Fatima Jinnah Women University

Department of Software Engineering

CLOUD COMPUTING

Assignment 2

Advanced Terraform & Nginx Multi-Tier Architecture

Submitted to: Waqas Saleem

Submitted by: Aimen Hafeez

Registration #: 2023-BSE-002

Deadline: 30th Dec 2025

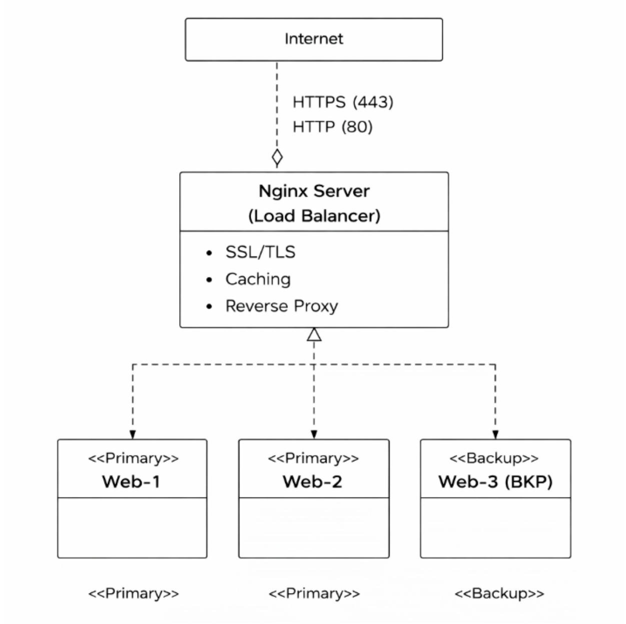
Table of Contents

Project Overview:

This project focuses on designing and deploying a robust, production-ready multi-tier web infrastructure on AWS using Terraform for Infrastructure as Code (IaC) and Nginx as a reverse proxy and load balancer with advanced configurations. The assignment involves creating a high-availability environment comprising a single Nginx server for HTTPS traffic management and load balancing, three backend web servers for application hosting, and comprehensive networking and security configurations to ensure reliability and performance. Key objectives include modular Terraform code organization, dynamic variable and local configurations, automated server provisioning with reusable webserver modules, implementation of caching mechanisms, SSL/TLS encryption, and backup server setup for failover scenarios. The project emphasizes practical application of AWS services, secure SSH management, Nginx advanced settings, and monitoring, culminating in a scalable, secure, and maintainable web architecture validated through deployment, configuration testing, load balancing verification, caching assessment, and high-availability simulations.

Arctitecture Diagram:

Here’s an archtitecture diagram for Multi-Tier Web Infrastructure:

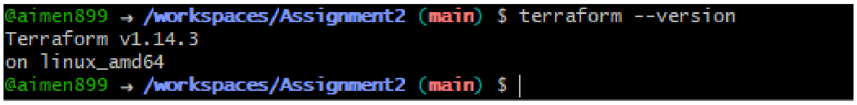


PART 0: GitHub Repository & Codespace Setup

1. Authenticate GitHub CLI
2. Create GitHub Repository
3. Create GitHub Codespace
4. SSH into Codespace

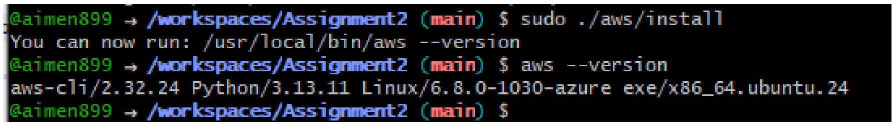
PART 1 – Infrastructure Setup:

Install Terraform in Codespace



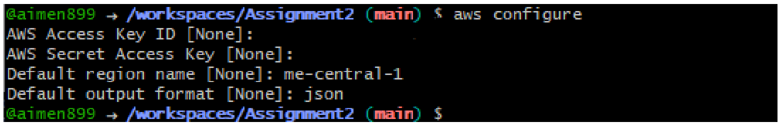
```
@aimen899 → /workspaces/Assignment2 (main) $ terraform --version
Terraform v1.14.3
on linux_amd64
@aimen899 → /workspaces/Assignment2 (main) $ |
```

Install aws



```
@aimen899 → /workspaces/Assignment2 (main) $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
@aimen899 → /workspaces/Assignment2 (main) $ aws --version
aws-cli/2.32.24 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
@aimen899 → /workspaces/Assignment2 (main) $
```

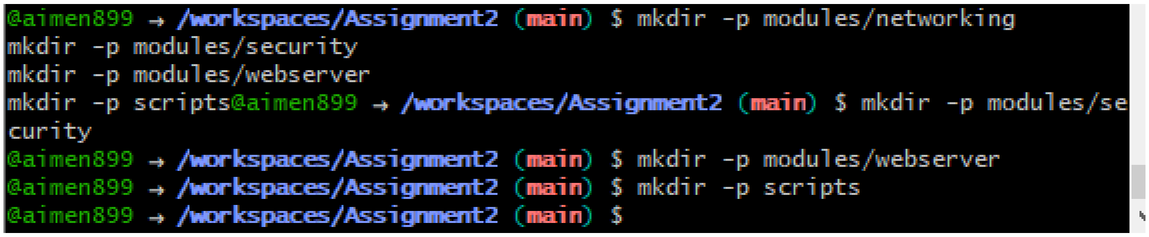
Configure AWS Credentials



```
@aimen899 → /workspaces/Assignment2 (main) $ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: me-central-1
Default output format [None]: json
@aimen899 → /workspaces/Assignment2 (main) $
```

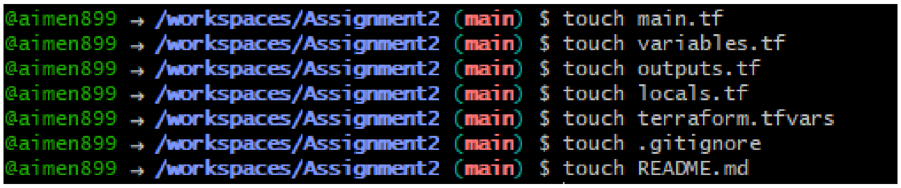
1.1 Create Complete Folder Structure

Create all directories



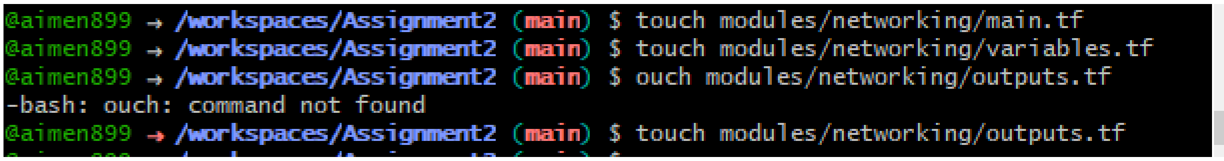
```
@aimen899 → /workspaces/Assignment2 (main) $ mkdir -p modules/networking
mkdir -p modules/security
mkdir -p modules/webserver
mkdir -p scripts@aimen899 → /workspaces/Assignment2 (main) $ mkdir -p modules/se
curity
@aimen899 → /workspaces/Assignment2 (main) $ mkdir -p modules/webserver
@aimen899 → /workspaces/Assignment2 (main) $ mkdir -p scripts
@aimen899 → /workspaces/Assignment2 (main) $
```

Create all root level files



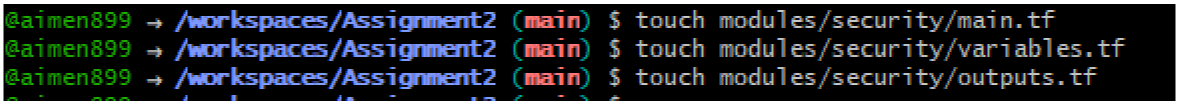
```
@aimen899 → /workspaces/Assignment2 (main) $ touch main.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch variables.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch outputs.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch locals.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch terraform.tfvars
@aimen899 → /workspaces/Assignment2 (main) $ touch .gitignore
@aimen899 → /workspaces/Assignment2 (main) $ touch README.md
```

Create networking module files



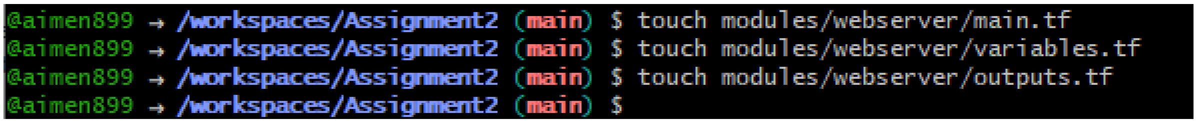
```
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/networking/main.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/networking/variables.tf
@aimen899 → /workspaces/Assignment2 (main) $ ouch modules/networking/outputs.tf
-bash: ouch: command not found
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/networking/outputs.tf
```

Create security module files



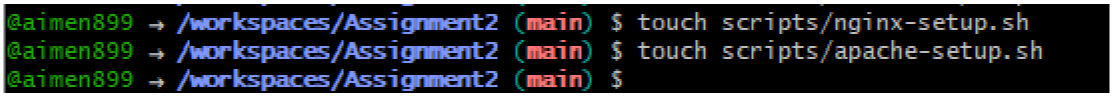
```
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/security/main.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/security/variables.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/security/outputs.tf
```

Create webserver module files



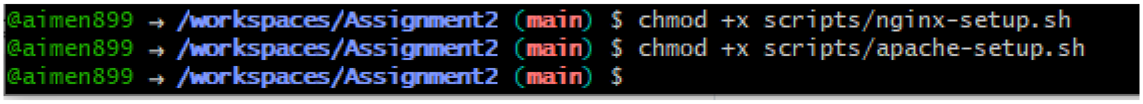
```
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/webserver/main.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/webserver/variables.tf
@aimen899 → /workspaces/Assignment2 (main) $ touch modules/webserver/outputs.tf
@aimen899 → /workspaces/Assignment2 (main) $
```

Create script files



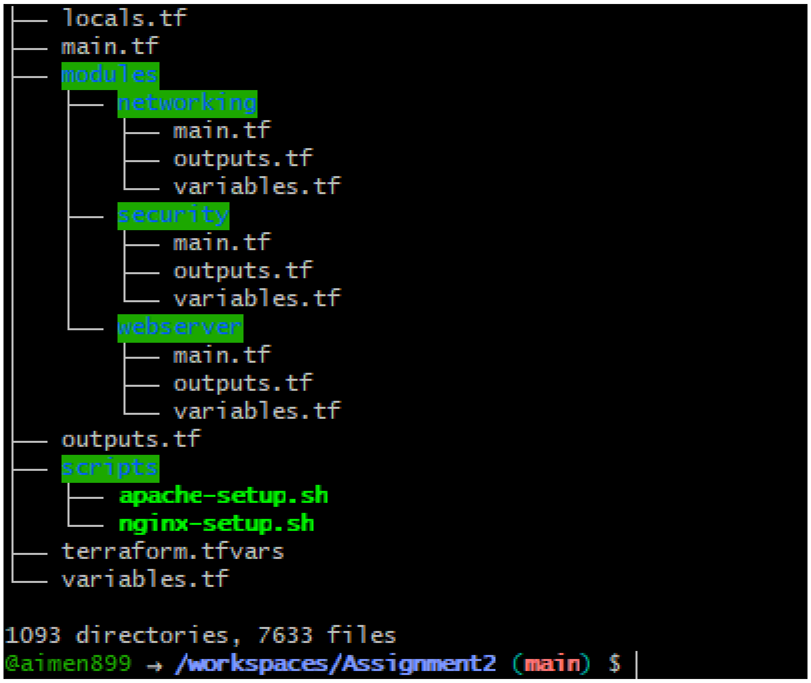
```
@aimen899 → /workspaces/Assignment2 (main) $ touch scripts/nginx-setup.sh
@aimen899 → /workspaces/Assignment2 (main) $ touch scripts/apache-setup.sh
@aimen899 → /workspaces/Assignment2 (main) $
```

Make scripts executable



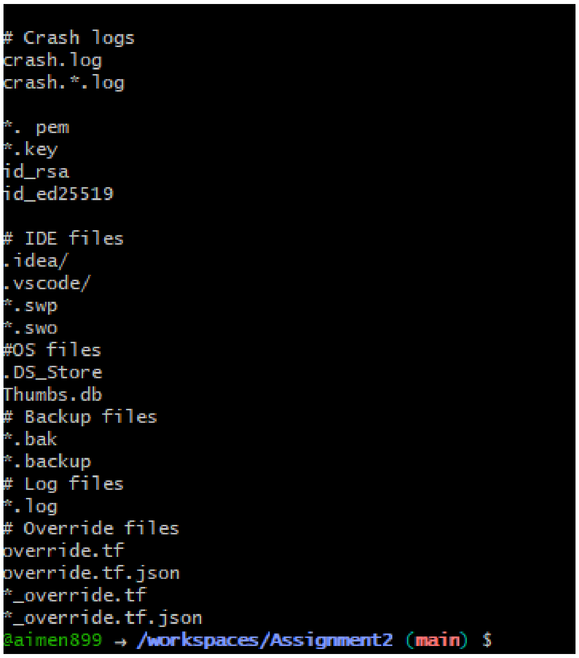
```
@aimen899 → /workspaces/Assignment2 (main) $ chmod +x scripts/nginx-setup.sh
@aimen899 → /workspaces/Assignment2 (main) $ chmod +x scripts/apache-setup.sh
@aimen899 → /workspaces/Assignment2 (main) $
```

Verify structure with tree command



```
├── 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

1093 directories, 7633 files
@aimen899 → /workspaces/Assignment2 (main) $
```

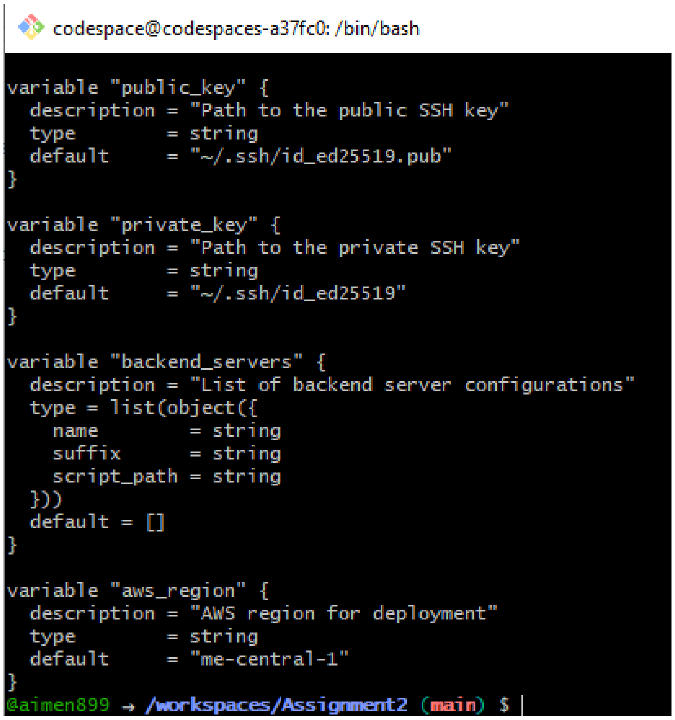
Create . gitignore File



```
# Crash logs
crash.log
crash.*.log

*. pem
*.key
id_rsa
id_ed25519

# IDE files
.idea/
.vscode/
*.swp
*.swo
#OS files
.DS_Store
Thumbs.db
# Backup files
*.bak
*.backup
# Log files
*.log
# Override files
override.tf
override.tf.json
*_override.tf
*_override.tf.json
@aimen899 → /workspaces/Assignment2 (main) $
```

1.2 Create File Contents

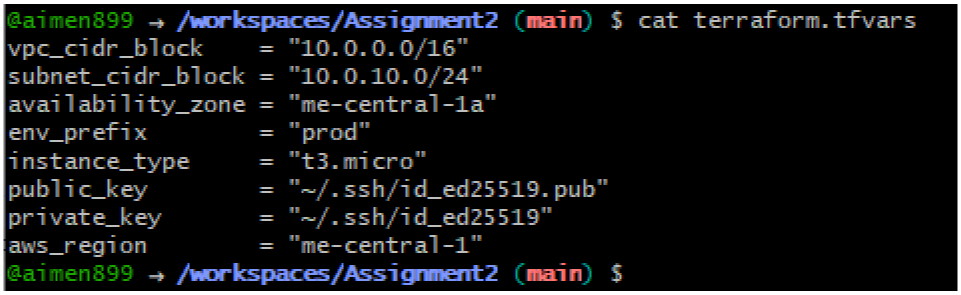
Create variables.tf



codespace@codespaces-a37fc0: /bin/bash

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

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

variable "backend_servers" {
  description = "List of backend server configurations"
  type = list(object({
    name        = string
    suffix      = string
    script_path = string
  }))
  default = []
}

variable "aws_region" {
  description = "AWS region for deployment"
  type        = string
  default     = "me-central-1"
}
@aimen899 → /workspaces/Assignment2 (main) $ 
```

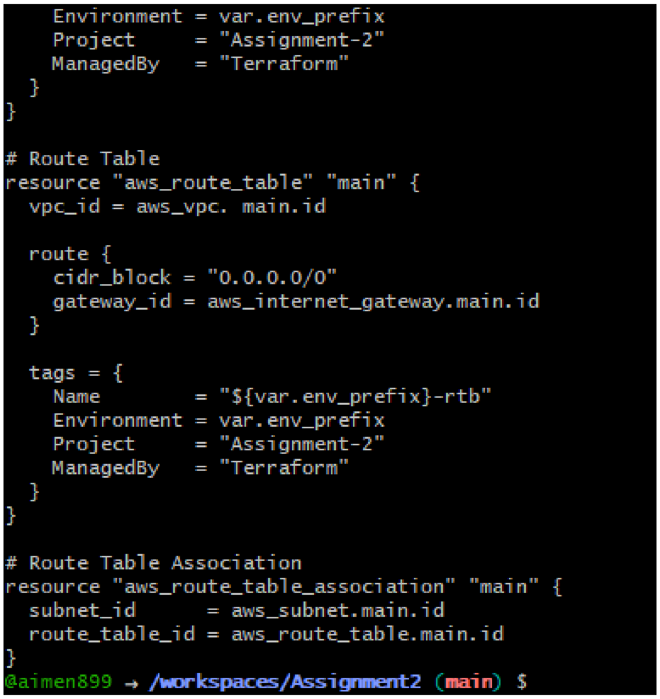
Create terraform.tfvars



```
@aimen899 → /workspaces/Assignment2 (main) $ cat 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"
aws_region        = "me-central-1"
@aimen899 → /workspaces/Assignment2 (main) $
```

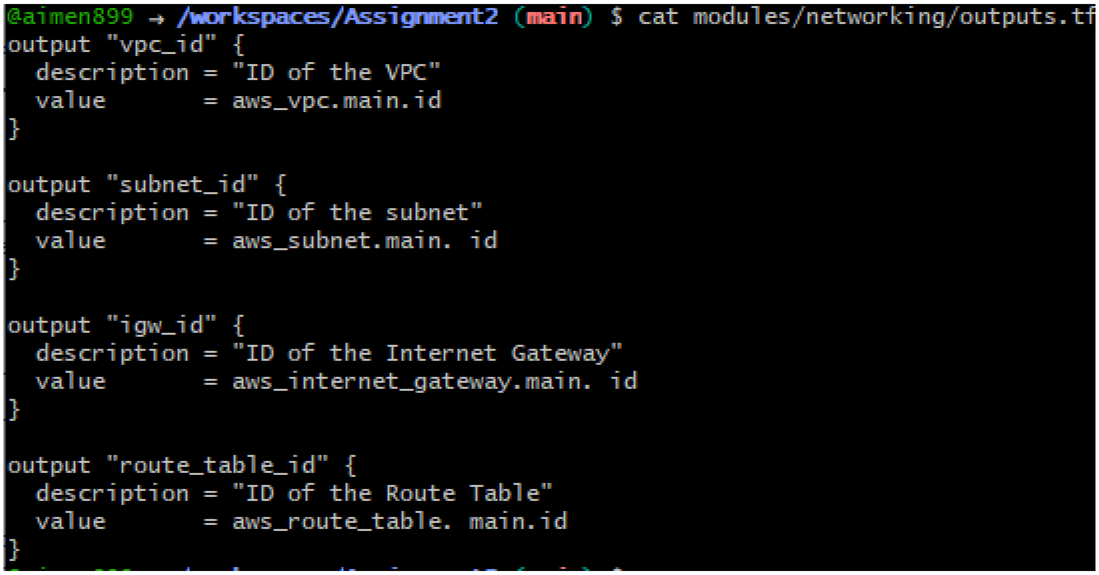
1.3 Create Networking Module

Main.tf



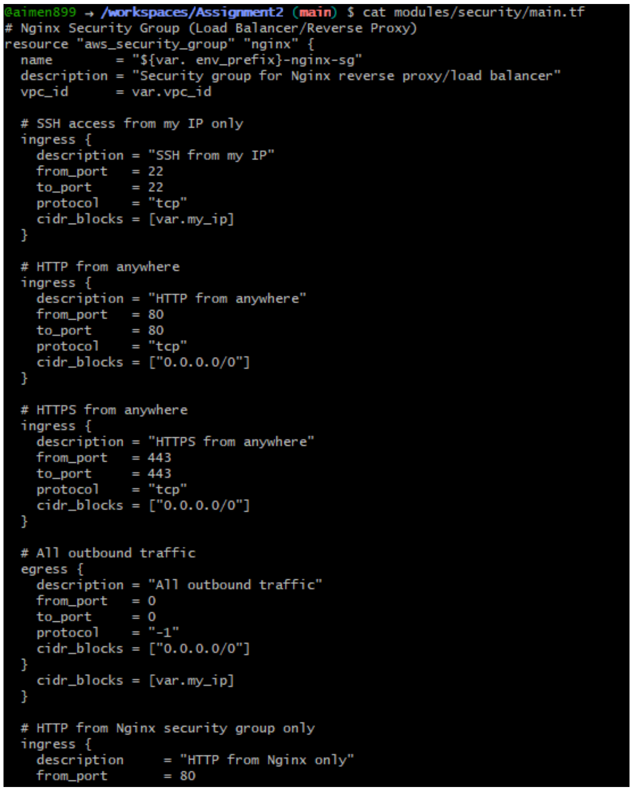
```
    Environment = var.env_prefix
    Project     = "Assignment-2"
    ManagedBy   = "Terraform"
  }
}

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

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

  tags = {
    Name        = "${var.env_prefix}-rtb"
    Environment = var.env_prefix
    Project     = "Assignment-2"
    ManagedBy   = "Terraform"
  }
}

