

## Assignment 2 – Advanced Terraform & Nginx Multi-Tier Architecture



**Name: Zunaira Noor**

**Roll Number: 2023-BSE-075**

**Course Name: Cloud Computing**

**Submission Date: 30-12-2025**

### 1.1 Project Structure

```
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ tree -a
.
├── .gitignore
├── README.md
├── locals.tf
└── main.tf
└── modules
    ├── networking
    │   ├── main.tf
    │   ├── outputs.tf
    │   └── variables.tf
    ├── security
    │   ├── main.tf
    │   ├── outputs.tf
    │   └── variables.tf
    └── webserver
        ├── main.tf
        ├── outputs.tf
        └── variables.tf
└── outputs.tf
└── scripts
    └── apache-setup.sh
    └── nginx-setup.sh
└── terraform.tfvars
└── variables.tf

6 directories, 18 files
```

Create Proper .gitignore

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

# SSH keys
*.pem
*.key
id_ed25519
id_ed25519.pub

# OS / Editor files
.DS_Store
.vscode/
.idea/

# Logs
*.log
|
~
~
```

## 1.2 Variable Configuration

variables.tf

```

@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ cat variables.tf
# VPC CIDR block
variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
  type        = string
  default     = "10.0.0.0/16"
  validation {
    condition   = can(regex("^(0-9){1,3}\.\{3\}[0-9]{1,3}/[0-9]{1,2}$", var.vpc_cidr_block))
    error_message = "vpc_cidr_block must be a valid CIDR, e.g., 10.0.0.0/16"
  }
}

# Subnet CIDR block
variable "subnet_cidr_block" {
  description = "CIDR block for the subnet"
  type        = string
  default     = "10.0.10.0/24"
  validation {
    condition   = can(regex("^(0-9){1,3}\.\{3\}[0-9]{1,3}/[0-9]{1,2}$", var.subnet_cidr_block))
    error_message = "subnet_cidr_block must be a valid CIDR, e.g., 10.0.10.0/24"
  }
}

# Availability Zone
variable "availability_zone" {
  description = "AWS Availability Zone"
  type        = string
  default     = "me-central-1a"
}

# Environment prefix
variable "env_prefix" {

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

  # SSH Public Key
  variable "public_key" {
    description = "Path to the public SSH key"
    type        = string
    default     = "~/.ssh/id_ed25519.pub"
  }

  # SSH Private Key
  variable "private_key" {
    description = "Path to the private SSH key"
    type        = string
    default     = "~/.ssh/id_ed25519"
  }

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

```

## terraform.tfvars

```
Windows PowerShell      X + | v

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" }
]

~ ~ ~ ~ ~ ~ ~ ~ ~ ~
```

### 1.3 Networking Module

modules/networking/main.tf

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

# Create Internet Gateway
resource "aws_internet_gateway" "this" {
  vpc_id = aws_vpc.this.id
  tags = {
    Name = "${var.env_prefix}-igw"
  }
}

# Create Route Table
resource "aws_route_table" "this" {
  vpc_id = aws_vpc.this.id
  tags = {
    Name = "${var.env_prefix}-rt"
  }

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

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

:wq|
```

modules/networking/variables.tf

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

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

variable "availability_zone" {
  description = "Availability Zone for the subnet"
  type        = string
}

variable "env_prefix" {
  description = "Environment prefix for naming resources"
  type        = string
}
|
~
~
```

modules/networking/outputs.tf

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

output "subnet_id" {
  description = "ID of the Subnet"
  value       = aws_subnet.this.id
}

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

output "route_table_id" {
  description = "ID of the Route Table"
  value       = aws_route_table.this.id
}

|
~  
~  
~  
~  
~  
~  
~  
~
```

Usage in main.tf (root module)

```
module "networking" {
    source          = "./modules/networking"
    vpc_cidr_block = var.vpc_cidr_block
    subnet_cidr_block = var.subnet_cidr_block
    availability_zone = var.availability_zone
    env_prefix       = var.env_prefix
}

~  
~  
~  
~  
~  
~
```

## **1.4 Security Module (5 marks)**

Create a security module that provisions:

## Two Security Groups:

```
# Nginx Security Group
resource "aws_security_group" "nginx_sg" {
  name          = "${var.env_prefix}-nginx-sg"
  description   = "Security group for Nginx reverse proxy/load balancer"
  vpc_id        = var.vpc_id

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

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

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

  egress {
    from_port   = 0
```

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

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

  ingress {
    description = "HTTP from Nginx SG 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"]
  }
}
```

## Outputs

Create outputs.tf in modules/security/:

```
> Windows PowerShell      X + <

output "nginx_sg_id" {
    value      = aws_security_group.nginx_sg.id
    description = "ID of the Nginx security group"
}

output "backend_sg_id" {
    value      = aws_security_group.backend_sg.id
    description = "ID of the backend security group"
}
|
~  
~  
~  
~  
~  
~  
~
```

Usage in Root Module (Lab12\_Assignment/main.tf)

```
{
module "security" {
    source      = "./modules/security"
    vpc_id      = module.networking.vpc_id
    env_prefix = var.env_prefix
    my_ip       = "YOUR_PUBLIC_IP/32" # replace with your actual public IP
}
|
~  
~  
~  
~  
~  
~  
~  
~  
~
```

After apply

```

@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ terraform apply
module.networking.aws_vpc.this: Refreshing state... [id=vpc-05e7c329893357003]
module.networking.aws_subnet.this: Refreshing state... [id=subnet-0c219c1554e9cf982]
module.networking.aws_internet_gateway.this: Refreshing state... [id=igw-0798c28387ced3a92]
module.security.aws_security_group.nginx_sg: Refreshing state... [id=sg-0de2455b17391ea20]
module.networking.aws_route_table.this: Refreshing state... [id=rtb-084608c678c195eb5]
module.security.aws_security_group.backend_sg: Refreshing state... [id=sg-074a8338b98c9af2e]
module.networking.aws_route_table_association.this: Refreshing state... [id=rtbassoc-00b053c9dd5ea0115]

Changes to Outputs:
+ backend_sg_id = "sg-074a8338b98c9af2e"
+ nginx_sg_id   = "sg-0de2455b17391ea20"

You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

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

Outputs:

backend_sg_id = "sg-074a8338b98c9af2e"
nginx_sg_id   = "sg-0de2455b17391ea20"
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ |

```



## AWS Console Security Groups

C

Name	Security group ID	Security group name	VPC ID
-	<a href="#">sg-07434edd9c6e95b46</a>	default	<a href="#">vpc-0dfd6367126ad5bc</a>
prod-backend-sg	<a href="#">sg-074a8338b98c9af2e</a>	prod-backend-sg	<a href="#">vpc-05e7c32989335700</a>
prod-nginx-sg	<a href="#">sg-0de2455b17391ea20</a>	prod-nginx-sg	<a href="#">vpc-05e7c32989335700</a>
-	<a href="#">sg-0b3bf7f2e95a68242</a>	default	<a href="#">vpc-05e7c32989335700</a>

Select a security group

## 1.5 Locals Configuration (5 marks)

Create locals.tf with:

```
locals {
    # Append /32 to your public IP for security group rules
    my_ip = "${chomp(data.http.my_ip.response_body)}/32"

    # Common tags for all resources
    common_tags = {
        Environment = var.env_prefix
        Project     = "Lab12-Assignment"
        ManagedBy   = "Terraform"
    }

    # Backend server configurations
    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"
        }
    ]
}

"locals.tf" 36L, 774B
```

## Part 2: Webserver Module (15 marks)

### 2.1 Module Design (10 marks)

