



**FATIMA JINNAH WOMEN UNIVERSITY**

**Department of Software Engineering**

**Assignment 2**

**Advanced Terraform & Nginx Multi-Tier Architecture**

**Course Title:**

Cloud Compting

**Submitted to:**

Mr. Waqas Saleem

**Submitted by:**

Rughma Malik

2023-BSE-054

V-B

**Submission Date:**

31-12-2025

# Table of Contents

<b>1. Executive Summary .....</b>	<b>4</b>
<b>2. Architecture Design .....</b>	<b>5</b>
2.1 Architecture Overview .....	5
2.2 Architecture Diagram .....	5
2.3 Network Topology.....	6
2.4 Security Design .....	6
<b>3. Implementation Details .....</b>	<b>6</b>
Part 1: Infrastructure Setup (25 marks) .....	6
1.1 Project Structure (5 marks) .....	6
1.2 Variable Configuration (5 marks) .....	7
1.3 Networking Module (5 marks) .....	8
1.4 Security Module (5 marks) .....	9
1.5 Locals Configuration (5 marks) .....	10
Part 2: Webserver Module (15 marks) .....	11
2.1 Module Design (10 marks) .....	11
2.2 Module Usage (5 marks) .....	12
Part 3: Server Configuration Scripts (20 marks) .....	13
3.1 Apache Backend Server Script (10 marks) .....	13
3.2 Nginx Server Setup Script (10 marks).....	13
Part 4: Infrastructure Deployment (15 marks).....	14
4.1 Initial Deployment (5 marks) .....	14
4.2 Output Configuration (5 marks).....	16
4.3 AWS Console Verification (5 marks) .....	17
Part 5: Nginx Configuration & Testing (25 marks).....	18
5.1 Update Nginx Backend Configuration (5 marks) .....	18
5.2 Test Load Balancing (5 marks).....	19
5.3 Test Cache Functionality (5 marks).....	21

5.4 Test High Availability (Backup Server) (5 marks).....	23
5.5 Security & Performance Analysis (5 marks) .....	25
Bonus Tasks (10 marks extra credit) .....	27
Bonus 1: Custom Error Pages (3 marks) .....	27
Bonus 2: Implement Rate Limiting (3 marks) .....	28
Bonus 3: Health Check Automation (4 marks) .....	29
Part 6: Documentation & Cleanup (10 marks) .....	30
6.1 README Documentation (5 marks) .....	30
6.2 Infrastructure Cleanup (5 marks) .....	30
<b>4. Testing Results .....</b>	<b>31</b>
4.1 Load Balancing Test.....	31
4.2 Cache Performance Test.....	32
4.3 High Availability Test.....	33
4.4 Security Testing.....	34
<b>5. Challenges &amp; Solutions .....</b>	<b>35</b>
Challenge 1: Terraform command not found .....	35
Challenge 2: AWS credential errors .....	35
Challenge 3: Incorrect Availability Zone.....	35
Challenge 4: Nginx Syntax Error .....	35
Challenge 5: Availability Zone Mismatch .....	35
<b>6. Conclusion.....</b>	<b>36</b>
<b>7. Appendices .....</b>	<b>36</b>
Appendix A – Repository Link with Complete Code Listings.....	36
Appendix B: Configuration Files .....	36
Appendix C: Additional Screenshots .....	36
Bonus 1: Custom 404 Error Page .....	36
Bonus 2: Rate Limit Testing (429 Too Many Requests).....	37
Bonus 3: Health Monitor Log .....	37
Appendix D: References .....	37

# **Advanced Terraform & Nginx Multi-Tier Architecture**

## **1. Executive Summary**

This assignment focuses on the design, deployment, and testing of a highly available multi-tier web infrastructure on Amazon Web Services (AWS) using Terraform as an Infrastructure as Code (IaC) tool.

The deployed infrastructure consists of a Virtual Private Cloud (VPC), a public subnet, security groups, three Apache backend web servers, and an Nginx reverse proxy acting as a load balancer. Secure communication is enforced using HTTPS with a self-signed SSL certificate, and performance is enhanced through Nginx caching.

Key achievements include:

- Automated infrastructure provisioning using Terraform
- Load balancing between multiple backend servers
- High availability using a backup web server
- Secure access with SSL/TLS and security headers
- Successful validation through extensive testing

This project demonstrates practical knowledge of cloud infrastructure, DevOps automation, and web server configuration.

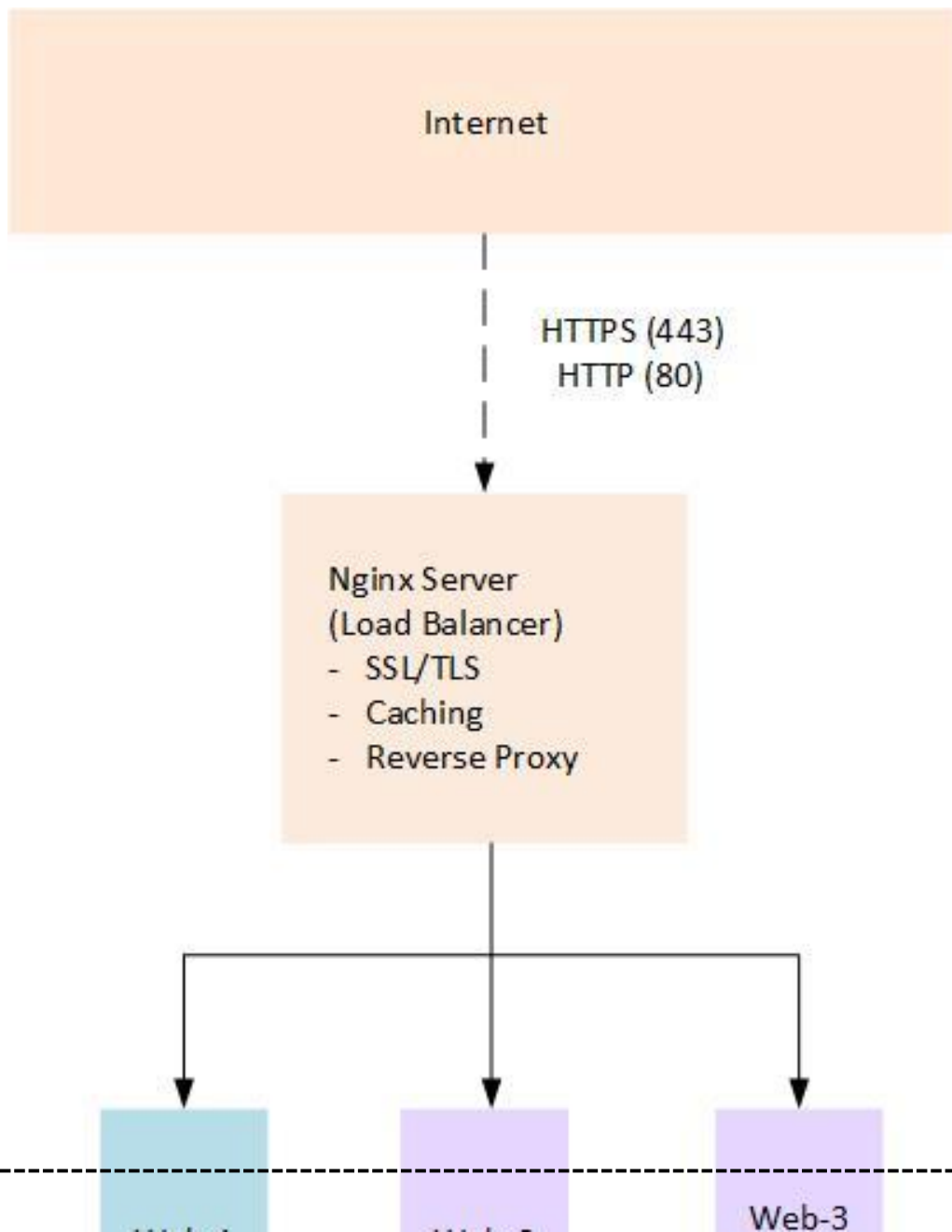
## 2. Architecture Design

### 2.1 Architecture Overview

The system follows a **three-tier architecture**:

- **Client Layer:** End users accessing the application via browser
- **Application Layer:** Nginx server acting as reverse proxy and load balancer
- **Backend Layer:** Apache web servers serving application content

### 2.2 Architecture Diagram



## 2.3 Network Topology

- Single VPC
- One public subnet
- Internet Gateway for external access
- Route table allowing outbound internet traffic

## 2.4 Security Design

- Separate security groups for Nginx and backend servers
- Restricted backend access (HTTP allowed only from Nginx)
- SSH access enabled for management
- HTTPS enforced using SSL/TLS

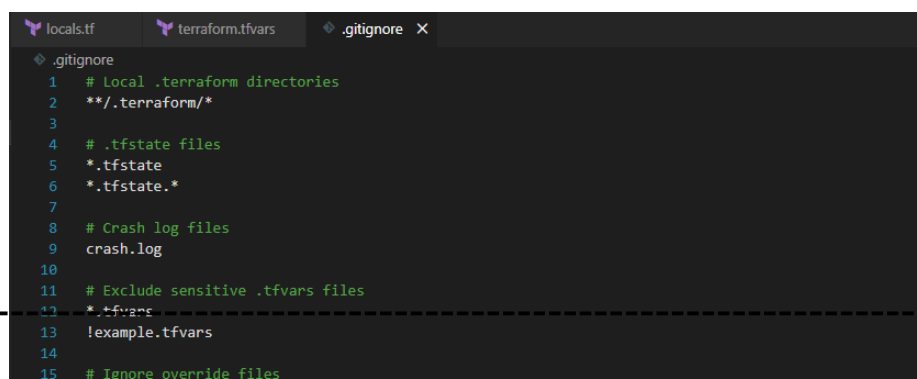
# 3. Implementation Details

## Part 1: Infrastructure Setup (25 marks)

### 1.1 Project Structure (5 marks)

#### Tasks:

- Create all necessary files and directories
- Implement proper .gitignore to exclude sensitive files
- Document the project structure in README.md



The screenshot shows a code editor with a dark theme. The top of the editor has three tabs: 'locals.tf', 'terraform.tfvars', and '.gitignore'. The '.gitignore' tab is active, showing the following content:

```
1 # Local .terraform directories
2 **/.terraform/*
3
4 # .tfstate files
5 *.tfstate
6 *.tfstate.*
7
8 # Crash log files
9 crash.log
10
11 # Exclude sensitive .tfvars files
12 *.tfvars
13 !example.tfvars
14
15 # Ignore override files
```

```
PS D:\docs\Assignment2> tree /F
Volume serial number is EE83-994E
.
├── .gitignore
├── locals.tf
├── main.tf
├── outputs.tf
├── README.md
├── terraform.tfvars
├── variables.tf
├── docs
├── modules
│   ├── networking
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   ├── security
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   └── webserver
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── screenshots
│   ├── bonus
│   ├── part1
│   ├── part2
│   ├── part3
│   ├── part4
│   ├── part5
│   └── part6
└── scripts
    ├── apache-setup.sh
    └── nginx-setup.sh
```

