



*Fatima Jinnah Women University*

Opening Portals of Excellence Through Higher Education

# **“Cloud Computing”**

## **Assignment 2**

### **BSE-V (A)**

**Submitted To: Waqas Saleem**

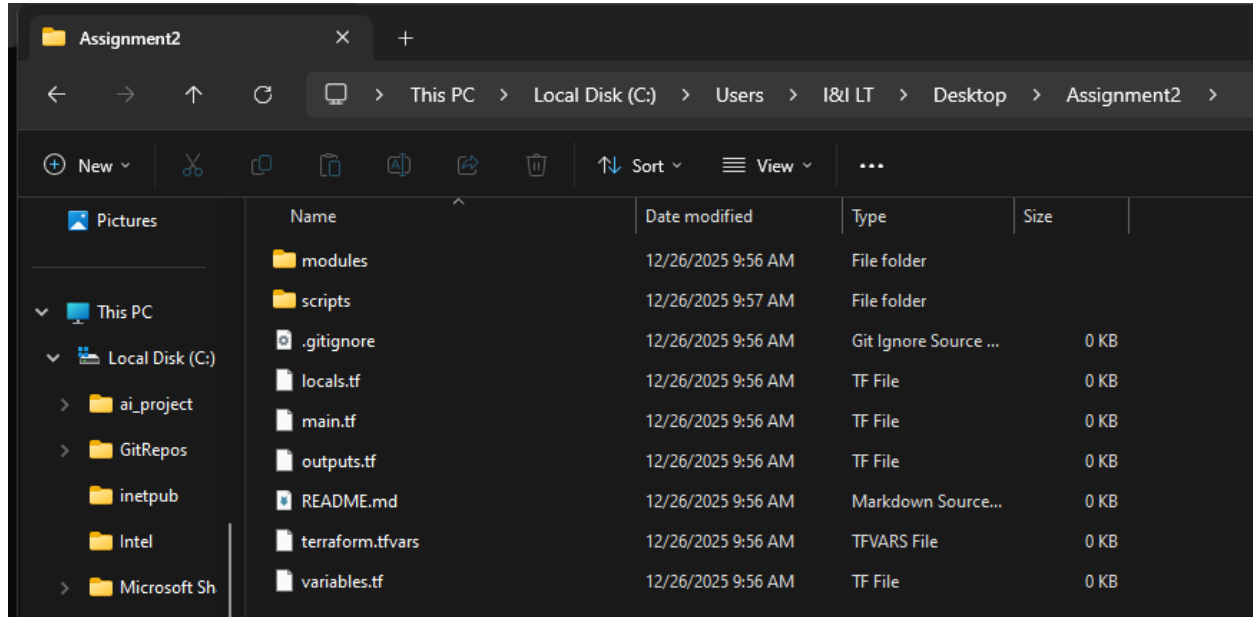
**Submitted By: Kainat Shakeel**

**Roll no: 2023-BSE-029**

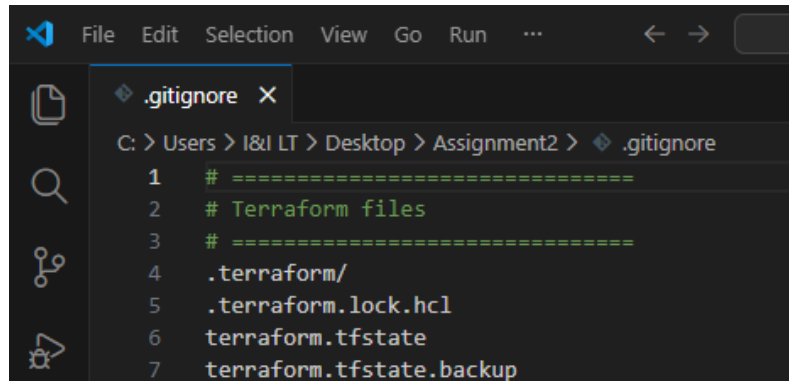
## Part 1: Infrastructure Setup

### 1.1 Project Structure

Create a well-organized Terraform project with the following structure:



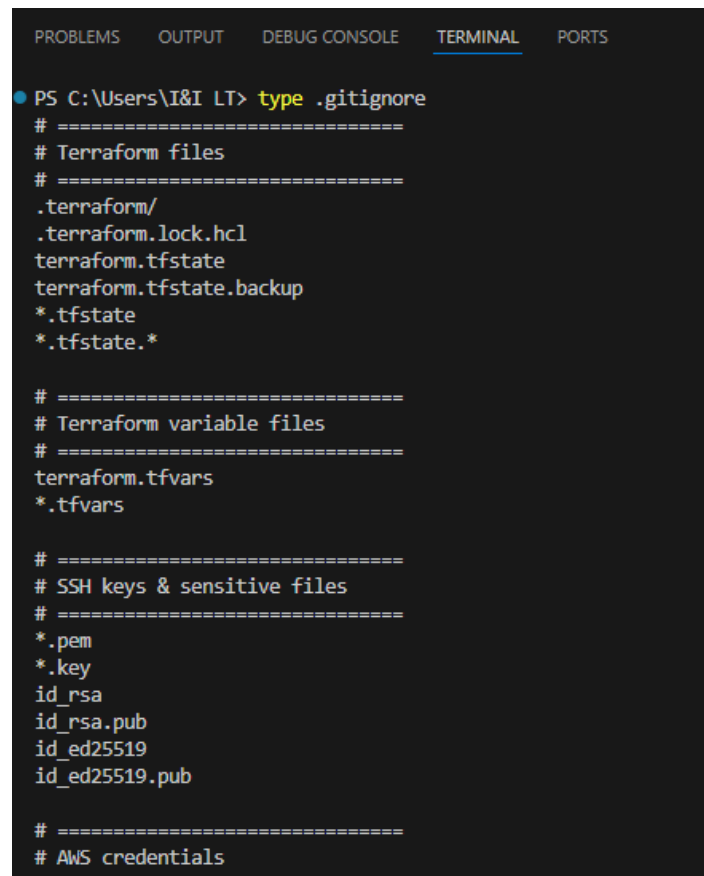
```
PS C:\Users\I&I LT\Desktop\Assignment2> tree /F
Folder PATH listing
Volume serial number is BEF9-1215
C:..
  .gitignore
  locals.tf
  main.tf
  outputs.tf
  README.md
  terraform.tfvars
  variables.tf
  modules
    networking
      main.tf
      outputs.tf
      variables.tf
    security
      main.tf
      outputs.tf
      variables.tf
    webserver
      main.tf
      outputs.tf
      variables.tf
  scripts
    apache-setup.sh
    nginx-setup.sh
```



A screenshot of the Visual Studio Code editor. The file explorer on the left shows the file `.gitignore` selected. The editor window displays the content of `.gitignore` with line numbers 1 through 7. The path in the title bar is `C:\Users\I&I LT\Desktop\Assignment2\> .gitignore`.

```
1 # =====
2 # Terraform files
3 # =====
4 .terraform/
5 .terraform.lock.hcl
6 terraform.tfstate
7 terraform.tfstate.backup
```

Verify .gitignore Content



A screenshot of a terminal window with the title bar showing `PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS`. The terminal shows the command `PS C:\Users\I&I LT> type .gitignore` and its output, which matches the content of the `.gitignore` file shown in the previous screenshot.

```
PS C:\Users\I&I LT> type .gitignore
# =====
# Terraform files
# =====
.terraform/
.terraform.lock.hcl
terraform.tfstate
terraform.tfstate.backup
*.tfstate
*.tfstate.*

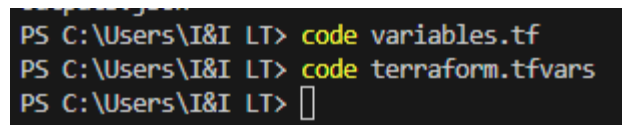
# =====
# Terraform variable files
# =====
terraform.tfvars
*.tfvars

# =====
# SSH keys & sensitive files
# =====
*.pem
*.key
id_rsa
id_rsa.pub
id_ed25519
id_ed25519.pub

# =====
# AWS credentials
# =====
```

## 1.2 Variable Configuration

Define all required variables in variables.tf:



A screenshot of a terminal window showing three commands being entered at the prompt `PS C:\Users\I&I LT>`:

```
PS C:\Users\I&I LT> code variables.tf
PS C:\Users\I&I LT> code terraform.tfvars
PS C:\Users\I&I LT> 
```

```
.gitignore X variables.tf X terraform.tfvars
C: > Users > I&I LT > variables.tf
1  # =====
2  # Networking Variables
3  # =====
4
5  variable "vpc_cidr_block" {
6      description = "CIDR block for the VPC"
7      type        = string
8
9      validation {
```

Verify the content for both

```
PS C:\Users\I&I LT> type .\variables.tf
# =====
# Networking Variables
# =====

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

    validation {
        condition     = can(cidrnetmask(var.vpc_cidr_block))
        error_message = "The VPC CIDR block must be a valid CIDR range."
    }
}

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

    validation {
        condition     = can(cidrnetmask(var.subnet_cidr_block))
        error_message = "The subnet CIDR block must be a valid CIDR range."
    }
}

variable "availability_zone" {
    description = "Availability zone where resources will be deployed"
```

```
PS C:\Users\I&I LT> type .\terraform.tfvars
vpc_cidr_block    = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"

env_prefix        = "prod"
instance_type     = "t3.micro"

public_key        = "~/ssh/id_ed25519.pub"
private_key       = "~/ssh/id_ed25519"
PS C:\Users\I&I LT>
```