```
Windows PowerShell

variable "env_prefix" {
  type      = string
  description = "Environment prefix (e.g., prod, dev)"
}

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

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

variable "availability_zone" {
  type      = string
  description = "AWS Availability Zone"
}

variable "vpc_id" {
  type      = string
  description = "VPC ID where instance will be launched"
}

variable "subnet_id" {
  type      = string
  description = "Subnet ID for the instance"
}

variable "security_group_id" {
  type      = string
}
-- INSERT --
```

## Step 2: Create main.tf

```
Windows PowerShell

# Create Key Pair
resource "aws_key_pair" "this" {
  key_name    = "${var.env_prefix}-${var.instance_name}-${var.instance_suffix}"
  public_key  = file(var.public_key)
  tags        = var.common_tags
}

# Create EC2 Instance
resource "aws_instance" "this" {
  ami           = "ami-0c9fc9b90d9c91f5b" # Amazon Linux 2023 (update if needed)
  instance_type = var.instance_type
  availability_zone = var.availability_zone
  subnet_id     = var.subnet_id
  vpc_security_group_ids = [var.security_group_id]
  key_name       = aws_key_pair.this.key_name
  associate_public_ip_address = true

  user_data = file(var.script_path)

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

## Step 2: Create main.tf

```
Windows PowerShell
```

```
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
}
|
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

Use the module in root main.tf

```
    value = module.security.backend_sg_id
}
module "web_1" {
  source          = "./modules/webserver"
  env_prefix      = var.env_prefix
  instance_name   = "backend"
  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      = "./scripts/apache-setup.sh"
  instance_suffix  = "1"
  common_tags     = local.common_tags
}
|
-- INSERT --
```

## EC2 Instance with:

The screenshot shows the AWS EC2 Instances page. The left sidebar is titled 'EC2' and includes sections for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main content area is titled 'Instances (1) Info' and shows a table with one row. The table columns are Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. The single instance listed is 'prod-backend-1' with Instance ID 'i-05c4bdda09dae50b9', Instance state 'Running', Instance type 't3.micro', Status check 'Initializing', and Alarm status 'View alarms'. A search bar at the top says 'Find Instance by attribute or tag (case-sensitive)' and a filter button says 'Clear filters'.

## 2.2 Module Usage (5 marks)

Use dynamic blocks or for\_each for backend servers:

```
# 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 (3)
module "backend_servers" {
  for_each = { for idx, 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
}
```

```
| (and 3 more similar warnings elsewhere)

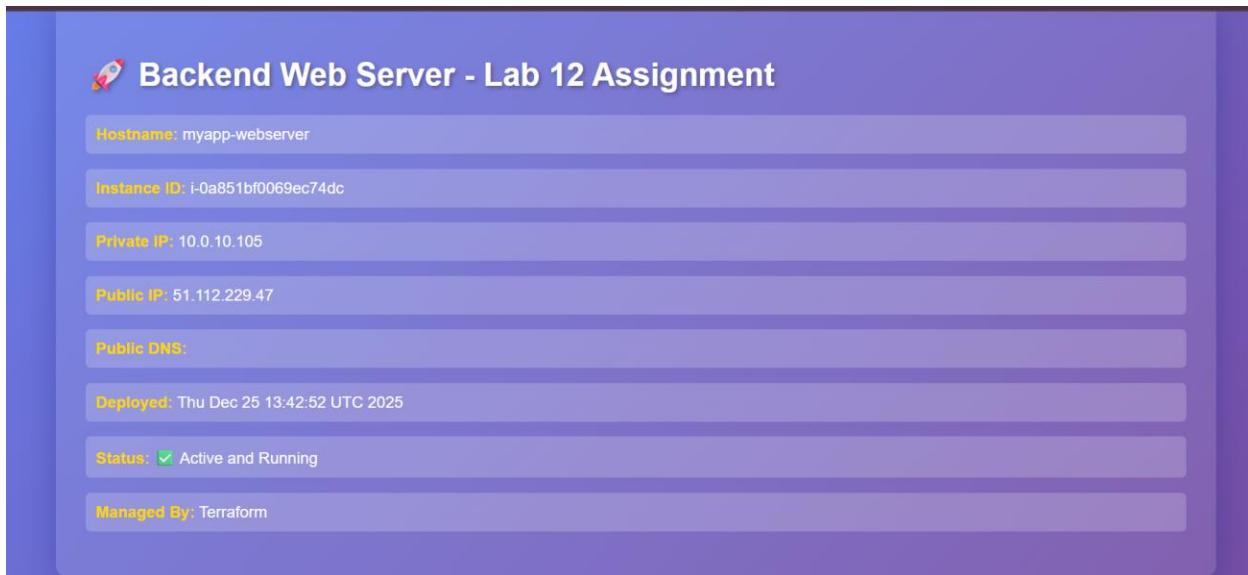
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.

Outputs:

backend_sg_id = "sg-074a8338b98c9af2e"
nginx_sg_id = "sg-0de2455b17391ea20"
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ |
```

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

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



The image shows two identical web pages displayed side-by-side in a browser. Both pages have a purple header with the title "Backend Web Server - Lab 12 Assignment" and a rocket icon. Below the header, there are several input fields containing configuration details:

- Hostname:** myapp-webserver
- Instance ID:** i-06a0255075461b3fd
- Private IP:** 10.0.10.249
- Public IP:** 40.172.187.20
- Public DNS:** (empty)
- Deployed:** Thu Dec 25 13:42:57 UTC 2025
- Status:**  Active and Running
- Managed By:** Terraform

The two pages are visually identical, except for the Public IP address (40.172.187.20) and Public DNS (empty) on the top page, and the Public IP address (3.28.41.123) and Public DNS (empty) on the bottom page.

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

scripts/nginx-setup.sh

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

# -----
# Nginx Setup Script with SSL
# -----

# Update system packages and install Nginx + OpenSSL
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 for EC2
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" \
```

```

< Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
< ETag: "603-646c6f40b5506"
< Accept-Ranges: bytes
<
<!DOCTYPE html>
<html>
<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 - Lab 12 Assignment</h1>
        <div class="info"><span class="label">Hostname:</span> myapp-webserver</div>
        <div class="info"><span class="label">Instance ID:</span> i-0a851bf0069ec74dc</div>
        <div class="info"><span class="label">Private IP:</span> 10.0.10.105</div>
        <div class="info"><span class="label">Public IP:</span> 51.112.229.47</div>
    </div>
</body>

```

## 4.1 Initial Deployment (5 marks)

### Steps to perform:

#### 1. Generate SSH key pair (if not exists)

```

@Zunaira-Noor123 ~ /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ ls ~/.ssh/id_ed25519
/home/codespace/.ssh/id_ed25519
@Zunaira-Noor123 ~ /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
/home/codespace/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:GVU2N5gRvnySFCcUyoBUbgAsrd33DKp4hv8ta04EmKE codespace@codespaces-5d58a1
The key's randomart image is:
++-[ED25519 256]-
|   ooooo .o@+*
|   . o .o + +o* .
|   + . + o o
|+. . .oo o o
|E .   oS+ = .
|   . . o o
|   o.o
|   o.* ..
|   B*...
+---[SHA256]---+
@Zunaira-Noor123 ~ /workspaces/CC_ZunairaNoor_075_Lab11 (main) $

```

### Initialize Terraform:

```
+----[SHA256]----+
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- 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.
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ |
```

### Validate configuration:

```
Commands will succeed if and when they are run in necessary.
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ terraform validate
Success! The configuration is valid.

@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ |
```