# Route Table Association
resource "aws_route_table_association" "main" {
  subnet_id      = aws_subnet.main.id
  route_table_id = aws_route_table.main.id
}
@aimen899 → /workspaces/Assignment2 (main) $
```

Outputs.tf



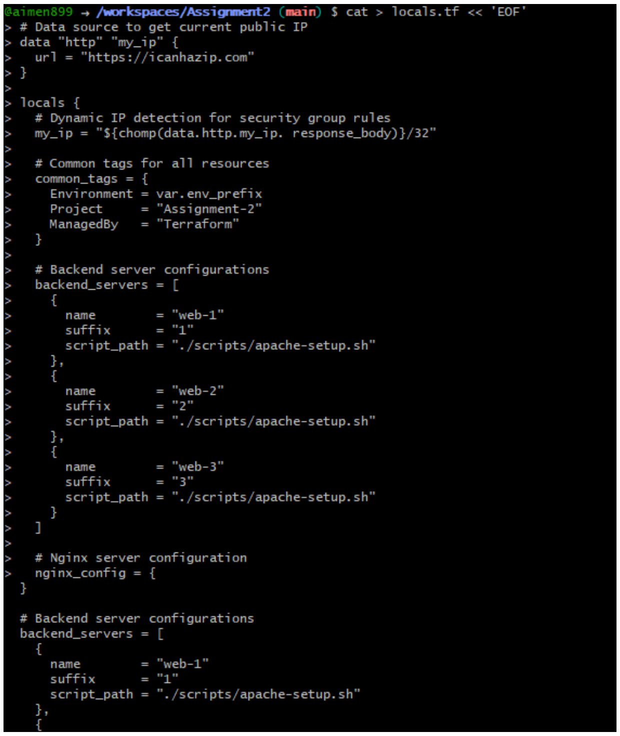
```
@aimen899 → /workspaces/Assignment2 (main) $ cat modules/networking/outputs.tf
output "vpc_id" {
  description = "ID of the VPC"
  value       = aws_vpc.main.id
}

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

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

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

1.4 Create Security Module



```
@aimen899 → /workspaces/Assignment2 (main) $ cat modules/security/main.tf
# Nginx Security Group (Load Balancer/Reverse Proxy)
resource "aws_security_group" "nginx" {
  name        = "${var. env_prefix}-nginx-sg"
  description = "Security group for Nginx reverse proxy/load balancer"
  vpc_id      = var.vpc_id

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

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

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

  # All outbound traffic
  egress {
    description = "All outbound traffic"
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
    cidr_blocks = [var.my_ip]
  }

  # HTTP from Nginx security group only
  ingress {
    description     = "HTTP from Nginx only"
    from_port       = 80
```

1.5 Create locals.tf

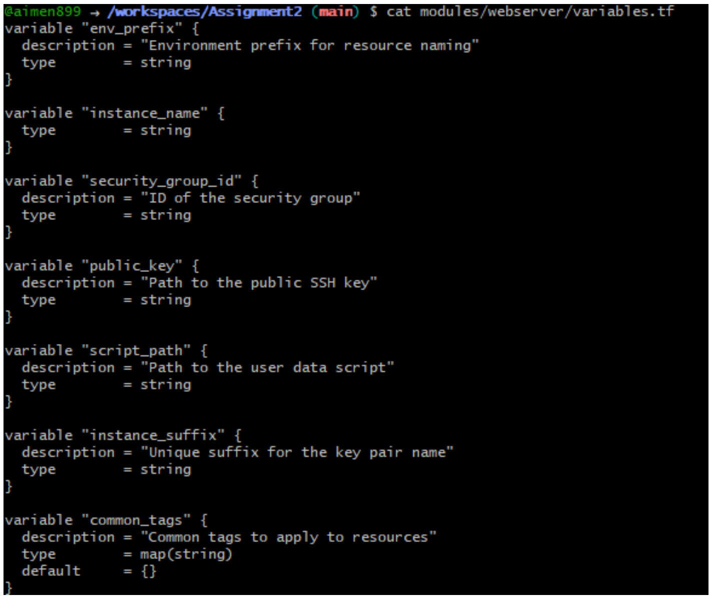


```
@aimen899 → /workspaces/Assignment2 (main) $ cat > locals.tf << 'EOF'
> # Data source to get current public IP
> data "http" "my_ip" {
>   url = "https://icanhazip.com"
> }
>
> locals {
>   # Dynamic IP detection for security group rules
>   my_ip = "${chomp(data.http.my_ip. response_body)}/32"
>
>   # Common tags for all resources
>   common_tags = {
>     Environment = var.env_prefix
>     Project     = "Assignment-2"
>     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"
>     }
>   ]
>
>   # Nginx server configuration
>   nginx_config = {
  }

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

Part 2: Webserver Module

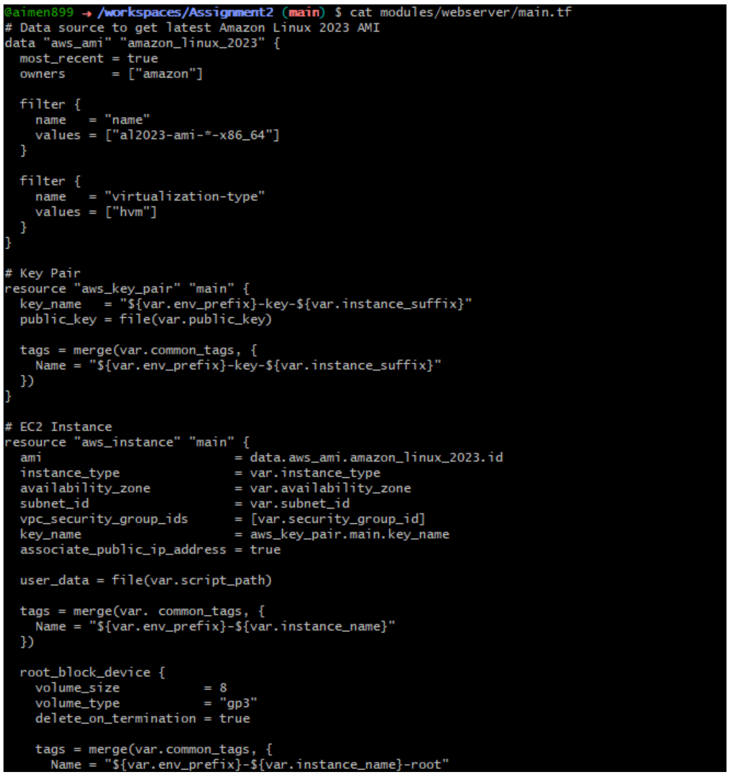
2.1 Create Webserver Module

modules/webserver/variables.tf



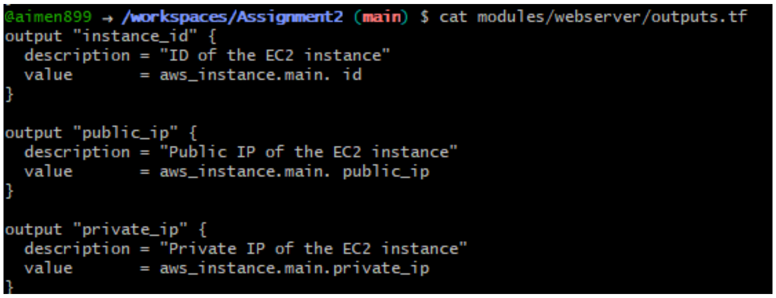
```
@aimen899 → /workspaces/Assignment2 (main) $ cat modules/webserver/variables.tf
variable "env_prefix" {
  description = "Environment prefix for resource naming"
  type        = string
}

variable "instance_name" {
  type        = string
}

variable "security_group_id" {
  description = "ID of the security group"
  type        = string
}

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

variable "script_path" {
  description = "Path to the user data script"
  type        = string
}

variable "instance_suffix" {
  description = "Unique suffix for the key pair name"
  type        = string
}

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

modules/webserver/ main.tf



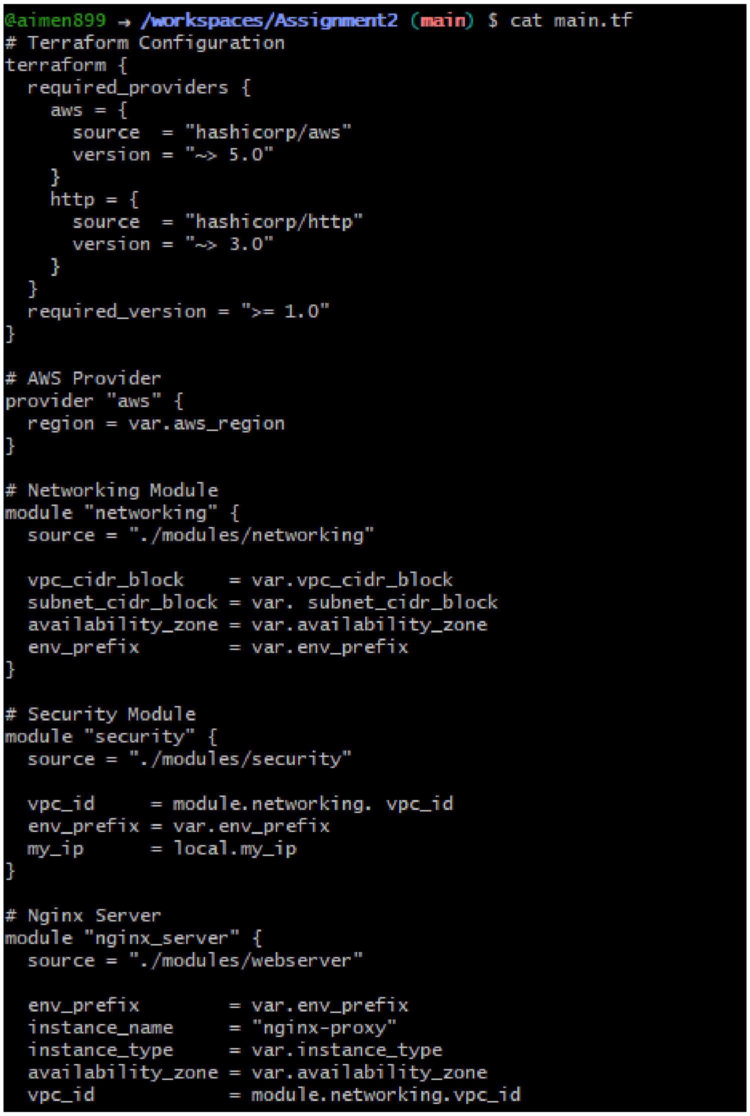
```
@aimen899 → /workspaces/Assignment2 (main) $ cat modules/webserver/main.tf
# Data source to get latest Amazon Linux 2023 AMI
data "aws_ami" "amazon_linux_2023" {
  most_recent = true
  owners      = ["amazon"]

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

  filter {
    name   = "virtualization-type"
    values = ["hvm"]
  }
}

# Key Pair
resource "aws_key_pair" "main" {
  key_name   = "${var.env_prefix}-key-${var.instance_suffix}"
  public_key = file(var.public_key)

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

# EC2 Instance
resource "aws_instance" "main" {
  ami                         = data.aws_ami.amazon_linux_2023.id
  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.main.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}"
  })

  root_block_device {
    volume_size           = 8
    volume_type           = "gp3"
    delete_on_termination = true

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

modules/webserver/ outputs.tf



```
@aimen899 → /workspaces/Assignment2 (main) $ cat modules/webserver/outputs.tf
output "instance_id" {
  description = "ID of the EC2 instance"
  value       = aws_instance.main. id
}

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

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

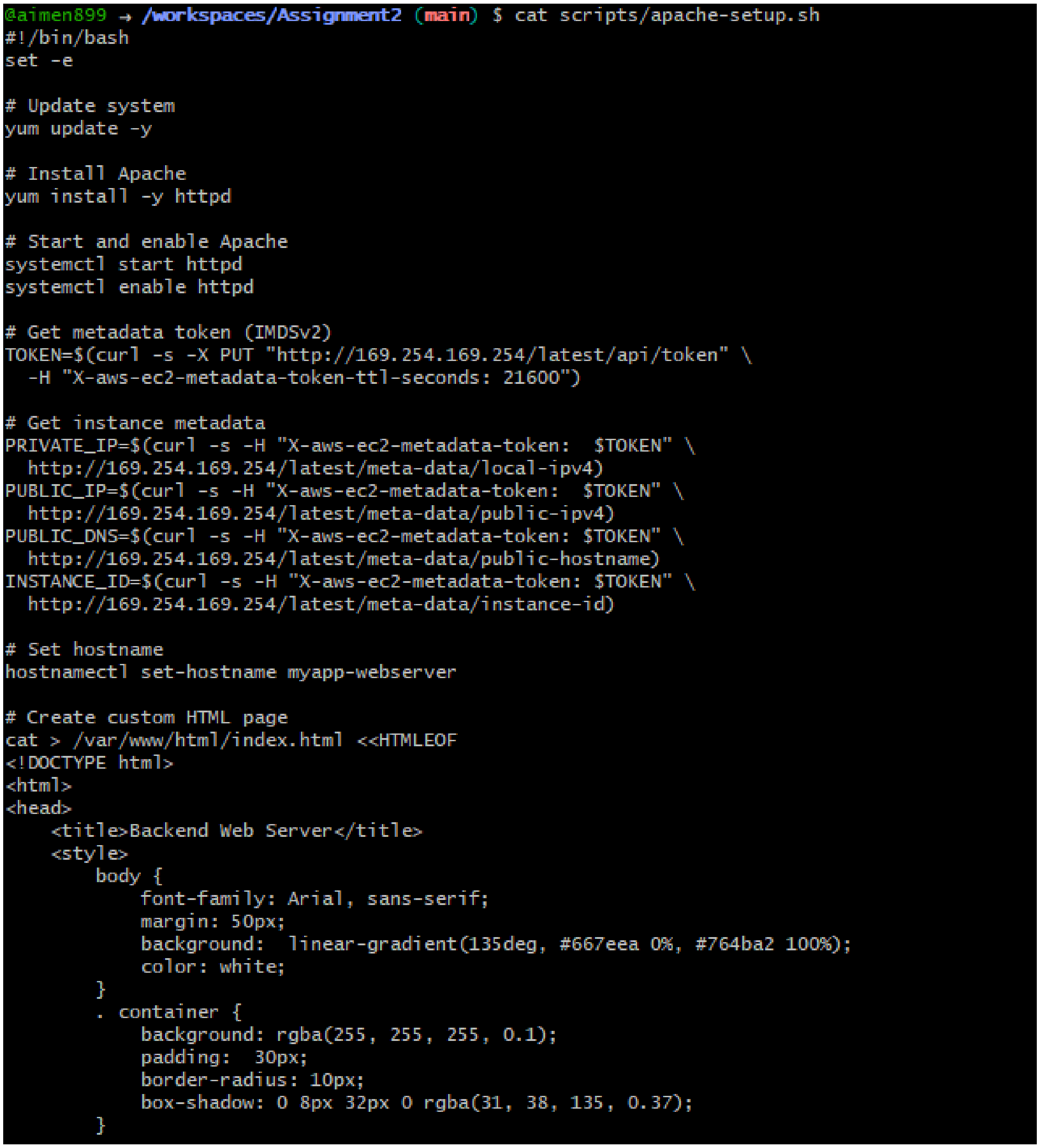
2.2 Create Root main.tf



```
@aimen899 → /workspaces/Assignment2 (main) $ cat main.tf
# Terraform Configuration
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
    http = {
      source  = "hashicorp/http"
      version = "~> 3.0"
    }
  }
  required_version = ">= 1.0"
}

# AWS Provider
provider "aws" {
  region = var.aws_region
}

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

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

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

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

# Nginx Server
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
```

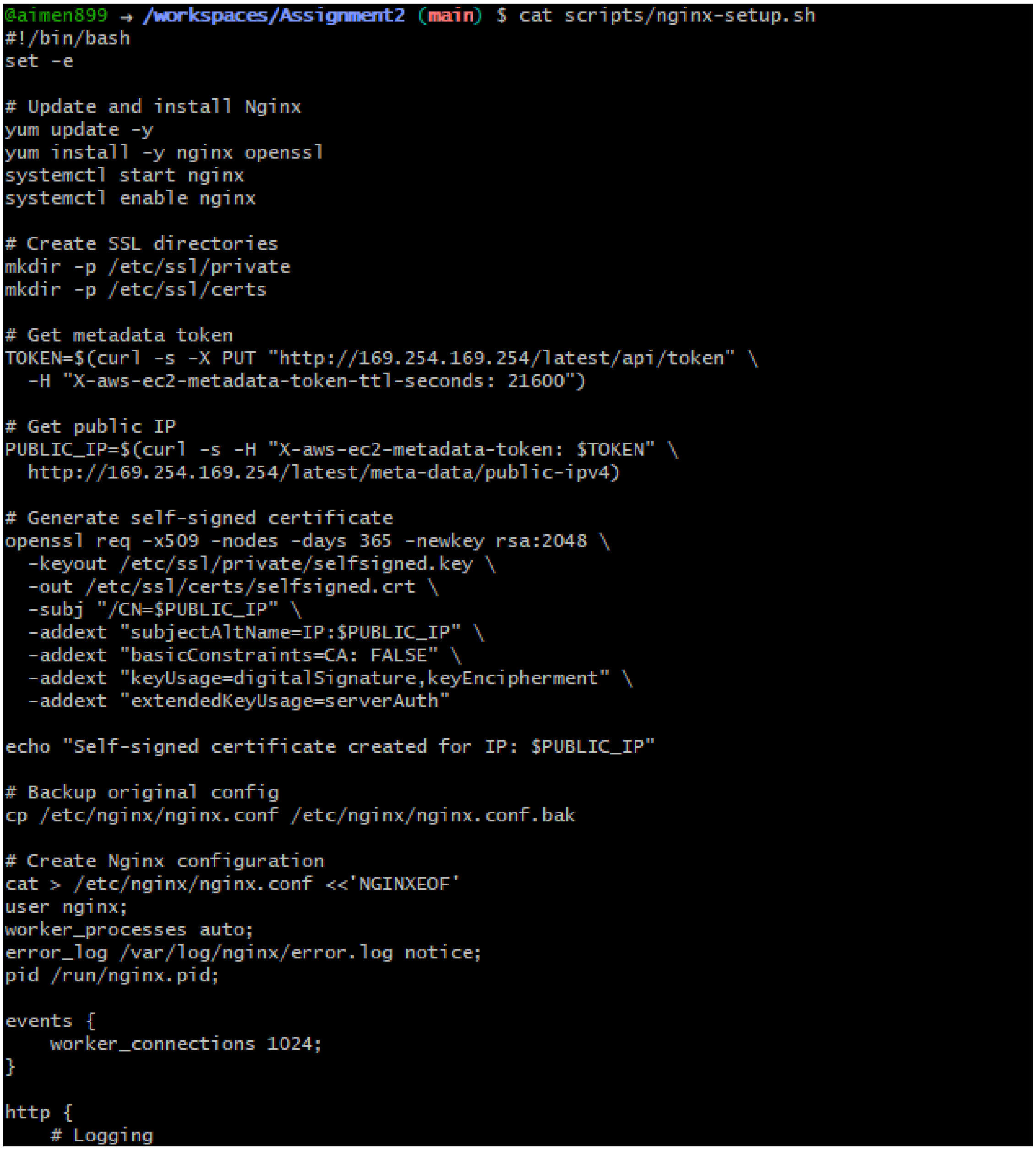
Part 3: Server Configuration Scripts

3.1 Create Apache Backend Script



```
@aimen899 → /workspaces/Assignment2 (main) $ cat scripts/apache-setup.sh
#!/bin/bash
set -e

# Update system
yum update -y

# Install Apache
yum install -y httpd

# Start and enable Apache
systemctl start httpd
systemctl enable httpd

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

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

# Set hostname
hostnamectl set-hostname myapp-webserver

# Create custom HTML page
cat > /var/www/html/index.html <<HTMLEOF
<!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);
        }
```

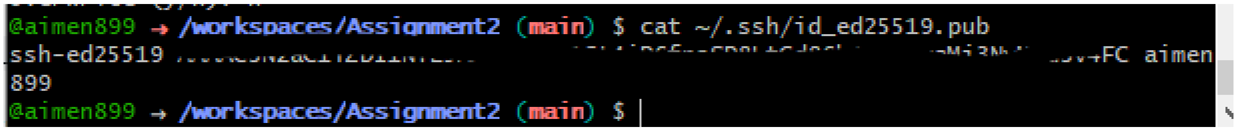
3.2 Create Nginx Setup Script



```
@aimen899 → /workspaces/Assignment2 (main) $ cat scripts/nginx-setup.sh
#!/bin/bash
set -e

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

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

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

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

# Generate self-signed certificate
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/private/selfsigned.key \
  -out /etc/ssl/certs/selfsigned.crt \
  -subj "/CN=$PUBLIC_IP" \
  -addext "subjectAltName=IP:$PUBLIC_IP" \
  -addext "basicConstraints=CA: FALSE" \
  -addext "keyUsage=digitalSignature,keyEncipherment" \
  -addext "extendedKeyUsage=serverAuth"

echo "Self-signed certificate created for IP: $PUBLIC_IP"

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

# Create Nginx configuration
cat > /etc/nginx/nginx.conf <<'NGINXEOF'
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    # Logging
```