Ln 24, Col 6   Spaces: 4   UTF-8   CRLF

## 1.2 Variable Configuration (5 marks)

### Tasks:

- Add validation rules for CIDR blocks
- Add descriptions for all variables
- Set appropriate defaults where applicable
- Populate terraform.tfvars with your values

```
locals.tf   terraform.tfvars X
terraform.tfvars
1  vpc_cidr_block      = "10.0.0.0/16"
2  subnet_cidr_block   = "10.0.10.0/24"
3  availability_zone    = "us-east-1a"
4  env_prefix          = "prod"
5  instance_type       = "t3.micro"
6
7  public_key_location  = "/c/Users/Rughma/.ssh/id_ed25519.pub"
8  private_key_location = "/c/Users/Rughma/.ssh/id_ed25519"
```

```
variables.tf
1 variable "vpc_cidr_block" {
2   type = string
3   description = "CIDR block for the VPC"
4   default = "10.0.0.0/16"
5
6   validation {
7     condition = can(cidrhost(var.vpc_cidr_block, 0))
8     error_message = "The vpc_cidr_block must be a valid CIDR n
9   }
10 }
11
12 variable "subnet_cidr_block" {
13   type = string
14   description = "CIDR block for the Subnet"
15   default = "10.0.10.0/24"
16
17   validation {
18     condition = can(cidrhost(var.subnet_cidr_block, 0))
19     error_message = "The subnet_cidr_block must be a valid CID
20   }
21 }
22
23 variable "availability_zone" {
24   type = string
25   description = "AWS Availability Zone"
26   default = "me-central-1a"
27 }
28
variables.tf X
28
29 variable "env_prefix" {
30   type = string
31   description = "Prefix for environment resources (e.g., dev,
32   default = "prod"
33 }
34
35 variable "instance_type" {
36   type = string
37   description = "EC2 Instance Type"
38   default = "t3.micro"
39 }
40
41 variable "public_key" {
42   type = string
43   description = "Path to the public SSH key"
44 }
45
46 variable "private_key" {
47   type = string
48   description = "Path to the private SSH key"
49 }
50
51 # The backend_servers variable is technically defined in local
52 # but if you wish to pass it as a variable, define it here.
53 # For this assignment, 1.5 strictly asks to define the list in
```

### 1.3 Networking Module (5 marks)

#### Tasks:

- Create VPC resource
- Create subnet with map\_public\_ip\_on\_launch = true
- Create and attach Internet Gateway
- Configure routing table
- Add proper tags to all resources using env\_prefix

```
variables.tf main.tf X outputs.tf
modules > networking > main.tf
1 resource "aws_vpc" "myapp_vpc" {
2   cidr_block = var.vpc_cidr_block
3   enable_dns_hostnames = true
4   enable_dns_support = true
5
6   tags = {
7     Name = "${var.env_prefix}-vpc"
8   }
9 }
10
11 resource "aws_subnet" "myapp_subnet" {
12   vpc_id = aws_vpc.myapp_vpc.id
13   cidr_block = var.subnet_cidr_block
14   availability_zone = var.availability_zone
15   map_public_ip_on_launch = true
16
17   tags = {
18     Name = "${var.env_prefix}-subnet-1"
19   }
20 }
21
22 resource "aws_internet_gateway" "myapp_igw" {
23   vpc_id = aws_vpc.myapp_vpc.id
24
25   tags = {
26     Name = "${var.env_prefix}-igw"
27   }
28 }
29
main.tf X
modules > networking > main.tf
29
30 resource "aws_route_table" "myapp_route_table" {
31   vpc_id = aws_vpc.myapp_vpc.id
32
33   route {
34     cidr_block = "0.0.0.0/0"
35     gateway_id = aws_internet_gateway.myapp_igw.id
36   }
37
38   tags = {
39     Name = "${var.env_prefix}-rtb"
40   }
41 }
42
43 resource "aws_route_table_association" "a-rtb-subnet" {
44   subnet_id = aws_subnet.myapp_subnet.id
45   route_table_id = aws_route_table.myapp_route_table.id
46 }
47
```



```
variables.tf × outputs.tf ×
modules > networking > outputs.tf
1  output "vpc_id" {
2    value = aws_vpc.myapp_vpc.id
3  }
4  output "subnet_id" {
5    value = aws_subnet.myapp_subnet.id
6  }
7  output "igw_id" {
8    value = aws_internet_gateway.myapp_igw.id
9  }
10 output "route_table_id" {
11   value = aws_route_table.myapp_route_table.id
12 }
```

## 1.4 Security Module (5 marks)

### Tasks:

- Create Nginx security group with appropriate rules
- Create backend security group with appropriate rules
- Use security group IDs for backend ingress (not CIDR blocks)
- Add descriptive names and tags

```
variables.tf main.tf × outputs.tf
modules > security > main.tf
1  # Nginx Security Group
2  resource "aws_security_group" "nginx_sg" {
3    name        = "nginx-sg"
4    description = "Allow SSH, HTTP, HTTPS"
5    vpc_id      = var.vpc_id
6
7    ingress {
8      description = "SSH from My IP"
9      from_port   = 22
10     to_port    = 22
11     protocol   = "tcp"
12     cidr_blocks = [var.my_ip]
13   }
14
15   ingress {
16     description = "HTTP from Anywhere"
17     from_port   = 80
18     to_port    = 80
19     protocol   = "tcp"
20     cidr_blocks = ["0.0.0.0/0"]
21   }
22
23   ingress {
24     description = "HTTPS from Anywhere"
25     from_port   = 443
26     to_port    = 443
27     protocol   = "tcp"
28     cidr_blocks = ["0.0.0.0/0"]
29   }
30 }
```

## 1.5 Locals Configuration (5 marks)

### Tasks:

- Implement dynamic IP detection
- Define common tags
- Define backend server list
- Add any other reusable local values

```
locals.tf
locals.tf
1 data "http" "my_ip" {
2   url = "https://icanhazip.com"
3 }
4
5 locals {
6   # Dynamically fetch current public IP and append CIDR mask
7   my_ip = "${chomp(data.http.my_ip.response_body)}/32"
8
9   common_tags = {
10    Environment = var.env_prefix
11    Project     = "Assignment-2"
12    ManagedBy   = "Terraform"
13  }
14
15  backend_servers = [
16    {
17      name       = "web-1"
18      suffix     = "1"
19      script_path = "./scripts/apache-setup.sh"
20    },
21    {
22      name       = "web-2"
23      suffix     = "2"
24      script_path = "./scripts/apache-setup.sh"
25    },
26    {
27      name       = "web-3"
28      suffix     = "3"
29      script_path = "./scripts/apache-setup.sh"
30    }
31  ]
32 }
```

VPC > Security Groups

Security Groups (1/5) Info

Find security groups by attribute or tag

Name	Security group ID	Security group name	VPC ID	Description
<input checked="" type="checkbox"/> prod-nginx-sg	sg-0e020fda3b655cd58	nginx-sg	vpc-011d0a54436bf2c22	Allow HTTP/HTTPS
<input type="checkbox"/> dev-default-sg	sg-0d441f326f220fd99	default	vpc-0884d2d72d498d73a	default VPC securit
<input type="checkbox"/> -	sg-09c99fd5a8f03e526	default	vpc-0c6ef73bc2d8b77a3	default VPC securit
<input type="checkbox"/> -	sg-0b0de8846f6137e4f	default	vpc-011d0a54436bf2c22	default VPC securit

sg-0e020fda3b655cd58 - nginx-sg

Inbound rules (3)

Search

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-016691b7bfc97941b	IPv4	HTTP	TCP	80	0.0.0.0/0
sgr-0ee0f0834521fc5733	IPv4	HTTPS	TCP	443	0.0.0.0/0
sgr-073dfa48e72e6e0ed	IPv4	SSH	TCP	22	154.192.0.128/32

## Part 2: Webserver Module (15 marks)

### 2.1 Module Design (10 marks)

#### Tasks:

- Create variables.tf with all required variables
- Create main.tf with key pair and instance resources
- Create outputs.tf with all required outputs
- Ensure module is reusable for both Nginx and backend servers

```
variables.tf X
modules > webserver > variables.tf
1 variable "env_prefix" {
2   type    = string
3   description = "Environment prefix (e.g., prod, dev)"
4 }
5
6 variable "instance_name" {
7   type    = string
8   description = "Name tag for the instance"
9 }
10
11 variable "instance_type" {
12   type    = string
13   description = "EC2 instance type"
14 }
15
16 variable "availability_zone" {
17   type    = string
18   description = "AWS Availability Zone"
19 }
20
21 variable "vpc_id" {
22   type    = string
23   description = "VPC ID"
24 }
25
26 variable "subnet_id" {
27   type    = string
28   description = "Subnet ID"
29 }
30

variables.tf X
modules > webserver > variables.tf
30
31 variable "security_group_id" {
32   type    = string
33   description = "Security Group ID to attach"
34 }
35
36 variable "public_key" {
37   type    = string
38   description = "Path to the public key file"
39 }
40
41 variable "script_path" {
42   type    = string
43   description = "Path to the bash script for user_data"
44 }
45
46 variable "instance_suffix" {
47   type    = string
48   description = "Unique suffix for resources like Key Pairs"
49 }
50
51 variable "common_tags" {
52   type    = map(string)
53   description = "Common tags to apply to resources"
54 }
```

```
main.tf X
modules > webserver > main.tf
1 # 1. Get latest Amazon Linux 2023 AMI
2 data "aws_ami" "amazon_linux_2023" {
3   most_recent = true
4   owners      = ["amazon"]
5
6   filter {
7     name   = "name"
8     values = ["al2023-ami-2023.*-x86_64"]
9   }
10
11   filter {
12     name   = "virtualization-type"
13     values = ["hvm"]
14   }
15 }
16
17 # 2. Create Unique Key Pair per Instance
18 resource "aws_key_pair" "ssh_key" {
19   key_name   = "server-key-${var.env_prefix}-${var.instance_name}"
20   public_key = file(var.public_key)
21 }
22

main.tf X
modules > webserver > main.tf
22
23 # 3. Create EC2 Instance
24 resource "aws_instance" "myapp_server" {
25   ami              = data.aws_ami.amazon_linux_2023
26   instance_type    = var.instance_type
27   subnet_id        = var.subnet_id
28   vpc_security_group_ids = [var.security_group_id]
29   availability_zone = var.availability_zone
30   associate_public_ip_address = true
31   key_name          = aws_key_pair.ssh_key.key_name
32
33   user_data = file(var.script_path)
34
35   tags = merge(
36     var.common_tags,
37     {
38       Name = "${var.env_prefix}-${var.instance_name}"
39     }
40   )
41 }
```

```
main.tf  outputs.tf X
modules > webserver > outputs.tf
1  output "instance_id" {
2    value = aws_instance.myapp_server.id
3  }
4
5  output "public_ip" {
6    value = aws_instance.myapp_server.public_ip
7  }
8
9  output "private_ip" {
10   value = aws_instance.myapp_server.private_ip
11 }
```

## 2.2 Module Usage (5 marks)

### Tasks:

- Create Nginx server module instance
- Create backend server module instances using for\_each
- Ensure proper security group assignment
- Pass all required variables

```
main.tf  X
main.tf
1  provider "aws" {
2    region = "me-central-1"
3  }
4
5  # --- Part 1: Networking & Security ---
6
7  module "networking" {
8    source      = "../modules/networking"
9    vpc_cidr_block = var.vpc_cidr_block
10   subnet_cidr_block = var.subnet_cidr_block
11   availability_zone = var.availability_zone
12   env_prefix      = var.env_prefix
13 }
14
15 module "security" {
16   source      = "../modules/security"
17   vpc_id      = module.networking.vpc_id
18   env_prefix  = var.env_prefix
19   my_ip      = local.my_ip
20 }
21
22 # --- Part 2: Webservers ---
23
24 # 1. Nginx Server (Single Instance)
25 module "nginx_server" {
26   source      = "../modules/webserver"
27   env_prefix  = var.env_prefix
28   instance_name = "nginx-proxy"
29   instance_type = var.instance_type
30   availability_zone = var.availability_zone
31
32   # ... other required variables ...
33 }
34
35 # 2. Backend Servers (Dynamic Loop using for_each)
36 module "backend_servers" {
37   for_each = { for idx, server in local.backend_servers : server.name, idx }
38
39   # ... variables for each backend server ...
40 }
41
42 # ... additional code ...
```