### Plan deployment:

```
~ private_dns_name_options (known after apply)
- private_dns_name_options {
  - enable_resource_name_dns_a_record    = false -> null
  - enable_resource_name_dns_aaaa_record = false -> null
  - hostname_type                      = "ip-name" -> null
}

~ root_block_device (known after apply)
- root_block_device {
  - delete_on_termination = true -> null
  - device_name          = "/dev/xvda" -> null
  - encrypted            = false -> null
  - iops                 = 3000 -> null
  - tags                 = {} -> null
  - tags_all              = {} -> null
  - throughput            = 125 -> null
  - volume_id             = "vol-0cba8b4f0ef4435da" -> null
  - volume_size            = 8 -> null
  - volume_type            = "gp3" -> null
  # (1 unchanged attribute hidden)
}
}

Plan: 1 to add, 0 to change, 1 to destroy.

Changes to Outputs:
~ aws_instance_public_ip = "3.29.244.20" -> (known after apply)

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.
```

### Apply deployment:

```

terraform apply now.
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11 (main) $ terraform apply -auto-approve
aws_key_pair.ssh_key: Refreshing state... [id=serverkey]
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-00a68f228a9315d1d]
aws_subnet.myapp_subnet_1: Refreshing state... [id=subnet-0ebd5874c01d71c35]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-017220bbaceab842a]
aws_default_security_group.myapp_sg: Refreshing state... [id=sg-0f292584bc7ff53aa]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-014a923a7b65bb9c7]
aws_instance.myapp_server: Refreshing state... [id=i-0444cd5f88c3369aa]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following
symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # aws_instance.myapp-server must be replaced
-/+ resource "aws_instance" "myapp-server" {
    ~ arn                               = "arn:aws:ec2:me-central-1:276995858123:instance/i-0444cd5f88c3369aa" -> (known after a
apply)
    ~ disable_api_stop                 = false -> (known after apply)
    ~ disable_api_termination          = false -> (known after apply)
    ~ ebs_optimized                   = false -> (known after apply)
    + enable_primary_ipv6              = (known after apply)
    - hibernation                     = false -> null
    + host_id                          = (known after apply)
    + host_resource_group_arn          = (known after apply)
    + iam_instance_profile             = (known after apply)
    ~ id                               = "i-0444cd5f88c3369aa" -> (known after apply)
    ~ instance_initiated_shutdown_behavior = "stop" -> (known after apply)
    + instance_lifecycle               = (known after apply)
        ~ instance_state                = "running" -> (known after apply)
}

```

## 4.2 Output Configuration (5 marks)

```

@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ vim outputs.tf
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ terraform output
backend_servers_private_ips = {
  "web-1" = "10.0.10.105"
  "web-2" = "10.0.10.249"
  "web-3" = "10.0.10.232"
}
backend_servers_public_ips = {
  "web-1" = "51.112.229.47"
  "web-2" = "40.172.187.20"
  "web-3" = "3.28.41.123"
}
backend_sg_id = "sg-04212b5d389bd5342"
nginx_private_ip = "10.0.10.185"
nginx_public_ip = "51.112.47.232"
nginx_sg_id = "sg-038da4178bc17d410"
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ 

```

```

@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ vim outputs.tf
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ terraform output
backend_servers_private_ips = {
  "web-1" = "10.0.10.105"
  "web-2" = "10.0.10.249"
  "web-3" = "10.0.10.232"
}
backend_servers_public_ips = {
  "web-1" = "51.112.229.47"
  "web-2" = "40.172.187.20"
  "web-3" = "3.28.41.123"
}
backend_sg_id = "sg-04212b5d389bd5342"
nginx_private_ip = "10.0.10.185"
nginx_public_ip = "51.112.47.232"
nginx_sg_id = "sg-038da4178bc17d410"
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ terraform output -json > outputs.json
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ cat outputs.json
{
  "backend_servers_private_ips": {
    "sensitive": false,
    "type": [
      "object",
      {
        "web-1": "string",
        "web-2": "string",
        "web-3": "string"
      }
    ]
  }
}

```

## 4.3 AWS Console Verification (5 marks)

Verify all resources in AWS Console.

## Verify VPC created

The screenshot shows the AWS VPC dashboard under the 'Your VPCs' tab. The table displays the following information:

Name	VPC ID	State	Encryption c...	Encryption contrc
dev-vpc	vpc-00a68f228a9315d1d	Available	-	-
<b>prod-vpc</b>	<b>vpc-047b1eb8c3d0e780e</b>	Available	-	-
-	vpc-0dfd6367126ad5bc9	Available	-	-

## Verify Subnet created

The screenshot shows the AWS VPC dashboard under the 'Subnets' tab. The table displays the following information:

Name	Subnet ID	State	VPC
-	subnet-056cc48cb248266d3	Available	vpc-0dfd6367126ad5bc9
dev-subnet-1	subnet-0ebd5874c01d71c35	Available	vpc-00a68f228a9315d1d   dev...
prod-subnet	subnet-0b87c585cb3ac7c7a	Available	vpc-047b1eb8c3d0e780e   prod...
-	subnet-0c352fea3f05561c1	Available	vpc-0dfd6367126ad5bc9

- Verify Internet Gateway attached

The screenshot shows the AWS VPC dashboard under the 'Internet gateways' tab. The table displays the following information:

Name	Internet gateway ID	State	VPC ID
dev-igw	igw-017220bbaceab842a	Attached	vpc-00a68f228a9315d1d   dev...
prod-igw	igw-095fd80411b95a0d	Attached	vpc-047b1eb8c3d0e780e   prod...
-	igw-0ec6044777e99c960	Attached	vpc-0dfd6367126ad5bc9

- Verify Route Table configured

Name	Route table ID	Explicit subnet associ...	Edge associations	Main
prod-rt	rtb-0bf52182d99a2051e	subnet-0b87c585cb3ac7...	-	No
dev-rt	rtb-014a923a7b65bb9c7	-	-	Yes
-	rtb-0d0f6009cdf6ba6d2	-	-	Yes
-	rtb-04748a9e046d15b70	-	-	Yes

- Verify Security Groups created with correct rules

Name	Security group ID	Security group name	VPC ID
prod-nginx-sg	sg-038da4178bc17d410	prod-nginx-sg	vpc-047b1eb8c3d0e780
prod-backend-sg	sg-04212b5d389bd5342	prod-backend-sg	vpc-047b1eb8c3d0e780
-	sg-07434edd9c6e95b46	default	vpc-0dfd6367126ad5bc1
dev-sg	sg-0f292584bc7ff53aa	default	vpc-00a68f228a9315d1

- Verify all 4 EC2 instances running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
prod-backend-1	i-0d72e9f63a5ee47b9	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>
prod-web-3-3	i-064446b8572997de1	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>
prod-nginx-pr...	i-03b1d01e2ee9b4c2f	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>
prod-web-2-2	i-06a0255075461b3fd	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>
prod-web-1-1	i-0a851bf0069ec74dc	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>
dev-ec2-insta...	i-0ff22350df9eddfef	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>

- Verify Key Pairs created

The screenshot shows the AWS EC2 console with the 'Key pairs' section selected. On the left, there's a navigation sidebar with sections like 'Network & Security' (Security Groups, Elastic IPs, Placement Groups, Key Pairs), 'Load Balancing' (Load Balancers, Target Groups, Trust Stores), and 'Auto Scaling' (Launch Configurations, Auto Scaling Groups). The main area is titled 'Key pairs (7)' and contains a table with the following data:

	Name	Type	Created	Fingerprint	ID
<input type="checkbox"/>	prod-backend-1	ed25519	2025/12/25 18:41 GMT+5	AmXViZQU+DsyTY/9JhvX...	key-04b23f39a2614f415
<input type="checkbox"/>	serverkey	ed25519	2025/12/25 21:01 GMT+5	AmXViZQU+DsyTY/9JhvX...	key-0b7654fee63bb6f0
<input type="checkbox"/>	prod-web-3-3	ed25519	2025/12/25 18:41 GMT+5	AmXViZQU+DsyTY/9JhvX...	key-017e05da8b53813b3
<input type="checkbox"/>	prod-nginx-proxy-nginx	ed25519	2025/12/25 18:41 GMT+5	AmXViZQU+DsyTY/9JhvX...	key-0d6c884d31f378269
<input type="checkbox"/>	MyED25519Key	ed25519	2025/12/25 00:49 GMT+5	QsmfSDaeAPdgOtQLFm8...	key-0d741fce12478f5e4
<input type="checkbox"/>	prod-web-2-2	ed25519	2025/12/25 18:41 GMT+5	AmXViZQU+DsyTY/9JhvX...	key-00c737cf2ad7ef472
<input type="checkbox"/>	prod-web-1-1	ed25519	2025/12/25 18:41 GMT+5	AmXViZQU+DsyTY/9JhvX...	key-01ff1f2bb6241099e5

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

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

```

tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
types_hash_max_size 4096;

include /usr/share/nginx/modules/*.conf;
include /etc/nginx/conf.d/*.conf;

# Upstream backend servers (outside any server block)
upstream backend_servers {
    server 10.0.10.105:80;
    server 10.0.10.249:80;
    server 10.0.10.232:80 backup;
}

```

### Test Nginx

```

[root@ip-10-0-10-185 ~]# sudo vim /etc/nginx/nginx.conf
[root@ip-10-0-10-185 ~]# sudo nginx -t
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:56
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@ip-10-0-10-185 ~]#

```

### Restart Nginx

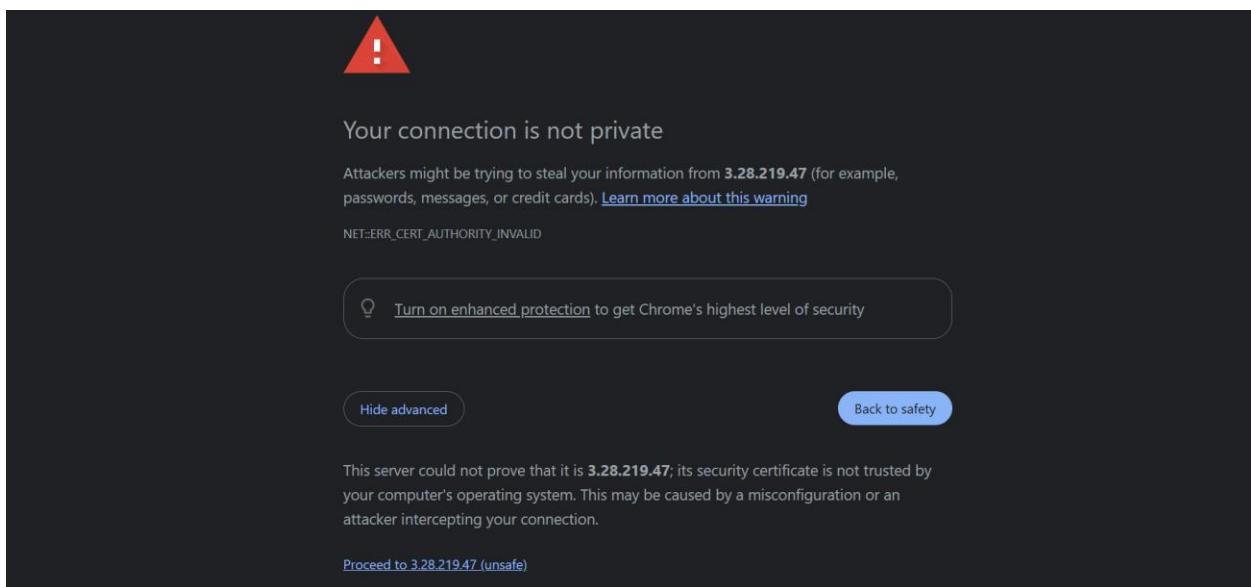
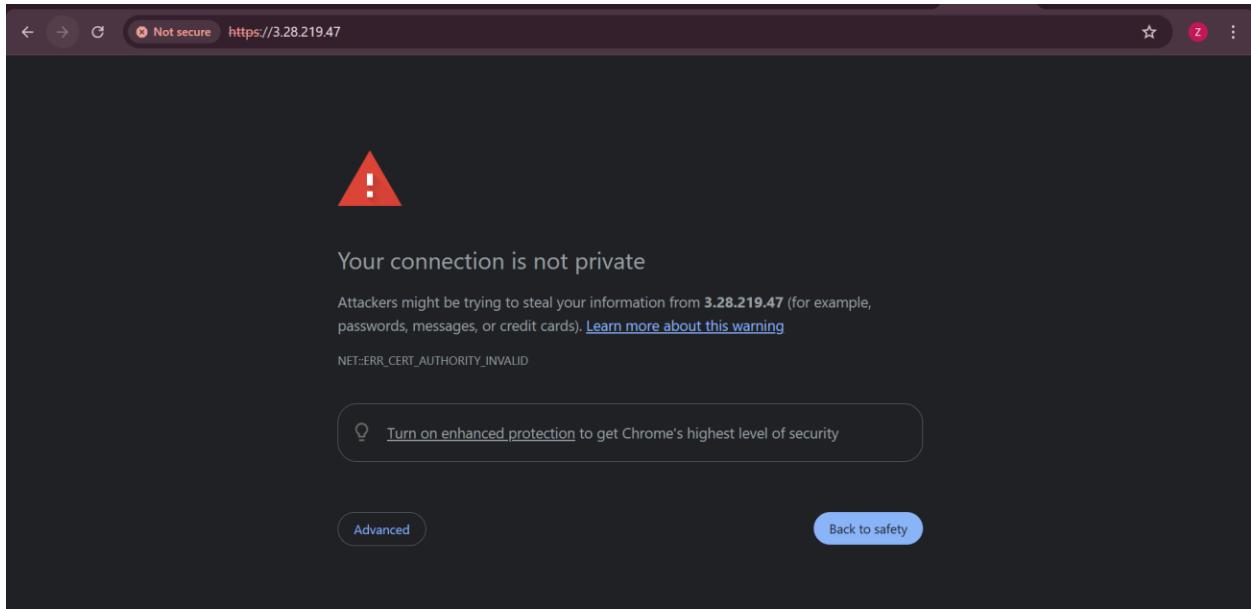
```

● 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 Thu 2025-12-25 18:11:22 UTC; 1min 25s ago
     Process: 2502 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Process: 2499 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 2497 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
  Main PID: 2504 (nginx)
     CGroup: /system.slice/nginx.service
             ├─2504 nginx: master process /usr/sbin/nginx
             ├─2505 nginx: worker process
             └─2506 nginx: worker process

Dec 25 18:11:22 ip-10-0-10-185.me-central-1.compute.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Dec 25 18:11:22 ip-10-0-10-185.me-central-1.compute.internal nginx[2499]: nginx: [warn] the "listen ... http2" directive is deprecated, us...nf
Dec 25 18:11:22 ip-10-0-10-185.me-central-1.compute.internal nginx[2499]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Dec 25 18:11:22 ip-10-0-10-185.me-central-1.compute.internal nginx[2499]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Dec 25 18:11:22 ip-10-0-10-185.me-central-1.compute.internal nginx[2502]: nginx: [warn] the "listen ... http2" directive is deprecated, us...nf
Dec 25 18:11:22 ip-10-0-10-185.me-central-1.compute.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-10-0-10-185 ~]#

