17:58:49 myapp-webserver systemd[1]: Starting The Apache HTTP Server...
Dec 29 17:58:49 myapp-webserver httpd[18364]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::88:31ff:f...is message
Dec 29 17:58:49 myapp-webserver systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@myapp-webserver ~]$
```

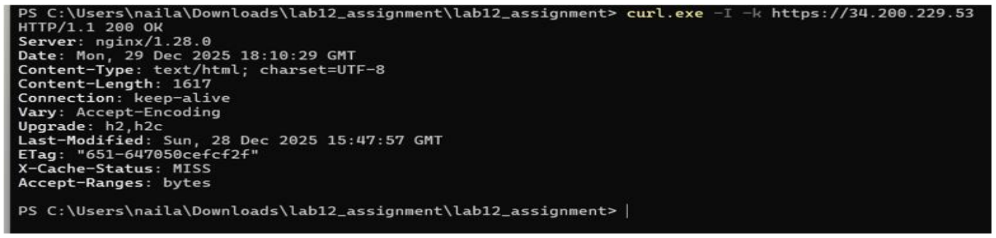
5.5



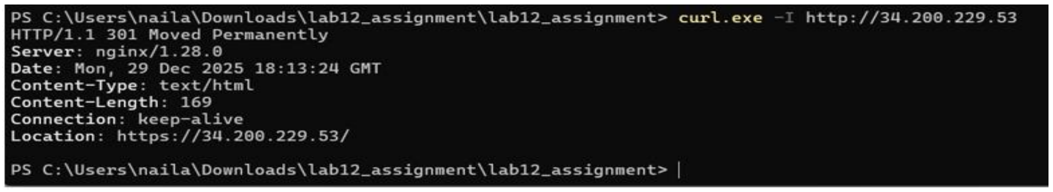
```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f2:08:30:f0:ab:ab:bd:bc
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=XX, L=Default City, O=Default Company Ltd
        Validity
            Not Before: Dec 28 07:54:46 2025 GMT
            Not After : Dec 28 07:54:46 2026 GMT
        Subject: C=XX, L=Default City, O=Default Company Ltd
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b2:08:87:fe:5f:68:20:70:96:be:42:73:c1:8c:
                    29:0c:66:52:18:4d:00:5f:6a:0e:96:e7:fc:ad:93:
                    a1:f4:7d:87:a2:69:9b:3a:f2:c8:9b:57:6b:80:fa:
                    e7:15:4c:1c:0b:25:e4:4e:ea:08:ab:77:5b:c5:39:
                    30:63:07:df:ed:96:4c:cd:ae:a3:54:bc:63:c9:fe:
                    f2:34:36:58:a8:ef:39:c8:23:d8:ab:54:f6:19:22:
                    fb:62:60:23:9a:88:1e:ee:21:ee:1b:e8:55:db:df:
                    bb:3d:05:04:06:6f:f3:72:eb:8f:da:1c:a2:d4:b9:
                    44:f5:6b:70:95:b4:57:4c:ac:96:5e:a8:b3:b8:70:
                    9b:90:0b:76:2f:13:44:03:5b:88:3d:8d:73:72:b0:
                    55:89:69:0b:73:2c:4e:3b:7b:f9:5d:be:7e:0b:58:
                    cb:6c:cb:0d:59:ea:51:78:4b:12:0a:ff:7f:5c:b9:
                    71:df:41:bc:dd:b6:91:0b:49:ff:56:ba:c6:a4:66:
```



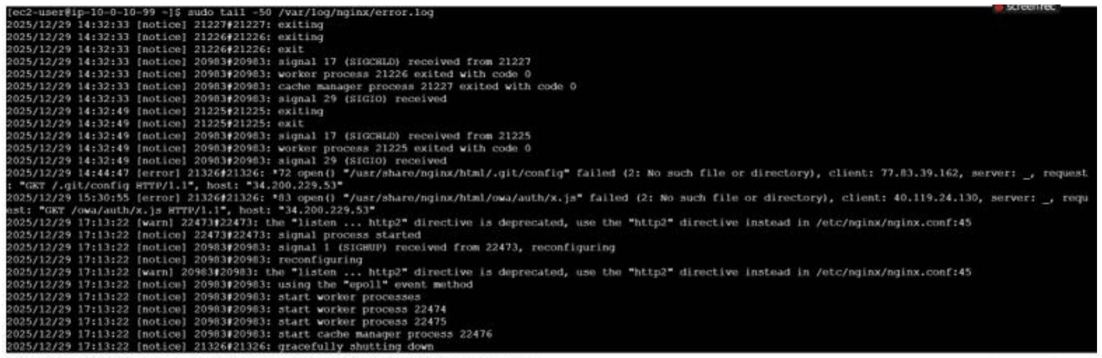
```
                    da:66:86:11:17:c6:c2:d9:d4:ca:8e:6a:42:d2:db:
                    cc:39
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                6D:52:3F:04:20:CD:0F:9A:83:22:E6:3B:63:D4:83:E4:3C:E5:79:EB
            X509v3 Authority Key Identifier:
                keyid:6D:52:3F:04:20:CD:0F:9A:83:22:E6:3B:63:D4:83:E4:3C:E5:79:EB

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
         ae:41:54:3a:da:0a:46:a4:fe:c6:68:8e:c7:b7:97:02:07:a9:
         ea:f8:bc:59:26:bb:81:76:90:09:06:2d:8f:66:7e:14:26:d9:
         57:77:a8:20:6e:c7:9c:61:5f:fa:8e:63:34:c9:71:53:85:9f:
         80:3c:a8:d8:ef:60:69:04:ef:84:5f:f4:b4:ed:40:99:47:86:
         8d:e4:5f:17:10:b8:26:9b:25:04:54:ff:ed:76:01:3d:f9:d9:
         3d:c6:06:18:4d:9e:98:2d:9a:d8:8e:19:a9:d7:3d:d3:3e:45:
         2d:e5:db:74:45:f5:8d:d4:22:ef:7c:b8:ae:87:ef:1d:78:5d:
         10:ea:09:79:bc:1f:42:a0:84:b5:0f:0c:58:8b:fd:30:b9:ee:
         1a:42:44:e2:a2:84:5f:cf:8c:71:71:d0:86:a0:78:5a:f2:44:
         20:f6:80:d2:06:f4:33:fb:e6:20:0d:d7:76:e8:a6:01:aa:43:
         ea:67:fb:34:fa:0f:ce:56:07:e4:6b:0c:1b:47:a8:b9:16:db:
         5f:ce:50:2d:07:9a:14:f7:95:fa:39:e5:11:a8:47:94:35:86:
         c4:75:9b:7f:11:57:64:1e:ec:07:8a:93:c3:32:c9:c8:20:a8:
         8f:74:2e:48:c7:46:b8:60:d5:44:7a:b7:a1:df:60:d0:44:ad:
         13:a0:42:83
[ec2-user@ip-10-0-10-99 ~]$
```



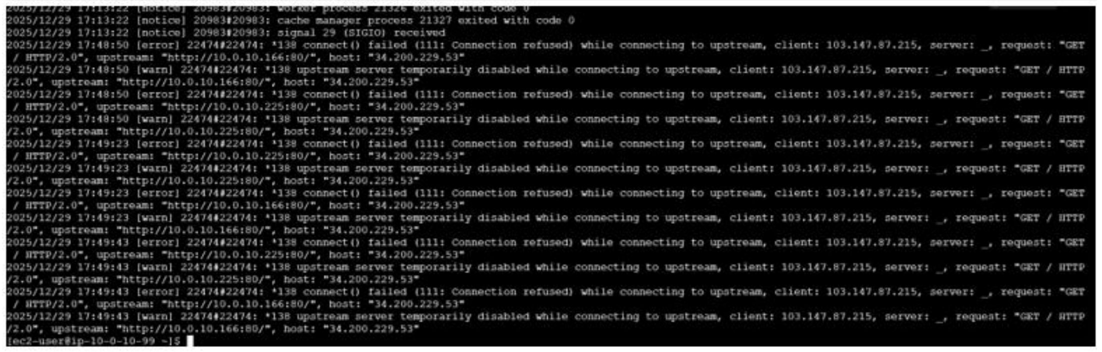
```
PS C:\Users\naila\Downloads\lab12_assignment\lab12_assignment> curl.exe -I -k https://34.200.229.53
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Mon, 29 Dec 2025 18:10:29 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1617
Connection: keep-alive
Vary: Accept-Encoding
Upgrade: h2,h2c
Last-Modified: Sun, 28 Dec 2025 15:47:57 GMT
ETag: "651-647050cefcf2f"
X-Cache-Status: MISS
Accept-Ranges: bytes