## Part 3: Server Configuration Scripts (20 marks)

### 3.1 Apache Backend Server Script (10 marks)

```
$ apache-setup.sh X
scripts > $ apache-setup.sh
1  #!/bin/bash
2  set -e
3
4  # Update system
5  yum update -y
6
7  # Install Apache
8  yum install httpd -y
9
10 # Start and enable Apache
11 systemctl start httpd
12 systemctl enable httpd
13
14 # Get metadata token (IMDSv2)
15 TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/tok
16 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
17
18 # Get instance metadata
19 PRIVATE_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
20 http://169.254.169.254/latest/meta-data/local-ipv4)
21 PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
22 http://169.254.169.254/latest/meta-data/public-ipv4)
23 PUBLIC_DNS=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
24 http://169.254.169.254/latest/meta-data/public-hostname)
25 INSTANCE_ID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
26 http://169.254.169.254/latest/meta-data/instance-id)
27
28 # Set hostname
29 hostnamectl set-hostname myapp-webserver
30
31 # Create custom HTML page
32 cat > /var/www/html/index.html <<EOF
33 <!DOCTYPE html>
34 <html>
35 <head>
36 <title>Backend Web Server - Assignment 2</title>
37 <style>
38
39 .container {
40     background: rgba(255, 255, 255, 0.1);
41     padding: 30px;
42     border-radius: 10px;
43     box-shadow: 0 8px 32px 0 rgba(31, 38, 135, 0.37);
44 }
45
46 h1 { color: #fff; text-shadow: 2px 2px 4px rgba(0,0,0,
47
48 .info { margin: 15px 0; padding: 10px; background: rgb
49
50 .label { font-weight: bold; color: #ffd700; }
51
52 </style>
53 </head>
54 <body>
55
56 <div class="container">
57
58 <h1> Backend Web Server - Assignment 2</h1>
59
60 <div class="info"><span class="label">Hostname:</span>
61
62 <div class="info"><span class="label">Instance ID:</span>
63
64 <div class="info"><span class="label">Private IP:</span>
65
66 <div class="info"><span class="label">Public IP:</span>
67
68 <div class="info"><span class="label">Public DNS:</span>
69
70 <div class="info"><span class="label">Deployed: </span>
71
72 <div class="info"><span class="label">Status:</span>
73
74 <div class="info"><span class="label">Managed By:</span>
75
76 </div>
77
78 </body>
79 </html>
80 EOF
81
82 # Set permissions
83 chmod 644 /var/www/html/index.html
84
85 echo "Apache setup completed successfully!"
```

### 3.2 Nginx Server Setup Script (10 marks)

#### Tasks:

- Create script with SSL certificate generation
- Configure upstream with placeholder IPs
- Implement caching
- Add security headers
- Configure HTTP to HTTPS redirect
- Add health check endpoint

```
$ nginx-setup.sh X
scripts > $ nginx-setup.sh
1  #!/bin/bash
2  set -e
3
4  # Update and install Nginx
5  yum update -y
6  yum install -y nginx openssl
7  systemctl start nginx
8  systemctl enable nginx
9
10 # Create SSL directories
11 mkdir -p /etc/ssl/private
12 mkdir -p /etc/ssl/certs
13
14 # Get metadata token
15 TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/tok
16 -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
17
18 # Get public IP
19 PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
20 http://169.254.169.254/latest/meta-data/public-ipv4)
21
22 # Generate self-signed certificate
23 openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
24 -keyout /etc/ssl/private/selfsigned.key \
25 -out /etc/ssl/certs/selfsigned.crt \
26 -subj "/CN=$PUBLIC_IP" \
27 -addext "subjectAltName=IP:$PUBLIC_IP" \
28 -addext "basicConstraints=CA:FALSE" \
29 -addext "keyUsage=digitalSignature,keyEncipherment" \
30 -addext "extendedKeyUsage=serverAuth"
31
32 echo "Self-signed certificate created for IP: $PUBLIC_IP"
33

$ nginx-setup.sh X
scripts > $ nginx-setup.sh
49
92 r {
138
139
140
141 P Server (redirect to HTTPS)
142 r {
143 listen 80;
144 server_name _;
145
146 location / {
147     return 301 https://$host$request_uri;
148
149
150 Allow health checks over HTTP
151 location /health {
152     access_log off;
153     return 200 "Nginx is healthy\n";
154     add_header Content-Type text/plain;
155
156
157
158
159
160 cache directory
161 /var/cache/nginx
162 nginx:nginx /var/cache/nginx
163
164 d restart Nginx
165 && systemctl restart nginx
166
167 nx setup completed successfully!"
168 ember to update backend server IPs in /etc/nginx/nginx.conf"
```

## Part 4: Infrastructure Deployment (15 marks)

### 4.1 Initial Deployment (5 marks)

Deploy the infrastructure using Terraform.

#### Tasks:

- Generate SSH key pair if not exists
- Initialize Terraform (terraform init)
- Validate configuration (terraform validate)
- Plan deployment (terraform plan)
- Apply configuration (terraform apply -auto-approve)

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
$ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
/c/Users/Rughma/.ssh/id_ed25519 already exists.
Overwrite (y/n)? n
```

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
• $ terraform 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.

Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
○ $
```

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
• $ terraform validate
Success! The configuration is valid.
```

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
$ terraform plan
```

```
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ primary_network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
```

Plan: 6 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
○ $
```

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
$ terraform apply -auto-approve
module.networking.aws_subnet.myapp_subnet: Still creating... [00m10s elapsed]
module.networking.aws_subnet.myapp_subnet: Creation complete after 12s [id=subnet-001cfc4f830f4d588]
module.networking.aws_route_table_association.a-rtb-subnet: Creating...
module.backend_servers["web-2"].aws_instance.myapp_server: Creating...
module.backend_servers["web-1"].aws_instance.myapp_server: Creating...
module.nginx_server.aws_instance.myapp_server: Creating...
module.backend_servers["web-3"].aws_instance.myapp_server: Creating...
module.networking.aws_route_table_association.a-rtb-subnet: Creation complete after 0s [id=rtbassoc-0a1ac9583a4d3fcb]
module.backend_servers["web-2"].aws_instance.myapp_server: Still creating... [00m10s elapsed]
module.backend_servers["web-1"].aws_instance.myapp_server: Still creating... [00m10s elapsed]
module.nginx_server.aws_instance.myapp_server: Still creating... [00m10s elapsed]
module.backend_servers["web-3"].aws_instance.myapp_server: Still creating... [00m10s elapsed]
module.backend_servers["web-1"].aws_instance.myapp_server: Creation complete after 13s [id=i-0bf37fdce3ea62d4d]
module.backend_servers["web-2"].aws_instance.myapp_server: Creation complete after 13s [id=i-0c87ebcfb4d5976fa]
module.backend_servers["web-3"].aws_instance.myapp_server: Creation complete after 13s [id=i-0ddead9a50b5e91b3]
module.nginx_server.aws_instance.myapp_server: Creation complete after 13s [id=i-0eff4e877ccb53f4b]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.

Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
○ $
```

## 4.2 Output Configuration (5 marks)

### Tasks:

- Create all required outputs
- Add helpful descriptions
- Include configuration guide
- Display outputs after apply

```
Rughma@DESKTOP-LJFKHJN MINGW64 /d/docs/Assignment2
$ terraform output

=====
DEPLOYMENT SUCCESSFUL!
=====

Next Steps:
1. SSH into Nginx server:
  ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.77.128

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

3. Update backend IPs in 'upstream backend_servers' block:
  - BACKEND_IP_1: 10.0.10.185
  - BACKEND_IP_2: 10.0.10.194
  - BACKEND_IP_3: 10.0.10.93

4. Restart Nginx:
  sudo systemctl restart nginx

5. Test:
  https://158.252.77.128

Backend Servers List:
- web-1: Public: 3.28.44.253 | Private: 10.0.10.185
- web-2: Public: 40.172.177.31 | Private: 10.0.10.194
- web-3: Public: 40.172.215.101 | Private: 10.0.10.93

=====

EOT
nginx_instance_id = "i-0eff4e877ccb53f4b"
nginx_public_ip = "158.252.77.128"
subnet_id = "subnet-001cfc4f830f4d588"
vpc_id = "vpc-011d0a54436bf2c22"

Rughma@DESKTOP-LJFKHJN MINGW64 /d/docs/Assignment2
```

```
terraform.tfvars  outputs.tf  {} outputs.json x  ▶ □ ...

{} outputs.json > ...
1  {
2    "backend_servers_info": {
3      "sensitive": false,
4      "type": [
5        "object",
6        {
7          "web-1": [
8            "object",
9            {
10             "instance_id": "string",
11             "private_ip": "string",
12             "public_ip": "string"
13           }
14         ],
15         "web-2": [
16           "object",
17           {
18             "instance_id": "string",
19             "private_ip": "string",
20             "public_ip": "string"
21           }
22         ]
23       }
24     }
25   }
26 }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Rughma@DESKTOP-LJFKHJN MINGW64 /d/docs/Assignment2
Rughma@DESKTOP-LJFKHJN MINGW64 /d/docs/Assignment2
$ terraform output -json > outputs.json
```



### 4.3 AWS Console Verification (5 marks)

#### Tasks:

- Verify VPC created
- Verify Subnet created
- Verify Internet Gateway attached
- Verify Route Table configured
- Verify Security Groups created with correct rules
- Verify all 4 EC2 instances running
- Verify Key Pairs created

#### Your VPCs

[VPCs](#) | [VPC encryption controls](#)

##### Your VPCs (1) [Info](#)

Last updated 40 minutes ago [Actions](#) [Create VPC](#)

VPC ID : vpc-011d0a54436bf2c22 [X](#) [Clear filters](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	VPC ID	State	Encryption c...	Encryption control ...
<input type="checkbox"/>	prod-vpc	<a href="#">vpc-011d0a54436bf2c22</a>	Available	-	-

#### Subnets (1) [Info](#)

Last updated less than a minute ago [Actions](#) [Create subnet](#)

VPC : vpc-011d0a54436bf2c22 [X](#) [Clear filters](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	Subnet ID	State	VPC	Blor
<input type="checkbox"/>	prod-subnet-1	<a href="#">subnet-001cfc4f830f4d588</a>	Available	<a href="#">vpc-011d0a54436bf2c22</a>   <a href="#">prod...</a>	<a href="#">Settings</a>

#### Security Groups (3) [Info](#)

[Actions](#) [Export security groups to CSV](#) [Create security group](#)

VPC ID = vpc-011d0a54436bf2c22 [X](#) [Clear filters](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID
<input type="checkbox"/>	prod-backend-sg	<a href="#">sg-00fe7ba84010e0a94</a>	backend-sg	<a href="#">vpc-011d0a54436bf2c22</a>
<input type="checkbox"/>	prod-nginx-sg	<a href="#">sg-0e020fda3b655cd58</a>	nginx-sg	<a href="#">vpc-011d0a54436bf2c22</a>
<input type="checkbox"/>	-	<a href="#">sg-0b0de8846f6137e4f</a>	default	<a href="#">vpc-011d0a54436bf2c22</a>