```

## browser security warning



showing web-1 content

The screenshot shows a web-based configuration or monitoring interface for a server. At the top, there is a header with a rocket icon and the text "Backend Web Server - Lab 12 Assignment". Below the header, there are several data fields listed in a vertical stack:

- Hostname:** myapp-webserver
- Instance ID:** i-0a851bf0069ec74dc
- Private IP:** 10.0.10.105
- Public IP:** 51.112.229.47
- Public DNS:** (empty)
- Deployed:** Thu Dec 25 13:42:52 UTC 2025
- Status:**  Active and Running
- Managed By:** Terraform

showing web-2 content

The screenshot shows a web-based configuration or monitoring interface for a server. At the top, there is a header with a rocket icon and the text "Backend Web Server - Lab 12 Assignment". Below the header, there are several data fields listed in a vertical stack:

- Hostname:** myapp-webserver
- Instance ID:** i-06a0255075461b3fd
- Private IP:** 10.0.10.249
- Public IP:** 40.172.187.20
- Public DNS:** (empty)
- Deployed:** Thu Dec 25 13:42:57 UTC 2025
- Status:**  Active and Running
- Managed By:** Terraform

### 5.3 Test Cache Functionality (5 marks)

Verify that Nginx caching is working correctly.

first request - MISS

```
[root@ip-10-0-10-185 ~]# curl -k -I https://localhost
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 19:50:09 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1539
Connection: keep-alive
Upgrade: h2,h2c
Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
ETag: "603-646c6f40b5506"
X-Cache-Status: MISS
Accept-Ranges: bytes
```

### second request - HIT

```
[root@ip-10-0-10-185 ~]# curl -k -I https://localhost
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 19:50:15 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1539
Connection: keep-alive
Upgrade: h2,h2c
Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
ETag: "603-646c6f40b5506"
X-Cache-Status: HIT
Accept-Ranges: bytes
```

### cache folder contents

```
[root@ip-10-0-10-185 ~]# ls -la /var/cache/nginx/
total 0
drwx----- 3 nginx root 15 Dec 25 19:50 .
drwxr-xr-x 7 root root 76 Dec 25 19:46 ..
drwx----- 3 nginx nginx 16 Dec 25 19:50 f
[root@ip-10-0-10-185 ~]# █
```

### access log showing cache status

```

drwxr-xr-x / root root 76 Dec 25 19:46 ..
drwx----- 3 nginx nginx 16 Dec 25 19:50 f
[root@ip-10-0-10-185 ~]# sudo tail -f /var/log/nginx/access.log
23.249.17.76 -- [25/Dec/2025:19:09:47 +0000] "POST / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36" "-"
23.249.17.76 -- [25/Dec/2025:19:09:48 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36" "-"
103.53.162.15 -- [25/Dec/2025:19:24:26 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 -- [25/Dec/2025:19:24:27 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "https://3.28.219.47/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 -- [25/Dec/2025:19:27:27 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 -- [25/Dec/2025:19:27:28 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "https://3.28.219.47/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
173.255.221.189 -- [25/Dec/2025:19:33:12 +0000] "GET / HTTP/1.1" 400 255 "-" "Mozilla/5.0 zgrab/0.x" "-"
204.76.203.219 -- [25/Dec/2025:19:34:55 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-"
127.0.0.1 -- [25/Dec/2025:19:50:09 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0" "-"
127.0.0.1 -- [25/Dec/2025:19:50:15 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0" "-"

```

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

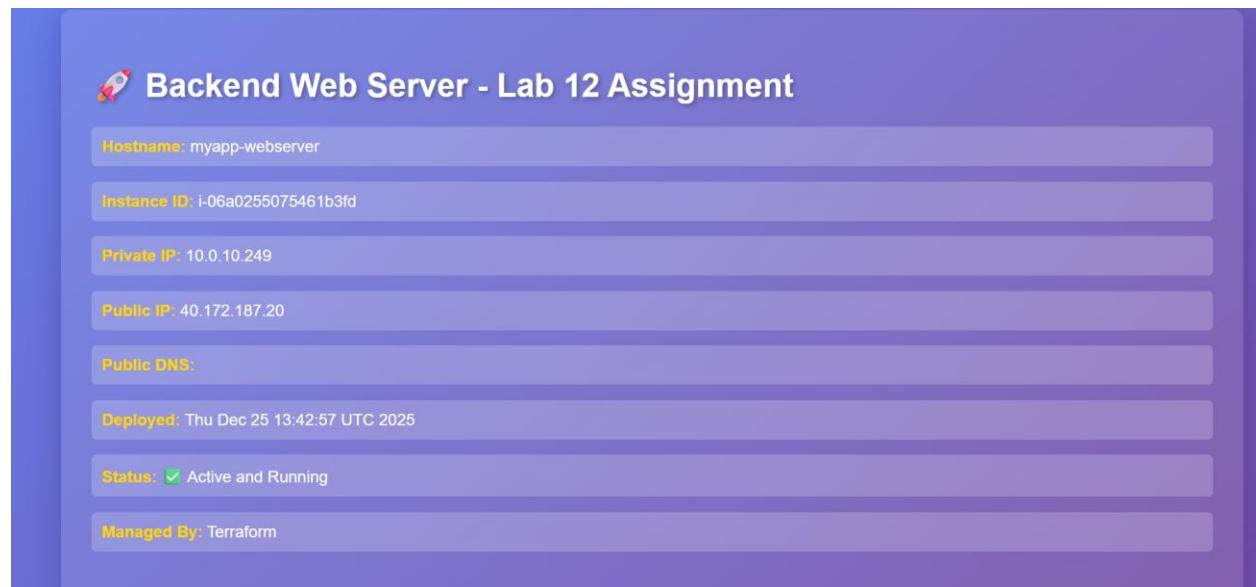
**SSH (Session Manager) into web-1 and stop Apache:**

```

[root@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 Thu 2025-12-25 21:11:11 UTC; 2min 30s ago
     Docs: man:httpd.service(8)
   Process: 16118 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 16118 (code=exited, status=0/SUCCESS)
    Status: "Total requests: 18; Idle/Busy workers 100/0;Requests/sec: 0.165; Bytes served/sec: 427 B/sec"

Dec 25 21:09:16 myapp-webserver systemd[1]: Starting The Apache HTTP Server...
Dec 25 21:09:16 myapp-webserver httpd[16118]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, u
Dec 25 21:09:16 myapp-webserver systemd[1]: Started The Apache HTTP Server.
Dec 25 21:11:10 myapp-webserver systemd[1]: Stopping The Apache HTTP Server...
Dec 25 21:11:11 myapp-webserver systemd[1]: Stopped The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[root@myapp-webserver ~]#

```



**Stop web-2 Apache:**

```

sh-4.2$ sudo -i
[root@myapp-webserver ~]# sudo systemctl stop httpd
[root@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 Thu 2025-12-25 21:09:58 UTC; 22s ago
        Docs: man:httpd.service(8)
     Process: 5834 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
    Main PID: 5834 (code=exited, status=0/SUCCESS)
       Status: "Total requests: 32; Idle/Busy workers 100/0;Requests/sec: 0.00119; Bytes served/sec: 2 B/sec"

Dec 25 13:42:57 ip-10-0-10-249.me-central-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Dec 25 13:42:57 ip-10-0-10-249.me-central-1.compute.internal systemd[1]: Started The Apache HTTP Server.
Dec 25 21:09:57 myapp-webserver systemd[1]: Stopping The Apache HTTP Server...
Dec 25 21:09:58 myapp-webserver systemd[1]: Stopped The Apache HTTP Server.
[root@myapp-webserver ~]#