### 1.3 Networking Module

Create a networking module that provisions:

- VPC with specified CIDR block
- Subnet with public IP assignment enabled
- Internet Gateway
- Route table with default route to IGW
- Associate route table with subnet

Open the files

```
PS C:\Users\I&I LT> code modules\networking\variables.tf
PS C:\Users\I&I LT> code modules\networking\main.tf
PS C:\Users\I&I LT> code modules\networking\outputs.tf
```

Write the code for each



The screenshot shows a code editor with four tabs: .gitignore, variables.tf C:\Users\I&I LT, variables.tf C:\Users\I&I\networking X, main.tf, and outputs.tf. The active tab is variables.tf C:\Users\I&I\networking X. The code in the editor is as follows:

```
C: > Users > I&I LT > modules > networking > variables.tf
1  variable "vpc_cidr_block" {
2      description = "CIDR block for the VPC"
3      type       = string
4  }
5
6  variable "subnet_cidr_block" {
7      description = "CIDR block for the public subnet"
8      type       = string
9  }
10
11 variable "availability_zone" {
12     description = "Availability zone for the subnet"
13     type       = string
14 }
15
16 variable "env_prefix" {
17     description = "Environment prefix for resource naming"
18     type       = string
```

Verify the content for each file

## Main.tf

```
● PS C:\Users\I&I LT> type .\modules\networking\main.tf
# =====
# VPC
# =====

resource "aws_vpc" "this" {
  cidr_block      = var.vpc_cidr_block
  enable_dns_support = true
  enable_dns_hostnames = true

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

# =====
# Public Subnet
# =====

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

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

## Output.tf

```
● PS C:\Users\I&I LT> type .\modules\networking\outputs.tf
output "vpc_id" {
  description = "ID of the VPC"
  value       = aws_vpc.this.id
}

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

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

output "route_table_id" {
  description = "ID of the route table"
  value       = aws_route_table.this.id
}
○ PS C:\Users\I&I LT> █
```

## Variable.tf

```
PS C:\Users\I&I LT> type .\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 public subnet"
  type        = string
}

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

variable "env_prefix" {
  description = "Environment prefix for resource naming"
  type        = string
}
```

## 1.4 Security Module

Create a security module that provisions:

### Two Security Groups:

1. **Nginx Security Group** (for reverse proxy/load balancer):
2. **Backend Security Group** (for web servers):

open the files

```
PS C:\Users\I&I LT> code modules\security\variables.tf
PS C:\Users\I&I LT> code modules\security\main.tf
PS C:\Users\I&I LT> code modules\security\outputs.tf
```

type the content for each:

```
variables.tf  main.tf  outputs.tf X
C: > Users > I&I LT > modules > security > outputs.tf
1  output "nginx_sg_id" {
2    description = "Security group ID for Nginx"
3    value       = aws_security_group.nginx.id
4  }
5
6  output "backend_sg_id" {
7    description = "Security group ID for backend servers"
8    value       = aws_security_group.backend.id
9  }
10
```

Verify the content:

### Variable.tf

```
PS C:\Users\I&I LT> type .\modules\security\variables.tf
variable "vpc_id" {
  description = "VPC ID where security groups will be created"
  type       = string
}

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

variable "my_ip" {
  description = "Your public IP address in CIDR format (e.g. 1.2.3.4/32)"
  type       = string
}
```

### Main.tf

```
PS C:\Users\I&I LT> type .\modules\security\main.tf
# =====
# NGINX Security Group
# =====

resource "aws_security_group" "nginx" {
  name        = "${var.env_prefix}-nginx-sg"
  description = "Security group for Nginx reverse proxy"
  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"]
  }
}
```

### Output.tf

```
PS C:\Users\I&I LT> type .\modules\security\outputs.tf
output "nginx_sg_id" {
  description = "Security group ID for Nginx"
  value       = aws_security_group.nginx.id
}

output "backend_sg_id" {
  description = "Security group ID for backend servers"
  value       = aws_security_group.backend.id
}
```



## 1.5 Locals Configuration

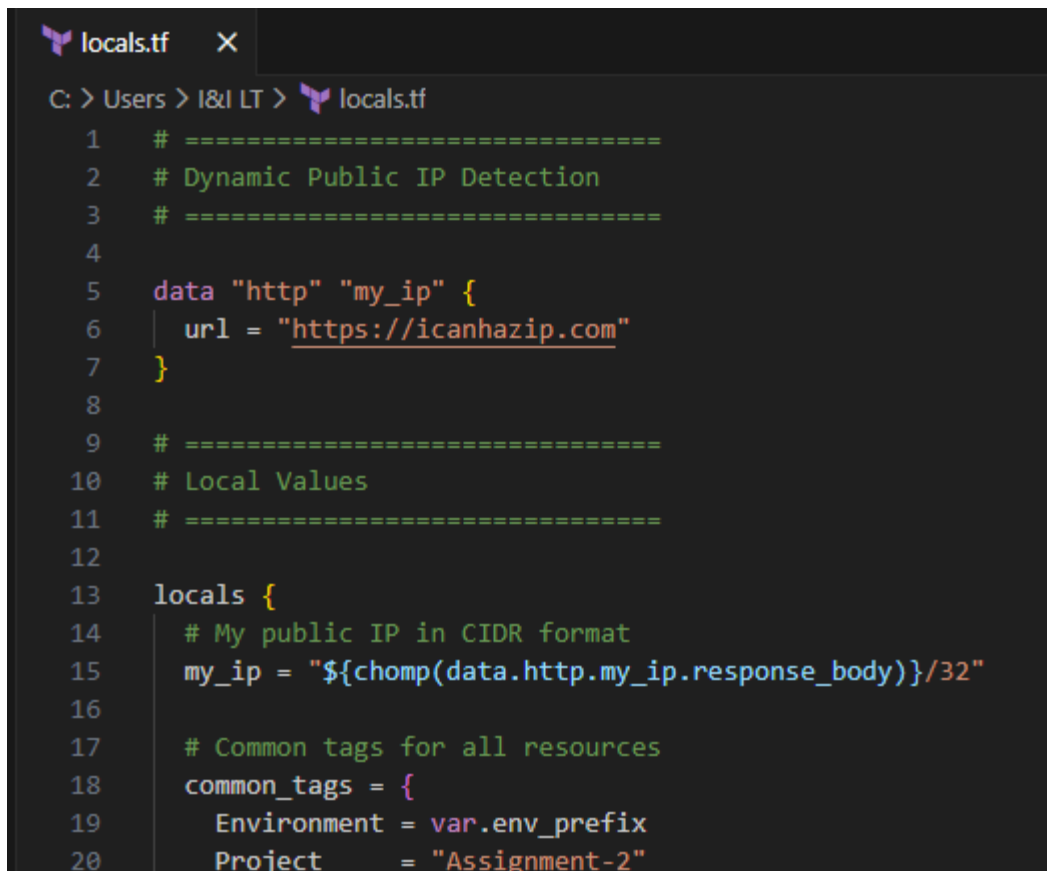
Create locals.tf with:

- Dynamic IP detection for my\_ip
- Resource naming conventions
- Common tags
- Backend server configurations

Open file

```
PS C:\Users\I&I LT> code locals.tf
PS C:\Users\I&I LT> 
```

Type the content

A screenshot of a code editor window titled 'locals.tf'. The editor shows the following content:

```
1  # =====
2  # Dynamic Public IP Detection
3  # =====
4
5  data "http" "my_ip" {
6    url = "https://icanhazip.com"
7  }
8
9  # =====
10 # Local Values
11 # =====
12
13 locals {
14   # My public IP in CIDR format
15   my_ip = "${chomp(data.http.my_ip.response_body)}/32"
16
17   # Common tags for all resources
18   common_tags = {
19     Environment = var.env_prefix
20     Project     = "Assignment-2"
```

Now verify the content

```

PS C:\Users\I&I LT> type .\locals.tf
# =====
# Dynamic Public IP Detection
# =====

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

# =====
# Local Values
# =====

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

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

  # Resource naming
  name_prefix = "${var.env_prefix}-assignment2"

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

```

\*\*\*\*\*

## Part 2: Webserver Module

### 2.1 – Webserver Module

Open the file

```

PS C:\Users\I&I LT> code modules\webserver\variables.tf
PS C:\Users\I&I LT> code modules\webserver\main.tf
PS C:\Users\I&I LT> code modules\webserver\outputs.tf

```

Type the content for each

```

variables.tf  main.tf  outputs.tf X
C: > Users > I&I LT > modules > webserver > outputs.tf
1  output "instance_id" {
2    description = "ID of the EC2 instance"
3    value       = aws_instance.this.id
4  }
5
6  output "public_ip" {
7    description = "Public IP address of the instance"
8    value       = aws_instance.this.public_ip
9  }

```

Verify the content

## Variables.tf