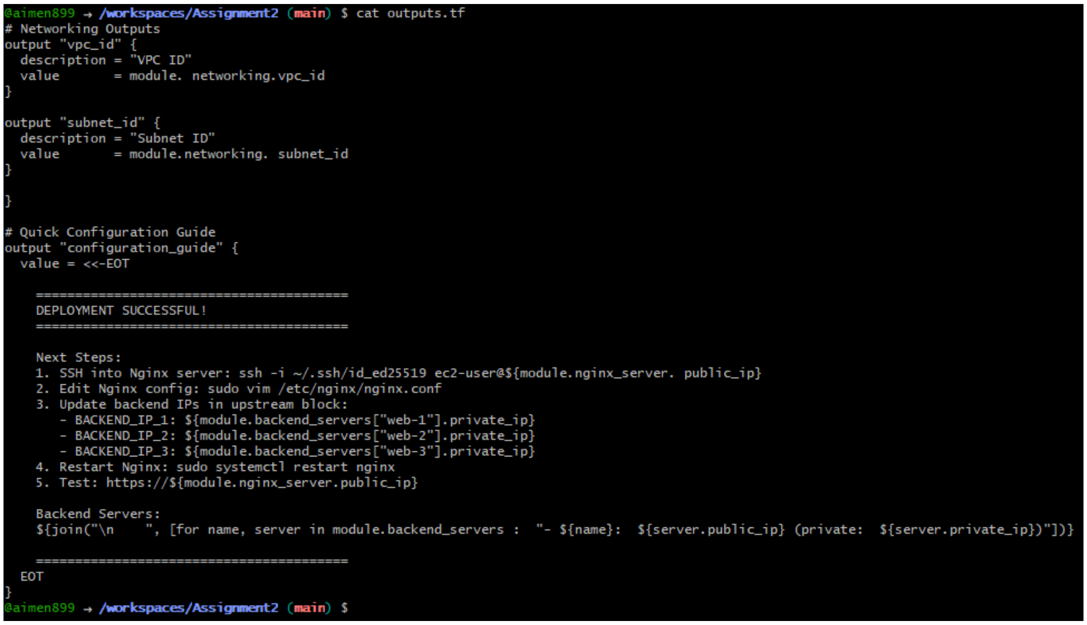
Part 4: Infrastructure Deployment

4.1 Generate SSH Key Pair



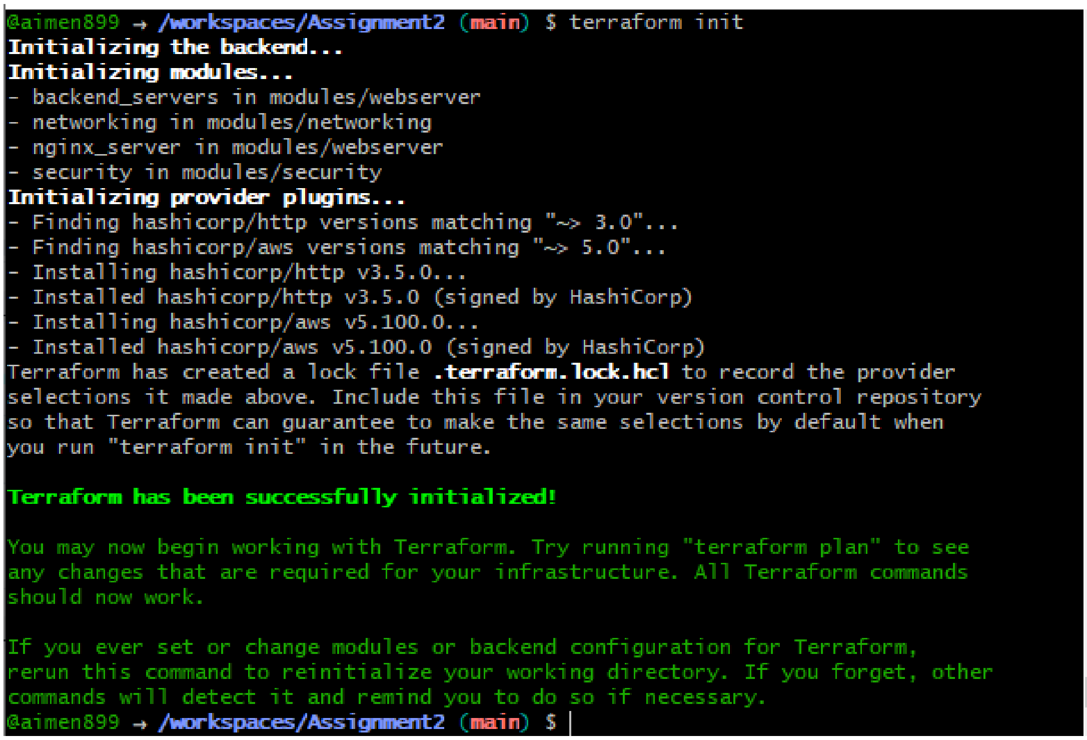
```
@aimen899 → /workspaces/Assignment2 (main) $ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 [redacted]+FC aimen
899
@aimen899 → /workspaces/Assignment2 (main) $ |
```

4.2 Create outputs.tf



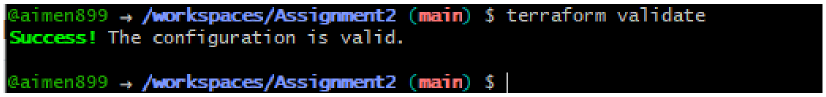
```
@aimen899 → /workspaces/Assignment2 (main) $ cat outputs.tf
# Networking Outputs
output "vpc_id" {
  description = "VPC ID"
  value       = module. networking.vpc_id
}

output "subnet_id" {
  description = "Subnet ID"
  value       = module.networking. subnet_id
}

}

# Quick Configuration Guide
output "configuration_guide" {
  value = <<-EOT

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

    Next Steps:
    1. SSH into Nginx server: ssh -i ~/.ssh/id_ed25519 ec2-user@${module.nginx_server. public_ip}
    2. Edit Nginx config: sudo vim /etc/nginx/nginx.conf
    3. Update backend IPs in upstream block:
       - BACKEND_IP_1: ${module.backend_servers["web-1"].private_ip}
       - BACKEND_IP_2: ${module.backend_servers["web-2"].private_ip}
       - BACKEND_IP_3: ${module.backend_servers["web-3"].private_ip}
    4. Restart Nginx: sudo systemctl restart nginx
    5. Test: https://${module.nginx_server.public_ip}

    Backend Servers:
    ${join("\n    ", [for name, server in module.backend_servers :  "- ${name}:  ${server.public_ip} (private:  ${server.private_ip})"])}

    ========================================
  EOT
}
@aimen899 → /workspaces/Assignment2 (main) $
```

Initialize Terraform



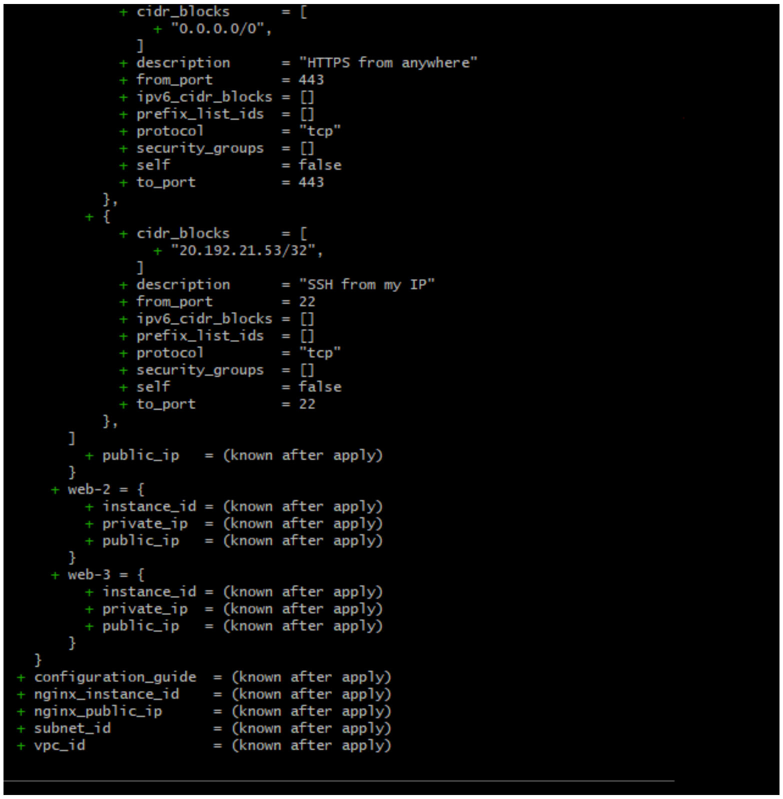
```
@aimen899 → /workspaces/Assignment2 (main) $ terraform init
Initializing the backend...
Initializing modules...
- backend_servers in modules/webserver
- networking in modules/networking
- nginx_server in modules/webserver
- security in modules/security
Initializing provider plugins...
- Finding hashicorp/http versions matching "~> 3.0"...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v5.100.0...
- Installed hashicorp/aws v5.100.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
@aimen899 → /workspaces/Assignment2 (main) $ |
```

Validate Terraform Configuration



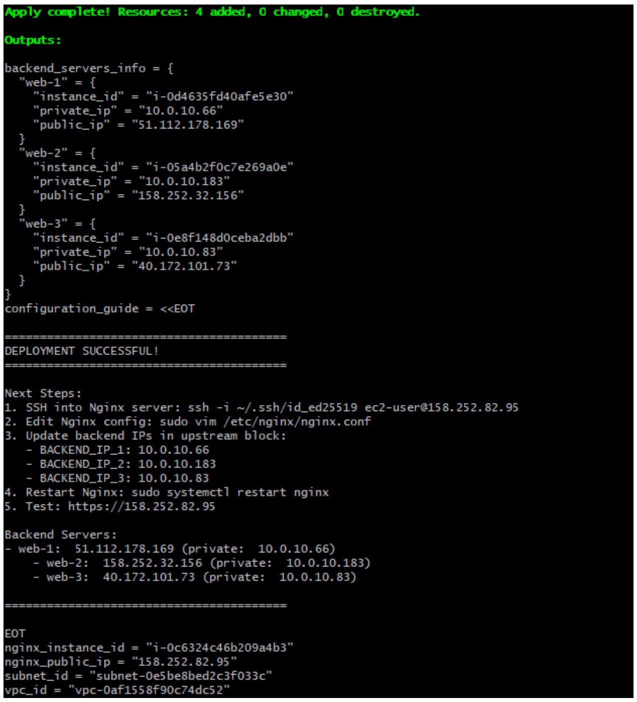
```
@aimen899 → /workspaces/Assignment2 (main) $ terraform validate
Success! The configuration is valid.

@aimen899 → /workspaces/Assignment2 (main) $ |
```

Plan Terraform Deployment



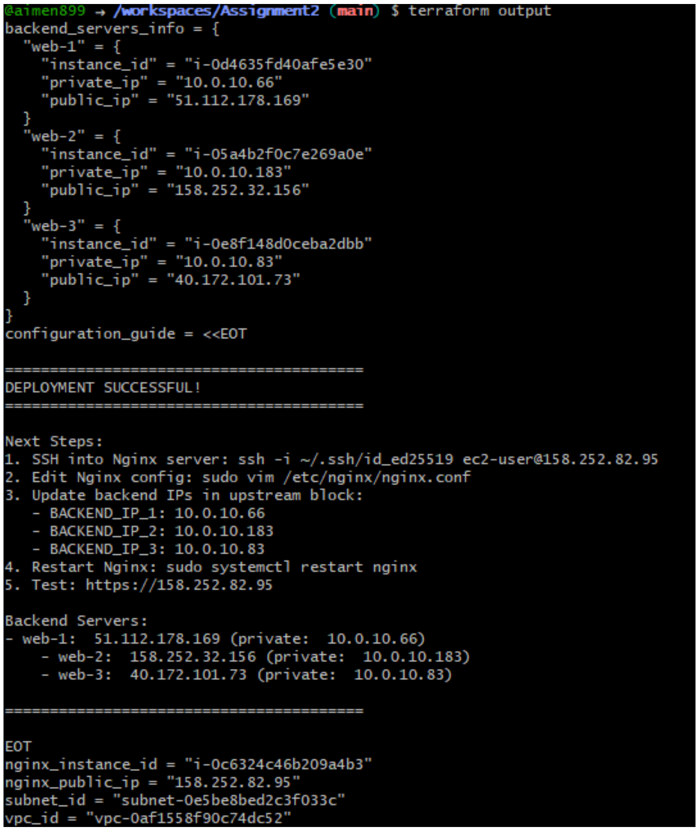
```
            + cidr_blocks      = [
                + "0.0.0.0/0",
              ]
            + description      = "HTTPS from anywhere"
            + from_port        = 443
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 443
          },
        + {
            + cidr_blocks      = [
                + "20.192.21.53/32",
              ]
            + description      = "SSH from my IP"
            + from_port        = 22
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
      ]
        + public_ip   = (known after apply)
      }
    + web-2 = {
        + instance_id = (known after apply)
        + private_ip  = (known after apply)
        + public_ip   = (known after apply)
      }
    + web-3 = {
        + instance_id = (known after apply)
        + private_ip  = (known after apply)
        + public_ip   = (known after apply)
      }
  }
+ configuration_guide  = (known after apply)
+ nginx_instance_id    = (known after apply)
+ nginx_public_ip      = (known after apply)
+ subnet_id            = (known after apply)
+ vpc_id               = (known after apply)
```

Apply Terraform Configuration



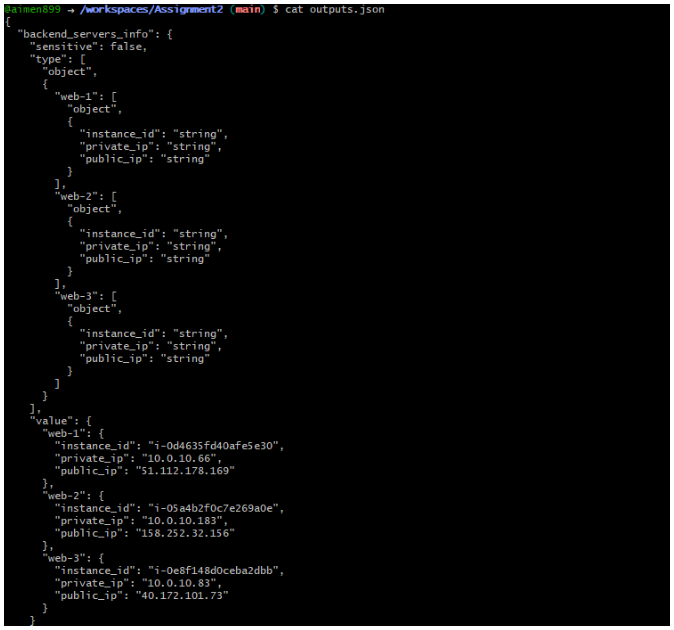
```
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:

backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-0d4635fd40afe5e30"
    "private_ip" = "10.0.10.66"
    "public_ip" = "51.112.178.169"
  }
  "web-2" = {
    "instance_id" = "i-05a4b2f0c7e269a0e"
    "private_ip" = "10.0.10.183"
    "public_ip" = "158.252.32.156"
  }
  "web-3" = {
    "instance_id" = "i-0e8f148d0ceba2dbb"
    "private_ip" = "10.0.10.83"
    "public_ip" = "40.172.101.73"
  }
}
configuration_guide = <<EOT

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

Next Steps:
1. SSH into Nginx server: ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.82.95
2. Edit Nginx config: sudo vim /etc/nginx/nginx.conf
3. Update backend IPs in upstream block:
   - BACKEND_IP_1: 10.0.10.66
   - BACKEND_IP_2: 10.0.10.183
   - BACKEND_IP_3: 10.0.10.83
4. Restart Nginx: sudo systemctl restart nginx
5. Test: https://158.252.82.95

Backend Servers:
- web-1:  51.112.178.169 (private:  10.0.10.66)
    - web-2:  158.252.32.156 (private:  10.0.10.183)
    - web-3:  40.172.101.73 (private:  10.0.10.83)

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

EOT
nginx_instance_id = "i-0c6324c46b209a4b3"
nginx_public_ip = "158.252.82.95"
subnet_id = "subnet-0e5be8bed2c3f033c"
vpc_id = "vpc-0af1558f90c74dc52"
```

Terraform output:



```
@aimen899 → /workspaces/Assignment2 (main) $ terraform output
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-0d4635fd40afe5e30"
    "private_ip" = "10.0.10.66"
    "public_ip" = "51.112.178.169"
  }
  "web-2" = {
    "instance_id" = "i-05a4b2f0c7e269a0e"
    "private_ip" = "10.0.10.183"
    "public_ip" = "158.252.32.156"
  }
  "web-3" = {
    "instance_id" = "i-0e8f148d0ceba2dbb"
    "private_ip" = "10.0.10.83"
    "public_ip" = "40.172.101.73"
  }
}
configuration_guide = <<EOT

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

Next Steps:
1. SSH into Nginx server: ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.82.95
2. Edit Nginx config: sudo vim /etc/nginx/nginx.conf
3. Update backend IPs in upstream block:
   - BACKEND_IP_1: 10.0.10.66
   - BACKEND_IP_2: 10.0.10.183
   - BACKEND_IP_3: 10.0.10.83
4. Restart Nginx: sudo systemctl restart nginx
5. Test: https://158.252.82.95

Backend Servers:
- web-1:  51.112.178.169 (private:  10.0.10.66)
    - web-2:  158.252.32.156 (private:  10.0.10.183)
    - web-3:  40.172.101.73 (private:  10.0.10.83)

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

EOT
nginx_instance_id = "i-0c6324c46b209a4b3"
nginx_public_ip = "158.252.82.95"
subnet_id = "subnet-0e5be8bed2c3f033c"
vpc_id = "vpc-0af1558f90c74dc52"
```

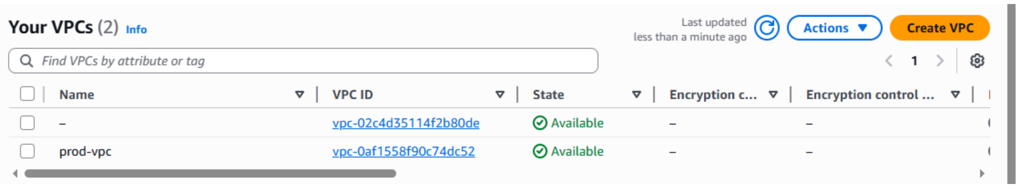
Output.json



```
@aimen899 → /workspaces/Assignment2 (main) $ cat outputs.json
{
  "backend_servers_info": {
    "sensitive": false,
    "type": [
      "object",
      {
        "web-1": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-2": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-3": [
          "object",
          {
            "instance_id": "string",
            "private_ip": "string",
            "public_ip": "string"
          }
        ]
      }
    ],
    "value": {
      "web-1": {
        "instance_id": "i-0d4635fd40afe5e30",
        "private_ip": "10.0.10.66",
        "public_ip": "51.112.178.169"
      },
      "web-2": {
        "instance_id": "i-05a4b2f0c7e269a0e",
        "private_ip": "10.0.10.183",
        "public_ip": "158.252.32.156"
      },
      "web-3": {
        "instance_id": "i-0e8f148d0ceba2dbb",
        "private_ip": "10.0.10.83",
        "public_ip": "40.172.101.73"
      }
    }
```

4.3 Verify Resources in AWS Console

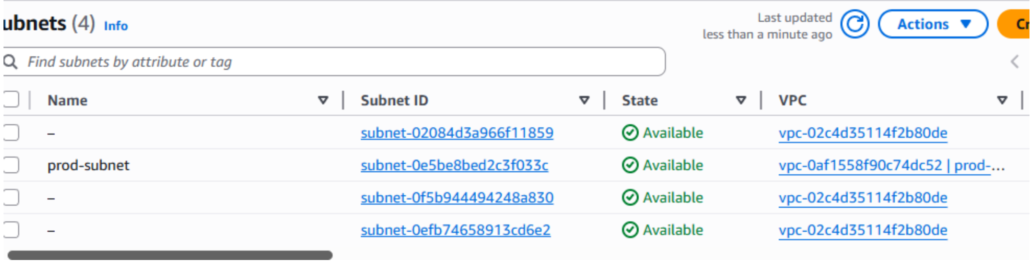
VPC:



Your VPCs (2) Info

□	Name	VPC ID	State	Encryption c...	Encryption control ...
□	–	vpc-02c4d35114f2b80de	✓ Available	–	–
□	prod-vpc	vpc-0af1558f90c74dc52	✓ Available	–	–

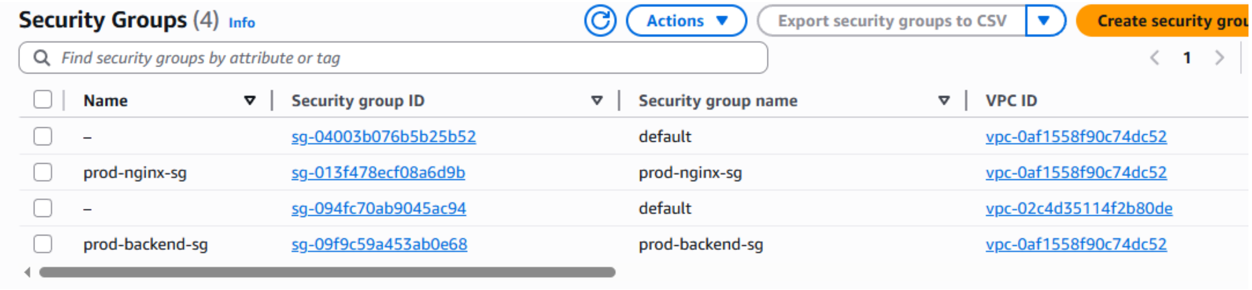
Subnet:



ubnets (4) Info

□	Name	Subnet ID	State	VPC
□	–	subnet-02084d3a966f11859	✓ Available	vpc-02c4d35114f2b80de
□	prod-subnet	subnet-0e5be8bed2c3f033c	✓ Available	vpc-0af1558f90c74dc52 | prod-...
□	–	subnet-0f5b944494248a830	✓ Available	vpc-02c4d35114f2b80de
□	–	subnet-0efb74658913cd6e2	✓ Available	vpc-02c4d35114f2b80de

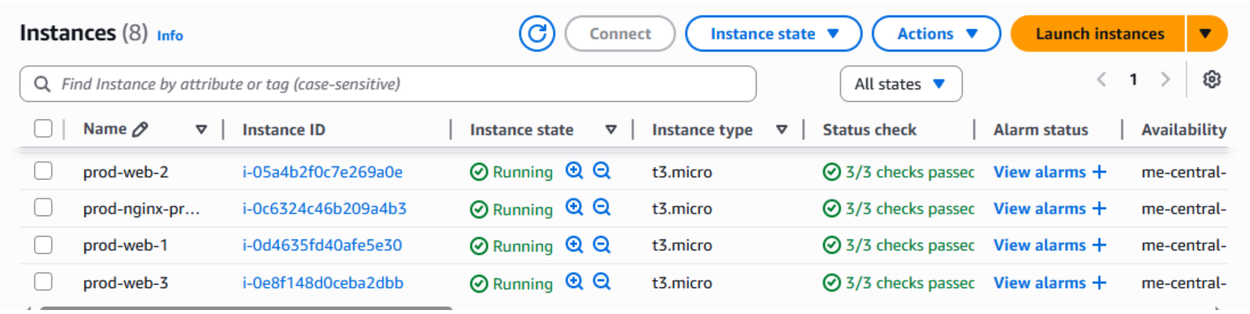
Security Groups:


Actions ▼
Export security groups to CSV
Create security grou
Find security groups by attribute or tag
1