PS C:\Users\naila\Downloads\lab12_assignment\lab12_assignment> |
```



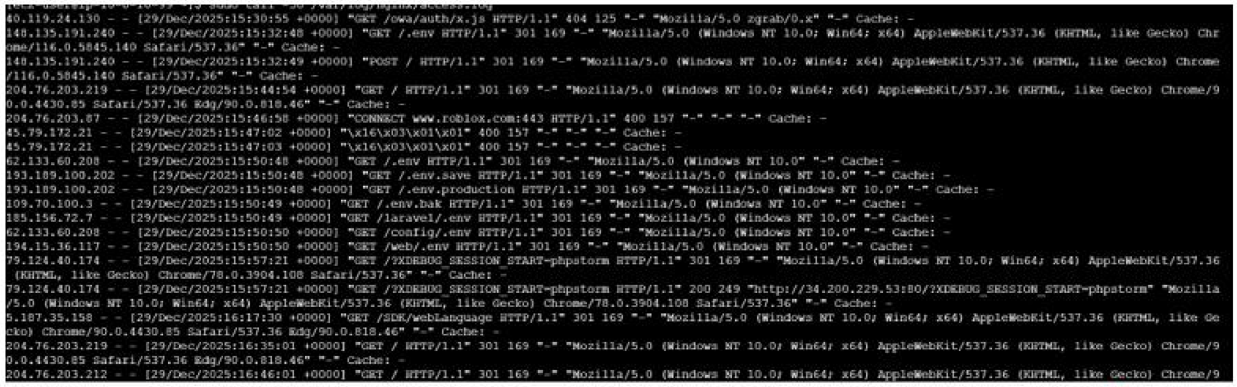
```
PS C:\Users\naila\Downloads\lab12_assignment\lab12_assignment> curl.exe -I http://34.200.229.53
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Mon, 29 Dec 2025 18:13:24 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://34.200.229.53/

PS C:\Users\naila\Downloads\lab12_assignment\lab12_assignment> |
```



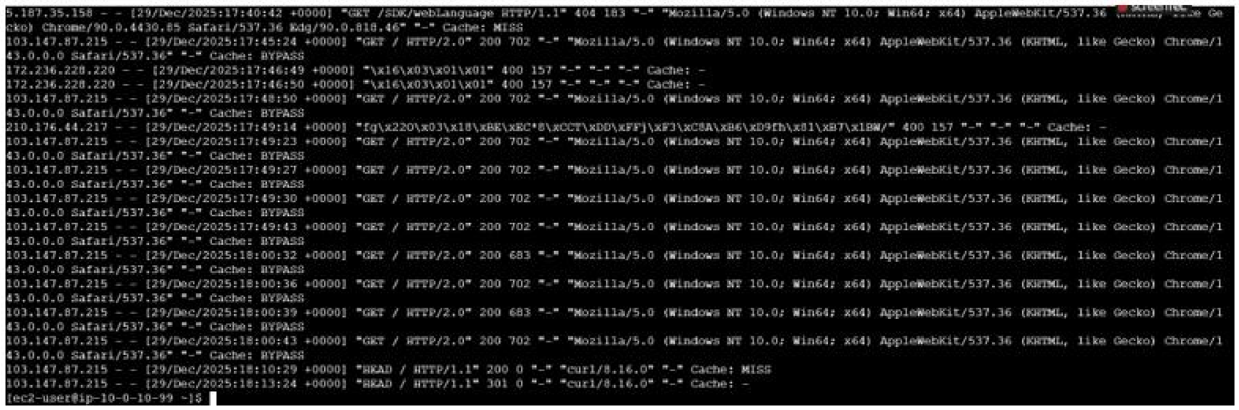
```
[ec2-user@ip-10-0-10-99 ~]$ sudo tail -50 /var/log/nginx/error.log
2025/12/29 14:32:33 [notice] 21227#21227: exiting
2025/12/29 14:32:33 [notice] 21226#21226: exiting
2025/12/29 14:32:33 [notice] 21226#21226: exit
2025/12/29 14:32:33 [notice] 20983#20983: signal 17 (SIGCHLD) received from 21227
2025/12/29 14:32:33 [notice] 20983#20983: worker process 21226 exited with code 0
2025/12/29 14:32:33 [notice] 20983#20983: cache manager process 21227 exited with code 0
2025/12/29 14:32:33 [notice] 20983#20983: signal 29 (SIGIO) received
2025/12/29 14:32:49 [notice] 21225#21225: exiting
2025/12/29 14:32:49 [notice] 21225#21225: exit
2025/12/29 14:32:49 [notice] 20983#20983: signal 17 (SIGCHLD) received from 21225
2025/12/29 14:32:49 [notice] 20983#20983: worker process 21225 exited with code 0
2025/12/29 14:32:49 [notice] 20983#20983: signal 29 (SIGIO) received
2025/12/29 14:44:47 [error] 21326#21326: *72 open() "/usr/share/nginx/html/.git/config" failed (2: No such file or directory), client: 77.83.39.162, server: _, request
: "GET /.git/config HTTP/1.1", host: "34.200.229.53"
2025/12/29 15:30:55 [error] 21326#21326: *83 open() "/usr/share/nginx/html/owa/auth/x.js" failed (2: No such file or directory), client: 40.119.24.130, server: _, requ
est: "GET /owa/auth/x.js HTTP/1.1", host: "34.200.229.53"
2025/12/29 17:13:22 [warn] 22473#22473: the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:45
2025/12/29 17:13:22 [notice] 22473#22473: signal process started
2025/12/29 17:13:22 [notice] 20983#20983: signal 1 (SIGHUP) received from 22473, reconfiguring
2025/12/29 17:13:22 [notice] 20983#20983: reconfiguring
2025/12/29 17:13:22 [warn] 20983#20983: the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:45
2025/12/29 17:13:22 [notice] 20983#20983: using the "epoll" event method
2025/12/29 17:13:22 [notice] 20983#20983: start worker processes
2025/12/29 17:13:22 [notice] 20983#20983: start worker process 22474
2025/12/29 17:13:22 [notice] 20983#20983: start worker process 22475
2025/12/29 17:13:22 [notice] 20983#20983: start cache manager process 22476
2025/12/29 17:13:22 [notice] 21326#21326: gracefully shutting down
```