```
PS C:\Users\I&I LT> type .\modules\webserver\variables.tf
variable "env_prefix" {
  description = "Environment prefix for resource naming"
  type        = string
}

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

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

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

variable "availability_zone" {
  description = "Availability zone for the instance"
  type        = string
}
```

## Main.tf

```
PS C:\Users\I&I LT> type .\modules\webserver\main.tf
# =====
# Latest Amazon Linux 2023 AMI
# =====

data "aws_ami" "amazon_linux_2023" {
  most_recent = true
  owners      = ["amazon"]

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

# =====
# Key Pair (Unique per Instance)
# =====

resource "aws_key_pair" "this" {
  key_name   = "${var.env_prefix}-${var.instance_name}-${var.instance_suffix}-key"
  public_key = file(var.public_key)

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

## Output.tf

```
PS C:\Users\I&I LT> type .\modules\webserver\outputs.tf
output "instance_id" {
  description = "ID of the EC2 instance"
  value       = aws_instance.this.id
}

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

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

## 2.2 Module Usage

In root main.tf, instantiate the webserver module for:

1. One Nginx server (using nginx-setup.sh)
2. Three backend servers (web-1, web-2, web-3 using apache-setup.sh)

Open the file

```
PS C:\Users\I&I LT> code main.tf
PS C:\Users\I&I LT> 
```

Type the content

```
main.tf x
C: > Users > I&I LT > main.tf
1  # =====
2  # NGINX SERVER (Reverse Proxy)
3  # =====
4
5  module "nginx_server" {
6      source          = "../modules/webserver"
7
8      env_prefix       = var.env_prefix
9      instance_name    = "nginx-proxy"
10     instance_type    = var.instance_type
11     availability_zone = var.availability_zone
12
13     vpc_id            = module.networking.vpc_id
14     subnet_id         = module.networking.subnet_id
15     security_group_id = module.security.nginx_sg_id
16
17     public_key        = var.public_key
18     script_path       = "../scripts/nginx-setup.sh"
19     instance_suffix   = "nginx"
20 }
```

Verify the content

```

PS C:\Users\I&I LT> type .\main.tf
# =====
# NGINX SERVER (Reverse Proxy)
# =====

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
}

```

\*\*\*\*\*

## Part 3: Server Configuration Scripts

### 3.1 Apache Backend Server Script

Open the file

```

● PS C:\Users\I&I LT> cd "C:\Users\I&I LT\Desktop\Assignment2\scripts"
● PS C:\Users\I&I LT\Desktop\Assignment2\scripts> notepad apache-setup.sh

```

Type the content

```

apache-setup.sh
File Edit View

#!/bin/bash
set -e

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

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

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

```

Verify the content

```

PS C:\Users\I&I LT\Desktop\Assignment2\scripts> Get-Content apache-setup.sh
#!/bin/bash
set -e

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

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

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

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

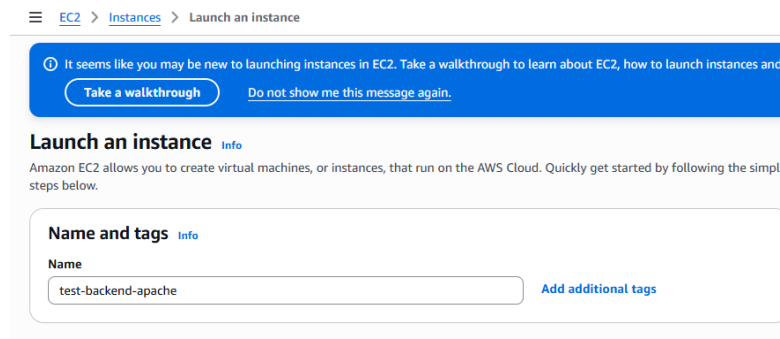
# -----
# Fetch instance metadata
# -----
PRIVATE_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \

```

## Test Script Independently

### Launch Test EC2 Instance

- Open **AWS Management Console** and navigate to **EC2 → Instances**.
- Click **Launch Instance**.
- **Name and Tags:**
  - Name: test-backend-apache



- **Choose Amazon Machine Image (AMI):**
  - Select **Amazon Linux 2023 AMI (x86\_64)**
- **Instance Type:**
  - Choose **t3.micro** (Free tier eligible)
- **Key Pair (login):**
  - Select existing key pair or create a new one
  - Download .pem file if creating new key
- **Network Settings:**
  - **VPC:** Select your backend VPC

- **Subnet:** Choose backend subnet (e.g., subnet-0e6477e4da4095d37)
  - **Auto-assign Public IP:** Enable
- **Firewall (Security Group):**
  - Allow **SSH (Port 22)** from your IP only
  - Allow **HTTP (Port 80)** from anywhere
  - Allow **HTTPS (Port 443)** from anywhere (optional)
- **Storage:**
  - Default 8 GiB (gp3) is sufficient

▼ Summary

Number of instances | Info

1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.9.2...[read more](#)  
ami-00ca570c1b6d79f36

Virtual server type (instance type)  
t3.micro

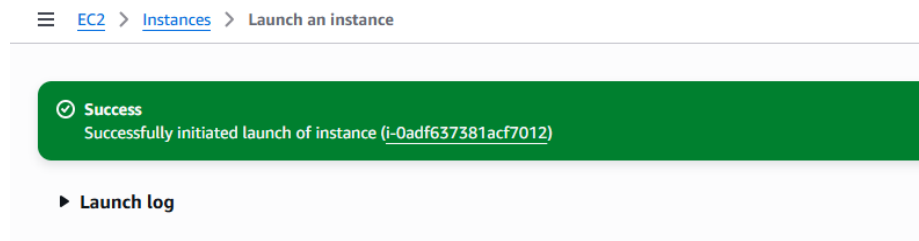
Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

Cancel Launch instance

[Preview code](#)

- Review all settings and click **Launch Instance**



- Verify the instance is in **running** state in the EC2 dashboard

Instances (1/1) Info Last updated 23 minutes ago [Refresh](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) [All states](#) [1](#) [Settings](#)

<input checked="" type="checkbox"/>	Name <a href="#">↗</a>	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/>	test-backend-...	i-0adf637381acf7012	Running <a href="#">🔍</a>	t3.micro	3/3 checks passed <a href="#">🔍</a>	<a href="#">View alarms +</a>	ap-south-1a

## Copy Script to EC2 Instance

- Navigate to your script folder: `cd "C:\Users\I&I LT\Desktop\Assignment2\scripts"` Copy the script to your EC2 backend instance using scp:
- `scp -i ".\assignment2-key.pem" ".\apache-setup.sh" ec2-user@<PUBLIC_IP>:~/`

- Replace <PUBLIC\_IP> with the public IP of your EC2 instance (e.g., 13.126.134.41).

```
PS C:\Users\I&I LT\Desktop\Assignment2\scripts> scp -i "C:\Users\I&I LT\Desktop\Assignment2\scripts\assignment2-key.pem" "C:\Users\I&I LT\Desktop\Assignment2\scripts\apache-setup.sh" ec2-user@13.126.134.41:~/
100% 3138 20.3KB/s 00:00
PS C:\Users\I&I LT\Desktop\Assignment2\scripts> ssh -i "C:\Users\I&I LT\Desktop\Assignment2\scripts\assignment2-key.pem" ec2-user@13.126.134.41
# Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
Last login: Sat Dec 27 14:40:21 2025 from 203.215.174.10
[ec2-user@ip-172-31-44-126 ~]$ chmod +x apache-setup.sh
[ec2-user@ip-172-31-44-126 ~]$ sudo ./apache-setup.sh
```

- Connect to EC2 Instance via SSH
- Make Script Executable: `chmod +x apache-setup.sh`
- Run the Apache Setup Script: `sudo ./apache-setup.sh`

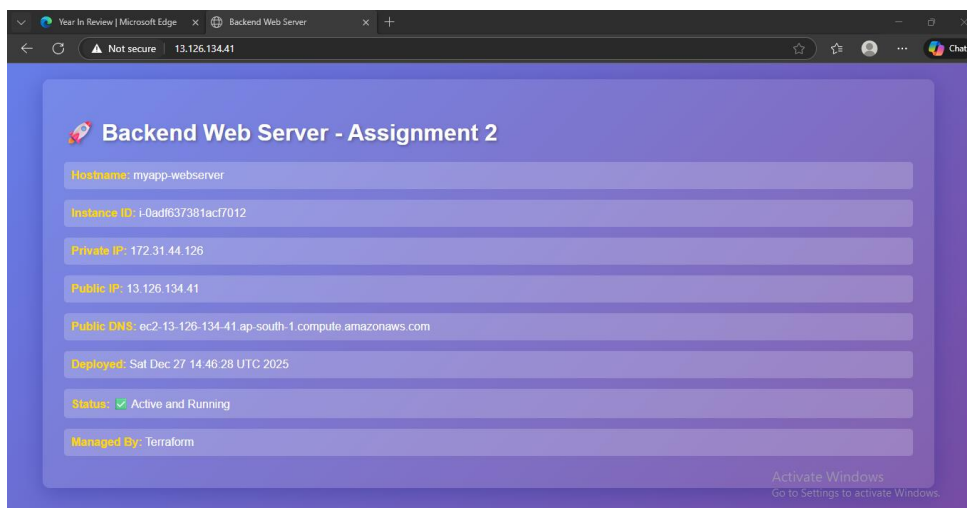
```
File Edit Selection View Go Run ... Search
C:\Users\I&I LT\Desktop\Assignment2\scripts > apache-setup.sh
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\I&I LT\Desktop\Assignment2\scripts> ssh -i "C:\Users\I&I LT\Desktop\Assignment2\scripts\assignment2-key.pem" ec2-user@13.126.134.41
Installing : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 11/13
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 12/13
Installing : httpd-2.4.65-1.amzn2023.0.2.x86_64 13/13
Running scriptlet: httpd-2.4.65-1.amzn2023.0.2.x86_64 13/13
Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64 1/13
Verifying : apr-util-1.6.3-1.amzn2023.0.2.x86_64 2/13
Verifying : apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64 3/13
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64 4/13
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 5/13
Verifying : httpd-2.4.65-1.amzn2023.0.2.x86_64 6/13
Verifying : httpd-core-2.4.65-1.amzn2023.0.2.x86_64 7/13
Verifying : httpdfilesystem-2.4.65-1.amzn2023.0.2.noarch 8/13
Verifying : httpd-tools-2.4.65-1.amzn2023.0.2.x86_64 9/13
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 10/13
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 11/13
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 12/13
Verifying : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 13/13