Security Groups (4) Info

□	Name	Security group ID	Security group name	VPC ID
□	–	sg-04003b076b5b25b52	default	vpc-0af1558f90c74dc52
□	prod-nginx-sg	sg-013f478ecf08a6d9b	prod-nginx-sg	vpc-0af1558f90c74dc52
□	–	sg-094fc70ab9045ac94	default	vpc-02c4d35114f2b80de
□	prod-backend-sg	sg-09f9c59a453ab0e68	prod-backend-sg	vpc-0af1558f90c74dc52

Instances:

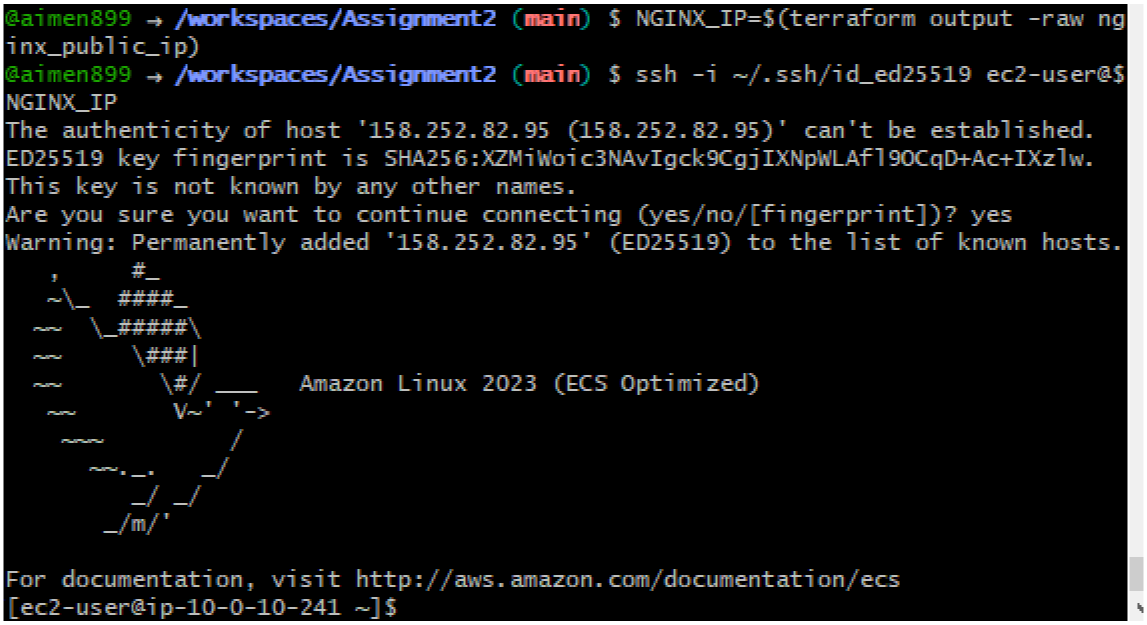


Instances (8) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
□	prod-web-2	i-05a4b2f0c7e269a0e	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	me-central-
□	prod-nginx-pr...	i-0c6324c46b209a4b3	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	me-central-
□	prod-web-1	i-0d4635fd40afe5e30	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	me-central-
□	prod-web-3	i-0e8f148d0ceba2dbb	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	me-central-

Part 5: Nginx Configuration & Testing

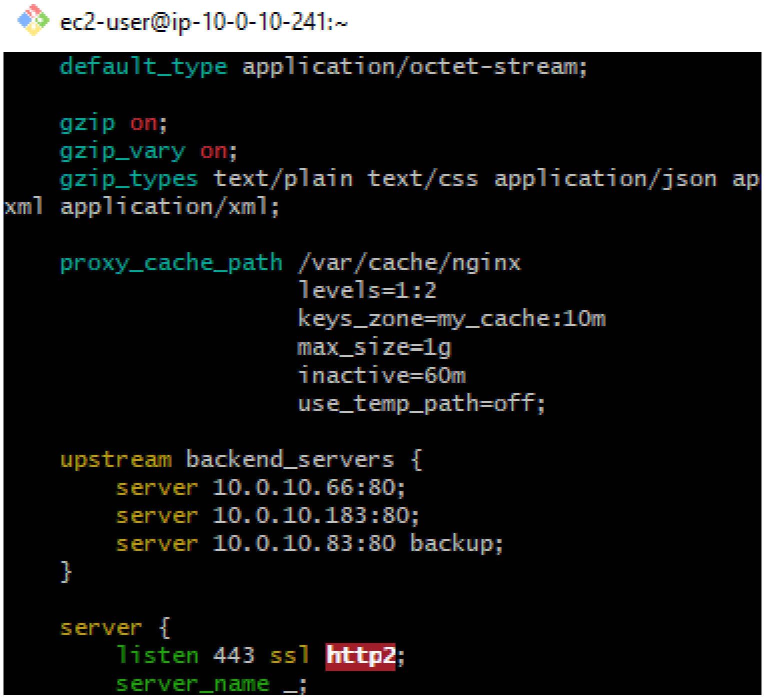
5.1 SSH into Nginx Server



```
@aimen899 → /workspaces/Assignment2 (main) $ NGINX_IP=$(terraform output -raw ng
inx_public_ip)
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@$
NGINX_IP
The authenticity of host '158.252.82.95 (158.252.82.95)' can't be established.
ED25519 key fingerprint is SHA256:XZMiWoic3NAvIgck9CgjIXNpWLAfl9OCqD+Ac+IXzlw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.252.82.95' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
[ec2-user@ip-10-0-10-241 ~]$
```

Update Nginx Configuration

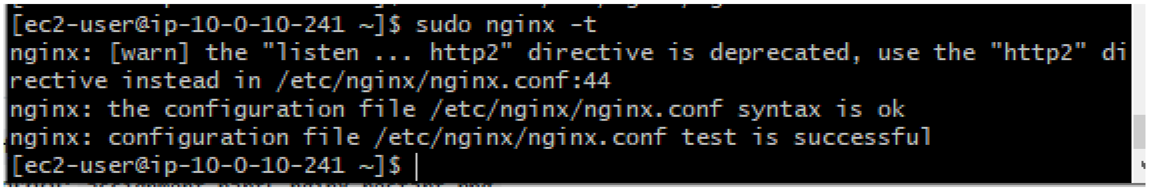
On the Nginx server:



ec2-user@ip-10-0-10-241:~

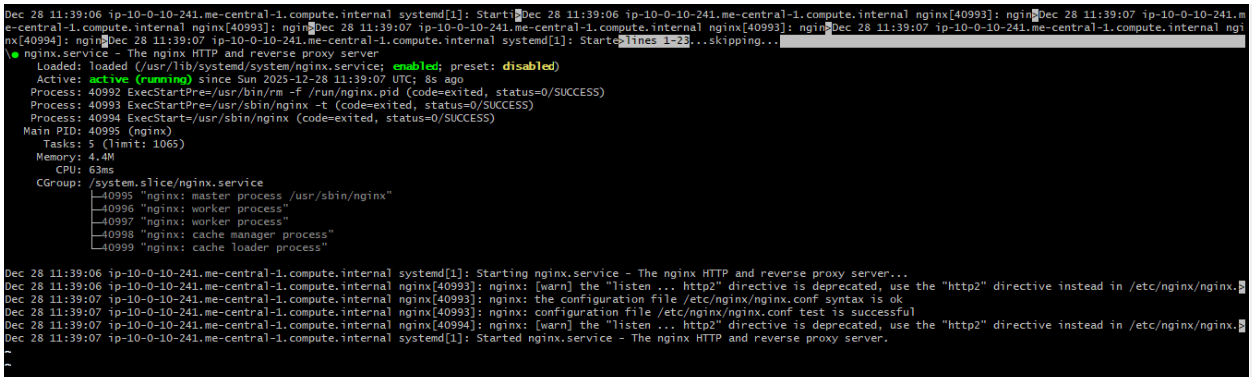
```
    default_type application/octet-stream;

    gzip on;
    gzip_vary on;
    gzip_types text/plain text/css application/json ap
xml application/xml;

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

    upstream backend_servers {
        server 10.0.10.66:80;
        server 10.0.10.183:80;
        server 10.0.10.83:80 backup;
    }

    server {
        listen 443 ssl http2;
        server_name _;
```

Test Nginx configuration



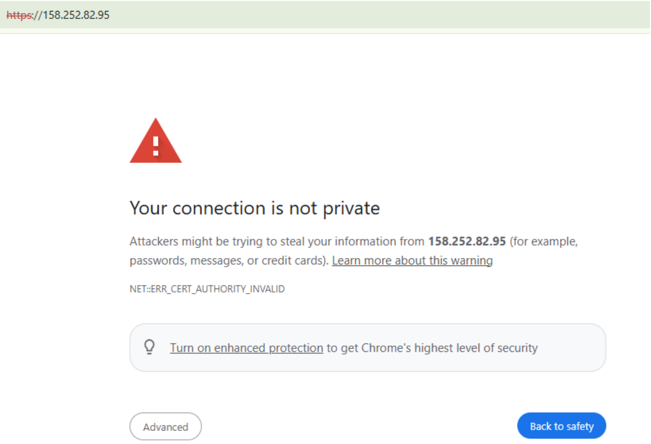
```
[ec2-user@ip-10-0-10-241 ~]$ sudo nginx -t
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" di
rective instead in /etc/nginx/nginx.conf:44
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-241 ~]$ |
```

Nginx Restart:



```
Dec 28 11:39:06 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Starti>Dec 28 11:39:06 ip-10-0-10-241.me-central-1.compute.internal nginx[40993]: ngin>Dec 28 11:39:07 ip-10-0-10-241.m
e-central-1.compute.internal nginx[40993]: ngin>Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal nginx[40993]: ngin>Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal ngi
nx[40994]: ngin>Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Starte>lines 1-23...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 11:39:07 UTC; 8s ago
    Process: 40992 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 40993 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 40994 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 40995 (nginx)
      Tasks: 5 (limit: 1065)
     Memory: 4.4M
        CPU: 63ms
     CGroup: /system.slice/nginx.service
             ├─40995 "nginx: master process /usr/sbin/nginx"
             ├─40996 "nginx: worker process"
             ├─40997 "nginx: worker process"
             ├─40998 "nginx: cache manager process"
             └─40999 "nginx: cache loader process"

Dec 28 11:39:06 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Dec 28 11:39:06 ip-10-0-10-241.me-central-1.compute.internal nginx[40993]: nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.>
Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal nginx[40993]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal nginx[40993]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal nginx[40994]: nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.>
Dec 28 11:39:07 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
~
~
```

5.2 Test Load Balancing



~~https~~://158.252.82.95

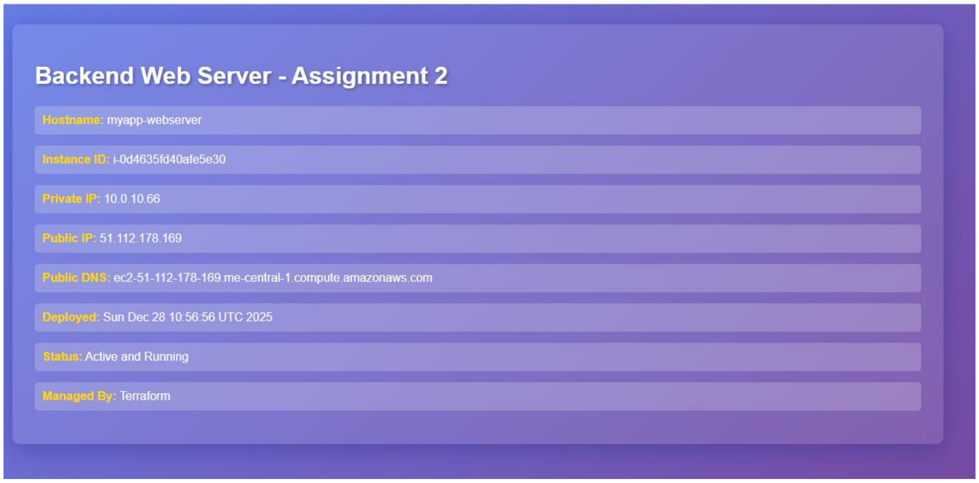
Your connection is not private

Attackers might be trying to steal your information from **158.252.82.95** (for example, passwords, messages, or credit cards). Learn more about this warning

NET::ERR_CERT_AUTHORITY_INVALID

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

Reload the page multiple times
Observe traffic alternating between web-1 and web-2
Web-1:



Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-0d4635fd40afe5e30

Private IP: 10.0.10.66

Public IP: 51.112.178.169

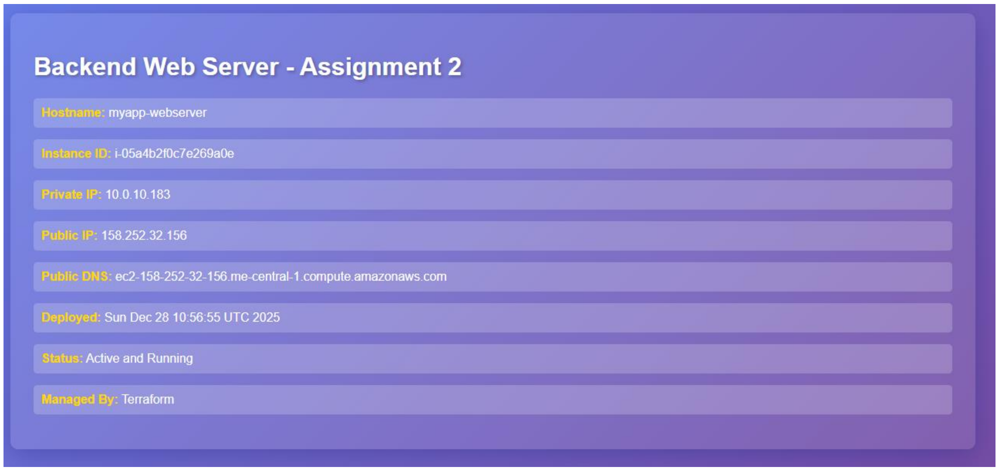
Public DNS: ec2-51-112-178-169.me-central-1.compute.amazonaws.com

Deployed: Sun Dec 28 10:56:56 UTC 2025

Status: Active and Running

Managed By: Terraform

Web-2:



Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-05a4b2f0c7e269a0e

Private IP: 10.0.10.183

Public IP: 158.252.32.156

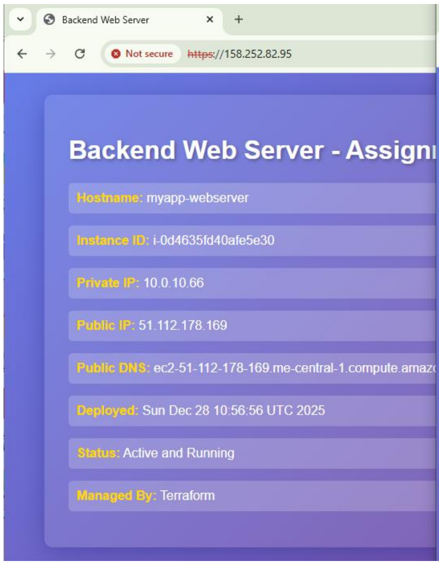
Public DNS: ec2-158-252-32-156.me-central-1.compute.amazonaws.com

Deployed: Sun Dec 28 10:56:55 UTC 2025

Status: Active and Running

Managed By: Terraform

Load Balancing Demo:



Backend Web Server - Assignı

Hostname: myapp-webserver

Instance ID: i-0d4635fd40afe5e30

Private IP: 10.0.10.66

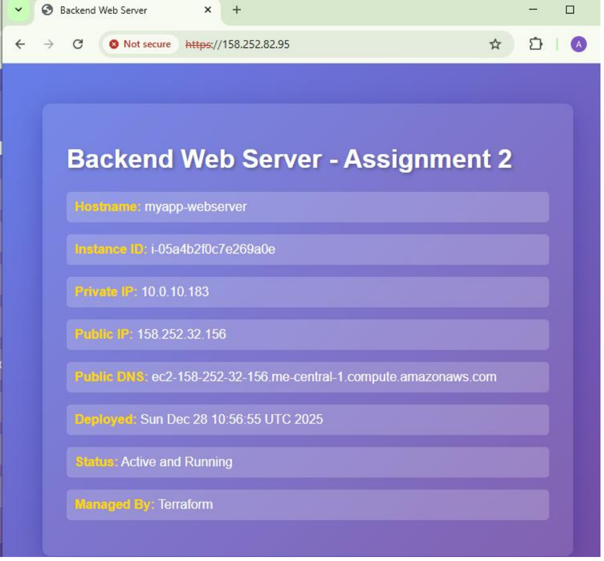
Public IP: 51.112.178.169

Public DNS: ec2-51-112-178-169.me-central-1.compute.amaz

Deployed: Sun Dec 28 10:56:56 UTC 2025

Status: Active and Running

Managed By: Terraform



Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-05a4b2f0c7e269a0e

Private IP: 10.0.10.183

Public IP: 158.252.32.156

Public DNS: ec2-158-252-32-156.me-central-1.compute.amazonaws.com

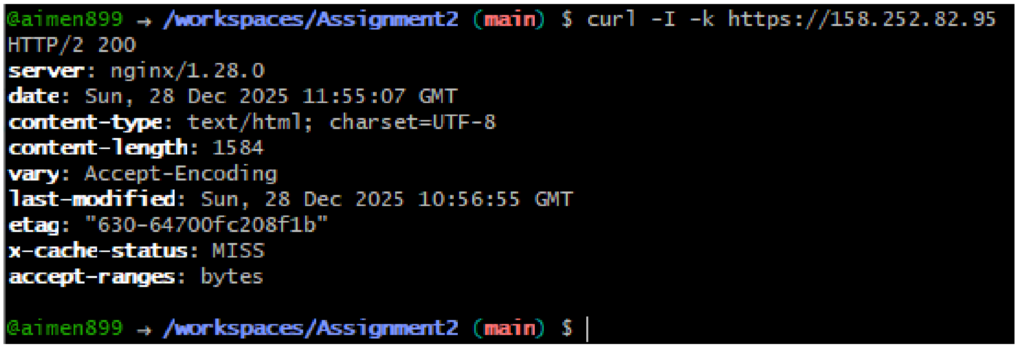
Deployed: Sun Dec 28 10:56:55 UTC 2025

Status: Active and Running

Managed By: Terraform

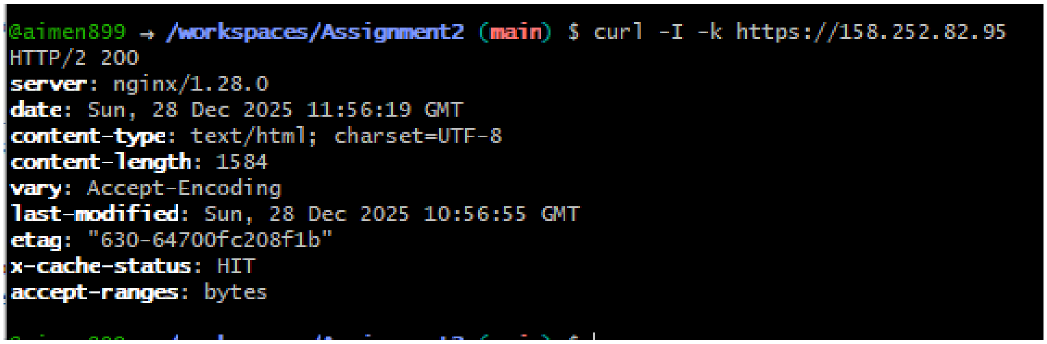
5.3 Test Cache Functionality

MISS on first request



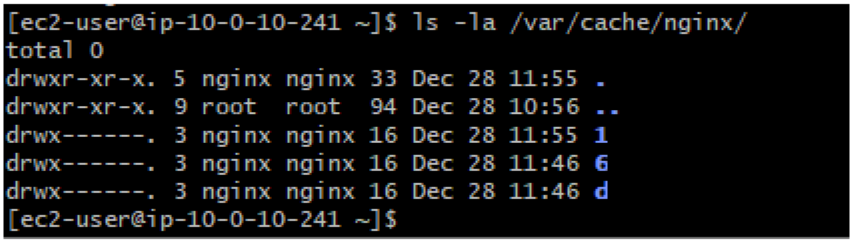
```
@aimen899 → /workspaces/Assignment2 (main) $ curl -I -k https://158.252.82.95
HTTP/2 200
server: nginx/1.28.0
date: Sun, 28 Dec 2025 11:55:07 GMT
content-type: text/html; charset=UTF-8
content-length: 1584
vary: Accept-Encoding
last-modified: Sun, 28 Dec 2025 10:56:55 GMT
etag: "630-64700fc208f1b"
x-cache-status: MISS
accept-ranges: bytes

@aimen899 → /workspaces/Assignment2 (main) $ |
```

HIT on subsequent request



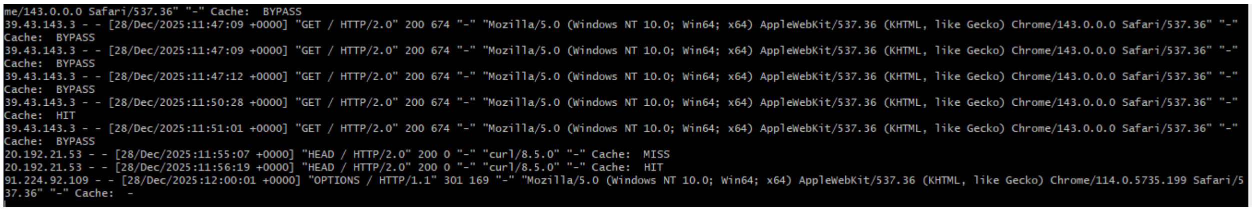
```
@aimen899 → /workspaces/Assignment2 (main) $ curl -I -k https://158.252.82.95
HTTP/2 200
server: nginx/1.28.0
date: Sun, 28 Dec 2025 11:56:19 GMT
content-type: text/html; charset=UTF-8
content-length: 1584
vary: Accept-Encoding
last-modified: Sun, 28 Dec 2025 10:56:55 GMT
etag: "630-64700fc208f1b"
x-cache-status: HIT
accept-ranges: bytes
```

Cache Directory:



```
[ec2-user@ip-10-0-10-241 ~]$ ls -la /var/cache/nginx/
total 0
drwxr-xr-x. 5 nginx nginx 33 Dec 28 11:55 .
drwxr-xr-x. 9 root  root  94 Dec 28 10:56 ..
drwx------. 3 nginx nginx 16 Dec 28 11:55 1
drwx------. 3 nginx nginx 16 Dec 28 11:46 6
drwx------. 3 nginx nginx 16 Dec 28 11:46 d
[ec2-user@ip-10-0-10-241 ~]$
```