```
2025/12/29 17:13:22 [notice] 20983#20983: worker process 21326 exited with code 0
2025/12/29 17:13:22 [notice] 20983#20983: cache manager process 21327 exited with code 0
2025/12/29 17:13:22 [notice] 20983#20983: signal 29 (SIGIO) received
2025/12/29 17:48:50 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:48:50 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:48:50 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:48:50 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:49:23 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.225:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [error] 22474#22474: *138 connect() failed (111: Connection refused) while connecting to upstream, client: 103.147.87.215, server: _, request: "GET
 / HTTP/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
2025/12/29 17:49:43 [warn] 22474#22474: *138 upstream server temporarily disabled while connecting to upstream, client: 103.147.87.215, server: _, request: "GET / HTTP
/2.0", upstream: "http://10.0.10.166:80/", host: "34.200.229.53"
[ec2-user@ip-10-0-10-99 ~]$
```

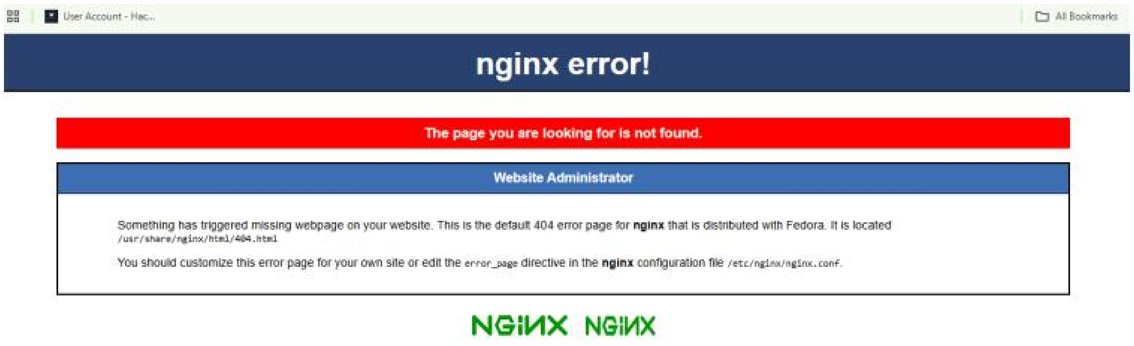


```
[ec2-user@ip-10-0-10-99 ~]$ sudo tail -50 /var/log/nginx/access.log
40.119.24.130 - - [29/Dec/2025:15:30:55 +0000] "GET /owa/auth/x.js HTTP/1.1" 404 125 "-" "Mozilla/5.0 zgrab/0.x" "-" Cache: -
148.135.191.240 - - [29/Dec/2025:15:32:48 +0000] "GET /.env HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
ome/116.0.5845.140 Safari/537.36" "-" Cache: -
148.135.191.240 - - [29/Dec/2025:15:32:49 +0000] "POST / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
/116.0.5845.140 Safari/537.36" "-" Cache: -
204.76.203.219 - - [29/Dec/2025:15:44:54 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
0.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
204.76.203.87 - - [29/Dec/2025:15:46:58 +0000] "CONNECT www.roblox.com:443 HTTP/1.1" 400 157 "-" "-" "-" Cache: -
45.79.172.21 - - [29/Dec/2025:15:47:02 +0000] "\x16\x03\x01\x01" 400 157 "-" "-" "-" Cache: -
45.79.172.21 - - [29/Dec/2025:15:47:03 +0000] "\x16\x03\x01\x01" 400 157 "-" "-" "-" Cache: -
62.133.60.208 - - [29/Dec/2025:15:50:48 +0000] "GET /.env HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
193.189.100.202 - - [29/Dec/2025:15:50:48 +0000] "GET /.env.save HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
193.189.100.202 - - [29/Dec/2025:15:50:48 +0000] "GET /.env.production HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
109.70.100.3 - - [29/Dec/2025:15:50:49 +0000] "GET /.env.bak HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
185.156.72.7 - - [29/Dec/2025:15:50:49 +0000] "GET /laravel/.env HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
62.133.60.208 - - [29/Dec/2025:15:50:50 +0000] "GET /config/.env HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
194.15.36.117 - - [29/Dec/2025:15:50:50 +0000] "GET /web/.env HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0" "-" Cache: -
79.124.40.174 - - [29/Dec/2025:15:57:21 +0000] "GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36" "-" Cache: -
79.124.40.174 - - [29/Dec/2025:15:57:21 +0000] "GET /?XDEBUG_SESSION_START=phpstorm HTTP/1.1" 200 249 "http://34.200.229.53:80/?XDEBUG_SESSION_START=phpstorm" "Mozilla
/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36" "-" Cache: -
5.187.35.158 - - [29/Dec/2025:16:17:30 +0000] "GET /SDK/webLanguage HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
cko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
204.76.203.219 - - [29/Dec/2025:16:35:01 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
0.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: -
204.76.203.212 - - [29/Dec/2025:16:46:01 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/9
```