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64      apr-util-1.6.3-1.amzn2023.0.2.x86_64      apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64      generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      httpd-2.4.65-1.amzn2023.0.2.x86_64
httpd-core-2.4.65-1.amzn2023.0.2.x86_64      httpdfilesystem-2.4.65-1.amzn2023.0.2.noarch      httpd-tools-2.4.65-1.amzn2023.0.2.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64      mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.27-1.amzn2023.0.3.x86_64
mod_lua-2.4.65-1.amzn2023.0.2.x86_64

Complete!
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service + /usr/lib/systemd/system/httpd.service.
Apache setup completed successfully!
[ec2-user@ip-172-31-44-126 ~]$
```

## Test the Webpage

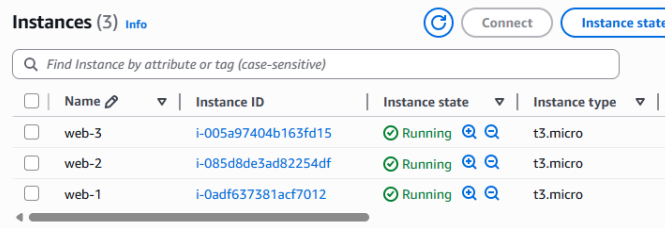
1. Open a **browser**.
2. Enter the **public IP** of your EC2 instance: [Backend Web Server](#)





## 3.2 Nginx Server Setup Script

- Connect to EC2 via SSH using the .pem key.



The screenshot shows the AWS Management Console 'Instances' page. It displays a table with three instances: 'web-3', 'web-2', and 'web-1'. All instances are in the 'Running' state and are of type 't3.micro'. The table includes columns for Name, Instance ID, Instance state, and Instance type. There are also buttons for 'Connect' and 'Instance state'.

	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	web-3	i-005a97404b163fd15	Running	t3.micro
<input type="checkbox"/>	web-2	i-085d8de3ad82254df	Running	t3.micro
<input type="checkbox"/>	web-1	i-0adf637381acf7012	Running	t3.micro

- Stop conflicting services (e.g., Apache) on port 80:
- Create SSL directories and generate a self-signed certificate:
- Create cache directory:

```
[ec2-user@myapp-webserver ~]$ sudo systemctl start nginx
sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Sun 2025-12-28 05:14:30 UTC; 55ms ago
     Process: 64383 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 64384 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 64385 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 64386 (nginx)
     Tasks: 3 (limit: 1067)
    Memory: 3.2M
       CPU: 55ms
    CGroup: /system.slice/nginx.service
            └─64386 "nginx: master process /usr/sbin/nginx"
              └─64387 "nginx: worker process"
                └─64388 "nginx: worker process"

Dec 28 05:14:30 myapp-webserver systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Dec 28 05:14:30 myapp-webserver nginx[64384]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Dec 28 05:14:30 myapp-webserver nginx[64384]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Dec 28 05:14:30 myapp-webserver systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@myapp-webserver ~]$
```

Test Nginx configuration:

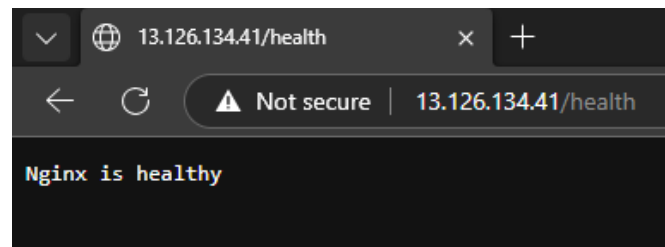
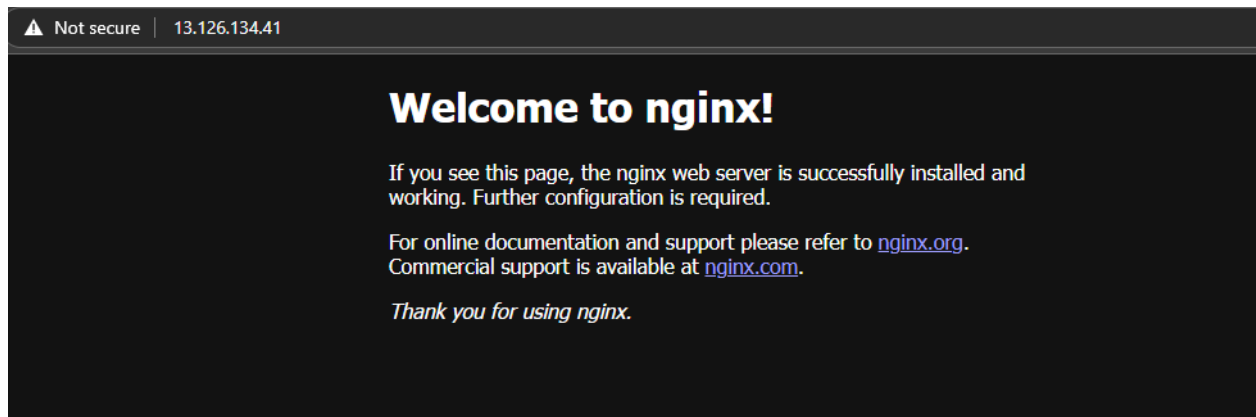
```
[ec2-user@myapp-webserver ~]$ sudo nginx -t
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:61
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:62
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@myapp-webserver ~]$
```

Verify endpoints:

- From EC2:

```
[ec2-user@myapp-webserver ~]$ curl -k https://localhost/
curl http://localhost/health
Nginx is healthy
Nginx is healthy
[ec2-user@myapp-webserver ~]$
```

- From browser:



\*\*\*\*\*

## Part 4: Infrastructure Deployment

### 4.1 Initial Deployment

Generate a new SSH key

```
I&I LT@Kainat-Shakeel MINGW64 ~ (master)
$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/assignment2-key
Generating public/private rsa key pair.
Enter passphrase for "/c/Users/I&I LT/.ssh/assignment2-key" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/I&I LT/.ssh/assignment2-key
Your public key has been saved in /c/Users/I&I LT/.ssh/assignment2-key.pub
The key fingerprint is:
SHA256:XZ3J5Vmxflp14PP1LME0RUEqA3GR4kqa/ZlKqN3oMxI I&I LT@Kainat-Shakeel
The key's randomart image is:
+----[RSA 4096]-----+
|          o.o. *O* |
|         .O. Bo* |
|        . .O. .@. + |
|       . O .O .== |
|      = S . . * |
|     E O.O . +. |
|    . . . O . |
|   .OO+ + |
|  .O+OO. |
+----[SHA256]-----+
```

## Initialize Terraform

```
I&I LT@Kainat-Shakeel MINGW64 ~ (master)
$ cd "C:\Users\I&I LT\Desktop\Assignment2"
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.