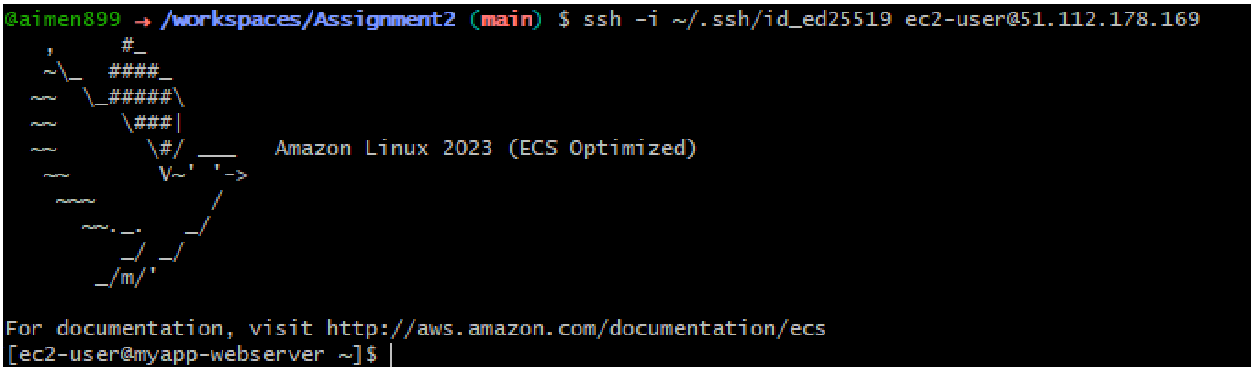
Access Log Cache:



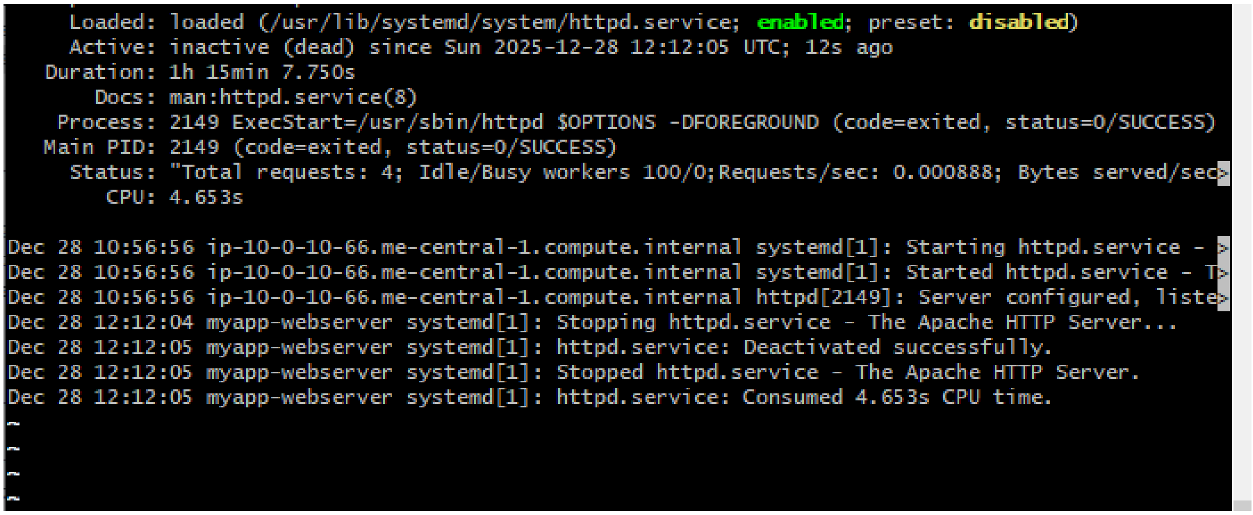
```
me/143.0.0.0 Safari/537.36" "-" Cache:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:47:09 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
Cache:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:47:09 +0000] "GET / HTTP/2.0" 200 674 "-" "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
39.43.143.3 - - [28/Dec/2025:11:47:12 +0000] "GET / HTTP/2.0" 200 674 "-" "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
39.43.143.3 - - [28/Dec/2025:11:50:28 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-"
Cache:  HIT
39.43.143.3 - - [28/Dec/2025:11:51:01 +0000] "GET / HTTP/2.0" 200 674 "-" "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
20.192.21.53 - - [28/Dec/2025:11:55:07 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.5.0" "-" Cache:  MISS
20.192.21.53 - - [28/Dec/2025:11:56:19 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.5.0" "-" Cache:  HIT
91.224.92.109 - - [28/Dec/2025:12:00:01 +0000] "OPTIONS / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199 Safari/5
37.36" "-" Cache:  -
```

5.4 Test High Availability

SSH into web-1 and stop Apache



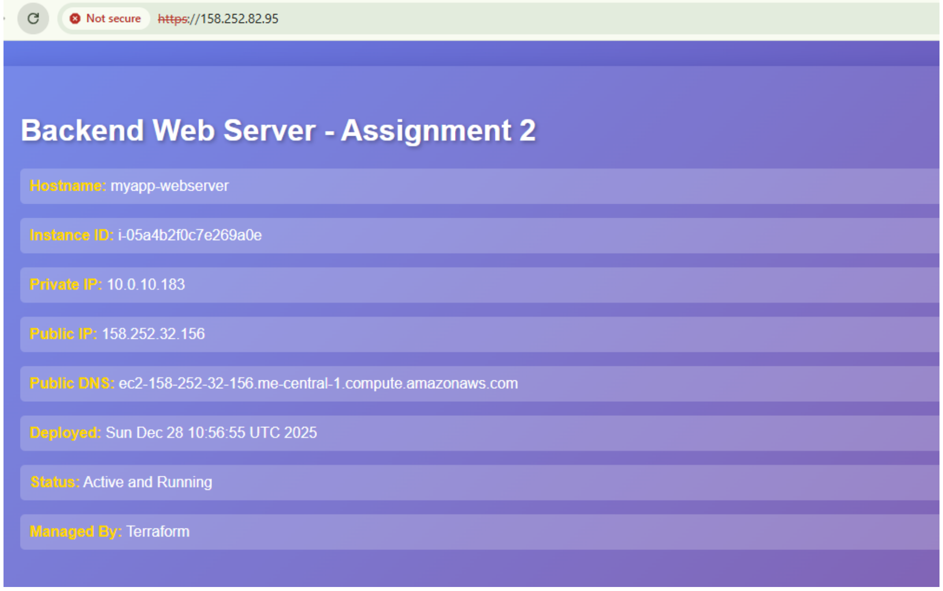
```
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@51.112.178.169
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
[ec2-user@myapp-webserver ~]$ |
```



```
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: inactive (dead) since Sun 2025-12-28 12:12:05 UTC; 12s ago
   Duration: 1h 15min 7.750s
       Docs: man:httpd.service(8)
    Process: 2149 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
   Main PID: 2149 (code=exited, status=0/SUCCESS)
     Status: "Total requests: 4; Idle/Busy workers 100/0;Requests/sec: 0.000888; Bytes served/sec>
        CPU: 4.653s

Dec 28 10:56:56 ip-10-0-10-66.me-central-1.compute.internal systemd[1]: Starting httpd.service - >
Dec 28 10:56:56 ip-10-0-10-66.me-central-1.compute.internal systemd[1]: Started httpd.service - T>
Dec 28 10:56:56 ip-10-0-10-66.me-central-1.compute.internal httpd[2149]: Server configured, liste>
Dec 28 12:12:04 myapp-webserver systemd[1]: Stopping httpd.service - The Apache HTTP Server...
Dec 28 12:12:05 myapp-webserver systemd[1]: httpd.service: Deactivated successfully.
Dec 28 12:12:05 myapp-webserver systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Dec 28 12:12:05 myapp-webserver systemd[1]: httpd.service: Consumed 4.653s CPU time.
~
~
~
~
```

Test - should only show web-2 now

On subsequent reloads, only web-2 server is shown:



~~https~~://158.252.82.95

Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-05a4b2f0c7e269a0e

Private IP: 10.0.10.183

Public IP: 158.252.32.156

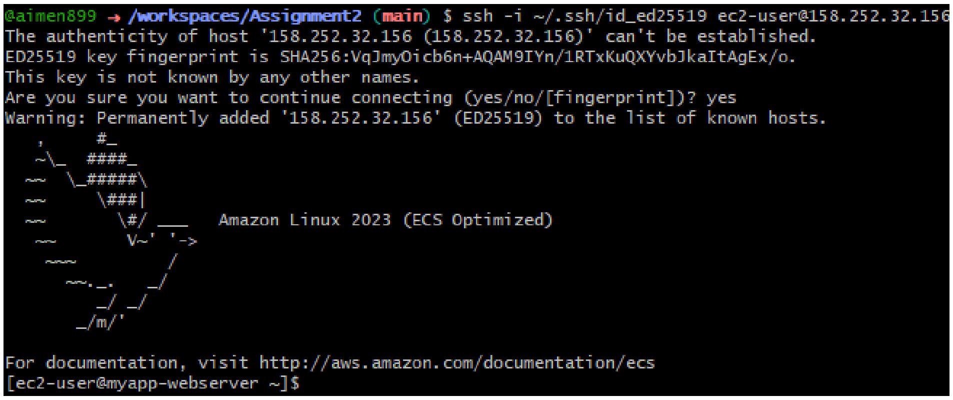
Public DNS: ec2-158-252-32-156.me-central-1.compute.amazonaws.com

Deployed: Sun Dec 28 10:56:55 UTC 2025

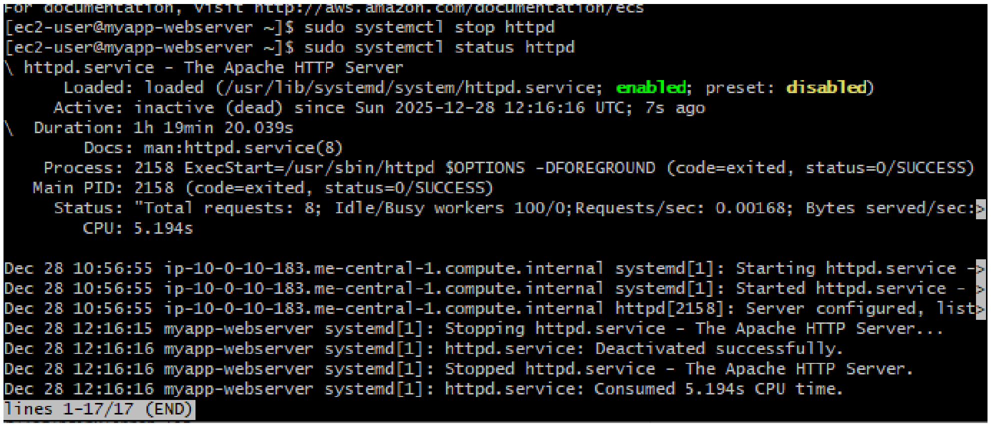
Status: Active and Running

Managed By: Terraform

SSH into web-2 and stop Apache

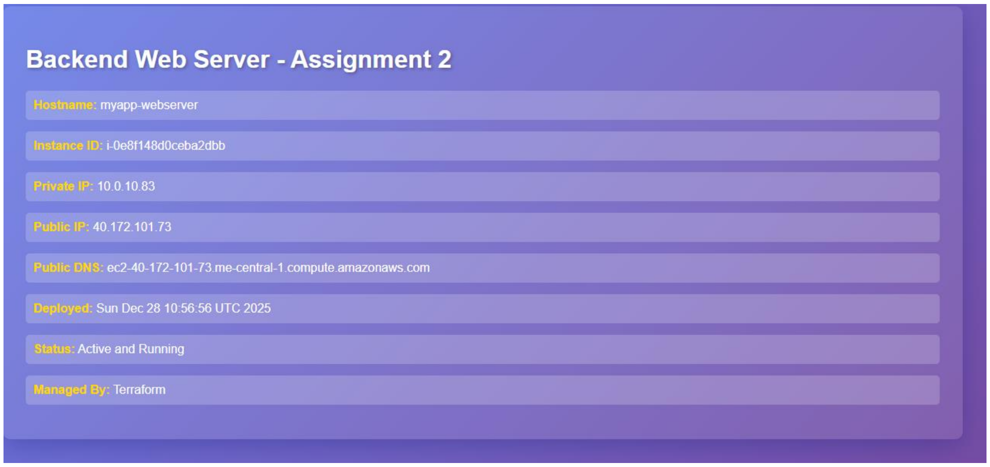


```
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.32.156
The authenticity of host '158.252.32.156 (158.252.32.156)' can't be established.
ED25519 key fingerprint is SHA256:VqJmyOicb6n+AQAM9IYn/1RTxKuQXYvbJkaItAgEx/o.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.252.32.156' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
[ec2-user@myapp-webserver ~]$
```



```
For documentation, visit http://aws.amazon.com/documentation/ecs
[ec2-user@myapp-webserver ~]$ sudo systemctl stop httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
\ httpd.service - The Apache HTTP Server
      Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: inactive (dead) since Sun 2025-12-28 12:16:16 UTC; 7s ago
\  Duration: 1h 19min 20.039s
        Docs: man:httpd.service(8)
    Process: 2158 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=0/SUCCESS)
   Main PID: 2158 (code=exited, status=0/SUCCESS)
     Status: "Total requests: 8; Idle/Busy workers 100/0;Requests/sec: 0.00168; Bytes served/sec:>
        CPU: 5.194s

Dec 28 10:56:55 ip-10-0-10-183.me-central-1.compute.internal systemd[1]: Starting httpd.service ->
Dec 28 10:56:55 ip-10-0-10-183.me-central-1.compute.internal systemd[1]: Started httpd.service - >
Dec 28 10:56:55 ip-10-0-10-183.me-central-1.compute.internal httpd[2158]: Server configured, list>
Dec 28 12:16:15 myapp-webserver systemd[1]: Stopping httpd.service - The Apache HTTP Server...
Dec 28 12:16:16 myapp-webserver systemd[1]: httpd.service: Deactivated successfully.
Dec 28 12:16:16 myapp-webserver systemd[1]: Stopped httpd.service - The Apache HTTP Server.
Dec 28 12:16:16 myapp-webserver systemd[1]: httpd.service: Consumed 5.194s CPU time.
lines 1-17/17 (END)
```

Test - should now show web-3 (backup activated)
Reload browser



Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-0e8f148d0ceba2dbb

Private IP: 10.0.10.83

Public IP: 40.172.101.73

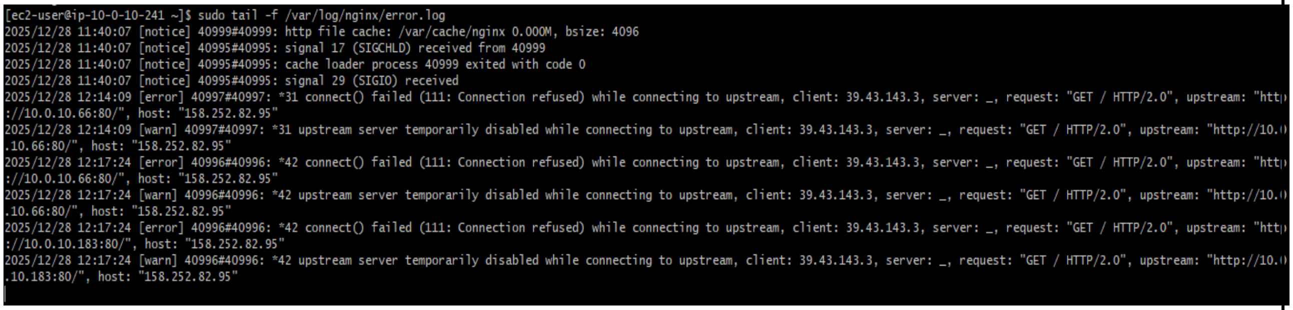
Public DNS: ec2-40-172-101-73.me-central-1.compute.amazonaws.com

Deployed: Sun Dec 28 10:56:56 UTC 2025

Status: Active and Running

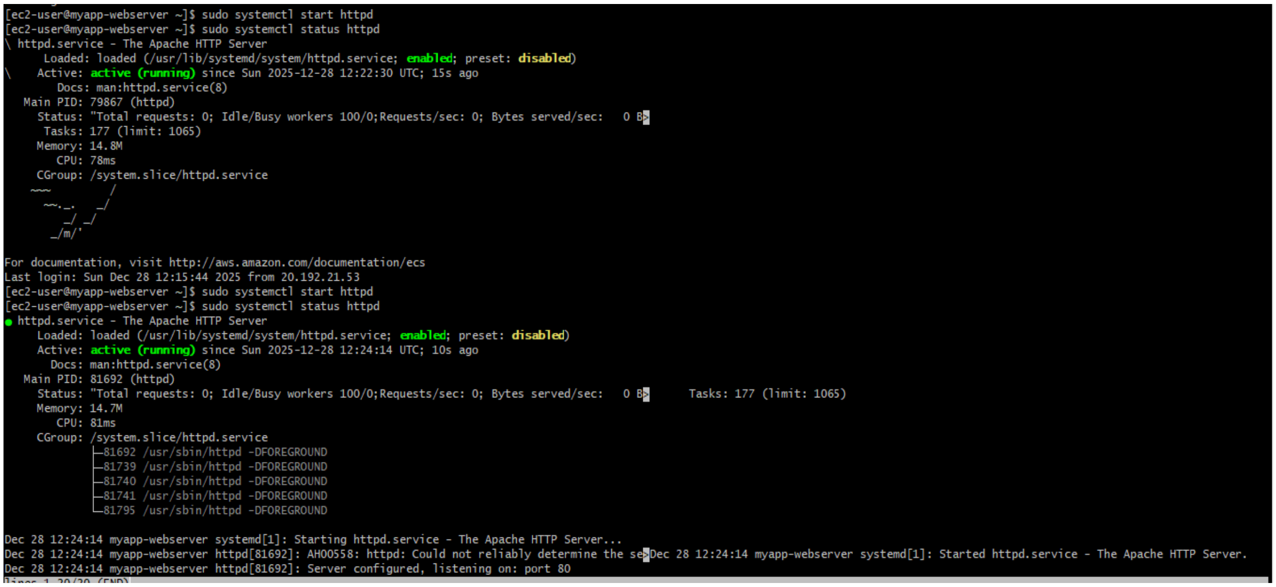
Managed By: Terraform

Check Nginx error logs



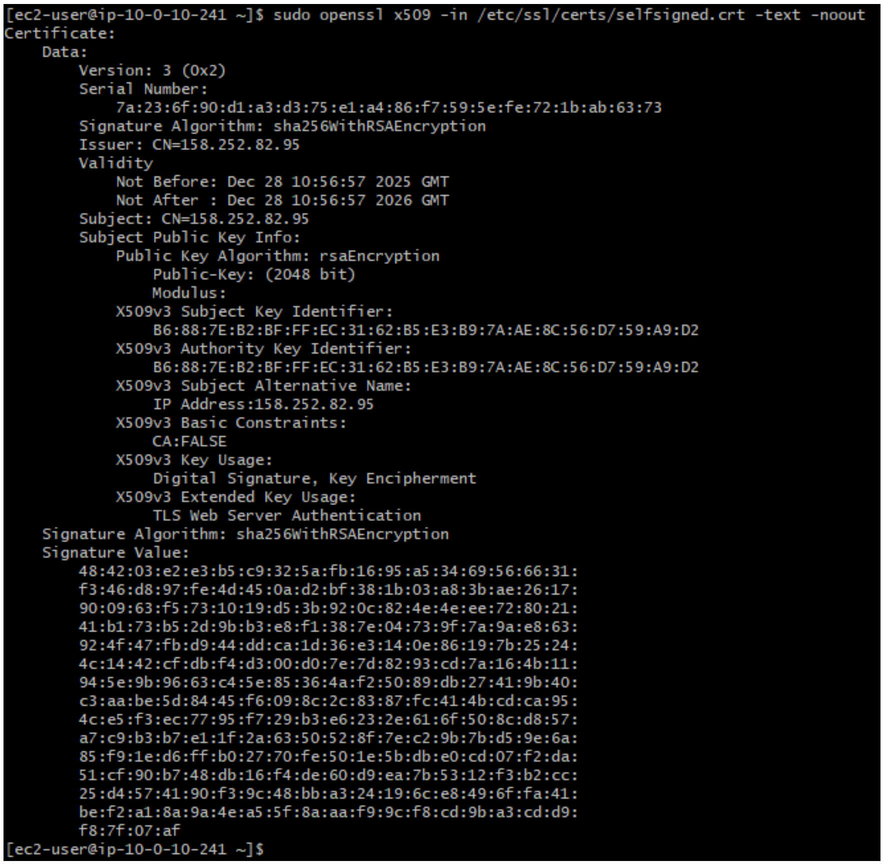
```
[ec2-user@ip-10-0-10-241 ~]$ sudo tail -f /var/log/nginx/error.log
2025/12/28 11:40:07 [notice] 40999#40999: http file cache: /var/cache/nginx 0.000M, bsize: 4096
2025/12/28 11:40:07 [notice] 40995#40995: signal 17 (SIGCHLD) received from 40999
2025/12/28 11:40:07 [notice] 40995#40995: cache loader process 40999 exited with code 0
2025/12/28 11:40:07 [notice] 40995#40995: signal 29 (SIGIO) received
2025/12/28 12:14:09 [error] 40997#40997: *31 connect() failed (111: Connection refused) while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "htt
://10.0.10.66:80/", host: "158.252.82.95"
2025/12/28 12:14:09 [warn] 40997#40997: *31 upstream server temporarily disabled while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0
.10.66:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [error] 40996#40996: *42 connect() failed (111: Connection refused) while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "htt
://10.0.10.66:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [warn] 40996#40996: *42 upstream server temporarily disabled while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0
.10.66:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [error] 40996#40996: *42 connect() failed (111: Connection refused) while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "htt
://10.0.10.183:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [warn] 40996#40996: *42 upstream server temporarily disabled while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0
.10.183:80/", host: "158.252.82.95"
```

Restart services on web-1 & web-2:



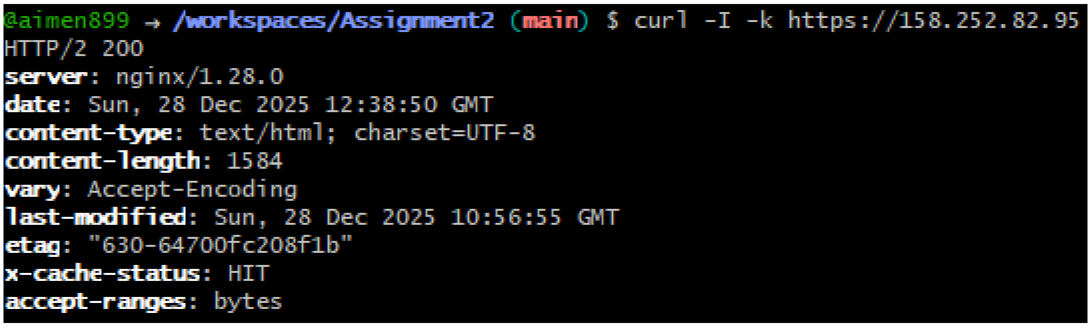
```
[ec2-user@myapp-webserver ~]$ sudo systemctl start httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
\ httpd.service - The Apache HTTP Server
      Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
\    Active: active (running) since Sun 2025-12-28 12:22:30 UTC; 15s ago
        Docs: man:httpd.service(8)
   Main PID: 79867 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B>
      Tasks: 177 (limit: 1065)
     Memory: 14.8M
        CPU: 78ms
     CGroup: /system.slice/httpd.service
    ~~~        /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 12:15:44 2025 from 20.192.21.53
[ec2-user@myapp-webserver ~]$ sudo systemctl start httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-12-28 12:24:14 UTC; 10s ago
       Docs: man:httpd.service(8)
   Main PID: 81692 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B>      Tasks: 177 (limit: 1065)
     Memory: 14.7M
        CPU: 81ms
     CGroup: /system.slice/httpd.service
             ├─81692 /usr/sbin/httpd -DFOREGROUND
             ├─81739 /usr/sbin/httpd -DFOREGROUND
             ├─81740 /usr/sbin/httpd -DFOREGROUND
             ├─81741 /usr/sbin/httpd -DFOREGROUND
             └─81795 /usr/sbin/httpd -DFOREGROUND

Dec 28 12:24:14 myapp-webserver systemd[1]: Starting httpd.service - The Apache HTTP Server...
Dec 28 12:24:14 myapp-webserver httpd[81692]: AH00558: httpd: Could not reliably determine the se>Dec 28 12:24:14 myapp-webserver systemd[1]: Started httpd.service - The Apache HTTP Server.
Dec 28 12:24:14 myapp-webserver httpd[81692]: Server configured, listening on: port 80
lines 1-20/20 (END)
```

5.5 Security & Performance Analysis



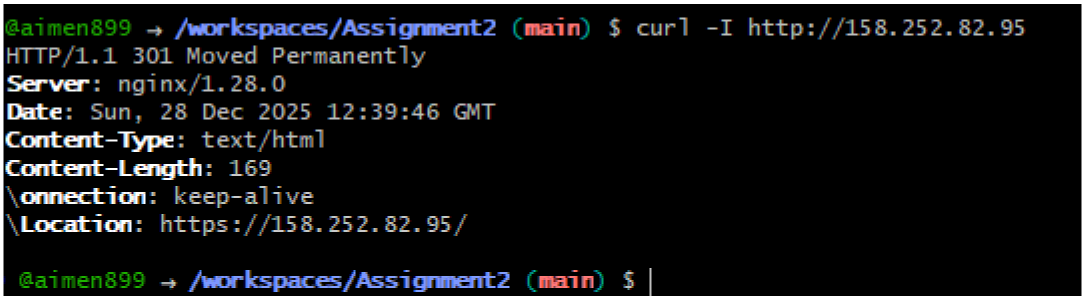
```
[ec2-user@ip-10-0-10-241 ~]$ sudo openssl x509 -in /etc/ssl/certs/selfsigned.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            7a:23:6f:90:d1:a3:d3:75:e1:a4:86:f7:59:5e:fe:72:1b:ab:63:73
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=158.252.82.95
        Validity
            Not Before: Dec 28 10:56:57 2025 GMT
            Not After : Dec 28 10:56:57 2026 GMT
        Subject: CN=158.252.82.95
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
            X509v3 Subject Key Identifier:
                B6:88:7E:B2:BF:FF:EC:31:62:B5:E3:B9:7A:AE:8C:56:D7:59:A9:D2
            X509v3 Authority Key Identifier:
                B6:88:7E:B2:BF:FF:EC:31:62:B5:E3:B9:7A:AE:8C:56:D7:59:A9:D2
            X509v3 Subject Alternative Name:
                IP Address:158.252.82.95
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        48:42:03:e2:e3:b5:c9:32:5a:fb:16:95:a5:34:69:56:66:31:
        f3:46:d8:97:fe:4d:45:0a:d2:bf:38:1b:03:a8:3b:ae:26:17:
        90:09:63:f5:73:10:19:d5:3b:92:0c:82:4e:4e:ee:72:80:21:
        41:b1:73:b5:2d:9b:b3:e8:f1:38:7e:04:73:9f:7a:9a:e8:63:
        92:4f:47:fb:d9:44:dd:ca:1d:36:e3:14:0e:86:19:7b:25:24:
        4c:14:42:cf:db:f4:d3:00:d0:7e:7d:82:93:cd:7a:16:4b:11:
        94:5e:9b:96:63:c4:5e:85:36:4a:f2:50:89:db:27:41:9b:40:
        c3:aa:be:5d:84:45:f6:09:8c:2c:83:87:fc:41:4b:cd:ca:95:
        4c:e5:f3:ec:77:95:f7:29:b3:e6:23:2e:61:6f:50:8c:d8:57:
        a7:c9:b3:b7:e1:1f:2a:63:50:52:8f:7e:c2:9b:7b:d5:9e:6a:
        85:f9:1e:d6:ff:b0:27:70:fe:50:1e:5b:db:e0:cd:07:f2:da:
        51:cf:90:b7:48:db:16:f4:de:60:d9:ea:7b:53:12:f3:b2:cc:
        25:d4:57:41:90:f3:9c:48:bb:a3:24:19:6c:e8:49:6f:fa:41:
        be:f2:a1:8a:9a:4e:a5:5f:8a:aa:f9:9c:f8:cd:9b:a3:cd:d9:
        f8:7f:07:af
[ec2-user@ip-10-0-10-241 ~]$
```

Check security headers



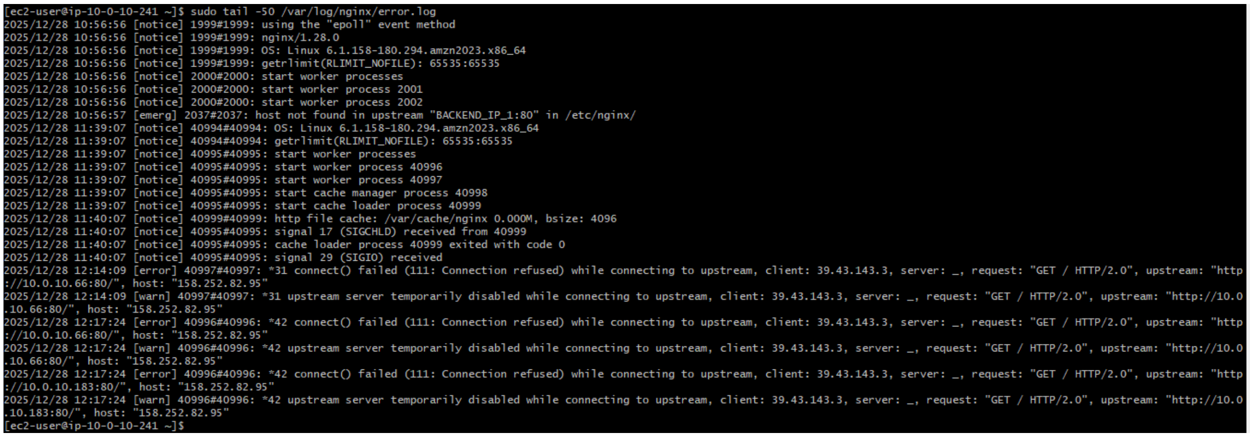
```
@aimen899 → /workspaces/Assignment2 (main) $ curl -I -k https://158.252.82.95
HTTP/2 200
server: nginx/1.28.0
date: Sun, 28 Dec 2025 12:38:50 GMT
content-type: text/html; charset=UTF-8
content-length: 1584
vary: Accept-Encoding
last-modified: Sun, 28 Dec 2025 10:56:55 GMT
etag: "630-64700fc208f1b"
x-cache-status: HIT
accept-ranges: bytes
```

Test HTTP redirect

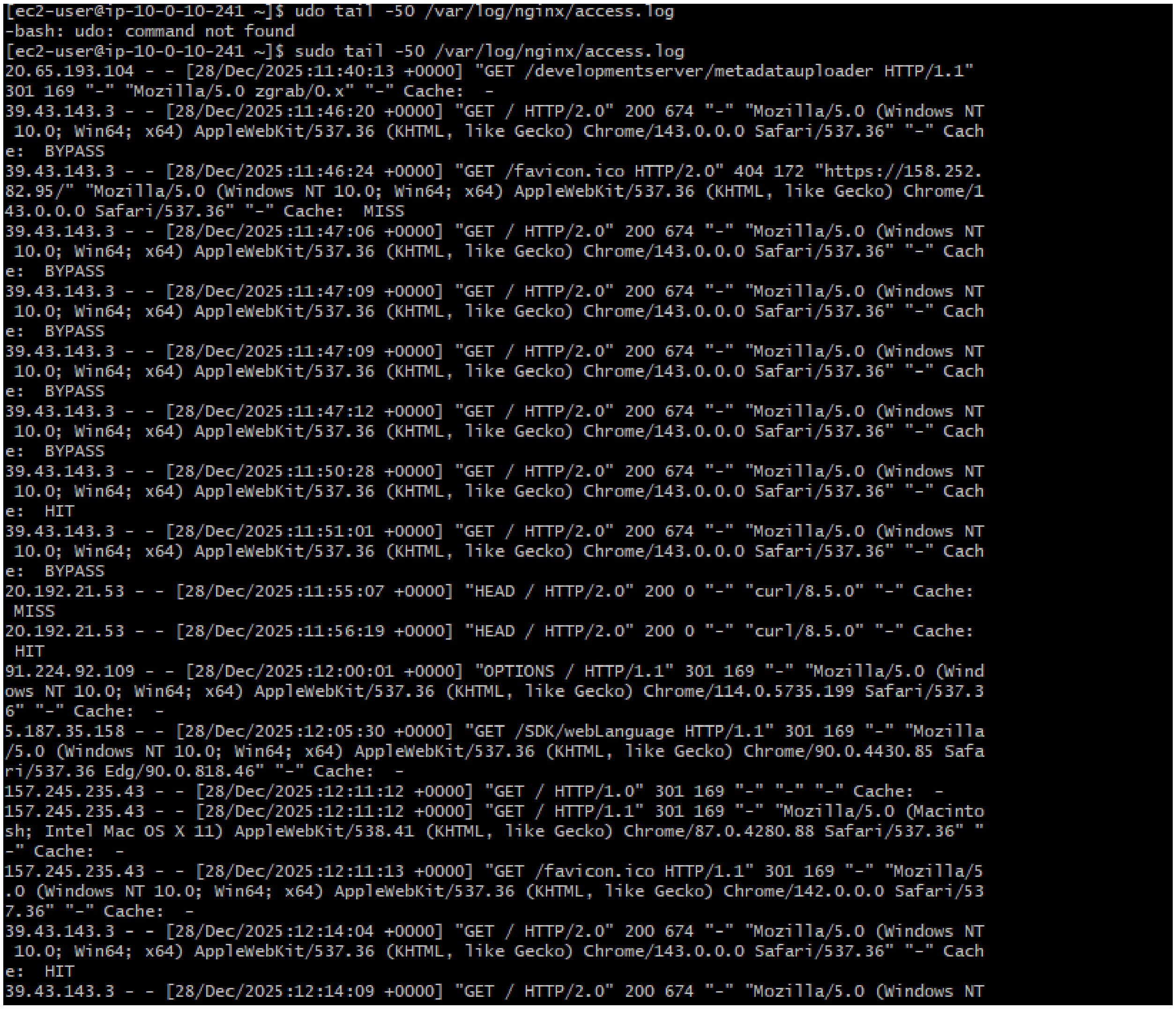


```
@aimen899 → /workspaces/Assignment2 (main) $ curl -I http://158.252.82.95
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Sun, 28 Dec 2025 12:39:46 GMT
Content-Type: text/html
Content-Length: 169
\onnection: keep-alive
\Location: https://158.252.82.95/

 @aimen899 → /workspaces/Assignment2 (main) $ |
```

View logs

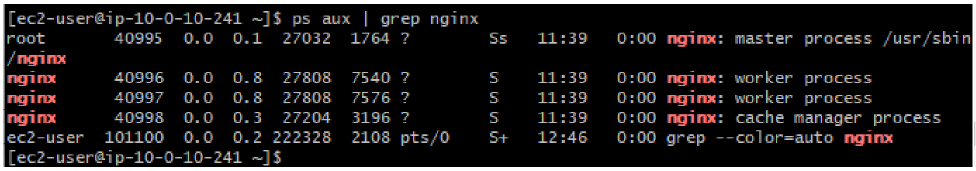


```
[ec2-user@ip-10-0-10-241 ~]$ sudo tail -50 /var/log/nginx/error.log
2025/12/28 10:56:56 [notice] 1999#1999: using the "epoll" event method
2025/12/28 10:56:56 [notice] 1999#1999: nginx/1.28.0
2025/12/28 10:56:56 [notice] 1999#1999: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/28 10:56:56 [notice] 1999#1999: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/28 10:56:56 [notice] 2000#2000: start worker processes
2025/12/28 10:56:56 [notice] 2000#2000: start worker process 2001
2025/12/28 10:56:56 [notice] 2000#2000: start worker process 2002
2025/12/28 10:56:57 [emerg] 2037#2037: host not found in upstream "BACKEND_IP_1:80" in /etc/nginx/
2025/12/28 11:39:07 [notice] 40994#40994: OS: Linux 6.1.158-180.294.amzn2023.x86_64
2025/12/28 11:39:07 [notice] 40994#40994: getrlimit(RLIMIT_NOFILE): 65535:65535
2025/12/28 11:39:07 [notice] 40995#40995: start worker processes
2025/12/28 11:39:07 [notice] 40995#40995: start worker process 40996
2025/12/28 11:39:07 [notice] 40995#40995: start worker process 40997
2025/12/28 11:39:07 [notice] 40995#40995: start cache manager process 40998
2025/12/28 11:39:07 [notice] 40995#40995: start cache loader process 40999
2025/12/28 11:40:07 [notice] 40999#40999: http file cache: /var/cache/nginx 0.000M, bsize: 4096
2025/12/28 11:40:07 [notice] 40995#40995: signal 17 (SIGCHLD) received from 40999
2025/12/28 11:40:07 [notice] 40995#40995: cache loader process 40999 exited with code 0
2025/12/28 11:40:07 [notice] 40995#40995: signal 29 (SIGIO) received
2025/12/28 12:14:09 [error] 40997#40997: *31 connect() failed (111: Connection refused) while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http
://10.0.10.66:80/", host: "158.252.82.95"
2025/12/28 12:14:09 [warn] 40997#40997: *31 upstream server temporarily disabled while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0
.10.66:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [error] 40996#40996: *42 connect() failed (111: Connection refused) while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http
://10.0.10.66:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [warn] 40996#40996: *42 upstream server temporarily disabled while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0
.10.66:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [error] 40996#40996: *42 connect() failed (111: Connection refused) while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http
://10.0.10.183:80/", host: "158.252.82.95"
2025/12/28 12:17:24 [warn] 40996#40996: *42 upstream server temporarily disabled while connecting to upstream, client: 39.43.143.3, server: _, request: "GET / HTTP/2.0", upstream: "http://10.0
.10.183:80/", host: "158.252.82.95"
[ec2-user@ip-10-0-10-241 ~]$
```



```
[ec2-user@ip-10-0-10-241 ~]$ udo tail -50 /var/log/nginx/access.log
-bash: udo: command not found
[ec2-user@ip-10-0-10-241 ~]$ sudo tail -50 /var/log/nginx/access.log
20.65.193.104 - - [28/Dec/2025:11:40:13 +0000] "GET /developmentserver/metadatauploader HTTP/1.1"
301 169 "-" "Mozilla/5.0 zgrab/0.x" "-" Cache:  -
39.43.143.3 - - [28/Dec/2025:11:46:20 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:46:24 +0000] "GET /favicon.ico HTTP/2.0" 404 172 "https://158.252.
82.95/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/1
43.0.0.0 Safari/537.36" "-" Cache:  MISS
39.43.143.3 - - [28/Dec/2025:11:47:06 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:47:09 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:47:09 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:47:12 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  BYPASS
39.43.143.3 - - [28/Dec/2025:11:50:28 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  HIT
39.43.143.3 - - [28/Dec/2025:11:51:01 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  BYPASS
20.192.21.53 - - [28/Dec/2025:11:55:07 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.5.0" "-" Cache:
 MISS
20.192.21.53 - - [28/Dec/2025:11:56:19 +0000] "HEAD / HTTP/2.0" 200 0 "-" "curl/8.5.0" "-" Cache:
 HIT
91.224.92.109 - - [28/Dec/2025:12:00:01 +0000] "OPTIONS / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199 Safari/537.3
6" "-" Cache:  -
5.187.35.158 - - [28/Dec/2025:12:05:30 +0000] "GET /SDK/webLanguage 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 Safa
ri/537.36 Edg/90.0.818.46" "-" Cache:  -
157.245.235.43 - - [28/Dec/2025:12:11:12 +0000] "GET / HTTP/1.0" 301 169 "-" "-" "-" Cache:  -
157.245.235.43 - - [28/Dec/2025:12:11:12 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 (Macinto
sh; Intel Mac OS X 11) AppleWebKit/538.41 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36" "
-" Cache:  -
157.245.235.43 - - [28/Dec/2025:12:11:13 +0000] "GET /favicon.ico HTTP/1.1" 301 169 "-" "Mozilla/5
.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/53
7.36" "-" Cache:  -
39.43.143.3 - - [28/Dec/2025:12:14:04 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" "-" Cach
e:  HIT
39.43.143.3 - - [28/Dec/2025:12:14:09 +0000] "GET / HTTP/2.0" 200 674 "-" "Mozilla/5.0 (Windows NT
```

Check Nginx worker processes

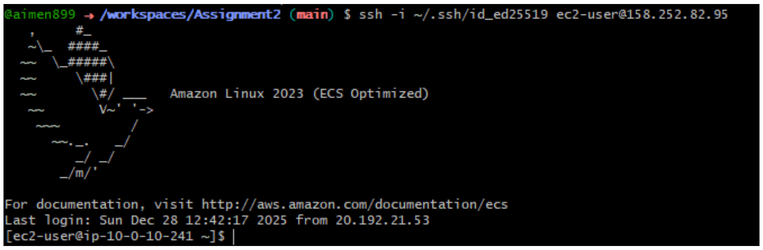


```
[ec2-user@ip-10-0-10-241 ~]$ ps aux | grep nginx
root       40995  0.0  0.1  27032  1764 ?        Ss   11:39   0:00 nginx: master process /usr/sbin
/nginx
nginx      40996  0.0  0.8  27808  7540 ?        S    11:39   0:00 nginx: worker process
nginx      40997  0.0  0.8  27808  7576 ?        S    11:39   0:00 nginx: worker process
nginx      40998  0.0  0.3  27204  3196 ?        S    11:39   0:00 nginx: cache manager process
ec2-user  101100  0.0  0.2 222328  2108 pts/0    S+   12:46   0:00 grep --color=auto nginx
[ec2-user@ip-10-0-10-241 ~]$
```

Bonus Tasks

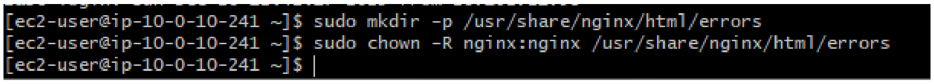
Bonus 1: Custom Error Pages

1. SSH into Nginx Server



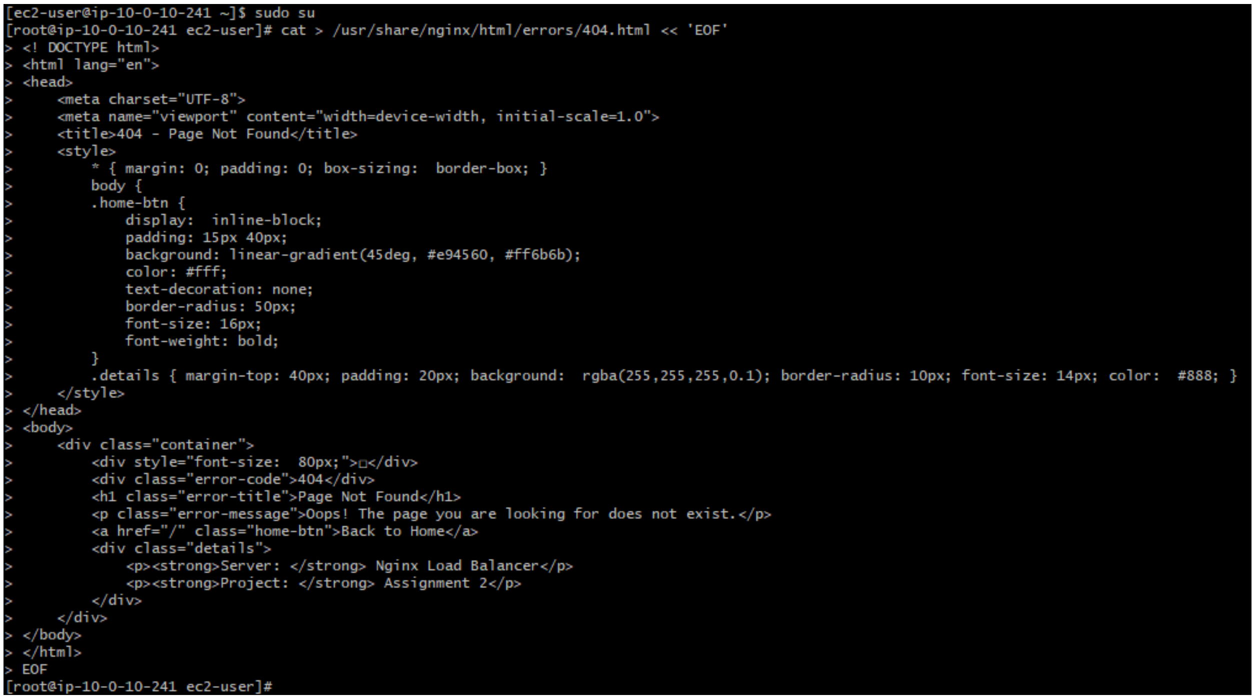
```
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.82.95
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 12:42:17 2025 from 20.192.21.53
[ec2-user@ip-10-0-10-241 ~]$ |
```