Instances (5) Info

Refresh

Connect

Instance state

Actions

Launch instances

All states

< 1 >

Settings

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	dev-ec2-insta...	i-02ced525da2d44ae6	Running	t3.micro	3/3 checks passed	View alarms	me-c
<input type="checkbox"/>	prod-web-2	i-0c87ebcfb4d5976fa	Running	t3.micro	3/3 checks passed	View alarms	me-c
<input type="checkbox"/>	prod-web-3	i-0ddead9a50b5e91b3	Running	t3.micro	3/3 checks passed	View alarms	me-c
<input type="checkbox"/>	prod-nginx-pr...	i-0eff4e877ccb53f4b	Running	t3.micro	3/3 checks passed	View alarms	me-c
<input type="checkbox"/>	prod-web-1	i-0bf37fde3ea62d4d	Running	t3.micro	3/3 checks passed	View alarms	me-c

## Part 5: Nginx Configuration & Testing (25 marks)

### 5.1 Update Nginx Backend Configuration (5 marks)

### Tasks:

- SSH into Nginx server
- Edit `/etc/nginx/nginx.conf`
- Replace placeholder IPs with actual private IPs of backend servers
- Test Nginx configuration
- Restart Nginx service

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
$ ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.77.128
The authenticity of host '158.252.77.128 (158.252.77.128)' can't be established.
ED25519 key fingerprint is SHA256:q0r7Q4ditp+mFTKqnfwBQkbQLCeXdhPoPqKSbcBCQA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.252.77.128' (ED25519) to the list of known hosts.
```

The terminal shows an ASCII art tree-like logo representing the AWS cloud infrastructure, with branches labeled with various services like S3, EC2, IAM, etc., leading down to the prompt [ec2-user@ip-10-0-10-89 ~]\$ .

```
[ec2-user@ip-10-0-10-89 ~]$
```

```
# Upstream backend servers
# PLACEHOLDER: Update these IPs after deployment
upstream backend_servers {
    # Primary servers (active load balancing)
    server 10.0.10.185:80;
    server 10.0.10.194:80;

    # Backup server (only used when primary servers are down)
    server 10.0.10.93:80 backup;
}

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

```
-- INSERT --
```

```
[ec2-user@ip-10-0-10-89 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-10-0-10-89 ~]$ sudo nginx -t
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:54
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-89 ~]$

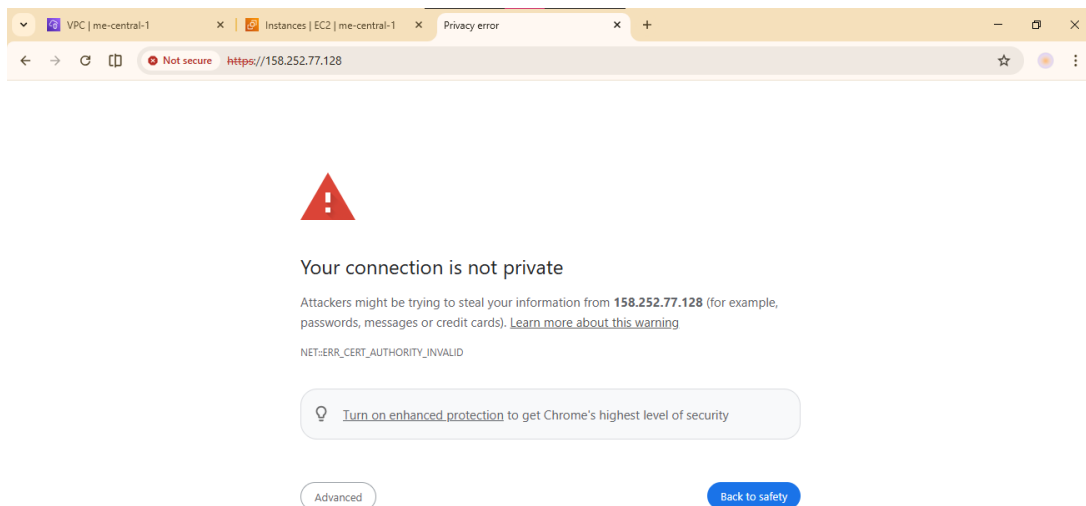
[ec2-user@ip-10-0-10-89 ~]$ sudo systemctl restart nginx
[ec2-user@ip-10-0-10-89 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Tue 2025-12-30 14:14:11 UTC; 8s ago
     Process: 27337 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 27338 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 27339 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 27340 (nginx)
      Tasks: 5 (limit: 1067)
     Memory: 4.5M
        CPU: 59ms
```

## 5.2 Test Load Balancing (5 marks)

Test that Nginx is properly load balancing between web-1 and web-2.

### Tasks:

- Open browser to <https://<nginx-public-ip>>
- Accept the security warning for self-signed certificate
- Reload page multiple times (at least 10 times)
- Verify traffic alternates between web-1 and web-2
- Verify web-3 is NOT serving traffic (it's backup only)
- Document the load balancing pattern



VPC | me-central-1 x Instances | EC2 | me-central-1 x Backend Web Server x +

Not secure https://158.252.77.128

## Backend Web Server - Assignment 2

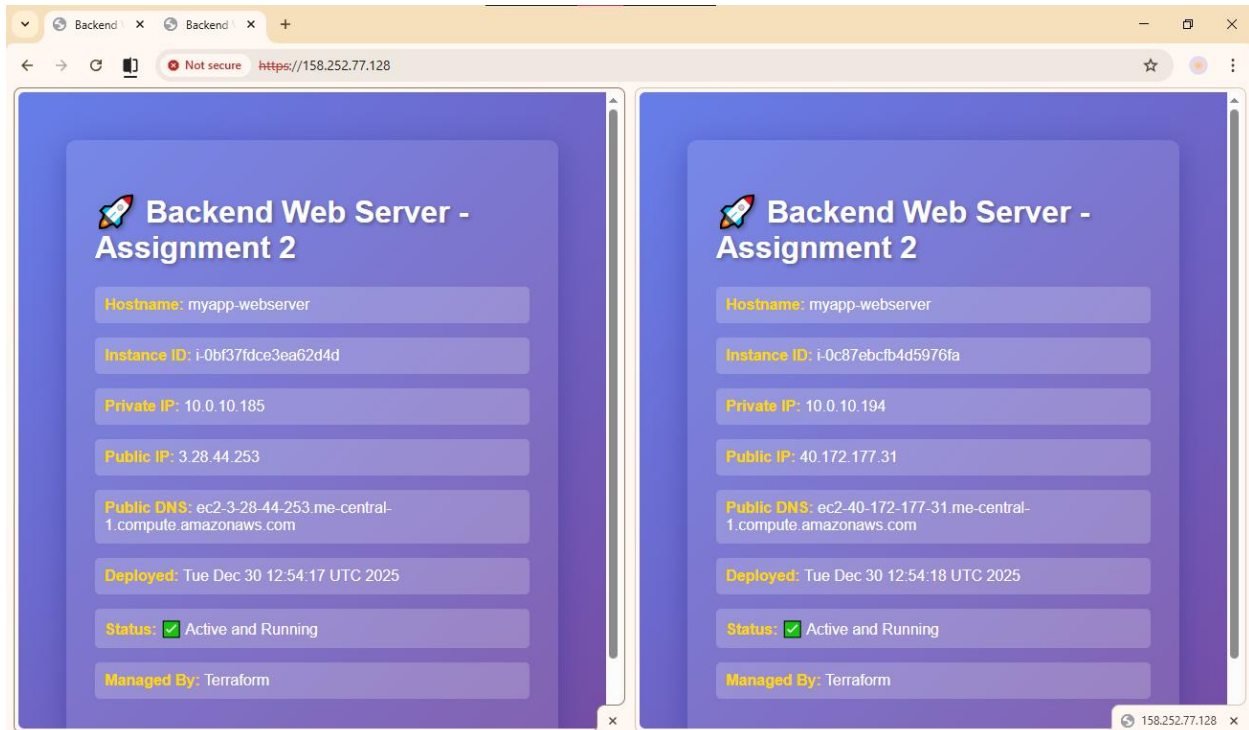
Hostname:	myapp-webserver
Instance ID:	i-0bf37fdce3ea62d4d
Private IP:	10.0.10.185
Public IP:	3.28.44.253
Public DNS:	ec2-3-28-44-253.me-central-1.compute.amazonaws.com
Deployed:	Tue Dec 30 12:54:17 UTC 2025
Status:	✓ Active and Running
Managed By:	Terraform

VPC | me-central-1 x Instances | EC2 | me-central-1 x Backend Web Server x +

Not secure https://158.252.77.128

## Backend Web Server - Assignment 2

Hostname:	myapp-webserver
Instance ID:	i-0c87ebcfb4d5976fa
Private IP:	10.0.10.194
Public IP:	40.172.177.31
Public DNS:	ec2-40-172-177-31.me-central-1.compute.amazonaws.com
Deployed:	Tue Dec 30 12:54:18 UTC 2025
Status:	✓ Active and Running
Managed By:	Terraform





### 5.3 Test Cache Functionality (5 marks)

#### Tasks:

- Open browser developer tools (F12)
- Navigate to Network tab
- Clear browser cache
- Load `https://<nginx-public-ip>`
- Check response headers for X-Cache-Status: MISS (first request)
- Reload page
- Check response headers for X-Cache-Status: HIT (cached request)
- Verify cache directory on Nginx server

Name	×	Headers	Preview	Response	Initiator	Timing
158.252.77.128	▼ General					
✖ favicon.ico	Request URL		https://158.252.77.128/			
	Request Method		GET			
	Status Code		200 OK			
	Remote Address		158.252.77.128:443			
	Referrer Policy		strict-origin-when-cross-origin			
	▼ Response Headers					
	Content-Encoding		gzip			
	Content-Type		text/html; charset=UTF-8			
	Date		Tue, 30 Dec 2025 14:51:43 GMT			
	Etag		W/"62e-6472adb8d59e1"			
	Last-Modified		Tue, 30 Dec 2025 12:54:17 GMT			
	Server		nginx/1.28.0			
	Vary		Accept-Encoding			
	X-Cache-Status		BYPASS			

Name	X	Headers	Preview	Response	Initiator	Timing
 158.252.77.128	▼ General					
		Request URL		https://158.252.77.128/		
		Request Method		GET		
		Status Code		 200 OK		
		Remote Address		158.252.77.128:443		
		Referrer Policy		strict-origin-when-cross-origin		
	▼ Response Headers					
		Content-Encoding		gzip		
		Content-Type		text/html; charset=UTF-8		
		Date		Tue, 30 Dec 2025 14:30:17 GMT		
		Etag		W/"62e-6472adb8d59e1"		
		Last-Modified		Tue, 30 Dec 2025 12:54:17 GMT		
		Server		nginx/1.28.0		
		Vary		Accept-Encoding		
		X-Cache-Status		HIT		

```
[ec2-user@ip-10-0-10-89 ~]$ ls -la /var/cache/nginx/
total 0
drwxr-xr-x. 6 nginx nginx 42 Dec 30 15:03 .
drwxr-xr-x. 9 root root 101 Dec 30 12:54 ..
drwx-----. 3 nginx nginx 16 Dec 30 15:03 3
drwx-----. 3 nginx nginx 16 Dec 30 14:14 7
drwx-----. 3 nginx nginx 16 Dec 30 14:39 8
drwx-----. 3 nginx nginx 16 Dec 30 14:17 a
[ec2-user@ip-10-0-10-89 ~]$
```