## Validate Terraform configuration

```
I&I LT@Kainat-Shakeel MINGW64 ~/Desktop/Assignment2 (master)
$ terraform validate
Success! The configuration is valid.
```

## Plan the

## deployment

```
I&I LT@Kainat-Shakeel MINGW64 ~/Desktop/Assignment2 (master)
$ terraform plan
aws_key_pair.assignment2_key: Refreshing state... [id=assignment2-key]
aws_security_group.web_sg: Refreshing state... [id=sg-016dc271b9a9179a1]

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

Terraform will perform the following actions:

# aws_instance.nginx will be created
+ resource "aws_instance" "nginx" {
  + ami                        = "ami-0de53d8956e8dcf80"
  + arn                       = (known after apply)
  + associate_public_ip_address = true
  + availability_zone         = (known after apply)
  + disable_api_stop          = (known after apply)
  + disable_api_termination   = (known after apply)
  + ebs_optimized              = (known after apply)
  + enable_primary_ipv6       = (known after apply)
  + force_destroy              = false
  + get_password_data          = false
  + host_id                   = (known after apply)
  + host_resource_group_arn    = (known after apply)
  + iam_instance_profile       = (known after apply)
  + id                        = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle         = (known after apply)
  + instance_state             = (known after apply)
  + instance_type              = "t3.micro"
  + ipv6_address_count         = (known after apply)
  + ipv6_addresses             = (known after apply)
  + key_name                   = "assignment2-key"
  + monitoring                 = (known after apply)
  + outpost_arn                = (known after apply)
  + password_data              = (known after apply)
  + placement_group            = (known after apply)
  + placement_group_id         = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns                = (known after apply)
  + private_ip                 = (known after apply)
  + public_dns                 = (known after apply)
  + public_ip                  = (known after apply)
  + region                     = "ap-south-1"
  + secondary_private_ips      = (known after apply)
  + security_groups             = [
```

## Apply the deployment

```
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

aws_key_pair.assignment2_key: Creating...
module.nginx_server.aws_instance.this: Creating...
module.backend_server.aws_instance.this: Creating...
aws_key_pair.assignment2_key: Creation complete after 1s [id=assignment2-key]
module.nginx_server.aws_instance.this: Still creating... [00m10s elapsed]
module.backend_server.aws_instance.this: Still creating... [00m10s elapsed]
module.nginx_server.aws_instance.this: Creation complete after 14s [id=i-05936bfb73fda8a7]
module.backend_server.aws_instance.this: Creation complete after 14s [id=i-08a4ccdb1457eb099]

Warning: Value is base64 encoded

  with module.nginx_server.aws_instance.this,
  on modules/webserver/main.tf line 25, in resource "aws_instance" "this":
  25:   user_data = file(var.script_path)

The value is base64 encoded. If you want to use base64 encoding, please use the user_data_base64 argument. user_data attribute is set as cleartext in state

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

Outputs:

backend_public_ip = "15.206.72.20"
nginx_public_ip = "43.204.23.253"
```

## 4.2 Output Configuration

### Apply Terraform

```
I&I LT@Kainat-Shakeel MINGW64 ~/Desktop/Assignment2 (master)
$ terraform apply
module.nginx_server.data.aws_ami.amazon_linux_2023: Reading...
aws_key_pair.assignment2_key: Refreshing state... [id=assignment2-key]
module.backend_server.data.aws_ami.amazon_linux_2023: Reading...
module.networking.aws_vpc.this: Refreshing state... [id=vpc-09bbc62788d0b971b]
module.nginx_server.data.aws_ami.amazon_linux_2023: Read complete after 0s [id=ami-01f662fe6044dfacb]
module.networking.aws_internet_gateway.this: Refreshing state... [id=igw-002ccb53120318e26]
module.networking.aws_subnet.this: Refreshing state... [id=subnet-0a3c420710935bb2c]
module.backend_server.data.aws_ami.amazon_linux_2023: Read complete after 1s [id=ami-01f662fe6044dfacb]
module.security.aws_security_group.backend: Refreshing state... [id=sg-0765585a8ea78bac1]
module.security.aws_security_group.nginx: Refreshing state... [id=sg-03bfb5e5eef166923]
module.networking.aws_route_table.this: Refreshing state... [id=rtb-04e04a84d822616c4]
module.backend_server.aws_instance.this: Refreshing state... [id=i-08a4ccdb1457eb099]
module.nginx_server.aws_instance.this: Refreshing state... [id=i-05936bfb73fda8a7]
module.networking.aws_route_table_association.this: Refreshing state... [id=rtbassoc-04d193779e850b74b]

No changes. Your infrastructure matches the configuration.

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

Warning: Value is base64 encoded

  with module.nginx_server.aws_instance.this,
  on modules/webserver/main.tf line 25, in resource "aws_instance" "this":
  25:   user_data = file(var.script_path)

The value is base64 encoded. If you want to use base64 encoding, please use the user_data_base64 argument. user_data attribute is set as cleartext in state

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

Outputs:

backend_public_ip = "15.206.72.20"
nginx_public_ip = "43.204.23.253"
```

### Check Outputs

```
I&I LT@Kainat-Shakeel MINGW64 ~/Desktop/Assignment2 (master)
$ terraform output
backend_public_ip = "15.206.72.20"
nginx_public_ip = "43.204.23.253"

I&I LT@Kainat-Shakeel MINGW64 ~/Desktop/Assignment2 (master)
$ terraform output -json > outputs.json
```

### Step 4.3: AWS Console Verification

aws\_vpc

**Your VPCs (1/2)** [Info](#)

Last updated 1 minute ago [Actions](#)

Find VPCs by attribute or tag

	Name	VPC ID	State	Encryption c...	Encryption contr
<input type="checkbox"/>	id <a href="#">🔗</a>	<a href="#">vpc-053a810b4de844c0f</a>	Available	–	–
<input checked="" type="checkbox"/>	assignment2-vpc	<a href="#">vpc-09bbc62788d0b971b</a>	Available	–	–

aws\_subnet

**Subnets (1/4)** [Info](#)

Last updated 6 minutes ago [Actions](#) [Create subnet](#)

Find subnets by attribute or tag

	Name	Subnet ID	State	VPC	Block Publ
<input type="checkbox"/>	–	<a href="#">subnet-0168f4ef4f561126f</a>	Available	<a href="#">vpc-053a810b4de844c0f</a>   <a href="#">id</a>	Off
<input type="checkbox"/>	–	<a href="#">subnet-0e6477e4da4095d37</a>	Available	<a href="#">vpc-053a810b4de844c0f</a>   <a href="#">id</a>	Off
<input checked="" type="checkbox"/>	assignment2-public-subnet	<a href="#">subnet-0a3c420710935bb2c</a>	Available	<a href="#">vpc-09bbc62788d0b971b</a>   <a href="#">assi...</a>	Off
<input type="checkbox"/>	–	<a href="#">subnet-063a6ea3a63804836</a>	Available	<a href="#">vpc-053a810b4de844c0f</a>   <a href="#">id</a>	Off

subnet-0a3c420710935bb2c / assignment2-public-subnet

aws\_security\_groups

**Internet gateways (1/2)** [Info](#)

[Actions](#) [Create internet gateway](#)

Find internet gateways by attribute or tag

	Name	Internet gateway ID	State	VPC ID
<input checked="" type="checkbox"/>	assignment2-igw	<a href="#">igw-002ccb53120318e26</a>	Attached	<a href="#">vpc-09bbc62788d0b971b</a>   <a href="#">assignment...</a>
<input type="checkbox"/>	–	<a href="#">igw-02d1e43a264008cd7</a>	Attached	<a href="#">vpc-053a810b4de844c0f</a>   <a href="#">id</a>

aws\_instances

**Route tables (1/3)** [Info](#)

Last updated 10 minutes ago [Actions](#) [Create route table](#)

Find route tables by attribute or tag

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input type="checkbox"/>	–	<a href="#">rtb-030c97f995b4ad928</a>	–	–	Yes	<a href="#">vpc-0</a>
<input checked="" type="checkbox"/>	assignment2-public-rt	<a href="#">rtb-04e04a84d822616c4</a>	<a href="#">subnet-0a3c420710935b...</a>	–	No	<a href="#">vpc-0</a>
<input type="checkbox"/>	–	<a href="#">rtb-0f72713d3b1391f02</a>	–	–	Yes	<a href="#">vpc-0</a>

\*\*\*\*\*

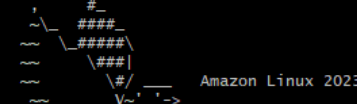
## Part 5: Nginx Configuration & Testing

## 5.1 Update Nginx Backend Configuration

## SSH into the Nginx

Server

```
t&I LT@kainat-Shakeel MINGW64 ~/Desktop/Assignment2 (master)
$ ssh -i ./scripts/assignment2-key.pem ec2-user@43.204.23.253
The authenticity of host '43.204.23.253 (43.204.23.253)' can't be established.
ED25519 key fingerprint is SHA256:rE7Sy1bwOq1EUe3dgbcy4ThhjAzPFiqmB667HyeldHs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '43.204.23.253' (ED25519) to the list of known hosts.
```



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

Amazon Linux 2023 (ECS Optimized)

```
For documentation, visit http://aws.amazon.com/documentation/ec2
[ec2-user@ip-10-0-1-245 ~]$
```

## nginx\_conf\_updated

```
ec2-user@ip-10-0-1-245:~  
GNU nano 8.3 /etc/nginx/nginx.conf  
# * Official English Documentation: http://nginx.org/en/docs/  
# * Official Russian Documentation: http://nginx.org/ru/docs/  
  
user nginx;  
worker_processes auto;  
error_log /var/log/nginx/error.log notice;  
pid /run/nginx.pid;  
  
# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.  
include /usr/share/nginx/modules/*.conf;  
  
events {  
    worker_connections 1024;  
}  
  
http {  
    # Define backend servers  
    upstream backend_servers {  
        server 10.0.1.101:80; # web-1 private IP  
        server 10.0.1.102:80; # web-2 private IP  
        server 10.0.1.103:80 backup; # web-3 private IP  
    }  
  
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '  
                    '$status $body_bytes_sent "$http_referer" '  
                    '"$http_user_agent" "$http_x_forwarded_for"';  
    access_log /var/log/nginx/access.log main;  
  
    sendfile            on;  
    tcp_nopush          on;  
    keepalive_timeout   65;  
    types_hash_max_size 4096;  
  
    include              /etc/nginx/mime.types;  
    default_type         application/octet-stream;  
  
    # Load modular configuration files from the /etc/nginx/conf.d directory  
    include /etc/nginx/conf.d/*.conf;  
  
    # Main server block  
    server {  
        listen            80;  
        listen            [::]:80;  
        server_name        _;  
        root               /usr/share/nginx/html;  
    }  
}
```

test the Nginx configuration

```
[ec2-user@ip-10-0-1-245 ~]$ sudo nano /etc/nginx/nginx.conf
[ec2-user@ip-10-0-1-245 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-1-245 ~]$ |
```