```
5.187.35.158 - - [29/Dec/2025:17:40:42 +0000] "GET /SDK/webLanguage HTTP/1.1" 404 183 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 [illegible] Ge
cko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-" Cache: MISS
103.147.87.215 - - [29/Dec/2025:17:45:24 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
172.236.228.220 - - [29/Dec/2025:17:46:49 +0000] "\x16\x03\x01\x01" 400 157 "-" "-" "-" Cache: -
172.236.228.220 - - [29/Dec/2025:17:46:50 +0000] "\x16\x03\x01\x01" 400 157 "-" "-" "-" Cache: -
103.147.87.215 - - [29/Dec/2025:17:48:50 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
210.176.44.217 - - [29/Dec/2025:17:49:14 +0000] "fq\x220\x03\x18\xBE\xEC*8\xCCT\xDD\xFFj\xFJ\xC8A\xB6\xD9fh\x81\xB7\x1BW/" 400 157 "-" "-" "-" Cache: -
103.147.87.215 - - [29/Dec/2025:17:49:23 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:17:49:27 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:17:49:30 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:17:49:43 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:18:00:32 +0000] "GET / HTTP/2.0" 200 683 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:18:00:36 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:18:00:39 +0000] "GET / HTTP/2.0" 200 683 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:18:00:43 +0000] "GET / HTTP/2.0" 200 702 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache: BYPASS
103.147.87.215 - - [29/Dec/2025:18:10:29 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.16.0" "-" Cache: MISS
103.147.87.215 - - [29/Dec/2025:18:13:24 +0000] "HEAD / HTTP/1.1" 301 0 "-" "curl/8.16.0" "-" Cache: -
[ec2-user@ip-10-0-10-99 ~]$
```

Bonus Task:



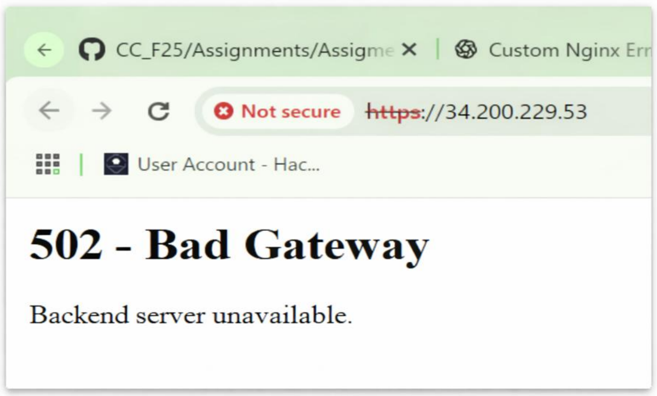
nginx error!

The page you are looking for is not found.

Website Administrator

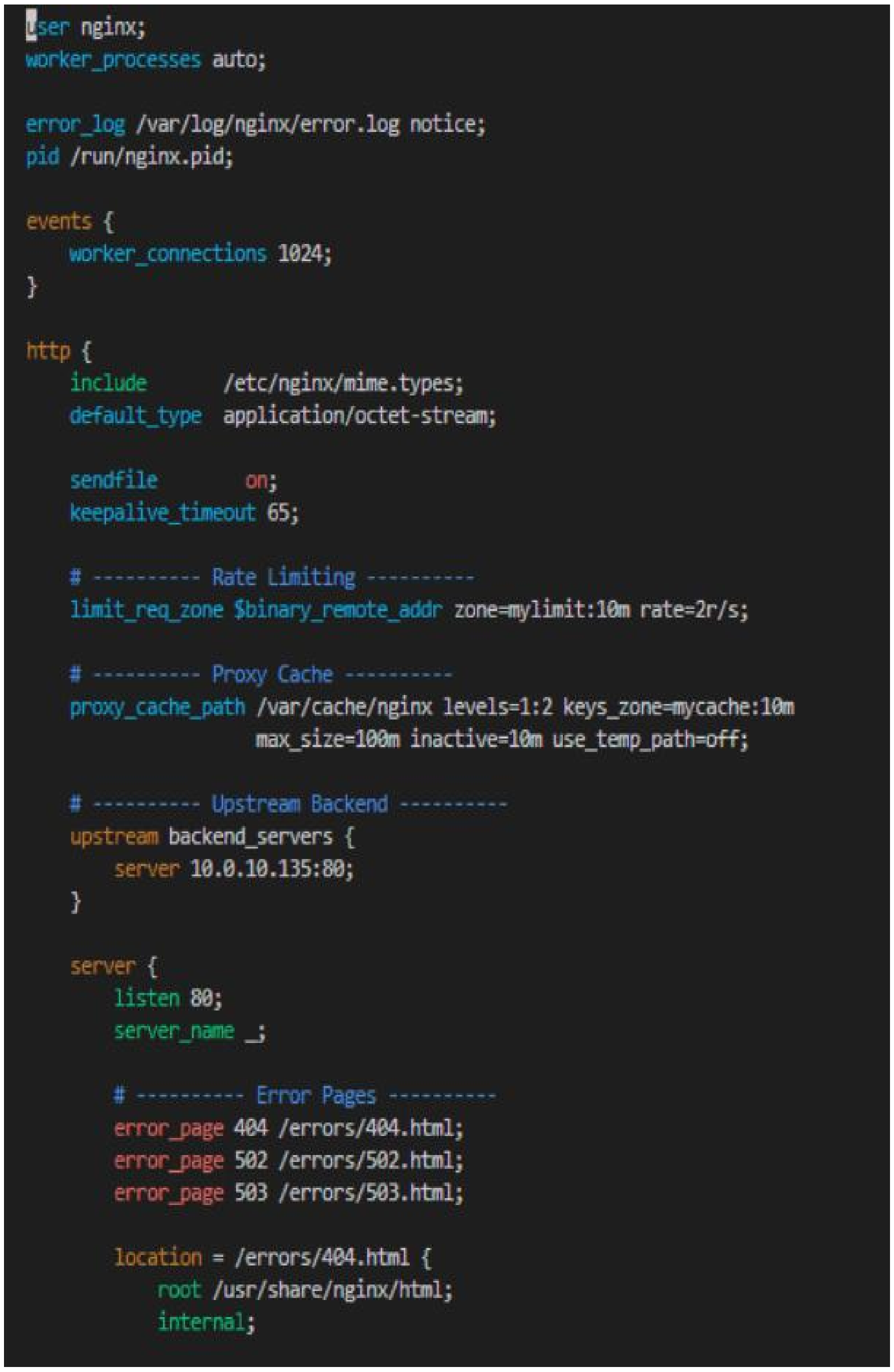
Something has triggered missing webpage on your website. This is the default 404 error page for **nginx** that is distributed with Fedora. It is located `/usr/share/nginx/html/404.html`

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`.

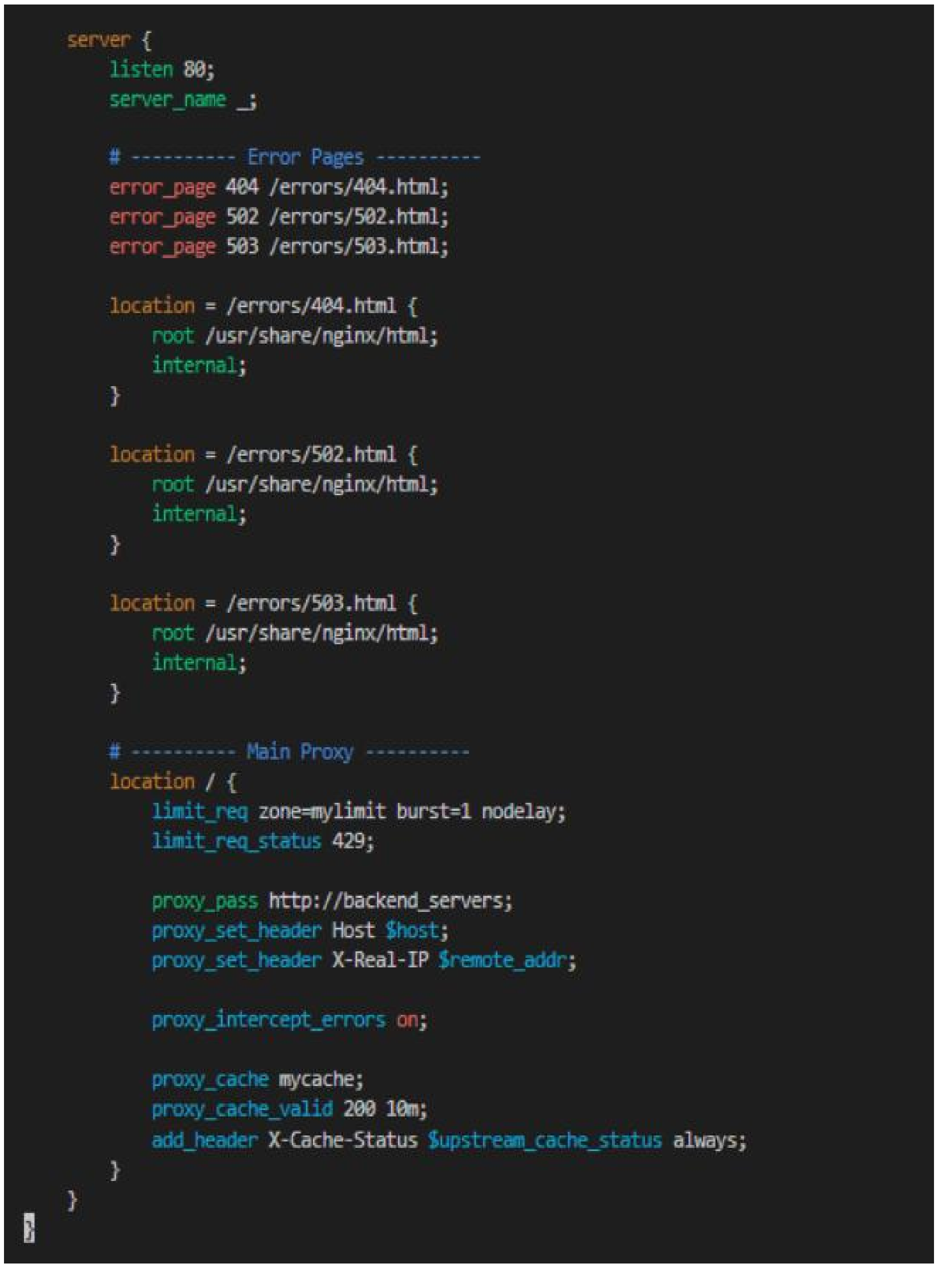


502 - Bad Gateway

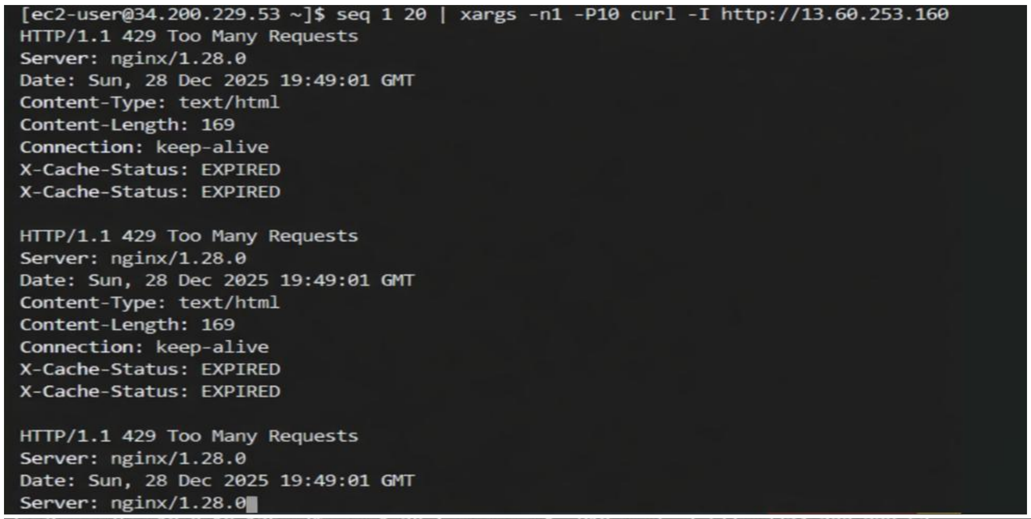
Backend server unavailable.



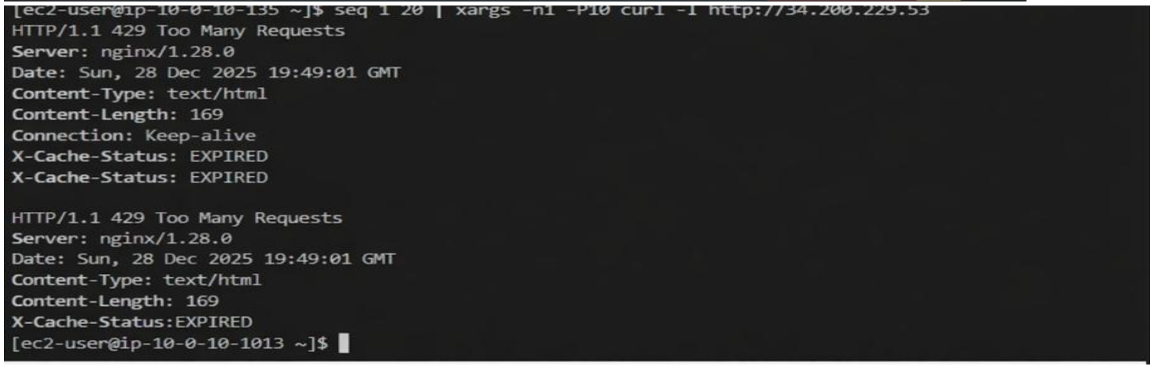
```
user nginx;
worker_processes auto;