2. Create Error Pages Directory



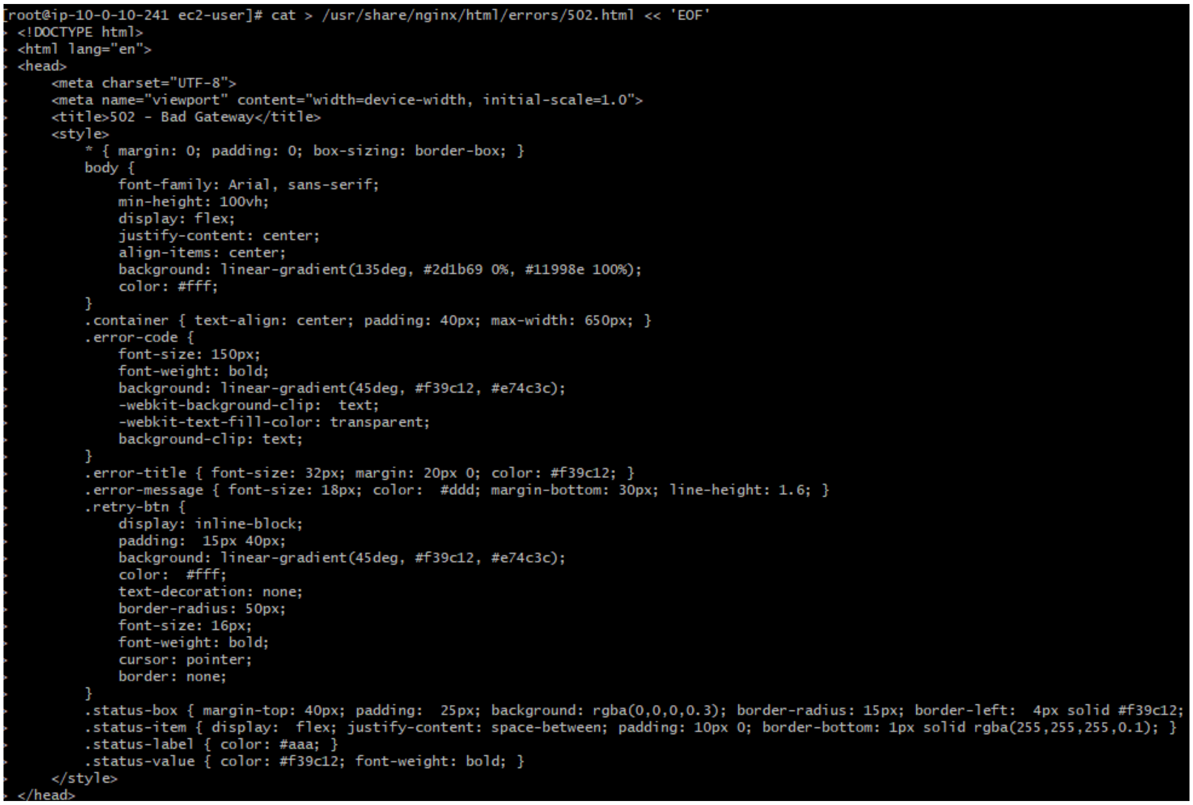
```
[ec2-user@ip-10-0-10-241 ~]$ sudo mkdir -p /usr/share/nginx/html/errors
[ec2-user@ip-10-0-10-241 ~]$ sudo chown -R nginx:nginx /usr/share/nginx/html/errors
[ec2-user@ip-10-0-10-241 ~]$ |
```

3. Create Custom 404 Error Page



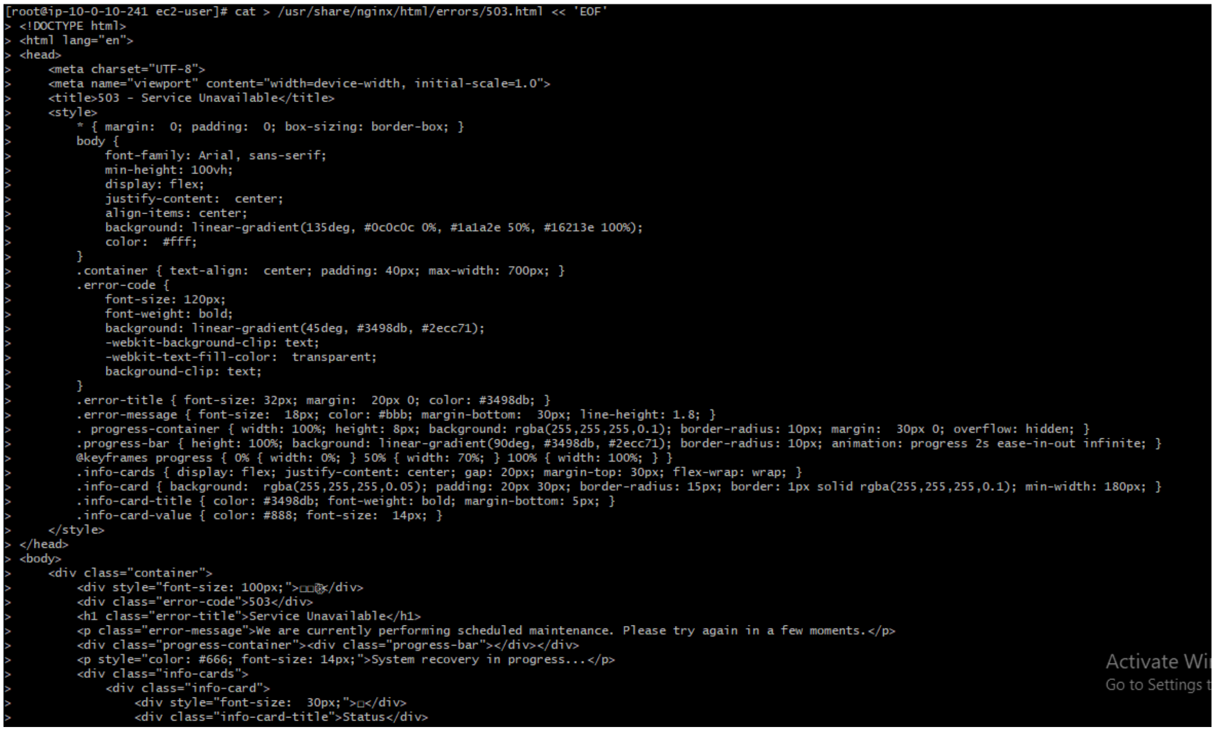
```
[ec2-user@ip-10-0-10-241 ~]$ sudo su
[root@ip-10-0-10-241 ec2-user]# cat > /usr/share/nginx/html/errors/404.html << 'EOF'
> <! DOCTYPE html>
> <html lang="en">
> <head>
>     <meta charset="UTF-8">
>     <meta name="viewport" content="width=device-width, initial-scale=1.0">
>     <title>404 - Page Not Found</title>
>     <style>
>         * { margin: 0; padding: 0; box-sizing:  border-box; }
>         body {
>         .home-btn {
>             display:  inline-block;
>             padding: 15px 40px;
>             background: linear-gradient(45deg, #e94560, #ff6b6b);
>             color: #fff;
>             text-decoration: none;
>             border-radius: 50px;
>             font-size: 16px;
>             font-weight: bold;
>         }
>         .details { margin-top: 40px; padding: 20px; background:  rgba(255,255,255,0.1); border-radius: 10px; font-size: 14px; color:  #888; }
>     </style>
> </head>
> <body>
>     <div class="container">
>         <div style="font-size:  80px;">□</div>
>         <div class="error-code">404</div>
>         <h1 class="error-title">Page Not Found</h1>
>         <p class="error-message">Oops! The page you are looking for does not exist.</p>
>         <a href="/" class="home-btn">Back to Home</a>
>         <div class="details">
>             <p><strong>Server: </strong> Nginx Load Balancer</p>
>             <p><strong>Project: </strong> Assignment 2</p>
>         </div>
>     </div>
> </body>
> </html>
> EOF
[root@ip-10-0-10-241 ec2-user]#
```

4. Create Custom 502 Error Page - Bad Gateway Error Page



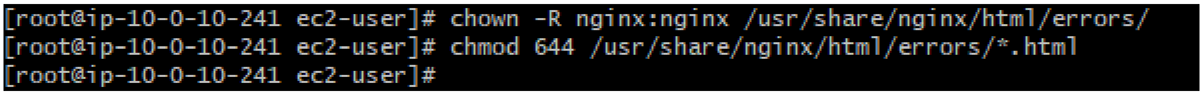
```
[root@ip-10-0-10-241 ec2-user]# cat > /usr/share/nginx/html/errors/502.html << 'EOF'
> <!DOCTYPE html>
> <html lang="en">
> <head>
>     <meta charset="UTF-8">
>     <meta name="viewport" content="width=device-width, initial-scale=1.0">
>     <title>502 - Bad Gateway</title>
>     <style>
>         * { margin: 0; padding: 0; box-sizing: border-box; }
>         body {
>             font-family: Arial, sans-serif;
>             min-height: 100vh;
>             display: flex;
>             justify-content: center;
>             align-items: center;
>             background: linear-gradient(135deg, #2d1b69 0%, #11998e 100%);
>             color: #fff;
>         }
>         .container { text-align: center; padding: 40px; max-width: 650px; }
>         .error-code {
>             font-size: 150px;
>             font-weight: bold;
>             background: linear-gradient(45deg, #f39c12, #e74c3c);
>             -webkit-background-clip:  text;
>             -webkit-text-fill-color: transparent;
>             background-clip: text;
>         }
>         .error-title { font-size: 32px; margin: 20px 0; color: #f39c12; }
>         .error-message { font-size: 18px; color:  #ddd; margin-bottom: 30px; line-height: 1.6; }
>         .retry-btn {
>             display: inline-block;
>             padding:  15px 40px;
>             background: linear-gradient(45deg, #f39c12, #e74c3c);
>             color:  #fff;
>             text-decoration: none;
>             border-radius: 50px;
>             font-size: 16px;
>             font-weight: bold;
>             cursor: pointer;
>             border: none;
>         }
>         .status-box { margin-top: 40px; padding:  25px; background: rgba(0,0,0,0.3); border-radius: 15px; border-left:  4px solid #f39c12;
>         .status-item { display:  flex; justify-content: space-between; padding: 10px 0; border-bottom: 1px solid rgba(255,255,255,0.1); }
>         .status-label { color: #aaa; }
>         .status-value { color: #f39c12; font-weight: bold; }
>     </style>
> </head>
```

5. Create 503 error Page - Service Unavailable Error Page



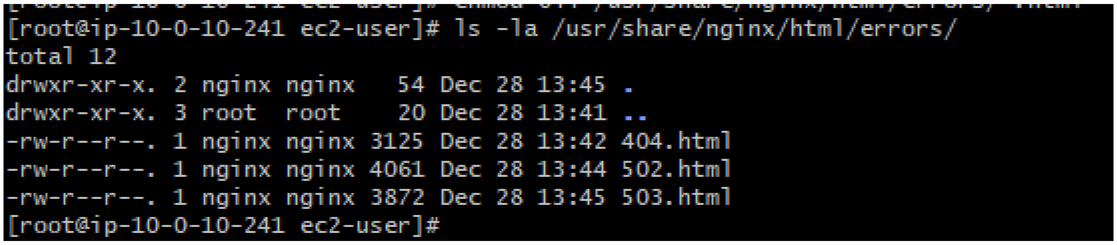
```
[root@ip-10-0-10-241 ec2-user]# cat > /usr/share/nginx/html/errors/503.html << 'EOF'
> <!DOCTYPE html>
> <html lang="en">
> <head>
>     <meta charset="UTF-8">
>     <meta name="viewport" content="width=device-width, initial-scale=1.0">
>     <title>503 - Service Unavailable</title>
>     <style>
>         * { margin:  0; padding:  0; box-sizing: border-box; }
>         body {
>             font-family: Arial, sans-serif;
>             min-height: 100vh;
>             display: flex;
>             justify-content:  center;
>             align-items: center;
>             background: linear-gradient(135deg, #0c0c0c 0%, #1a1a2e 50%, #16213e 100%);
>             color:  #fff;
>         }
>         .container { text-align:  center; padding: 40px; max-width: 700px; }
>         .error-code {
>             font-size: 120px;
>             font-weight: bold;
>             background: linear-gradient(45deg, #3498db, #2ecc71);
>             -webkit-background-clip: text;
>             -webkit-text-fill-color:  transparent;
>             background-clip: text;
>         }
>         .error-title { font-size: 32px; margin:  20px 0; color: #3498db; }
>         .error-message { font-size:  18px; color: #bbb; margin-bottom:  30px; line-height: 1.8; }
>         . progress-container { width: 100%; height: 8px; background: rgba(255,255,255,0.1); border-radius: 10px; margin:  30px 0; overflow: hidden; }
>         .progress-bar { height: 100%; background: linear-gradient(90deg, #3498db, #2ecc71); border-radius: 10px; animation: progress 2s ease-in-out infinite; }
>         @keyframes progress { 0% { width: 0%; } 50% { width: 70%; } 100% { width: 100%; } }
>         .info-cards { display: flex; justify-content: center; gap: 20px; margin-top: 30px; flex-wrap: wrap; }
>         .info-card { background:  rgba(255,255,255,0.05); padding: 20px 30px; border-radius: 15px; border: 1px solid rgba(255,255,255,0.1); min-width: 180px; }
>         .info-card-title { color: #3498db; font-weight: bold; margin-bottom: 5px; }
>         .info-card-value { color: #888; font-size:  14px; }
>     </style>
> </head>
> <body>
>     <div class="container">
>         <div style="font-size: 100px;">□□</div>
>         <div class="error-code">503</div>
>         <h1 class="error-title">Service Unavailable</h1>
>         <p class="error-message">We are currently performing scheduled maintenance. Please try again in a few moments.</p>
>         <div class="progress-container"><div class="progress-bar"></div></div>
>         <p style="color: #666; font-size: 14px;">System recovery in progress...</p>
>         <div class="info-cards">
>             <div class="info-card">
>                 <div style="font-size:  30px;">□</div>
>                 <div class="info-card-title">Status</div>
```

6. Set Proper Permissions



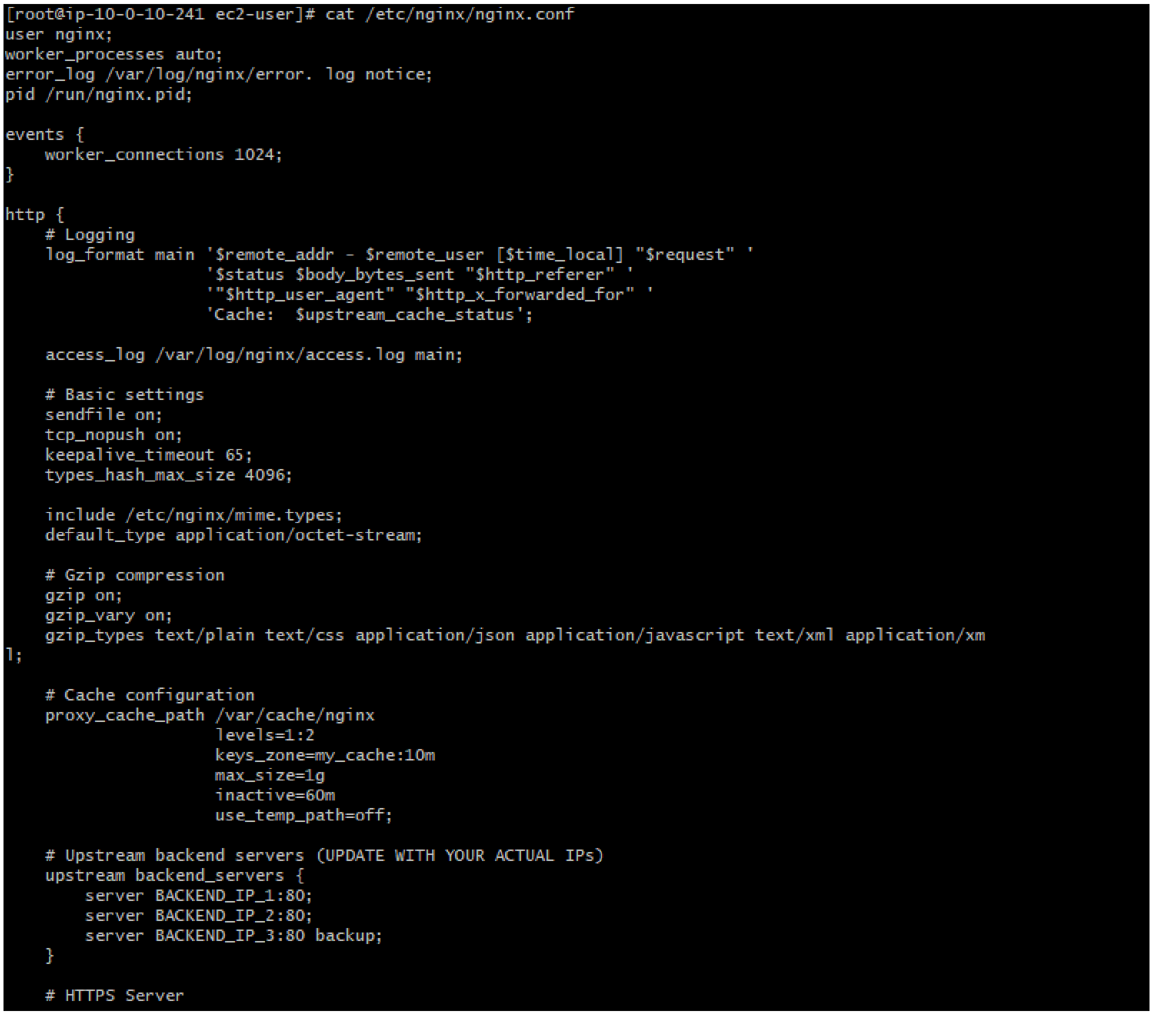
```
[root@ip-10-0-10-241 ec2-user]# chown -R nginx:nginx /usr/share/nginx/html/errors/
[root@ip-10-0-10-241 ec2-user]# chmod 644 /usr/share/nginx/html/errors/*.html
[root@ip-10-0-10-241 ec2-user]#
```

7. Verify files permissions



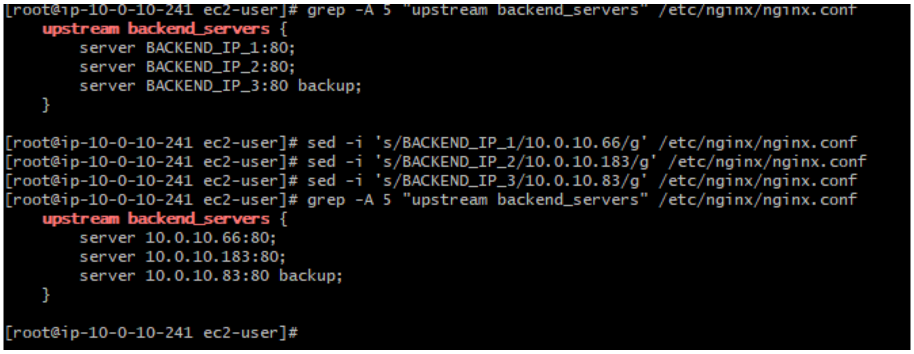
```
[root@ip-10-0-10-241 ec2-user]# ls -la /usr/share/nginx/html/errors/
total 12
drwxr-xr-x. 2 nginx nginx   54 Dec 28 13:45 .
drwxr-xr-x. 3 root  root    20 Dec 28 13:41 ..
-rw-r--r--. 1 nginx nginx 3125 Dec 28 13:42 404.html
-rw-r--r--. 1 nginx nginx 4061 Dec 28 13:44 502.html
-rw-r--r--. 1 nginx nginx 3872 Dec 28 13:45 503.html
[root@ip-10-0-10-241 ec2-user]#
```

8. Update Nginx Configuration for Custom Error Pages



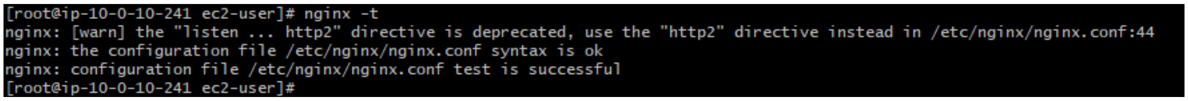
```
[root@ip-10-0-10-241 ec2-user]# cat /etc/nginx/nginx.conf
user nginx;
worker_processes auto;
error_log /var/log/nginx/error. log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

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

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

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

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

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

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

    # Upstream backend servers (UPDATE WITH YOUR ACTUAL IPs)
    upstream backend_servers {
        server BACKEND_IP_1:80;
        server BACKEND_IP_2:80;
        server BACKEND_IP_3:80 backup;
    }

    # HTTPS Server
```

9. Update Backend IPs in Configuration

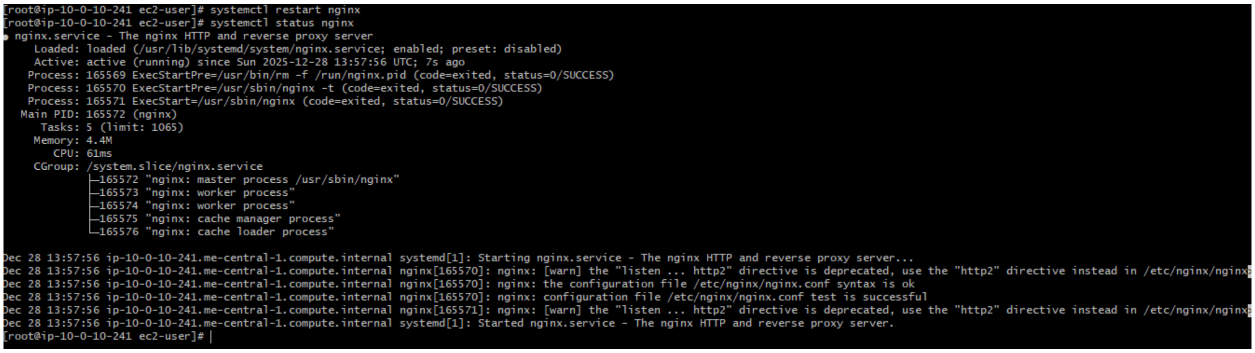


```
[root@ip-10-0-10-241 ec2-user]# grep -A 5 "upstream backend_servers" /etc/nginx/nginx.conf
    upstream backend_servers {
        server BACKEND_IP_1:80;
        server BACKEND_IP_2:80;
        server BACKEND_IP_3:80 backup;
    }

[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_1/10.0.10.66/g' /etc/nginx/nginx.conf
[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_2/10.0.10.183/g' /etc/nginx/nginx.conf
[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_3/10.0.10.83/g' /etc/nginx/nginx.conf
[root@ip-10-0-10-241 ec2-user]# grep -A 5 "upstream backend_servers" /etc/nginx/nginx.conf
    upstream backend_servers {
        server 10.0.10.66:80;
        server 10.0.10.183:80;
        server 10.0.10.83:80 backup;
    }

[root@ip-10-0-10-241 ec2-user]#
```