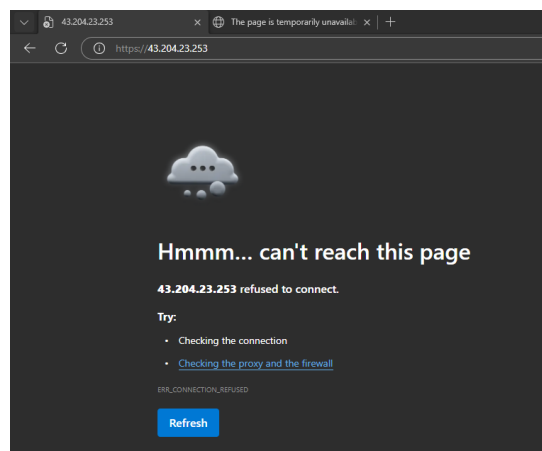
restart Nginx

```
ec2-user@ip-10-0-1-245:~
For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 15:12:12 2025 from 203.215.174.10
[ec2-user@ip-10-0-1-245 ~]$ sudo nano /etc/nginx/nginx.conf
[ec2-user@ip-10-0-1-245 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-1-245 ~]$ sudo systemctl restart nginx
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 Sun 2025-12-28 16:25:57 UTC; 56ms ago
     Process: 204684 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 204685 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 204686 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 204687 (nginx)
       Tasks: 3 (limit: 1067)
      Memory: 3.3M
         CPU: 59ms
    CGroup: /system.slice/nginx.service
            └─204687 "nginx: master process /usr/sbin/nginx"
              └─204688 "nginx: worker process"
                └─204689 "nginx: worker process"

Dec 28 16:25:57 ip-10-0-1-245.ap-south-1.compute.internal systemd[1]: Starting nginx[204685]: nginx:
Dec 28 16:25:57 ip-10-0-1-245.ap-south-1.compute.internal nginx[204685]: nginx:
Dec 28 16:25:57 ip-10-0-1-245.ap-south-1.compute.internal systemd[1]: Started nginx[204685]: nginx:
lines 1-19/19 (END)...skipping...
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Sun 2025-12-28 16:25:57 UTC; 56ms ago
     Process: 204684 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 204685 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 204686 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 204687 (nginx)
       Tasks: 3 (limit: 1067)
      Memory: 3.3M
         CPU: 59ms
    CGroup: /system.slice/nginx.service
            └─204687 "nginx: master process /usr/sbin/nginx"
              └─204688 "nginx: worker process"
                └─204689 "nginx: worker process"
```

## 5.2 Test Load Balancing

browser security warning





showing web-1 content

```
[ec2-user@ip-10-0-10-75 ~]$  
[ec2-user@ip-10-0-10-75 ~]$ curl -k -s "https://localhost/?r=$RANDOM" | grep -oE 'i-[0-9a-f]+' | head -n 1  
i-0f0428b79f1efc916  
[ec2-user@ip-10-0-10-75 ~]$
```

showing web-2 content

## WEB-2

Hostname: ip-10-0-10-107.me-central-1.compute.internal  
Instance ID: i-0ce73e90b08689349  
Time: Thu Dec 25 16:16:46 UTC 2025

multiple reloads showing alternation

```
i-0cc21e871d1d28299  
[ec2-user@ip-10-0-10-75 ~]$ for i in {1..12}; do curl -k -s "https://localhost/?r=$RANDOM" | grep -oE 'i-[0-9a-f]+' | head -n 1; done  
i-0f0428b79f1efc916  
i-0ce73e90b08689349  
i-0f0428b79f1efc916  
i-0ce73e90b08689349  
i-0f0428b79f1efc916  
i-0ce73e90b08689349  
i-0f0428b79f1efc916  
i-0ce73e90b08689349  
i-0f0428b79f1efc916  
i-0ce73e90b08689349  
i-0f0428b79f1efc916  
i-0ce73e90b08689349  
i-0f0428b79f1efc916  
[ec2-user@ip-10-0-10-75 ~]$
```

## 5.3 Test Cache Functionality

first request - MISS

```
[ec2-user@ip-10-0-10-75 ~]$ curl -k -I "https://localhost/?x=miss1" | grep -i "x-cache-status"  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
             Dload  Upload    Total   Spent    Left   Speed  
0   159    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0  
x-cache-status: MISS
```

second request – HIT

```
[ec2-user@ip-10-0-10-75 ~]$ curl -k -I "https://localhost/?x=miss1" | grep -i "x-cache-status"  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
             Dload  Upload    Total   Spent    Left   Speed  
0   159    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0  
x-cache-status: HIT
```

cache folder contents

```
x-cache-status: HIT  
[ec2-user@ip-10-0-10-75 ~]$ sudo ls -lah /var/cache/nginx  
-type f | head  
sudo find /var/cache/nginx -type f | head  
total 0  
drwx-----. 17 nginx root 141 Dec 25 15:12 .  
drwxr-xr-x. 9 root root 101 Dec 24 17:00 ..  
drwx-----. 5 nginx nginx 36 Dec 25 14:55 0  
drwx-----. 8 nginx nginx 66 Dec 25 15:12 1  
drwx-----. 7 nginx nginx 56 Dec 25 14:55 2  
drwx-----. 6 nginx nginx 46 Dec 25 15:12 4  
drwx-----. 9 nginx nginx 76 Dec 25 15:12 5  
drwx-----. 5 nginx nginx 36 Dec 25 15:12 6  
drwx-----. 7 nginx nginx 56 Dec 25 15:12 7  
drwx-----. 4 nginx nginx 26 Dec 25 14:55 8  
drwx-----. 4 nginx nginx 26 Dec 25 15:12 9  
drwx-----. 6 nginx nginx 46 Dec 25 14:55 a  
drwx-----. 8 nginx nginx 66 Dec 25 16:41 b  
drwx-----. 8 nginx nginx 66 Dec 25 15:12 c  
drwx-----. 6 nginx nginx 46 Dec 25 15:12 d  
drwx-----. 3 nginx nginx 16 Dec 25 15:12 e  
drwx-----. 5 nginx nginx 36 Dec 25 14:54 f  
[ec2-user@ip-10-0-10-75 ~]$
```



access log showing cache status

```
Chrome/143.0.0.0 Safari/537.36" "-" Cache:MISS
39.62.158.42 - - [25/Dec/2025:16:33:48 +0000] "GET /?r=2 HTTP/2.0" 200 153 "-" "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
39.62.158.42 - - [25/Dec/2025:16:33:49 +0000] "GET /?r=2 HTTP/2.0" 200 723 "-" "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
39.62.158.42 - - [25/Dec/2025:16:33:49 +0000] "GET /favicon.ico HTTP/2.0" 404 172 "https://51.112.45.72/?r=2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:MISS
39.62.158.42 - - [25/Dec/2025:16:33:49 +0000] "GET /?r=2 HTTP/2.0" 200 723 "-" "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
39.62.158.42 - - [25/Dec/2025:16:33:49 +0000] "GET /favicon.ico HTTP/2.0" 404 172 "https://51.112.45.72/?r=2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cache:MISS
127.0.0.1 - - [25/Dec/2025:16:36:18 +0000] "HEAD /?r=cachetest1 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:BYPASS
127.0.0.1 - - [25/Dec/2025:16:39:52 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MIT
127.0.0.1 - - [25/Dec/2025:16:40:05 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MIT
91.224.92.34 - - [25/Dec/2025:16:40:52 +0000] "GET / HTTP/1.1" 301 169 "-" "-" Cache:-
127.0.0.1 - - [25/Dec/2025:16:41:52 +0000] "HEAD /?r=miss1 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MISS
127.0.0.1 - - [25/Dec/2025:16:42:03 +0000] "HEAD /?r=miss1 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MIT
5.281.188.219 - - [25/Dec/2025:16:43:20 +0000] "POST /boaforn/admin/forlogin HTTP/1.1" 301 169 "http://51.112.45.72:80/admin/login.asp" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0" "-" Cache:-
5.181.188.219 - - [25/Dec/2025:16:43:20 +0000] "" 400 0 "-" "-" Cache:-
194.187.176.192 - - [25/Dec/2025:16:50:52 +0000] "GET / HTTP/1.1" 400 255 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/65.0" "-" Cache:-
194.187.176.16 - - [25/Dec/2025:16:50:52 +0000] "GET /favicon.ico HTTP/1.1" 400 255 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/65.0" "-" Cache:-
127.0.0.1 - - [25/Dec/2025:16:54:36 +0000] "HEAD /?r=123 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MISS
127.0.0.1 - - [25/Dec/2025:16:54:36 +0000] "HEAD /?r=123 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MIT
1ec2-user@ip-10-0-10-78 ~$
```