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;
```



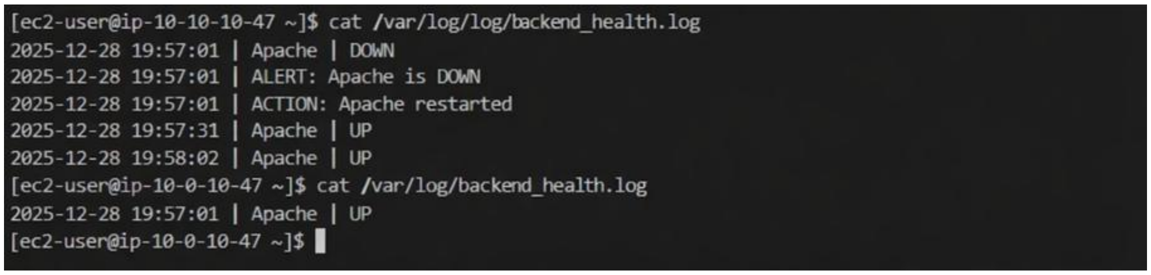
```
    server {
        listen 80;
        server_name _;

        # ---------- 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;
        }
    }
}
```



```
[ec2-user@34.200.229.53 ~]$ seq 1 20 | xargs -n1 -P10 curl -I http://13.60.253.160
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
Server: nginx/1.28.0
```



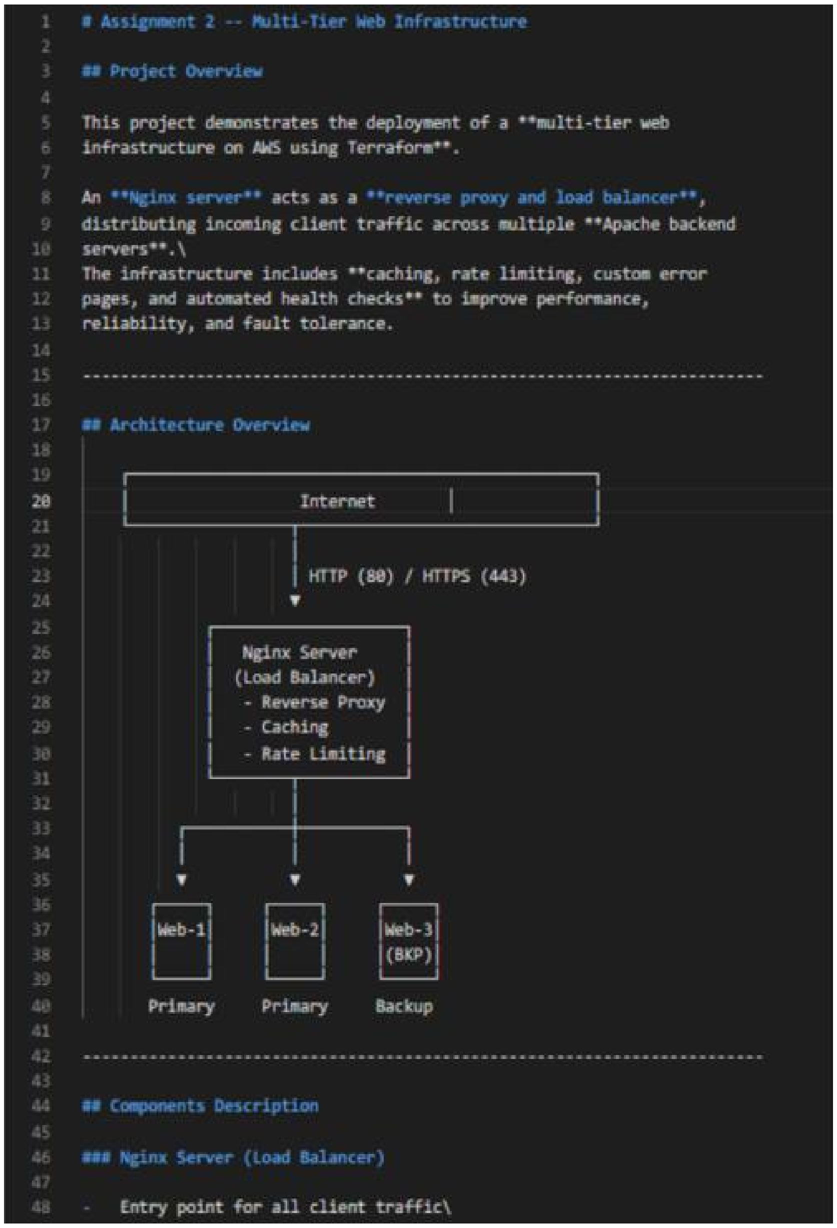
```
[ec2-user@ip-10-0-10-135 ~]$ seq 1 20 | xargs -n1 -P10 curl -I http://34.200.229.53
HTTP/1.1 429 Too Many Requests
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 19:49:01 GMT
Content-Type: text/html
Content-Length: 169
Connection: Keep-alive
X-Cache-Status: EXPIRED
X-Cache-Status: EXPIRED

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
X-Cache-Status:EXPIRED
[ec2-user@ip-10-0-10-1013 ~]$
```



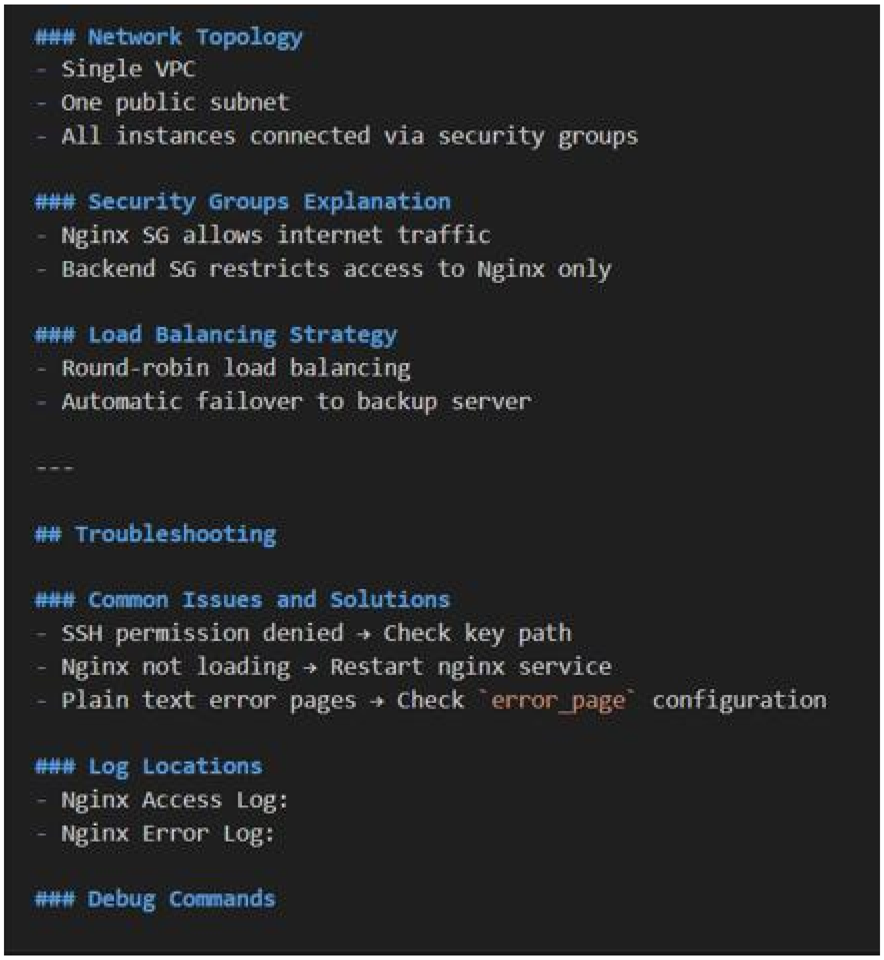
```
[ec2-user@ip-10-10-10-47 ~]$ cat /var/log/log/backend_health.log
2025-12-28 19:57:01 | Apache | DOWN
2025-12-28 19:57:01 | ALERT: Apache is DOWN
2025-12-28 19:57:01 | ACTION: Apache restarted
2025-12-28 19:57:31 | Apache | UP
2025-12-28 19:58:02 | Apache | UP
[ec2-user@ip-10-0-10-47 ~]$ cat /var/log/backend_health.log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