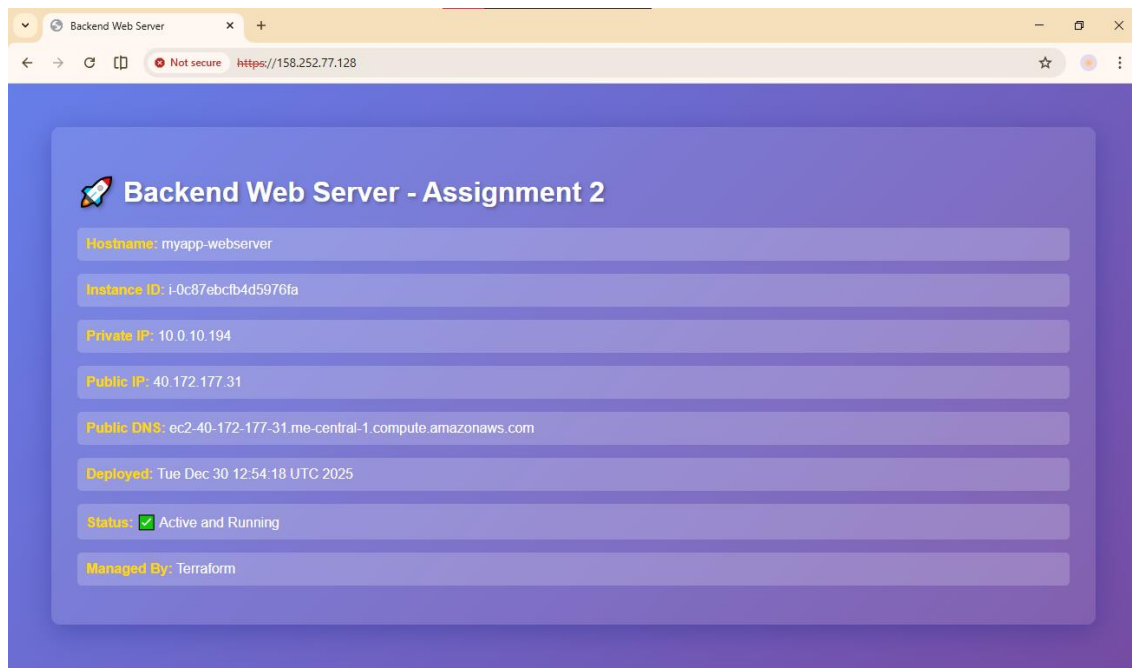
```
[ec2-user@ip-10-0-10-89 ~]$ sudo grep "Cache:" /var/log/nginx/access.log | tail -5
154.192.0.128 - - [30/Dec/2025:14:51:43 +0000] "GET /favicon.ico HTTP/2.0" 404 172 "https://158.252.77.128/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache: EXPIRED
154.192.0.128 - - [30/Dec/2025:14:53:40 +0000] "GET / HTTP/2.0" 200 691 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache: BYPASS
154.192.0.128 - - [30/Dec/2025:14:54:05 +0000] "GET / HTTP/2.0" 200 693 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache: BYPASS
154.192.0.128 - - [30/Dec/2025:14:54:06 +0000] "GET / HTTP/2.0" 200 691 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache: BYPASS
48.217.87.78 - - [30/Dec/2025:15:03:43 +0000] "GET /ecp/Current/exporttool/microsoft.exchange.ediscovery.exporttool.application HTTP/1.1" 404 183 "-" "Mozilla/5.0 zgrab/0.x" "-" Cache: MISS
[ec2-user@ip-10-0-10-89 ~]$
```

## 5.4 Test High Availability (Backup Server) (5 marks)

Test the backup server functionality by simulating primary server failure.

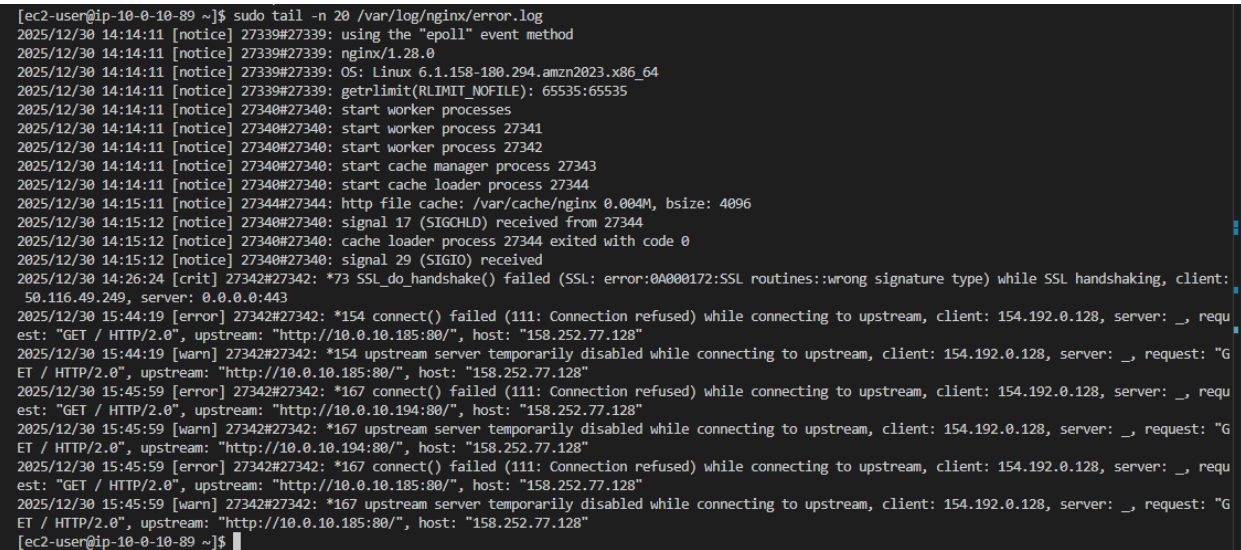
### Tasks:

- SSH into web-1 and stop Apache
- Reload Nginx page - should show web-2 only
- SSH into web-2 and stop Apache
- Reload Nginx page - should now show web-3 (backup activated)
- Restart web-1 and web-2
- Verify traffic returns to web-1 and web-2

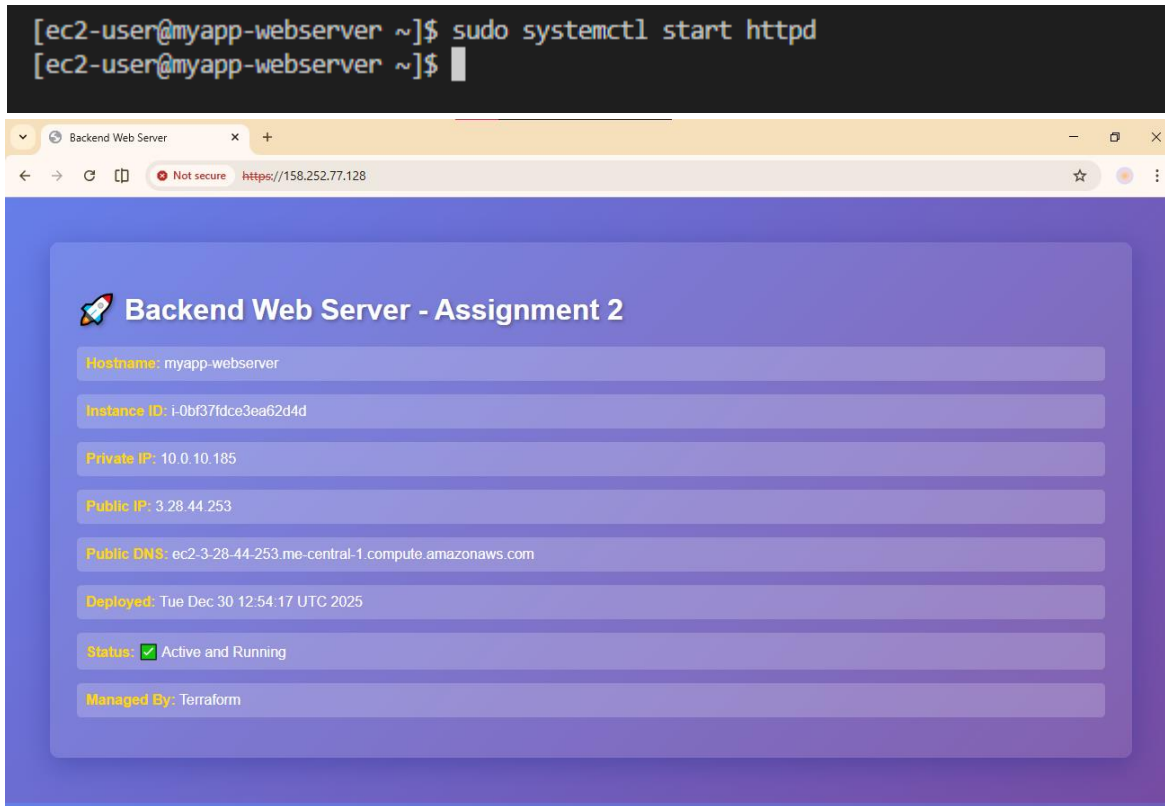


```
#  
~$ sudo yum install -y amazon-linux-extras-2023  
  
#  
~$ curl https://aws.amazon.com/linux/amazon-linux-2023
```

[ec2-user@myapp-webserver ~]\$ sudo systemctl stop httpd  
[ec2-user@myapp-webserver ~]\$







## 5.5 Security & Performance Analysis (5 marks)

Analyze the security headers and performance of your Nginx setup.