```

The screenshot shows a web-based interface for managing a backend web server. The title is 'Backend Web Server - Lab 12 Assignment'. Below the title, there is a list of configuration parameters:

- Hostname:** myapp-webserver
- Instance ID:** i-064446b8572997de1
- Private IP:** 10.0.10.232
- Public IP:** 3.28.41.123
- Public DNS:** (empty)
- Deployed:** Thu Dec 25 13:42:53 UTC 2025
- Status:**  Active and Running
- Managed By:** Terraform

## Check Nginx error logs

```

[root@ip-10-0-10-185 ~]# sudo tail -f /var/log/nginx/error.log
2025/12/25 21:10:30 [error] 2899#2899: *31 connect() failed (111: Connection refused) while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.249:80/", host: "3.28.219.47"
2025/12/25 21:10:30 [warn] 2899#2899: *31 upstream server temporarily disabled while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.249:80/", host: "3.28.219.47"
2025/12/25 21:11:15 [error] 2899#2899: *31 connect() failed (111: Connection refused) while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.105:80/", host: "3.28.219.47"
2025/12/25 21:11:15 [warn] 2899#2899: *31 upstream server temporarily disabled while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.105:80/", host: "3.28.219.47"
2025/12/25 21:11:15 [error] 2899#2899: *31 connect() failed (111: Connection refused) while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.249:80/", host: "3.28.219.47"
2025/12/25 21:11:15 [warn] 2899#2899: *31 upstream server temporarily disabled while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.249:80/", host: "3.28.219.47"
2025/12/25 21:12:28 [error] 2900#2900: *76 connect() failed (111: Connection refused) while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.249:80/", host: "3.28.219.47"
2025/12/25 21:12:28 [warn] 2900#2900: *76 upstream server temporarily disabled while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.249:80/", host: "3.28.219.47"
2025/12/25 21:12:28 [error] 2900#2900: *76 connect() failed (111: Connection refused) while connecting to upstream, client: 103.53.162.15, server: _, request: "GET / HTTP/1.1", upstream: "http://10.0.10.105:80/", host: "3.28.219.47"
2025/12/25 21:12:28 [warn] 2900#2900: *76 upstream server temporarily disabled while connecting to upstream, client: 103.53.162.15, server: _

```

## Restart web-1 and web-2

```
[root@ip-10-0-10-185 ~]# curl -k https://3.28.219.47 --http1.1
<!DOCTYPE html>
<html>
<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 - Lab 12 Assignment</h1>
        <div class="info"><span class="label">Hostname:</span> myapp-webserver</div>
        <div class="info"><span class="label">Instance ID:</span> i-064446b8572997de1</div>
        <div class="info"><span class="label">Private IP:</span> 10.0.10.232</div>
    </div>
</body>
```

## Verify traffic returns to web-1 and web-2

### web-1



### web-2

## Backend Web Server - Lab 12 Assignment

**Hostname:** myapp-webserver

**Instance ID:** i-06a0255075461b3fd

**Private IP:** 10.0.10.249

**Public IP:** 40.172.187.20

**Public DNS:**

**Deployed:** Thu Dec 25 13:42:57 UTC 2025

**Status:**  Active and Running

**Managed By:** Terraform

### 5.5 Security & Performance Analysis (5 marks)

Analyze the security headers and performance of your Nginx setup.

Check SSL/TLS certificate details

```
[root@ip-10-0-10-185 ~]# openssl s_client -connect 3.28.219.47:443 -showcerts
CONNECTED(00000003)
depth=0 CN = localhost
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = localhost
verify return:1
---
Certificate chain
  0 s:/CN=localhost
    i:/CN=localhost
-----BEGIN CERTIFICATE-----
MIIC+zCCAeOgAwIBAgIJAk6tjDPKiigMA0GCSqGSIb3DQEBCwUAMBQxEjAQBgNV
BAMMCWxvY2FsaG9zdDaeFw0yNTBvMjUxNjE5MDZaFw0yNjEyMjUxNjE5MDZaMBQx
EjAQBgNVBAMMCWxvY2FsaG9zdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggERANz196ecfeg0P3gvRAM1npqtTqaJ500g8wwVK0RMTdy0/b/+f/Zxm3+rPqv8z
Fcshm5XWOAhQqc0iNeWNKAEEUhXbjfamG8rUEEXX54xgo+lqYiv6QdxNKXfDDRByz
RQDUPV19qZaIgZuP4I8XLePG+Rb2tjktni/6UNUHJ4BUS1zo02ZMySy+PltMhK9
yAKDCMshYK3FI6ws62H3pNIkAVMMffw4z4sBb+exUJ/HExVnd0GePVKw+nv3e4v
xzjMiRv/vI3MX0C/TbepoyM/cPLUXFiY/99KTCYhp2MtqVJHjut/cUrBTTV6AFt
Ya6/Ex3hEHq7k7jilwzDYNRAaQECAwFAAaNQME4wHQYDVR0ORBYEF1qbeT70tKor
LnVFa56m5FD1ADdBMB8GA1UdIwQYMBaAF1qbeT70tKorLnVFa56m5FD1ADdBMAwG
A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAHyl88ism+oJ6fSMoixGZJoo
5L3wwGDTf5wp6fQO8gUU6YXmkZMLbnAE/4VIRzJr5N0dz6cy2wQ0kGG/uuqcpF0
tyaQCBeo2kTnT12K8eZTMODVxeJ3JLsEE3O1mi4s23Bdz0Dpn+irIDIFJLwl6jY+
hUTZJv7yIX5T/f0MfeCSz0unNnTPJS6GWl1.8zNVym62T4cd2ZDcVtrzgPKHZURi
bSekcpaj03NJIgQ+oSMXjh33X9GdZ9Knkou9tXp22JbzXrn9WkP0DbseyRyAcL7w
CbtmYBTHOdBVJfDQMsk7UtzgFivGEyxhQ7oySAoNXFSueXBIWNI8sW+bfD+jy5s=
-----END CERTIFICATE-----
```

View certificate on Nginx server

```
[root@ip-10-0-10-185 ~]# sudo openssl x509 -in /etc/ssl/certs/selfsigned.crt -text -noout
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number:
        a8:fa:b6:30:cf:2a:28:a0
Signature Algorithm: sha256WithRSAEncryption
    Issuer: CN=localhost
    Validity
        Not Before: Dec 25 16:19:06 2025 GMT
        Not After : Dec 25 16:19:06 2026 GMT
Subject: CN=localhost
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
        Modulus:
            00:dc:e5:f7:a7:9c:7d:e1:b4:3f:71:af:44:03:35:
            9e:9a:93:a9:a2:79:3b:48:3c:c3:05:4a:d1:13:13:
            77:2d:3f:6f:ff:9f:fd:9c:66:df:ea:cf:42:ff:33:
            15:cb:21:9b:95:d6:38:0c:a1:42:a7:34:88:d7:96:
            36:40:04:52:15:db:8d:f6:a6:1b:ca:d4:38:45:d7:
            e7:8c:60:a3:ed:6a:62:2b:fa:41:dc:4d:29:77:c3:
            0d:10:72:cd:14:03:50:fb:f5:f6:a6:5a:22:06:6e:
            3f:82:3c:5c:b7:8f:1b:e4:5b:da:d8:e4:b6:78:bf:
            e9:43:54:1c:9e:01:51:2d:73:a3:4d:99:33:24:b2:
            f8:f9:6d:32:12:bd:c8:02:83:08:c4:a1:60:ad:c5:
            23:ac:2c:eb:61:f7:a4:dd:64:01:53:0c:7d:ff:30:
            e3:3e:2c:05:bf:9e:c5:42:7f:1c:4a:15:35:dd:06:

```

## Check security headers

```
[root@ip-10-0-10-185 ~]# curl -I -k https://3.28.219.47
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 21:34:10 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1539
Connection: keep-alive
Upgrade: h2,h2c
Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
ETag: "603-646c6f40b5506"
X-Cache-Status: MISS
Accept-Ranges: bytes

```

## Test HTTP to HTTPS redirect

```
[root@ip-10-0-10-185 ~]# curl -I http://3.28.219.47
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 21:35:13 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://3.28.219.47/
```

## Step 5: Analyze Nginx logs

```
sudo tail -50 /var/log/nginx/error.log
```

```
[root@ip-10-0-10-185 ~]# sudo tail -50 /var/log/nginx/error.log
2025/12/25 18:45:08 [notice] 2504#2504: exit
2025/12/25 18:45:08 [notice] 2630#2630: using the "epoll" event method
2025/12/25 18:45:08 [notice] 2630#2630: nginx/1.28.0
2025/12/25 18:45:08 [notice] 2630#2630: built by gcc 7.3.1 20180712 (Red Hat 7.3.1-17) (GCC)
2025/12/25 18:45:08 [notice] 2630#2630: OS: Linux 5.10.228-219.884.amzn2.x86_64
2025/12/25 18:45:08 [notice] 2630#2630: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/25 18:45:08 [notice] 2632#2632: start worker processes
2025/12/25 18:45:08 [notice] 2632#2632: start worker process 2633
2025/12/25 18:45:08 [notice] 2632#2632: start worker process 2634
2025/12/25 19:36:19 [emerg] 2824#2824: "proxy_cache" zone "my_cache" is unknown in /etc/nginx/nginx.conf:89
2025/12/25 19:40:44 [emerg] 2836#2836: "proxy_cache_path" directive is not allowed here in /etc/nginx/nginx.conf:50
2025/12/25 19:47:05 [notice] 2632#2632: signal 3 (SIGQUIT) received from 1, shutting down
2025/12/25 19:47:05 [notice] 2633#2633: gracefully shutting down
2025/12/25 19:47:05 [notice] 2633#2633: exiting
2025/12/25 19:47:05 [notice] 2634#2634: gracefully shutting down
2025/12/25 19:47:05 [notice] 2634#2634: exiting
2025/12/25 19:47:05 [notice] 2633#2633: exit
2025/12/25 19:47:05 [notice] 2634#2634: exit
2025/12/25 19:47:05 [notice] 2632#2632: signal 17 (SIGCHLD) received from 2633
2025/12/25 19:47:05 [notice] 2632#2632: worker process 2633 exited with code 0
2025/12/25 19:47:05 [notice] 2632#2632: worker process 2634 exited with code 0
2025/12/25 19:47:05 [notice] 2632#2632: exit
2025/12/25 19:47:05 [notice] 2896#2896: using the "epoll" event method
2025/12/25 19:47:05 [notice] 2896#2896: nginx/1.28.0
2025/12/25 19:47:05 [notice] 2896#2896: built by gcc 7.3.1 20180712 (Red Hat 7.3.1-17) (GCC)
2025/12/25 19:47:05 [notice] 2896#2896: OS: Linux 5.10.228-219.884.amzn2.x86_64
2025/12/25 19:47:05 [notice] 2896#2896: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/25 19:47:05 [notice] 2898#2898: start worker processes
```

## **sudo tail -50 /var/log/nginx/access.log**

```
[root@ip-10-0-10-185 ~]# sudo tail -50 /var/log/nginx/access.log
103.53.162.15 - - [25/Dec/2025:19:27:28 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "https://3.28.219.47/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
173.255.221.189 - - [25/Dec/2025:19:33:12 +0000] "GET / HTTP/1.1" 400 255 "-" "Mozilla/5.0 zgrab/b0/x" "-"
204.76.203.219 - - [25/Dec/2025:19:34:55 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46" "-"
127.0.0.1 - - [25/Dec/2025:19:50:09 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0" "-"
127.0.0.1 - - [25/Dec/2025:19:50:15 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0" "-"
101.32.192.203 - - [25/Dec/2025:20:35:01 +0000] "HEAD /Core/Skin/Login.aspx HTTP/1.1" 301 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/106.0.0.0 Safari/537.36" "-"
101.32.192.203 - - [25/Dec/2025:20:35:02 +0000] "HEAD /Core/Skin/Login.aspx HTTP/1.1" 404 0 "http://3.28.219.47:80/Core/Skin/Login.aspx" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/106.0.0.0 Safari/537.36" "-"
110.38.199.135 - - [25/Dec/2025:20:35:21 +0000] "POST /GponForm/diag_Form?images/ HTTP/1.1" 301 169 "-" "Hello, World" "-"
110.38.199.135 - - [25/Dec/2025:20:35:22 +0000] "zsh+/tmp/gpon80&ipv=0" 400 157 "-" "-" "-"
103.53.162.15 - - [25/Dec/2025:21:03:07 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 - - [25/Dec/2025:21:03:11 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 - - [25/Dec/2025:21:03:13 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 - - [25/Dec/2025:21:08:55 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 - - [25/Dec/2025:21:08:56 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 - - [25/Dec/2025:21:08:57 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
103.53.162.15 - - [25/Dec/2025:21:10:29 +0000] "GET / HTTP/1.1" 200 1539 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/143.0.0.0 Safari/537.36" "
```

## **Step 6: Check Nginx worker processes**

```
3.28.219.47 - - [25/Dec/2025:21:35:13 +0000] "HEAD / HTTP/1.1" 301 0 "-" "curl/8.3.0" "-"
[root@ip-10-0-10-185 ~]# ps aux | grep nginx
root      2898  0.0  0.1 49052 1076 ?        Ss   19:47  0:00 nginx: master process /usr/sbin/nginx
nginx     2899  0.0  0.6 49604 5976 ?        S    19:47  0:00 nginx: worker process
nginx     2900  0.0  0.6 49604 5976 ?        S    19:47  0:00 nginx: worker process
nginx     2901  0.0  0.2 49264 2592 ?        S    19:47  0:00 nginx: cache manager process
root      3287  0.0  0.8 237824 7712 pts/0   S+   21:24  0:00 sudo tail -f /var/log/nginx/error.log
root      3289  0.0  0.0 114676 792 pts/0   S+   21:24  0:00 tail -f /var/log/nginx/error.log
root      3390  0.0  0.0 119420 908 pts/1   S+   21:40  0:00 grep --color=auto nginx
[root@ip-10-0-10-185 ~]#
```

## **Bonus Tasks (10 marks extra credit)**

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

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

**Verify custom error pages**

## 1. 404 Test:

```
[root@ip-10-0-10-185 ~]# curl -k https://curl -k https://3.28.219.47/nonexistentpage
curl: (6) Could not resolve host: curl
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
</body></html>
[root@ip-10-0-10-185 ~]#
```

## Access Nginx:

```
[root@ip-10-0-10-185 ~]# curl -k https://3.28.219.47
<!DOCTYPE html>
<html>
<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 - Lab 12 Assignment</h1>
    <div class="info"><span class="label">Hostname:</span> myapp-webserver</div>
    <div class="info"><span class="label">Instance ID:</span> i-06a0255075461b3fd</div>
    <div class="info"><span class="label">Private IP:</span> 10.0.10.249</div>
</div>

```

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

Add rate limiting to prevent abuse.