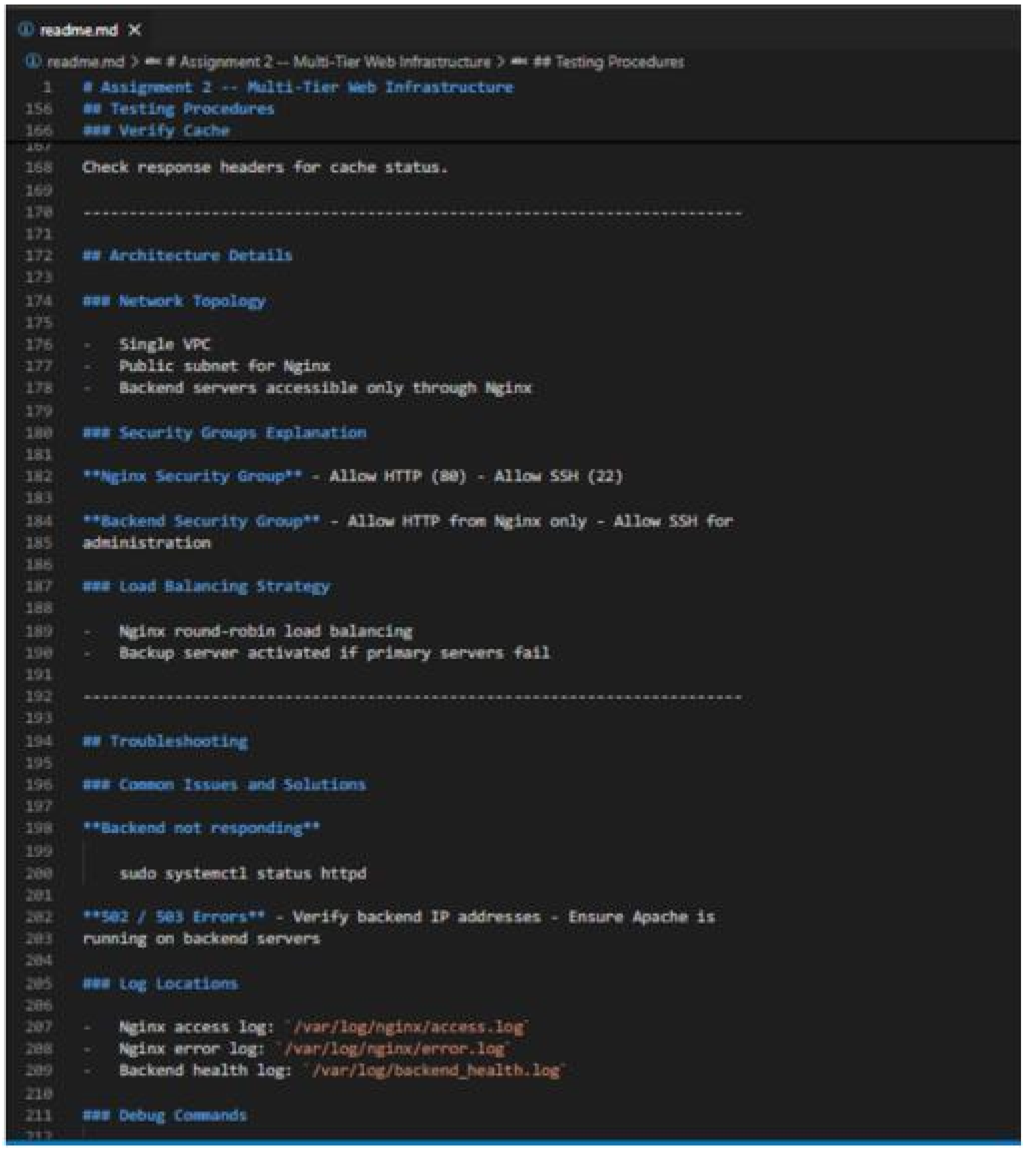
Part 6:
Cleanup All resources were destroyed using Terraform to avoid unnecessary AWS charges. The cleanup process removed EC2 instances, networking components, and security groups. Command used: terraform destroy The Terraform state file confirmed that no resources remained.

Task 6:

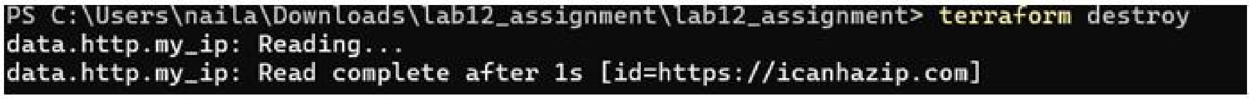


```
# Assignment 2 -- Multi-Tier Web Infrastructure

## 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\
```

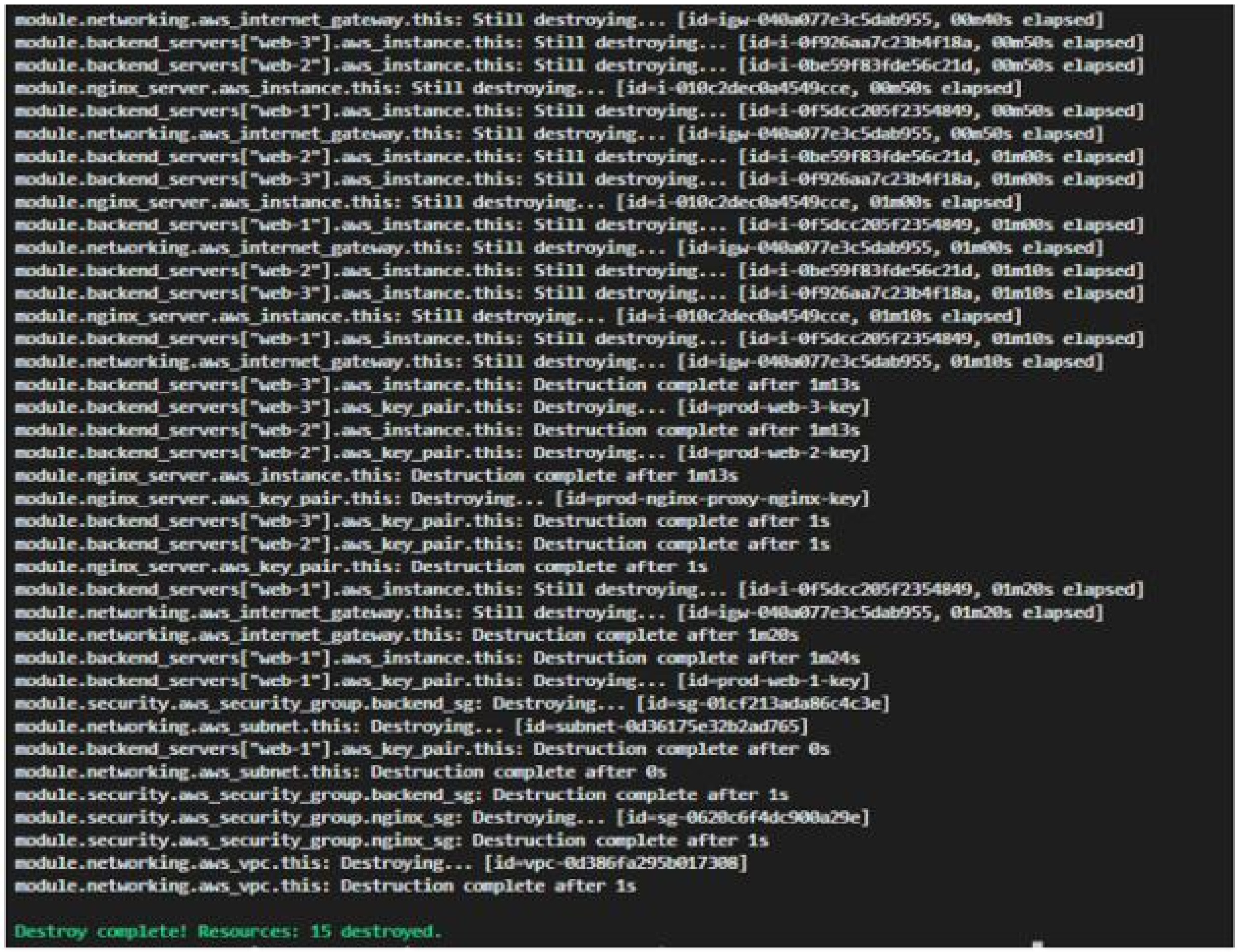

```
### Network Topology
- Single VPC
- One public subnet
- All instances connected via security groups

### Security Groups Explanation
- Nginx SG allows internet traffic
- Backend SG restricts access to Nginx only

### Load Balancing Strategy
- Round-robin load balancing
- Automatic failover to backup server

---

## Troubleshooting

### Common Issues and Solutions
- SSH permission denied → Check key path
- Nginx not loading → Restart nginx service
- Plain text error pages → Check `error_page` configuration

### Log Locations
- Nginx Access Log:
- Nginx Error Log:

### Debug Commands
```