### Tasks:

- Check SSL/TLS certificate details
- Verify security headers in response
- Test HTTP to HTTPS redirect
- Check response times
- Analyze Nginx logs

```
[ec2-user@ip-10-0-10-89 ~]$ curl -I http://158.252.77.128
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Tue, 30 Dec 2025 16:04:38 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://158.252.77.128/

[ec2-user@ip-10-0-10-89 ~]$
```

```
[ec2-user@ip-10-0-10-89 ~]$
```

```

Chrome/143.0.0.0 Safari/537.36 "-" Cache: BYPASS
154.192.0.128 - - [30/Dec/2025:15:50:45 +0000] "GET / HTTP/2.0" 200 693 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36 "-" Cache: BYPASS
154.192.0.128 - - [30/Dec/2025:15:50:46 +0000] "GET / HTTP/2.0" 200 693 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36 "-" Cache: BYPASS
154.192.0.128 - - [30/Dec/2025:15:50:47 +0000] "GET / HTTP/2.0" 200 691 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36 "-" Cache: BYPASS
176.32.195.85 - - [30/Dec/2025:15:53:43 +0000] "\x16\x03\x02\x010\x01\x00\x01k\x03\x02fH\xC5\x1A#x7F:\x0F\xE2\xB4\x82\xFF\x09T\x9F\xA7\xC4y\xB0h\xC6\x13\x
8C\x44\x1C=\x22\xE1\x1A\x98 \x84\xB4, \x85\xAFn\xE3Y\xBBbhh\xFF=':\xA9\x82\x090\xC8\xA2\xD7\x93\x98\xB4\xEF\x80\xE5\xB9\x90\x00(\xC0" 400 157 "-" "-" "-" Cache: -
95.215.0.144 - - [30/Dec/2025:15:54:36 +0000] "GET / HTTP/1.1" 301 169 "-" "fasthttp" "-" Cache: -
46.161.50.108 - - [30/Dec/2025:15:54:38 +0000] "\x16\x03\x01\x05\xA8\x01\x00\x05\xA4\x03\x035= \x0FY\x89_\xD5\xC3\x12\xC5N\xF3\xB4(\xE4\x8EF\xA3t~BM\x99]" 4
00 157 "-" "-" "-" Cache: -
46.161.50.108 - - [30/Dec/2025:15:54:38 +0000] "GET /aaa HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
cko) Chrome/70.0.3538.77 Safari/537.36" "-" Cache: -
46.161.50.108 - - [30/Dec/2025:15:54:38 +0000] "\x16\x03\x01\x05\xA8\x01\x00\x05\xA4\x03\x03\x7F\x04V\xD2g\xD82\x80\xD0" 400 157 "-" "-" "-" Cache: -
46.161.50.108 - - [30/Dec/2025:15:54:38 +0000] "GET /aab HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Ge
cko) Chrome/70.0.3538.77 Safari/537.36" "-" Cache: -
45.91.64.8 - - [30/Dec/2025:15:55:00 +0000] "\x16\x03\x01\x05\xA8\x01\x00\x05\xA4\x03\x03\x1E\x03\xA1\xE2\x0A\xFf\x8C, \x98\x0F\x01\xB6\xD5\x89Pe\xE3\xC5? \xB1F\xF1
\x22\xB0z\xB4\xAC\xEC, \xA4 \x95\xC4\x94(\xAEU\x08\x8F\xCBd\xFF\x0C\x83\xCA\xF3Dy" 400 157 "-" "-" "-" Cache: -
45.91.64.8 - - [30/Dec/2025:15:55:00 +0000] "GET /aaa HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
) Chrome/70.0.3538.77 Safari/537.36" "-" Cache: -
45.91.64.8 - - [30/Dec/2025:15:55:00 +0000] "\x16\x03\x01\x05\xA8\x01\x00\x05\xA4\x03\x03\xD8\xF9\xF7\xB4\x90\x89\x99\xDf\x91s\xBD\xA6\xCF\xB4K\x12\xEE\xC2
q\xA09\xFC\x0DBa\xB9j\xFA\xB7\x89\xCF\xD6 v\xA1Zm2\xA5\xC5) \xA2J$ \xC2\xD0T\xF3e\x9E\x08\xE7\x1D\x41\x8F\x1F\xBA\xED\xB4\xFE\xE9\xB7\x00\x1A\xC0+ \xC0/\xC0, \xC00
\xCC\xA9\xCC\xA8\xC0\x89\xC0\x13\xC0" 400 157 "-" "-" "-" Cache: -
45.91.64.8 - - [30/Dec/2025:15:55:00 +0000] "GET /aab HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
) Chrome/70.0.3538.77 Safari/537.36" "-" Cache: -
93.174.93.12 - - [30/Dec/2025:15:58:16 +0000] "GET / HTTP/1.0" 301 169 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.
0.3889.0 Safari/537.36" "-" Cache: -
129.212.186.47 - - [30/Dec/2025:16:01:00 +0000] "SSTP_DUPLEX_POST /sra [BA195980-CD49-458b-9E23-C84EE0ADC75]/ HTTP/1.1" 400 157 "-" "-" "-" Cache: -
158.252.77.128 - - [30/Dec/2025:16:04:38 +0000] "HEAD / HTTP/1.1" 301 0 "-" "curl/8.11.1" "-" Cache: -
158.252.77.128 - - [30/Dec/2025:16:05:34 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache: MISS
20.119.75.60 - - [30/Dec/2025:16:05:39 +0000] "GET /login HTTP/1.1" 404 183 "-" "Mozilla/5.0 zgrab/0.x" "-" Cache: MISS
[ec2-user@ip-10-0-10-89 ~]$

```

```
0.3889.0 Safari/537.36" "-" Cache: -
129.212.186.47 - - [30/Dec/2025:16:01:00 +0000] "SSTP DUPLEX_POST /sra_{BA195980-CD49-458b-9E23-C84EE0ADC75}/ HTTP/1.1" 400 157 "-" "-" "-" Cache: -
158.252.77.128 - - [30/Dec/2025:16:04:38 +0000] "HEAD / HTTP/1.1" 301 0 "-" "curl/8.11.1" "-" "-" Cache: -
158.252.77.128 - - [30/Dec/2025:16:05:34 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" "-" Cache: MISS
20.119.75.60 - - [30/Dec/2025:16:05:39 +0000] "GET /login HTTP/1.1" 404 183 "-" "Mozilla/5.0 zgrab/0.x" "-" "-" Cache: MISS
[ec2-user@ip-10-0-10-89 ~]$ sudo tail -n 20 /var/log/nginx/error.log
2025/12/30 14:14:11 [notice] 27339#27339: using the "epoll" event method
2025/12/30 14:14:11 [notice] 27339#27339: nginx/1.28.0
2025/12/30 14:14:11 [notice] 27339#27339: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/30 14:14:11 [notice] 27339#27339: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/30 14:14:11 [notice] 27340#27340: start worker processes
2025/12/30 14:14:11 [notice] 27340#27340: start worker process 27341
2025/12/30 14:14:11 [notice] 27340#27340: start worker process 27342
2025/12/30 14:14:11 [notice] 27340#27340: start cache manager process 27343
2025/12/30 14:14:11 [notice] 27340#27340: start cache loader process 27344
2025/12/30 14:15:11 [notice] 27344#27344: http file cache: /var/cache/nginx 0.004M, bsize: 4096
2025/12/30 14:15:12 [notice] 27340#27340: signal 17 (SIGCHLD) received from 27344
2025/12/30 14:15:12 [notice] 27340#27340: cache loader process 27344 exited with code 0
2025/12/30 14:15:12 [notice] 27340#27340: signal 29 (SIGIO) received
2025/12/30 14:26:24 [crit] 27342#27342: *73 SSL_do_handshake() failed (SSL: error:0A000172:SSL routines::wrong signature type) while SSL handshaking, client:
50.116.49.249, server: 0.0.0.0:443
2025/12/30 15:44:19 [error] 27342#27342: *154 connect() failed (111: Connection refused) while connecting to upstream, client: 154.192.0.128, server: _, requ
est: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:44:19 [warn] 27342#27342: *154 upstream server temporarily disabled while connecting to upstream, client: 154.192.0.128, server: _, request: "G
ET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [error] 27342#27342: *167 connect() failed (111: Connection refused) while connecting to upstream, client: 154.192.0.128, server: _, requ
est: "GET / HTTP/2.0", upstream: "http://10.0.10.194:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [warn] 27342#27342: *167 upstream server temporarily disabled while connecting to upstream, client: 154.192.0.128, server: _, request: "G
ET / HTTP/2.0", upstream: "http://10.0.10.194:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [error] 27342#27342: *167 connect() failed (111: Connection refused) while connecting to upstream, client: 154.192.0.128, server: _, requ
est: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [warn] 27342#27342: *167 upstream server temporarily disabled while connecting to upstream, client: 154.192.0.128, server: _, request: "G
ET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
[ec2-user@ip-10-0-10-89 ~]$
```

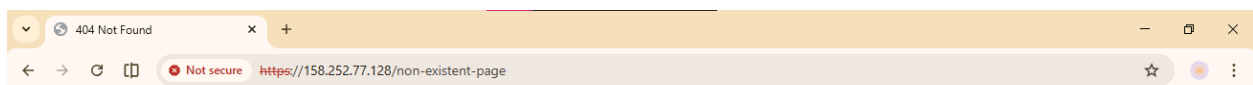
## Bonus Tasks (10 marks extra credit)

### Bonus 1: Custom Error Pages (3 marks)

Create custom error pages for Nginx (404, 502, 503).

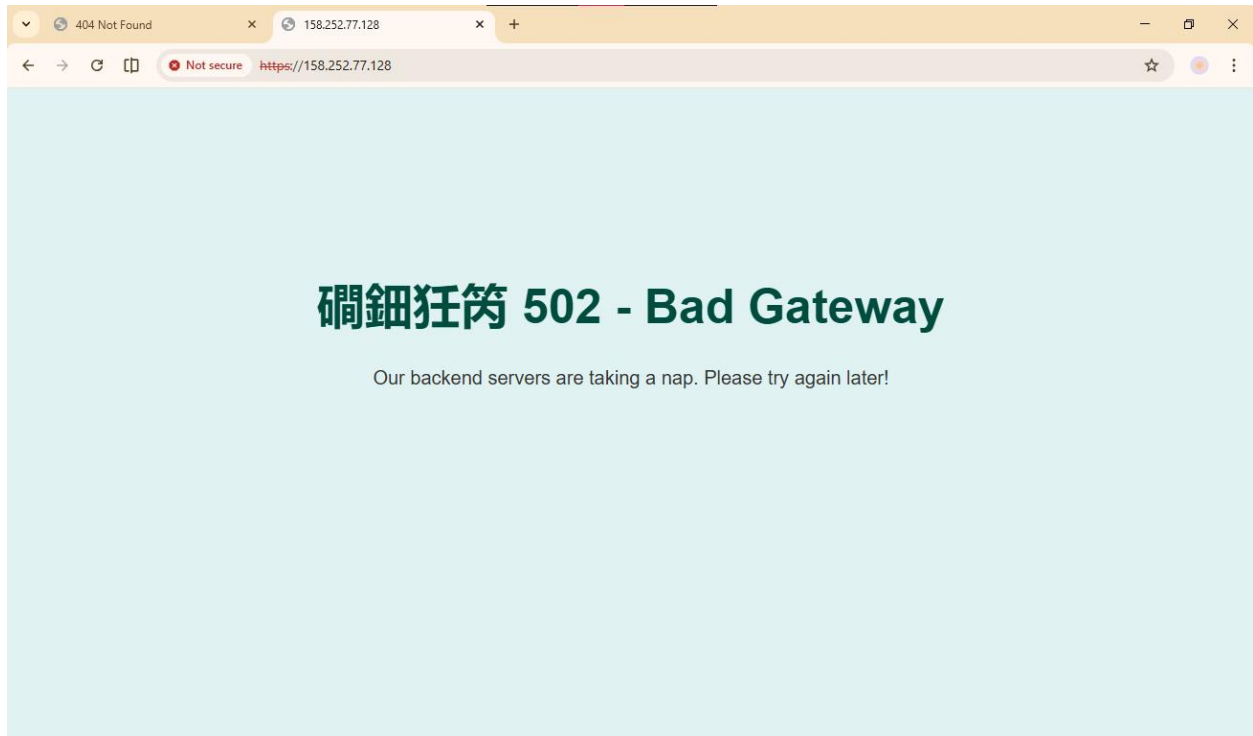
#### Tasks:

- Create custom HTML error pages
- Configure Nginx to use custom error pages
- Test by accessing non-existent URL and stopping backend servers



## Not Found

The requested URL was not found on this server.