### Verify Rate Limiting Configuration

```
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    add_header X-Cache-Status $upstream_cache_status;
    limit_req zone=mylimit burst=20 nodelay;
}
error_page 404 /errors/404.html;
error_page 502 /errors/502.html;
error_page 503 /errors/503.html;
location /errors/ {
    internal;
```

### Test Rate Limiting

```
[root@ip-10-0-10-185 ~]# for i in {1..30}; do curl -i -k https://3.28.219.47; done
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 22:32:20 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1539
Connection: keep-alive
Upgrade: h2,h2c
Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
ETag: "603-646c6f40b5506"
X-Cache-Status: EXPIRED
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<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;
        }
    </style>
</head>
<body>
    <div class="info"><span class="label">Status:</span> <span>Active and Running</span></div>
    <div class="info"><span class="label">Managed By:</span> Terraform</div>
</body>
</html>
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 22:32:20 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1539
Connection: keep-alive
Upgrade: h2,h2c
Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
ETag: "603-646c6f40b5506"
X-Cache-Status: HIT
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<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;
        }
    </style>
</head>
<body>
    <div class="info"><span class="label">Status:</span> <span>Active and Running</span></div>
    <div class="info"><span class="label">Managed By:</span> Terraform</div>
</body>
</html>
```

```
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 22:32:20 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1539
Connection: keep-alive
Upgrade: h2,h2c
Last-Modified: Thu, 25 Dec 2025 13:42:52 GMT
ETag: "603-646c6f40b5506"
X-Cache-Status: HIT
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<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);
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Backend Web Server</h1>
        <p>This is the Backend Web Server. It is running smoothly!</p>
        <hr>
        <small>Powered by Nginx 1.28.0</small>
    </div>
</body>
</html>
```

```
HTTP/1.1 404 Not Found
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 22:32:21 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive

<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.28.0</center>
</body>
</html>
HTTP/1.1 404 Not Found
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 22:32:21 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive

<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.28.0</center>
</body>
</html>
[root@ip-10-0-10-185 ~]#
```

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

Create a shell script that monitors backend server health.

**Script Screenshot → open the script file:**

```
[root@ip-10-0-10-185 ~]# sudo vim /usr/local/bin/health_check.sh
```

**2. Log Screenshot → show log output:**

```
[root@ip-10-0-10-185 ~]# cat /var/log/backend_health.log
2025-12-25 22:39:11 - 10.0.10.105 is UP
2025-12-25 22:39:11 - 10.0.10.249 is UP
2025-12-25 22:39:11 - 10.0.10.232 is UP
[root@ip-10-0-10-185 ~]# █
```

```
GNU nano 7.2                                     README.md
Assignment 2 - Multi-Tier Web Infrastructure
Project Overview

This project implements a secure multi-tier web infrastructure on AWS using Terraform and Nginx.
Nginx acts as a reverse proxy and load balancer distributing traffic across multiple backend Apache web servers.

The setup includes:
HTTPS with SSL/TLS
Load balancing
Health checks
Security headers
Rate limiting
Custom error pages
Monitoring and logging

Architecture Overview
    Internet
        |
        | HTTPS (443)
        | HTTP (80)
[ Wrote 429 lines ]
^G Help      ^O Write Out     ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo      M-A Set Mark
^Y Exit      ^R Read File     ^V Replace      ^I Paste         ^J Justify      ^L Go To Line   M-F Redo      M-G Copy
```

GNS3 Nano 7.2

## Components Description

Nginx Server

Reverse proxy

SSL termination

Load balancing

Security headers

Rate limiting

## Backend Web Servers

Apache HTTP server

Serve application content

Health-checked by script

Terraform

Infrastructure provisioning

EC2, security groups, networking

AWS Systems Manager (SSM)

**^G** Help  
**^X** Exit

**^O** Write Out  
**^R** Read File

**^W** Where Is  
**^\\** Replace

**^K** Cut  
**^U** Paste

**Part 6.2 – Infrastructure Cleanup (5 marks)**

```

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

Next Steps:
1. SSH into Nginx server: ssh ec2-user@3.28.219.47
2. Edit Nginx config: sudo vim /etc/nginx/nginx.conf
3. Update backend IPs in upstream block:
   - BACKEND_IP_1: 10.0.10.105
   - BACKEND_IP_2: 10.0.10.249
   - BACKEND_IP_3: 10.0.10.232
4. Restart Nginx: sudo systemctl restart nginx
5. Test: https://3.28.219.47

Backend Servers:
- web-1: 51.112.229.47 (private: 10.0.10.105)
  - web-2: 40.172.187.20 (private: 10.0.10.249)
  - web-3: 3.28.41.123 (private: 10.0.10.232)

=====
EOT -> null
- nginx_instance_id      = "i-03b1d01e2ee9b4c2f" -> null
- nginx_private_ip       = "10.0.10.185" -> null
- nginx_public_ip        = "3.28.219.47" -> null
- nginx_sg_id            = "sg-038da4178bc17d410" -> null
- subnet_id               = "subnet-0b87c585cb3ac7c7a" -> null
- vpc_id                  = "vpc-047b1eb8c3d0e780e" -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```

## Step 2: Confirm Completion

```

module.aws_instance.this: Still destroying... [id=i-0a851bf0069ec74dc, 0m0s elapsed]
module.backend_servers["web-1"].aws_instance.this: Still destroying... [id=i-0a851bf0069ec74dc, 0
module.backend_servers["web-3"].aws_instance.this: Still destroying... [id=i-064446b8572997de1, 0
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-06a0255075461b3fd, 0
module.backend_servers["web-3"].aws_instance.this: Destruction complete after 50s
module.backend_servers["web-3"].aws_key_pair.this: Destroying... [id=prod-web-3-3]
module.web_1.aws_instance.this: Destruction complete after 50s
module.web_1.aws_key_pair.this: Destroying... [id=prod-backend-1]
module.backend_servers["web-3"].aws_key_pair.this: Destruction complete after 0s
module.web_1.aws_key_pair.this: Destruction complete after 0s
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-095fdf80411b95a0d, 00m50
module.networking.aws_internet_gateway.this: Destruction complete after 57s
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-06a0255075461b3fd, 0
module.backend_servers["web-1"].aws_instance.this: Still destroying... [id=i-0a851bf0069ec74dc, 0
module.backend_servers["web-2"].aws_instance.this: Destruction complete after 1m0s
module.backend_servers["web-2"].aws_key_pair.this: Destroying... [id=prod-web-2-2]
module.backend_servers["web-1"].aws_instance.this: Destruction complete after 1m0s
module.backend_servers["web-1"].aws_key_pair.this: Destroying... [id=prod-web-1-1]
module.networking.aws_subnet.this: Destroying... [id=subnet-0b87c585cb3ac7c7a]
module.security.aws_security_group.backend_sg: Destroying... [id=sg-04212b5d389bd5342]
module.backend_servers["web-2"].aws_key_pair.this: Destruction complete after 0s
module.backend_servers["web-1"].aws_key_pair.this: Destruction complete after 0s
module.networking.aws_subnet.this: Destruction complete after 1s
module.security.aws_security_group.backend_sg: Destruction complete after 1s
module.security.aws_security_group.nginx_sg: Destroying... [id=sg-038da4178bc17d410]
module.security.aws_security_group.nginx_sg: Destruction complete after 1s
module.networking.aws_vpc.this: Destroying... [id=vpc-047b1eb8c3d0e780e]
module.networking.aws_vpc.this: Destruction complete after 0s

Destroy complete! Resources: 17 destroyed.
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ |

```

## Step 2: Confirm Completion

```
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 111,
  "lineage": "c07899d9-348f-9281-e4f7-083a74ddbda7",
  "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
    }
  ]
}
@Zunaira-Noor123 → /workspaces/CC_ZunairaNoor_075_Lab11/Lab12_Assignment (main) $ |
```