## 5.4 Test High Availability

web-1 Apache stopped

```
sudo systemctl status httpd --no-pager
[ec2-user@ip-10-0-10-243 ~]$ sudo systemctl stop httpd
[ec2-user@ip-10-0-10-243 ~]$ sudo systemctl status httpd --no-pager
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: inactive (dead) since Thu 2025-12-25 17:00:09 UTC; 3min 54s ago
   Duration: 3h 17min 20.275s
   Docs: man:httpd.service(8)
  Process: 83858 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 83858 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 68; Idle/Busy workers 100/0;Requests/sec: 0.00575; Bytes served/sec: 9 B/sec"
    CPU: 9.538s
```

web-2 Apache stopped

```
[ec2-user@ip-10-0-10-107 ~]$ sudo systemctl stop httpd
sud[ec2-user@ip-10-0-10-107 ~]$ sudo systemctl status httpd --no-pager
o httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: inactive (dead) since Thu 2025-12-25 17:08:07 UTC; 14s ago
   Duration: 3h 21min 4.638s
   Docs: man:httpd.service(8)
  Process: 83259 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
 Main PID: 83259 (code=exited, status=0/SUCCESS)
   Status: "Total requests: 68; Idle/Busy workers 100/0;Requests/sec: 0.00564; Bytes served/sec: 9 B/sec"
    CPU: 9.625s
```

web-3 serving traffic

## WEB-3 (BACKUP)

Hostname: ip-10-0-10-98.me-central-1.compute.internal

Instance ID: i-0cc21e871d1d28299

error log showing backend failures

```
125/12/25 15:54:58 [notice] 64188#64188: signal 29 (SIGIO) received
125/12/25 16:35:49 [warn] 65463#65463: the "listen ... http2" directive is deprecated, use the "http2" directive instead
125/12/25 17:10:51 [error] 64189#64189: *127 connect() failed (111: Connection refused) while connecting to upstream, client: 39.10.107.80, host: "51.112.45.72"
125/12/25 17:10:51 [warn] 64189#64189: *127 upstream server temporarily disabled while connecting to upstream, client: 39.10.107.80, host: "51.112.45.72"
125/12/25 17:10:51 [error] 64189#64189: *127 connect() failed (111: Connection refused) while connecting to upstream, client: 39.10.107.80, host: "51.112.45.72"
125/12/25 17:10:51 [warn] 64189#64189: *127 upstream server temporarily disabled while connecting to upstream, client: 39.10.107.80, host: "51.112.45.72"
```

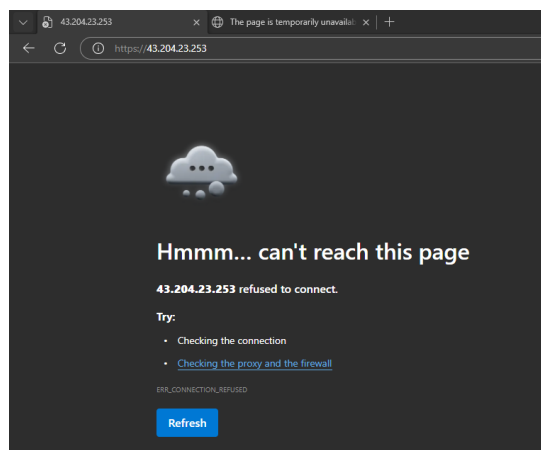
all services back online

```
[ec2-user@ip-10-0-10-243 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-10-243 ~]$ sudo systemctl status httpd --no-pager
• httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2025-12-25 17:15:52 UTC; 27s ago
     Docs: man:httpd.service(8)
  Main PID: 93130 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:
      Tasks: 177 (limit: 1067)
     Memory: 13.3M
        CPU: 86ms
    CGroup: /system.slice/httpd.service
            └─93130 /usr/sbin/httpd -DFOREGROUND
               └─93131 /usr/sbin/httpd -DFOREGROUND
                  └─93132 /usr/sbin/httpd -DFOREGROUND
                     └─93133 /usr/sbin/httpd -DFOREGROUND
                        └─93139 /usr/sbin/httpd -DFOREGROUND
```

```
[ec2-user@ip-10-0-10-107 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-10-107 ~]$ sudo systemctl status httpd --no-pager
• httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2025-12-25 17:17:31 UTC; 14s ago
     Docs: man:httpd.service(8)
  Main PID: 92565 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:
      Tasks: 177 (limit: 1067)
     Memory: 13.3M
        CPU: 75ms
    CGroup: /system.slice/httpd.service
            └─92565 /usr/sbin/httpd -DFOREGROUND
               └─92566 /usr/sbin/httpd -DFOREGROUND
                  └─92567 /usr/sbin/httpd -DFOREGROUND
                     └─92568 /usr/sbin/httpd -DFOREGROUND
                        └─92569 /usr/sbin/httpd -DFOREGROUND
```

## 5.5 Security & Performance Analysis

certificate details



response headers showing security headers

```
[ec2-user@ip-10-0-10-75 ~]$ curl -k -I https://localhost/ | egrep -i "x-content-type-options|x-xss-protection|:"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total      Spent    Left     Speed
0   167    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
```

301 redirect test

```
[ec2-user@ip-10-0-10-75 ~]$ curl -I http://localhost/ | head -n 5
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total      Spent    Left     Speed
0   169    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Thu, 25 Dec 2025 18:06:19 GMT
Content-Type: text/html
Content-Length: 169
```

error log review

```
2025/12/25 15:53:57 [notice] 64188#64188: start cache loader process 64192
2025/12/25 15:54:20 [warn] 64197#64197: the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:36
2025/12/25 15:54:58 [notice] 64192#64192: http file cache: /var/cache/nginx 0.219M, bsize: 4096
2025/12/25 15:54:58 [notice] 64188#64188: signal 17 (SIGCHLD) received from 64192
2025/12/25 15:54:58 [notice] 64188#64188: cache loader process 64192 exited with code 0
2025/12/25 15:54:58 [notice] 64188#64188: signal 29 (SIGIO) received
2025/12/25 16:35:49 [warn] 65463#65463: the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:36
2025/12/25 17:10:51 [error] 64189#64189: *127 connect() failed (111: Connection refused) while connecting to upstream, client: 39.62.158.42, server: _, request: "GET / HTTP/1.1", host: "51.112.45.72"
2025/12/25 17:10:51 [warn] 64189#64189: *127 upstream server temporarily disabled while connecting to upstream, client: 39.62.158.42, server: _, request: "GET / HTTP/1.1", host: "51.112.45.72"
2025/12/25 17:10:51 [error] 64189#64189: *127 connect() failed (111: Connection refused) while connecting to upstream, client: 39.62.158.42, server: _, request: "GET / HTTP/1.1", host: "51.112.45.72"
2025/12/25 17:10:51 [warn] 64189#64189: *127 upstream server temporarily disabled while connecting to upstream, client: 39.62.158.42, server: _, request: "GET / HTTP/1.1", host: "51.112.45.72"
```

access log patterns

```
39.62.158.42 - - [25/Dec/2025:16:33:49 +0000] "GET /?r=2 HTTP/2.0" 200 723 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:BYPASS
39.62.158.42 - - [25/Dec/2025:16:33:49 +0000] "GET /favicon.ico HTTP/2.0" 404 172 "https://51.112.45.72/?r=2" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:MISS
127.0.0.1 - - [25/Dec/2025:16:36:10 +0000] "HEAD /?r=cachetest1 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:BYPASS
127.0.0.1 - - [25/Dec/2025:16:39:52 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
127.0.0.1 - - [25/Dec/2025:16:40:05 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
11.224.92.14 - - [25/Dec/2025:16:40:52 +0000] "GET / HTTP/1.1" 301 169 "-" "-" Cache:-
127.0.0.1 - - [25/Dec/2025:16:41:52 +0000] "HEAD /?x=miss1 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MISS
127.0.0.1 - - [25/Dec/2025:16:42:03 +0000] "HEAD /?x=miss1 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
3.181.188.219 - - [25/Dec/2025:16:43:20 +0000] "POST /boaform/admin/formLogin HTTP/1.1" 301 169 "http://51.112.45.72:80/admin/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:-
3.181.188.219 - - [25/Dec/2025:16:43:20 +0000] "" 400 0 "-" "-" Cache:-
39.187.176.192 - - [25/Dec/2025:16:50:52 +0000] "GET / HTTP/1.1" 400 255 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/71.0" "-" Cache:-
39.187.176.16 - - [25/Dec/2025:16:50:52 +0000] "GET /favicon.ico HTTP/1.1" 400 255 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/71.0" "-" Cache:-
127.0.0.1 - - [25/Dec/2025:16:54:36 +0000] "HEAD /?t=123 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:MISS
127.0.0.1 - - [25/Dec/2025:16:54:36 +0000] "HEAD /?t=123 HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
39.62.158.42 - - [25/Dec/2025:17:10:51 +0000] "GET / HTTP/2.0" 200 162 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:EXPIRED
39.62.158.42 - - [25/Dec/2025:17:23:22 +0000] "GET / HTTP/2.0" 200 162 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:HIT
39.62.158.42 - - [25/Dec/2025:17:38:57 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:-
39.62.158.42 - - [25/Dec/2025:18:00:58 +0000] "GET / HTTP/2.0" 200 162 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:HIT
39.62.158.42 - - [25/Dec/2025:18:03:22 +0000] "GET / HTTP/2.0" 200 162 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36" "-" Cache:HIT
127.0.0.1 - - [25/Dec/2025:18:05:05 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.11.1" "-" Cache:HIT
127.0.0.1 - - [25/Dec/2025:18:06:19 +0000] "HEAD / HTTP/1.1" 301 0 "-" "curl/8.11.1" "-" Cache:-
[ec2-user@ip-10-0-10-75 ~]$
```

\*\*\*\*\*

## Bonus Tasks

### Bonus 1: Custom Error Pages



## Custom 404 - Page Not Found

Served by NGINX (Lab 12).