## Bonus 2: Implement Rate Limiting (3 marks)

### Tasks:

- Implement rate limiting
- Test with rapid requests
- Show 429 (Too Many Requests) response

```
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;
    limit_req_zone $binary_remote_addr zone=mylimit:10m rate=1r/s;

    # Basic settings
    sendfile on;
    tcp_nopush on;
    keepalive_timeout 65;
    "/etc/nginx/nginx.conf" 133L, 4019B
```



### Bonus 3: Health Check Automation (4 marks)

```
[ec2-user@myapp-webserver ~]$ cat health_monitor.sh
#!/bin/bash

LOG_FILE="/home/ec2-user/health_monitor.log"

echo "Starting Health Monitor..." >> $LOG_FILE

while true; do
    # Check HTTP status code
    HTTP_STATUS=$(curl -o /dev/null -s -w "%{http_code}\n" http://localhost)
    TIMESTAMP=$(date +%Y-%m-%d %H:%M:%S)

    if [ "$HTTP_STATUS" == "200" ]; then
        echo "[${TIMESTAMP}] ✓ Status: $HTTP_STATUS - Healthy" >> $LOG_FILE
    else
        echo "[${TIMESTAMP}] ✗ Status: $HTTP_STATUS - DOWN! Restarting Apache..." >> $LOG_FILE
        sudo systemctl start httpd

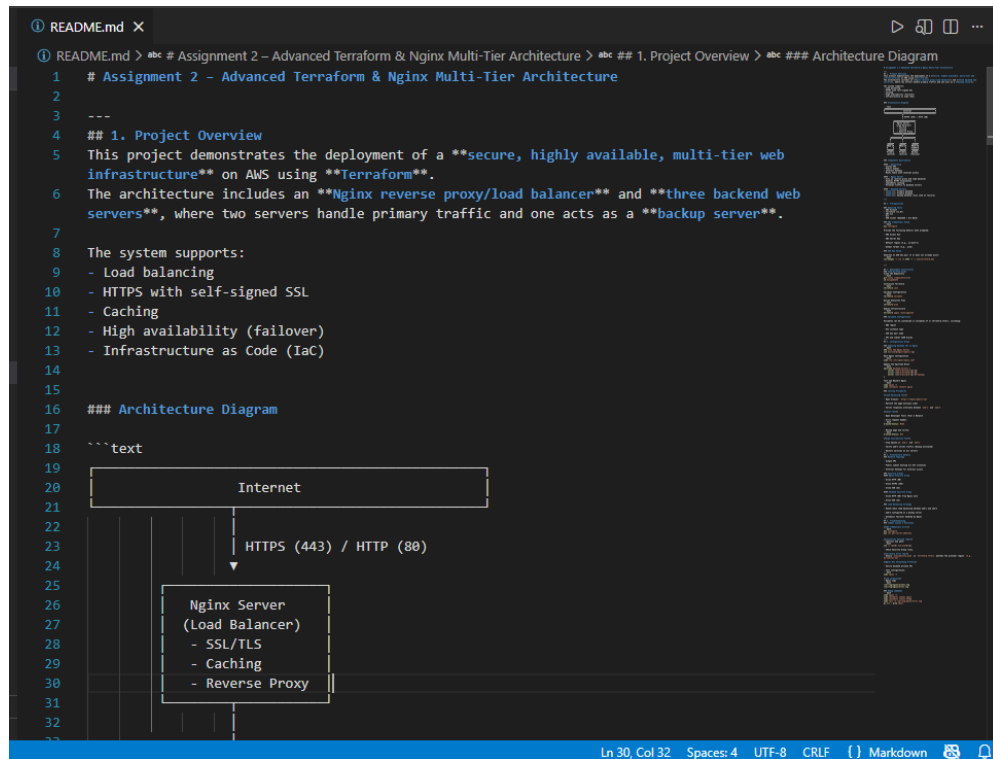
        # Verify restart
        sleep 2
        NEW_STATUS=$(curl -o /dev/null -s -w "%{http_code}\n" http://localhost)
        echo "[${TIMESTAMP}] Restart result: $NEW_STATUS" >> $LOG_FILE
    fi

    # Wait 10 seconds (using 10 instead of 30 for faster testing)
    sleep 10
done
[ec2-user@myapp-webserver ~]$ ./health_monitor.sh &
[2] 37861
[ec2-user@myapp-webserver ~]$
```

```
[ec2-user@myapp-webserver ~]$ tail -f health_monitor.log
[2025-12-30 17:38:56] ✓ Status: 200 - Healthy
[2025-12-30 17:39:02] ✓ Status: 200 - Healthy
[2025-12-30 17:39:06] ✓ Status: 200 - Healthy
[2025-12-30 17:39:12] ✓ Status: 200 - Healthy
[2025-12-30 17:39:16] ✓ Status: 200 - Healthy
[2025-12-30 17:39:22] ✓ Status: 200 - Healthy
[2025-12-30 17:39:26] ✓ Status: 200 - Healthy
[2025-12-30 17:39:32] ✗ Status: 000 - DOWN! Restarting Apache...
[2025-12-30 17:39:32] Restart result: 200
[2025-12-30 17:39:36] ✓ Status: 200 - Healthy
[2025-12-30 17:39:44] ✓ Status: 200 - Healthy
[2025-12-30 17:39:46] ✓ Status: 200 - Healthy
[2025-12-30 17:39:54] ✓ Status: 200 - Healthy
[2025-12-30 17:39:56] ✓ Status: 200 - Healthy
[2025-12-30 17:40:04] ✓ Status: 200 - Healthy
[2025-12-30 17:40:07] ✓ Status: 200 - Healthy
[2025-12-30 17:40:14] ✓ Status: 200 - Healthy
[2025-12-30 17:40:17] ✓ Status: 200 - Healthy
[2025-12-30 17:40:24] ✓ Status: 200 - Healthy
[2025-12-30 17:40:27] ✓ Status: 200 - Healthy
```

## Part 6: Documentation & Cleanup (10 marks)

### 6.1 README Documentation (5 marks)



```
1 # Assignment 2 - Advanced Terraform & Nginx Multi-Tier Architecture
2
3 ---
4 ## 1. Project Overview
5 This project demonstrates the deployment of a **secure, highly available, multi-tier web
6 infrastructure** on AWS using **Terraform**.
7 The architecture includes an **Nginx reverse proxy/load balancer** and **three backend web
8 servers**, where two servers handle primary traffic and one acts as a **backup server**.
9
10 The system supports:
11 - Load balancing
12 - HTTPS with self-signed SSL
13 - Caching
14 - High availability (failover)
15 - Infrastructure as Code (IaC)
16
17 ### Architecture Diagram
18
19 ```text
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
```

```
Rughma@DESKTOP-LJFKHUN MINGW64 /d/docs/Assignment2
$ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 40,
  "lineage": "5e4c4843-6ab3-3a11-0dde-f9b10742df8f",
  "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
    }
  ]
}
```

Instances (5) Info Refresh Connect Instance state Actions Launch instances

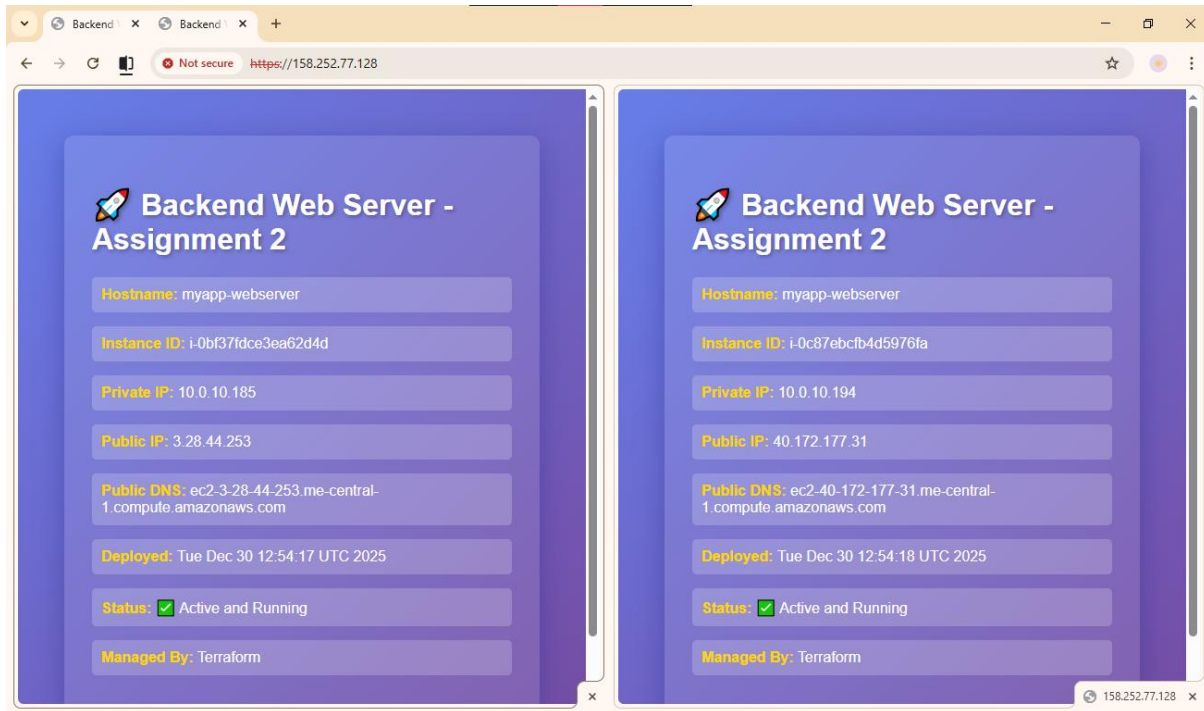
Find Instance by attribute or tag (case-sensitive) All states < 1 > Settings

<input type="checkbox"/>	Name <span>↗</span>	Instance ID	Instance state <span>↕</span>	Instance type <span>↕</span>	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	dev-ec2-insta...	i-02ced525da2d44ae6	Running <span>🔍</span> <span>🔍</span>	t3.micro	3/3 checks passed	<a href="#">View alarms</a> <span>+</span>	me-c
<input type="checkbox"/>	prod-web-2	i-0c87ebcfb4d5976fa	Terminated <span>🔍</span> <span>🔍</span>	t3.micro	-	<a href="#">View alarms</a> <span>+</span>	me-c
<input type="checkbox"/>	prod-web-3	i-0ddead9a50b5e91b3	Terminated <span>🔍</span> <span>🔍</span>	t3.micro	-	<a href="#">View alarms</a> <span>+</span>	me-c
<input type="checkbox"/>	prod-nginx-pr...	i-0eff4e877ccb53f4b	Terminated <span>🔍</span> <span>🔍</span>	t3.micro	-	<a href="#">View alarms</a> <span>+</span>	me-c
<input type="checkbox"/>	prod-web-1	i-0bf37fdce3ea62d4d	Terminated <span>🔍</span> <span>🔍</span>	t3.micro	-	<a href="#">View alarms</a> <span>+</span>	me-c

## 4. Testing Results

### 4.1 Load Balancing Test

- Requests alternated between web-1 and web-2
- web-3 remained inactive as backup



## 4.2 Cache Performance Test

- First request: X-Cache-Status: MISS
- Subsequent requests: X-Cache-Status: HIT

Name	X	Headers	Preview	Response	Initiator	Timing
158.252.77.128		General				
		Request URL		https://158.252.77.128/		
		Request Method		GET		
		Status Code		200 OK		
		Remote Address		158.252.77.128:443		
		Referrer Policy		strict-origin-when-cross-origin		
		Response Headers				
		Content-Encoding		gzip		
		Content-Type		text/html; charset=UTF-8		
		Date		Tue, 30 Dec 2025 14:30:17 GMT		
		Etag		W/"62e-6472adb8d59e1"		
		Last-Modified		Tue, 30 Dec 2025 12:54:17 GMT		
		Server		nginx/1.28.0		
		Vary		Accept-Encoding		
		X-Cache-Status		HIT		