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

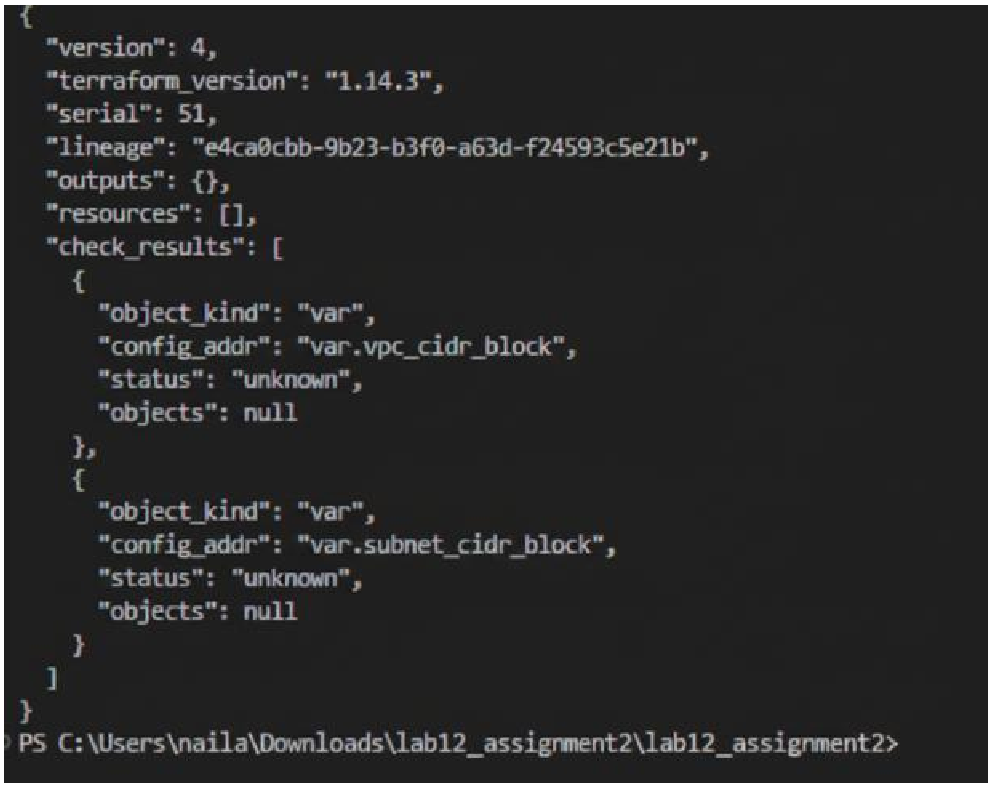
```
# Assignment 2 -- Multi-Tier Web Infrastructure
## Testing Procedures
### Verify Cache

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
```



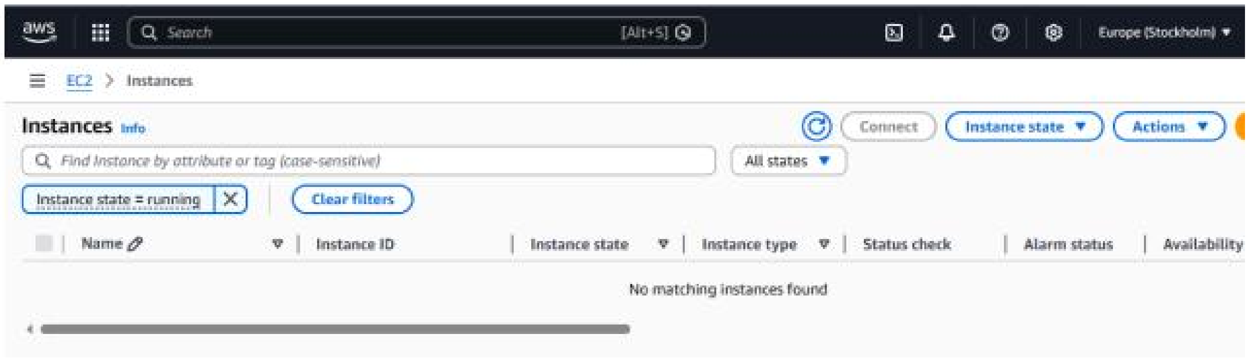
```
PS C:\Users\naila\Downloads\lab12_assignment\lab12_assignment> terraform destroy
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 1s [id=https://icanhazip.com]
```



```
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.
```



```
{
  "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\naila\Downloads\lab12_assignment2\lab12_assignment2>
```



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability

No matching instances found

4. Testing Results

4.1 Load Balancing Tests

The load balancing functionality was verified by sending multiple requests to the application. The requests were handled alternately by Web-1 and Web-2 backend servers, which confirmed that traffic was being distributed correctly by the Nginx load balancer.

4.2 Cache Performance Tests

During cache testing, the first request resulted in an **X-Cache-Status: MISS**, while repeated requests showed **X-Cache-Status: HIT**. This behavior confirmed that Nginx caching was working as expected and reduced repeated backend processing.

4.3 High Availability Tests

To test high availability, one of the primary backend servers was intentionally stopped. The Nginx server successfully redirected incoming traffic to the remaining active server, ensuring uninterrupted service.

4.4 Security Tests

Security was validated by attempting direct access to backend servers. These attempts were blocked due to security group rules, confirming that only the Nginx server was publicly accessible.

4.5 Performance Metrics

Overall performance improved as caching reduced response times and load balancing minimized the workload on individual backend servers, resulting in better stability and efficiency.

5. Challenges & Solutions

Terraform Initialization and Dependency Issues

During the initial setup, Terraform failed to initialize properly due to missing provider configurations and incorrect module references. This caused errors while running `terraform init` and `terraform plan`.

Solution:
The issue was resolved by carefully verifying provider blocks, correcting module paths, and re-initializing Terraform. Proper file organization and consistent naming helped ensure successful initialization.

SSH Access and Key Pair Errors

SSH access to EC2 instances initially failed due to incorrect key pair usage and improper file permissions on the private key file.

Solution:
The correct `.pem` key file was used, and file permissions were fixed using appropriate commands. Security group rules were also verified to allow SSH access from the correct IP address.

Nginx Reverse Proxy and Load Balancing Configuration

Configuring Nginx as a reverse proxy with load balancing was challenging due to syntax errors and incorrect backend server definitions, which resulted in failed request routing.

Solution:
Nginx configuration files were reviewed and validated using syntax checks. Backend server IPs were corrected, and Nginx logs were used to debug and confirm proper request forwarding.

Cache Behavior Verification

Initially, it was difficult to confirm whether caching was working correctly, as responses appeared similar across requests.

Solution:
HTTP response headers were analyzed to verify cache status. Repeated requests confirmed cache behavior through MISS and HIT responses

Lessons Learned:

Through these challenges, I learned the importance of careful Terraform configuration and module management to avoid initialization and deployment issues. I also gained hands-on experience in troubleshooting cloud infrastructure by analyzing logs, verifying permissions, and debugging configuration files. Additionally, I learned how proper testing and verification techniques, such as checking response headers and failover behavior, are essential to ensure system reliability and performance.

6. Conclusion

This assignment successfully demonstrated the deployment of a multi-tier web infrastructure on AWS using Terraform. The project strengthened my understanding of cloud architecture concepts such as load balancing, caching, security, and automation. Hands-on experience was gained in Terraform provisioning, Linux scripting, and Nginx configuration. Overall, the assignment provided valuable practical exposure to real-world cloud deployment scenarios. Future improvements could include implementing auto-scaling groups, integrating CloudWatch monitoring, and using AWS ACM certificates for managed HTTPS.

7. Appendices

- Complete Terraform configuration files

- Apache and Nginx setup scripts

- Additional screenshots for verification
- References