## Custom 502 - Bad Gateway

NGINX could not reach the backend servers.

### Bonus 2 - Rate Limiting

rate limit config

```
[ec2-user@ip-10-0-10-75 ~]$ sudo nginx -t
sudo systemctl reload nginx
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:3
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Rate limit test

```
[ec2-user@ip-10-0-10-75 ~]$ sudo systemctl reload nginx
[ec2-user@ip-10-0-10-75 ~]$ seq 1 120 | xargs -I{} -P40 sh -c 'curl -k -s -o /dev/null -w "%{http_code}\n" '
3 200
117 429
```

### Bonus 3: Health Check Automation

Health check script

```
[ec2-user@ip-10-0-10-75 ~]$ sudo nginx -t
nginx: [warn] duplicate MIME type "text/html" in /etc/nginx/nginx.conf:12
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2"
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-75 ~]$ sudo systemctl reload nginx
```

Health log

```
[ec2-user@ip-10-0-10-75 ~]$ sudo systemctl reload nginx
[ec2-user@ip-10-0-10-75 ~]$ curl -k -I -H "Accept-Encoding: gzip" https://localhost/ | grep -i content-encoding
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0    0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
content-encoding: gzip
```

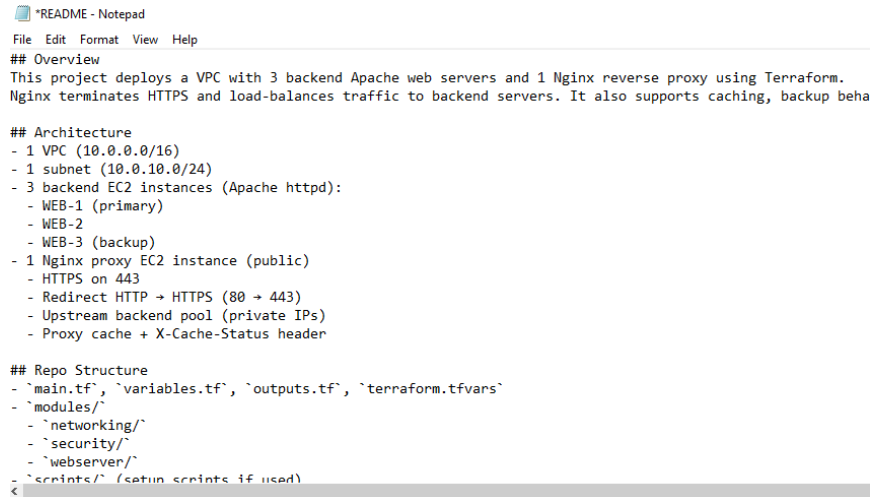
\*\*\*\*\*

## Part 6: Documentation & Cleanup

### 6.1 README Documentation

Documented architecture, prerequisites, deployment steps, configuration guide, and troubleshooting in README.md.

#### README.md content



```
*README - Notepad
File Edit Format View Help

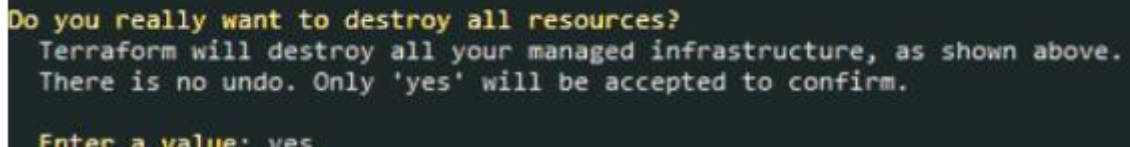
## Overview
This project deploys a VPC with 3 backend Apache web servers and 1 Nginx reverse proxy using Terraform.
Nginx terminates HTTPS and load-balances traffic to backend servers. It also supports caching, backup beha

## Architecture
- 1 VPC (10.0.0.0/16)
- 1 subnet (10.0.10.0/24)
- 3 backend EC2 instances (Apache httpd):
  - WEB-1 (primary)
  - WEB-2
  - WEB-3 (backup)
- 1 Nginx proxy EC2 instance (public)
  - HTTPS on 443
  - Redirect HTTP → HTTPS (80 → 443)
  - Upstream backend pool (private IPs)
  - Proxy cache + X-Cache-Status header

## Repo Structure
- `main.tf`, `variables.tf`, `outputs.tf`, `terraform.tfvars`
- `modules/`
  - `networking/`
  - `security/`
  - `webserver/`
- `scripts/` (setup scripts if used)
```

#### Infrastructure Cleanup

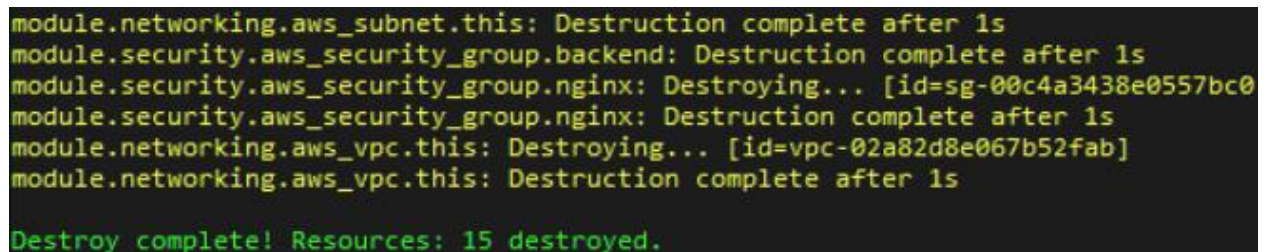
Destroyed all Terraform-managed resources and verified cleanup in AWS Console.



```
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
```

Terraform destroy complete



```
module.networking.aws_subnet.this: Destruction complete after 1s
module.security.aws_security_group.backend: Destruction complete after 1s
module.security.aws_security_group.nginx: Destroying... [id=sg-00c4a3438e0557bc0]
module.security.aws_security_group.nginx: Destruction complete after 1s
module.networking.aws_vpc.this: Destroying... [id=vpc-02a82d8e067b52fab]
module.networking.aws_vpc.this: Destruction complete after 1s

Destroy complete! Resources: 15 destroyed.
```

Empty state

## Instances (5) [Info](#)

[Connect](#)[Instance state ▼](#)[Actions](#)[All states ▼](#)

<input type="checkbox"/>	Name <a href="#">✎</a> ▼	Instance ID	Instance state ▼	Instance type ▼	Status check
<input type="checkbox"/>	web-1	<a href="#">i-0adf637381acf7012</a>	Terminated <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	–
<input type="checkbox"/>	web-3	<a href="#">i-005a97404b163fd15</a>	Terminated <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	–
<input type="checkbox"/>	web-2	<a href="#">i-085d8de3ad82254df</a>	Terminated <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	–
<input type="checkbox"/>	assignment2-...	<a href="#">i-05936bfb73fda8a7</a>	Terminated <a href="#">🔍</a> <a href="#">🔍</a>	t3.micro	–

terraform.tfstate after destroy showing no resources.

```
terraform.tfstate - Notepad
File Edit Format View Help
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 38,
  "lineage": "8176b985-1058-d742-a585-c1c5b93bfc37",
  "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
    }
  ]
}
```

## Testing Results Summary

- Load balancing alternated between web-1 and web-2 across multiple reloads.

## WEB-2

Hostname: ip-10-0-10-107.me-central-1.compute.internal

Instance ID: i-0ce73e90b08689349

Time: Thu Dec 25 16:16:46 UTC 2025

- Cache status changed from MISS (first request) to HIT (subsequent requests).

```
[ec2-user@ip-10-0-10-75 ~]$ curl -k -I "https://localhost/?x=miss1" | grep -i "x-cache-status"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0   159    0     0    0     0      0      0  --:--:--  --:--:--  --:--:--     0
x-cache-status: MISS
```

Backup

```
[ec2-user@ip-10-0-10-75 ~]$ curl -k -I "https://localhost/?x=miss1" | grep -i "x-cache-status"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0   159    0     0    0     0      0      0  --:--:--  --:--:--  --:--:--     0
x-cache-status: HIT
```

\*\*\*\*\*