Name	Headers	Preview	Response	Initiator	Timing
158.252.77.128	▼ General				
✖ favicon.ico	<div> <div>Request URL</div> <div>https://158.252.77.128/</div> </div> <div> <div>Request Method</div> <div>GET</div> </div> <div> <div>Status Code</div> <div>200 OK</div> </div> <div> <div>Remote Address</div> <div>158.252.77.128:443</div> </div> <div> <div>Referrer Policy</div> <div>strict-origin-when-cross-origin</div> </div>				
	▼ Response Headers				
	<div> <div>Content-Encoding</div> <div>gzip</div> </div> <div> <div>Content-Type</div> <div>text/html; charset=UTF-8</div> </div> <div> <div>Date</div> <div>Tue, 30 Dec 2025 14:51:43 GMT</div> </div> <div> <div>Etag</div> <div>W/"62e-6472adb8d59e1"</div> </div> <div> <div>Last-Modified</div> <div>Tue, 30 Dec 2025 12:54:17 GMT</div> </div> <div> <div>Server</div> <div>nginx/1.28.0</div> </div> <div> <div>Vary</div> <div>Accept-Encoding</div> </div> <div> <div>X-Cache-Status</div> <div>BYPASS</div> </div>				

### 4.3 High Availability Test

- Stopping Apache on web-1 and web-2 activated web-3
- Traffic resumed normally after restart

Backend Web Server

Not secure
https://158.252.77.128



## Backend Web Server - Assignment 2

**Hostname:** myapp-webserver

**Instance ID:** i-0ddead9a50b5e91b3

**Private IP:** 10.0.10.93

**Public IP:** 40.172.215.101

**Public DNS:** ec2-40-172-215-101.me-central-1.compute.amazonaws.com

**Deployed:** Tue Dec 30 12:54:17 UTC 2025

**Status:** ✔ Active and Running

**Managed By:** Terraform

```
[ec2-user@ip-10-0-10-89 ~]$ sudo tail -n 20 /var/log/nginx/error.log
2025/12/30 14:14:11 [notice] 27339#27339: using the "epoll" event method
2025/12/30 14:14:11 [notice] 27339#27339: nginx/1.28.0
2025/12/30 14:14:11 [notice] 27339#27339: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/30 14:14:11 [notice] 27339#27339: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/30 14:14:11 [notice] 27340#27340: start worker processes
2025/12/30 14:14:11 [notice] 27340#27340: start worker process 27341
2025/12/30 14:14:11 [notice] 27340#27340: start worker process 27342
2025/12/30 14:14:11 [notice] 27340#27340: start cache manager process 27343
2025/12/30 14:14:11 [notice] 27340#27340: start cache loader process 27344
2025/12/30 14:15:11 [notice] 27344#27344: http file cache: /var/cache/nginx 0.004M, bsize: 4096
2025/12/30 14:15:12 [notice] 27340#27340: signal 17 (SIGCHLD) received from 27344
2025/12/30 14:15:12 [notice] 27340#27340: cache loader process 27344 exited with code 0
2025/12/30 14:15:12 [notice] 27340#27340: signal 29 (SIGIO) received
2025/12/30 14:26:24 [crit] 27342#27342: *73 SSL_do_handshake() failed (SSL: error:0A000172:SSL routines::wrong signature type) while SSL handshaking, client: 50.116.49.249, server: 0.0.0.0:443
2025/12/30 15:44:19 [error] 27342#27342: *154 connect() failed (111: Connection refused) while connecting to upstream, client: 154.192.0.128, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:44:19 [warn] 27342#27342: *154 upstream server temporarily disabled while connecting to upstream, client: 154.192.0.128, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [error] 27342#27342: *167 connect() failed (111: Connection refused) while connecting to upstream, client: 154.192.0.128, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [warn] 27342#27342: *167 upstream server temporarily disabled while connecting to upstream, client: 154.192.0.128, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [error] 27342#27342: *167 connect() failed (111: Connection refused) while connecting to upstream, client: 154.192.0.128, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
2025/12/30 15:45:59 [warn] 27342#27342: *167 upstream server temporarily disabled while connecting to upstream, client: 154.192.0.128, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0.10.185:80/", host: "158.252.77.128"
[ec2-user@ip-10-0-10-89 ~]$
```

## 4.4 Security Testing

- SSL certificate verified
- Security headers confirmed
- HTTP redirected to HTTPS

```
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 729F5EF620634B1BF39F82402855B836AAF4B32363F063B76AF0B81672D3AC9B
    Session-ID-ctx:
    Resumption PSK: A3B74206CC70757A8EE63791524EBB721976E40A3794429D99EEA558193D1BB358DD53AC47B1D3EAF88FA5602AADED09
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - a5 80 d2 62 3f 60 da 16-fa 8f c4 47 46 fe de e2 ..b?".....GF...
    0010 - 6b 58 7f 8f 43 02 b8 aa-27 d7 c6 5b b1 42 49 f4 kX..C....'[.BI.
    0020 - 36 b0 ca f5 bf 60 22 b1-8a b6 e3 6b 0c 48 6b 9b 6.....".....k.Hk.
    0030 - dd 59 68 07 78 cf bc 02-74 69 a8 70 5f c0 4c 08 .Yh.x...ti.p..L.
    0040 - 24 33 25 48 8c c7 c8 b5-3f 66 04 77 19 e9 c2 f0 $3H....?f.w....
    0050 - be d2 79 b5 53 0b 48 0b-81 82 3f 09 c3 a3 5b cc ..y.S.H...?....[.
    0060 - 54 96 a3 f0 ae 22 2f bb-b2 c1 cc 05 ee cc 0f 34 T...."/.....4
    0070 - 91 fc af 2d e6 23 45 9e-9c 2f cc 0a cc a9 60 d7 ....#E../.....".
    0080 - cf 39 72 dc 94 78 54 76-67 f1 84 7f a4 95 f9 ba .9r..xTvg.....
    0090 - ed 08 ec ce 22 1f 03 df-fb 7e 2b df 79 9b 86 b6 ....~...+..y....
    00a0 - f9 b3 85 c3 83 76 80 88-00 56 b1 3a be 0d 41 09 .....v...V...A.
    00b0 - 4c 6e a5 63 1a 4d bd 79-9e b9 c7 88 6c 29 4e e0 Ln.c.M.y....l)N.
    00c0 - b4 0c c7 08 1f fb bf 43-ad 88 1c d4 e0 9a e2 47 .....C.....G
    00d0 - aa 49 82 bd 49 60 6a 42-b3 e3 6b eb 18 33 9a e2 .I..I'jB..k..3..

    Start Time: 1767110810
    Timeout    : 7200 (sec)
    Verify return code: 18 (self-signed certificate)
    Extended master secret: no
    Max Early Data: 0
---
```

```
[ec2-user@ip-10-0-10-89 ~]$ curl -I -k https://158.252.77.128
HTTP/2 200
server: nginx/1.28.0
date: Tue, 30 Dec 2025 16:05:34 GMT
content-type: text/html; charset=UTF-8
content-length: 1586
vary: Accept-Encoding
last-modified: Tue, 30 Dec 2025 12:54:18 GMT
etag: "632-6472adb9ca5aa"
x-cache-status: MISS
accept-ranges: bytes

[ec2-user@ip-10-0-10-89 ~]$
```

```
[ec2-user@ip-10-0-10-89 ~]$ curl -I http://158.252.77.128
\\HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Tue, 30 Dec 2025 16:04:38 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://158.252.77.128/

[ec2-user@ip-10-0-10-89 ~]$
```

## 5. Challenges & Solutions

### Challenge 1: Terraform command not found

- **Solution:** Added Terraform to system PATH and restarted terminal

### Challenge 2: AWS credential errors

- **Solution:** Installed AWS CLI and configured credentials

### Challenge 3: Incorrect Availability Zone

- **Solution:** Updated AZ to match the configured region

### Challenge 4: Nginx Syntax Error

- **Solution:** Sudo nginx -t which pointed to a syntax error on line 53. Upon inspection, I found an accidental typo (i}) caused by a vim keystroke error.

### Challenge 5: Availability Zone Mismatch

- **Solution:** Updated the availability\_zone variable to me-central-1a to match the provider's region

### Lessons Learned

- Importance of region consistency
- Benefits of Infrastructure as Code
- Debugging using logs is critical

## 6. Conclusion

This assignment successfully demonstrated the deployment of a secure, scalable, and highly available web infrastructure using Terraform and AWS. Through this project, practical skills in cloud networking, automation, server configuration, and testing were developed.

Future improvements may include:

- Using ACM certificates
- Auto Scaling Groups
- CloudWatch monitoring
- CI/CD integration

## 7. Appendices

### Appendix A – Repository Link with Complete Code Listings

[Github Repository-assignment2](#)

### Appendix B: Configuration Files

Final Nginx Configuration (/etc/nginx/nginx.conf) *The upstream block modified after deployment:*

```
# Upstream backend servers
# PLACEHOLDER: Update these IPs after deployment
upstream backend_servers {
    # Primary servers (active load balancing)
    server 10.0.10.185:80;
    server 10.0.10.194:80;

    # Backup server (only used when primary servers are down)
    server 10.0.10.93:80 backup;
}
```

### Appendix C: Additional Screenshots

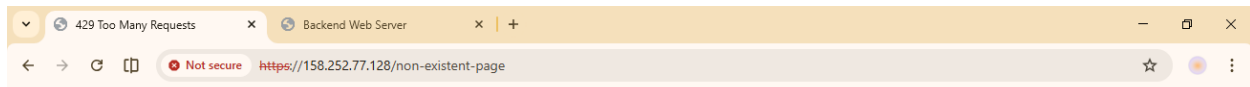
#### Bonus 1: Custom 404 Error Page



#### Not Found

The requested URL was not found on this server.

## Bonus 2: Rate Limit Testing (429 Too Many Requests)



### 429 Too Many Requests

nginx/1.28.0

## Bonus 3: Health Monitor Log

```
[ec2-user@myapp-webserver ~]$ tail -f health_monitor.log
[2025-12-30 17:38:56] ✓ Status: 200 - Healthy
[2025-12-30 17:39:02] ✓ Status: 200 - Healthy
[2025-12-30 17:39:06] ✓ Status: 200 - Healthy
[2025-12-30 17:39:12] ✓ Status: 200 - Healthy
[2025-12-30 17:39:16] ✓ Status: 200 - Healthy
[2025-12-30 17:39:22] ✓ Status: 200 - Healthy
[2025-12-30 17:39:26] ✓ Status: 200 - Healthy
[2025-12-30 17:39:32] ✗ Status: 000 - DOWN! Restarting Apache...
[2025-12-30 17:39:32] Restart result: 200
[2025-12-30 17:39:36] ✓ Status: 200 - Healthy
[2025-12-30 17:39:44] ✓ Status: 200 - Healthy
[2025-12-30 17:39:46] ✓ Status: 200 - Healthy
[2025-12-30 17:39:54] ✓ Status: 200 - Healthy
[2025-12-30 17:39:56] ✓ Status: 200 - Healthy
[2025-12-30 17:40:04] ✓ Status: 200 - Healthy
[2025-12-30 17:40:07] ✓ Status: 200 - Healthy
[2025-12-30 17:40:14] ✓ Status: 200 - Healthy
[2025-12-30 17:40:17] ✓ Status: 200 - Healthy
[2025-12-30 17:40:24] ✓ Status: 200 - Healthy
[2025-12-30 17:40:27] ✓ Status: 200 - Healthy
```

## Appendix D: References

1. **Terraform Documentation:** <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>
2. **Nginx Documentation:** <https://nginx.org/en/docs/>
3. **AWS EC2 User Guide:** <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/>
4. **Amazon Linux 2023:** <https://docs.aws.amazon.com/linux/al2023/>

---

THE END