10. Test and Restart Nginx

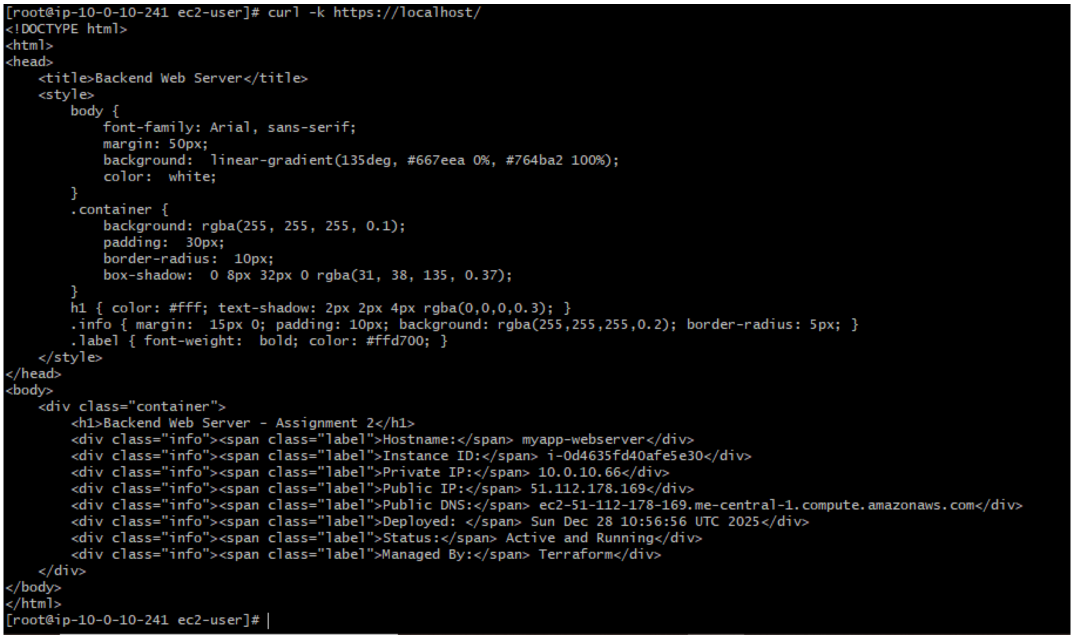


```
[root@ip-10-0-10-241 ec2-user]# nginx -t
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx.conf:44
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@ip-10-0-10-241 ec2-user]#
```



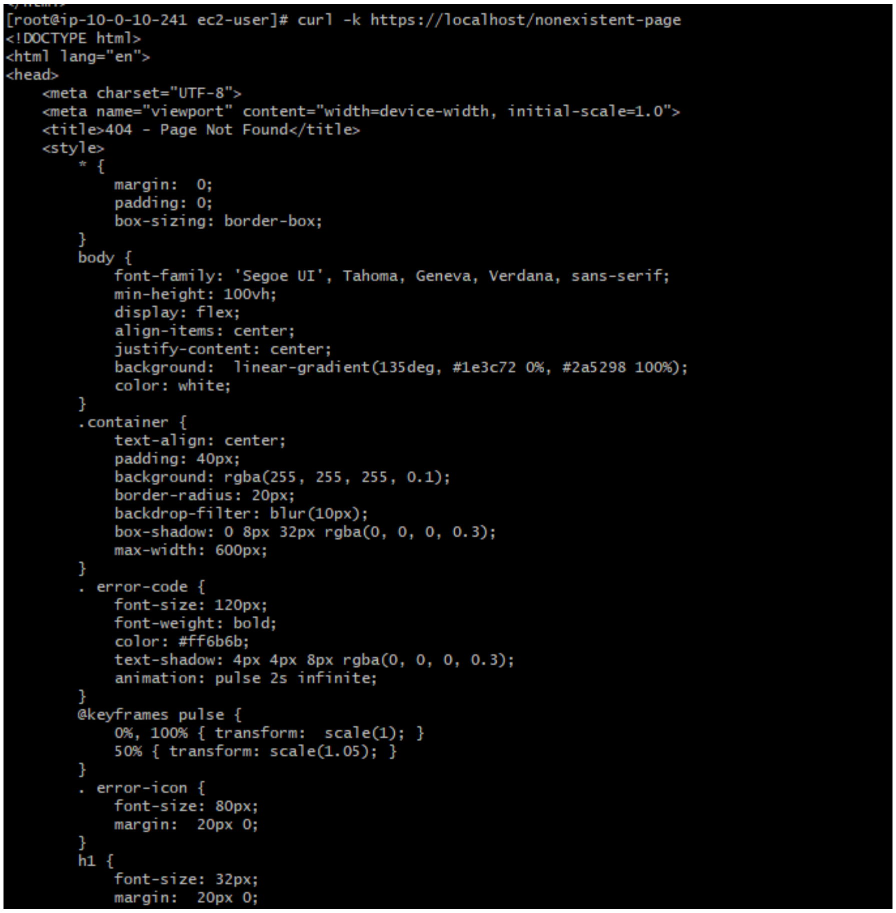
```
[root@ip-10-0-10-241 ec2-user]# systemctl restart nginx
[root@ip-10-0-10-241 ec2-user]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-12-28 13:57:56 UTC; 7s ago
    Process: 165569 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 165570 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 165571 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 165572 (nginx)
      Tasks: 5 (limit: 1065)
     Memory: 4.4M
        CPU: 61ms
     CGroup: /system.slice/nginx.service
             ├─165572 "nginx: master process /usr/sbin/nginx"
             ├─165573 "nginx: worker process"
             ├─165574 "nginx: worker process"
             ├─165575 "nginx: cache manager process"
             └─165576 "nginx: cache loader process"

Dec 28 13:57:56 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Dec 28 13:57:56 ip-10-0-10-241.me-central-1.compute.internal nginx[165570]: nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx>
Dec 28 13:57:56 ip-10-0-10-241.me-central-1.compute.internal nginx[165570]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Dec 28 13:57:56 ip-10-0-10-241.me-central-1.compute.internal nginx[165570]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Dec 28 13:57:56 ip-10-0-10-241.me-central-1.compute.internal nginx[165571]: nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in /etc/nginx/nginx>
Dec 28 13:57:56 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-10-0-10-241 ec2-user]# |
```

11. Verify Everything is Working



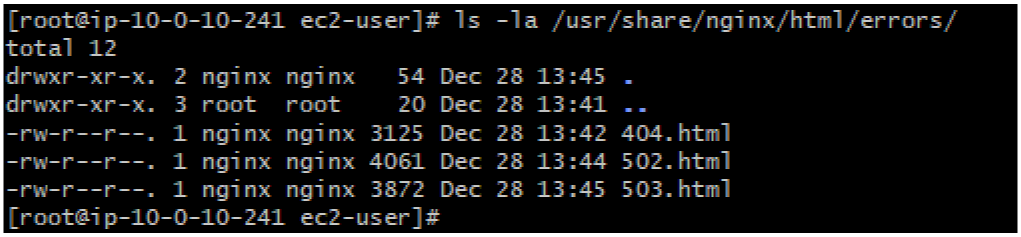
```
[root@ip-10-0-10-241 ec2-user]# curl -k https://localhost/
<!DOCTYPE html>
<html>
<head>
    <title>Backend Web Server</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 50px;
            background:  linear-gradient(135deg, #667eea 0%, #764ba2 100%);
            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 - Assignment 2</h1>
        <div class="info"><span class="label">Hostname:</span> myapp-webserver</div>
        <div class="info"><span class="label">Instance ID:</span> i-0d4635fd40afe5e30</div>
        <div class="info"><span class="label">Private IP:</span> 10.0.10.66</div>
        <div class="info"><span class="label">Public IP:</span> 51.112.178.169</div>
        <div class="info"><span class="label">Public DNS:</span> ec2-51-112-178-169.me-central-1.compute.amazonaws.com</div>
        <div class="info"><span class="label">Deployed: </span> Sun Dec 28 10:56:56 UTC 2025</div>
        <div class="info"><span class="label">Status:</span> Active and Running</div>
        <div class="info"><span class="label">Managed By:</span> Terraform</div>
    </div>
</body>
</html>
[root@ip-10-0-10-241 ec2-user]# |
```

Test 404 error page



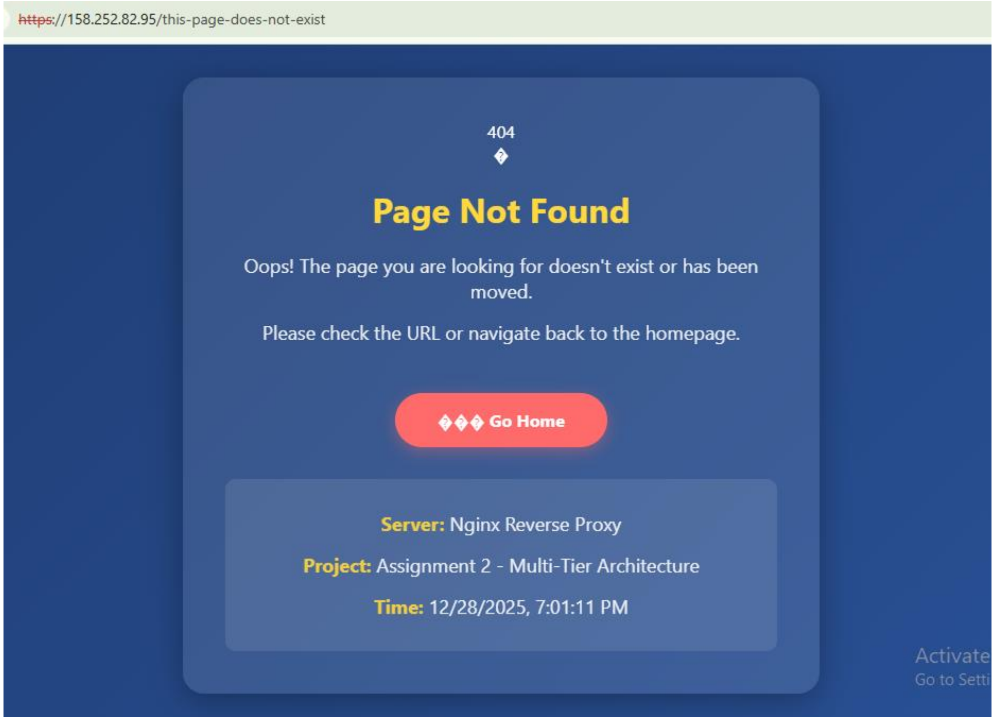
```
</html>
[root@ip-10-0-10-241 ec2-user]# curl -k https://localhost/nonexistent-page
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>404 - Page Not Found</title>
    <style>
        * {
            margin:  0;
            padding: 0;
            box-sizing: border-box;
        }
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            min-height: 100vh;
            display: flex;
            align-items: center;
            justify-content: center;
            background:  linear-gradient(135deg, #1e3c72 0%, #2a5298 100%);
            color: white;
        }
        .container {
            text-align: center;
            padding: 40px;
            background: rgba(255, 255, 255, 0.1);
            border-radius: 20px;
            backdrop-filter: blur(10px);
            box-shadow: 0 8px 32px rgba(0, 0, 0, 0.3);
            max-width: 600px;
        }
        . error-code {
            font-size: 120px;
            font-weight: bold;
            color: #ff6b6b;
            text-shadow: 4px 4px 8px rgba(0, 0, 0, 0.3);
            animation: pulse 2s infinite;
        }
        @keyframes pulse {
            0%, 100% { transform:  scale(1); }
            50% { transform: scale(1.05); }
        }
        . error-icon {
            font-size: 80px;
            margin:  20px 0;
        }
        h1 {
            font-size: 32px;
            margin:  20px 0;
```

Check error pages directory



```
[root@ip-10-0-10-241 ec2-user]# ls -la /usr/share/nginx/html/errors/
total 12
drwxr-xr-x. 2 nginx nginx   54 Dec 28 13:45 .
drwxr-xr-x. 3 root  root    20 Dec 28 13:41 ..
-rw-r--r--. 1 nginx nginx 3125 Dec 28 13:42 404.html
-rw-r--r--. 1 nginx nginx 4061 Dec 28 13:44 502.html
-rw-r--r--. 1 nginx nginx 3872 Dec 28 13:45 503.html
[root@ip-10-0-10-241 ec2-user]#
```

12. Test 404 Page



~~https~~://158.252.82.95/this-page-does-not-exist

404

�

Page Not Found

Oops! The page you are looking for doesn't exist or has been moved.

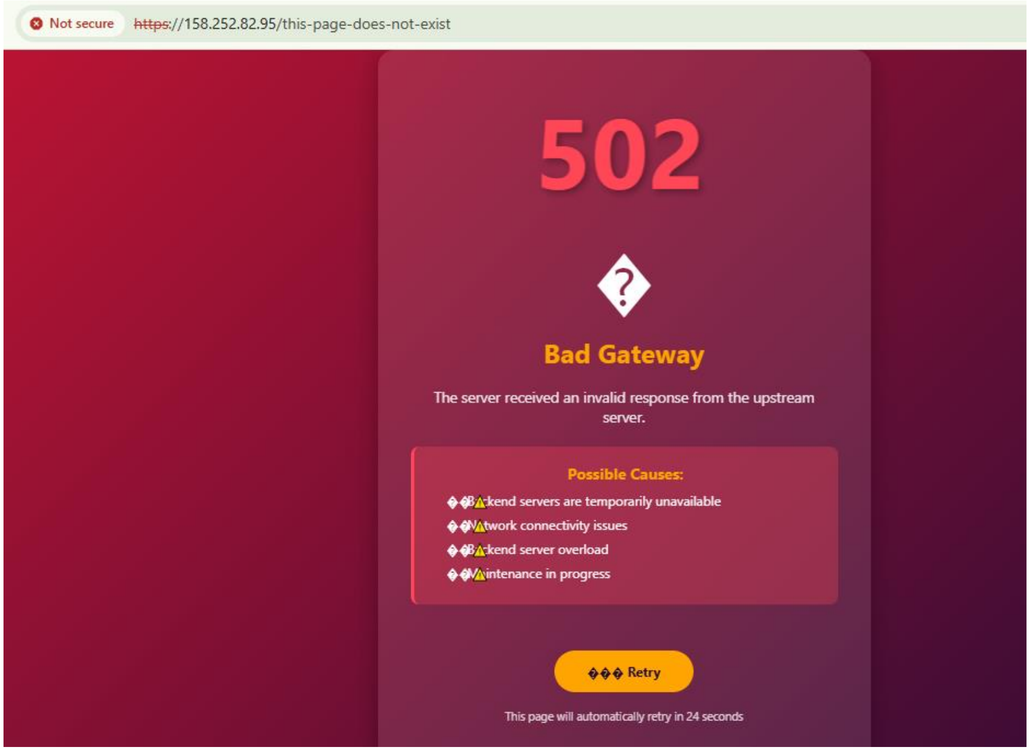
Please check the URL or navigate back to the homepage.

Server: Nginx Reverse Proxy

Project: Assignment 2 - Multi-Tier Architecture

Time: 12/28/2025, 7:01:11 PM

13. Test 502 Page



~~https~~://158.252.82.95/this-page-does-not-exist

502

Bad Gateway

The server received an invalid response from the upstream server.

Possible Causes:

��B⚠kend servers are temporarily unavailable

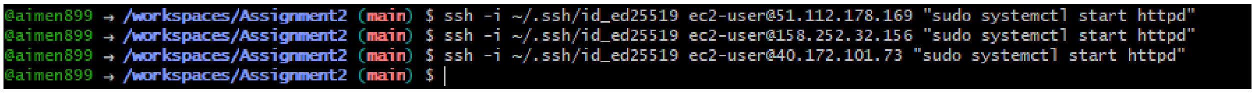
��N⚠twork connectivity issues

��B⚠kend server overload

��M⚠intenance in progress

This page will automatically retry in 24 seconds

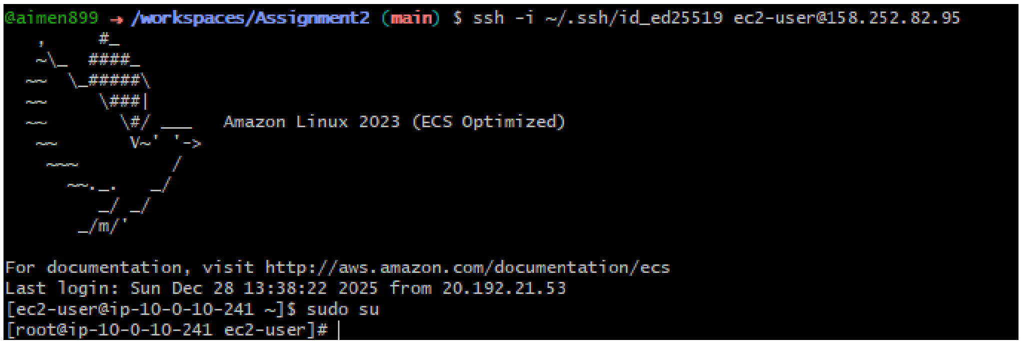
14. Restart Backend Servers



```
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@51.112.178.169 "sudo systemctl start httpd"
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.32.156 "sudo systemctl start httpd"
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@40.172.101.73 "sudo systemctl start httpd"
@aimen899 → /workspaces/Assignment2 (main) $ |
```

Bonus 2: Implement Rate Limiting

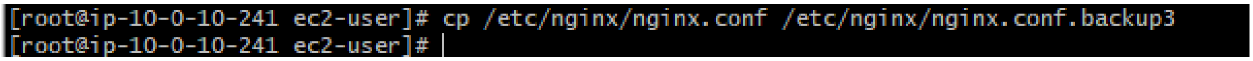
1. SSH into Nginx Server and Become Root



```
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.82.95
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 13:38:22 2025 from 20.192.21.53
[ec2-user@ip-10-0-10-241 ~]$ sudo su
[root@ip-10-0-10-241 ec2-user]# |
```

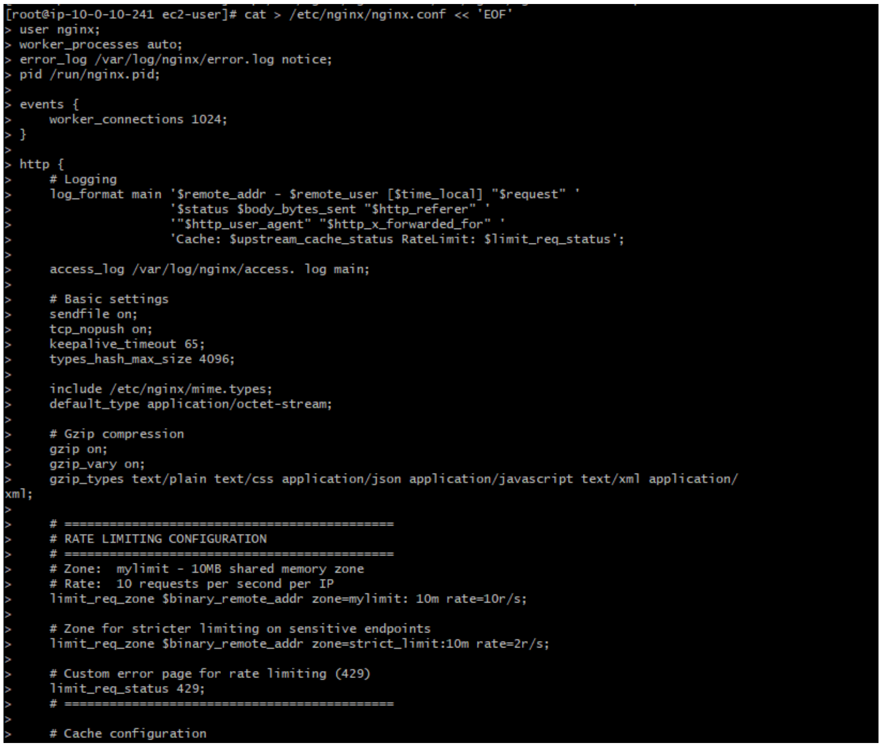
2. Update Nginx Configuration with Rate Limiting

   Backup current config



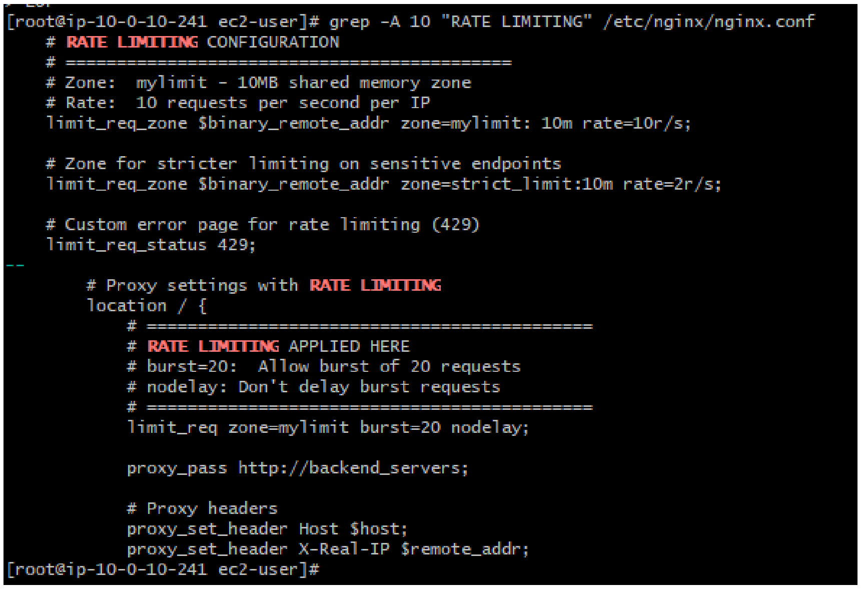
```
[root@ip-10-0-10-241 ec2-user]# cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.backup3
[root@ip-10-0-10-241 ec2-user]# |
```

3. Create updated Nginx configuration with rate limiting



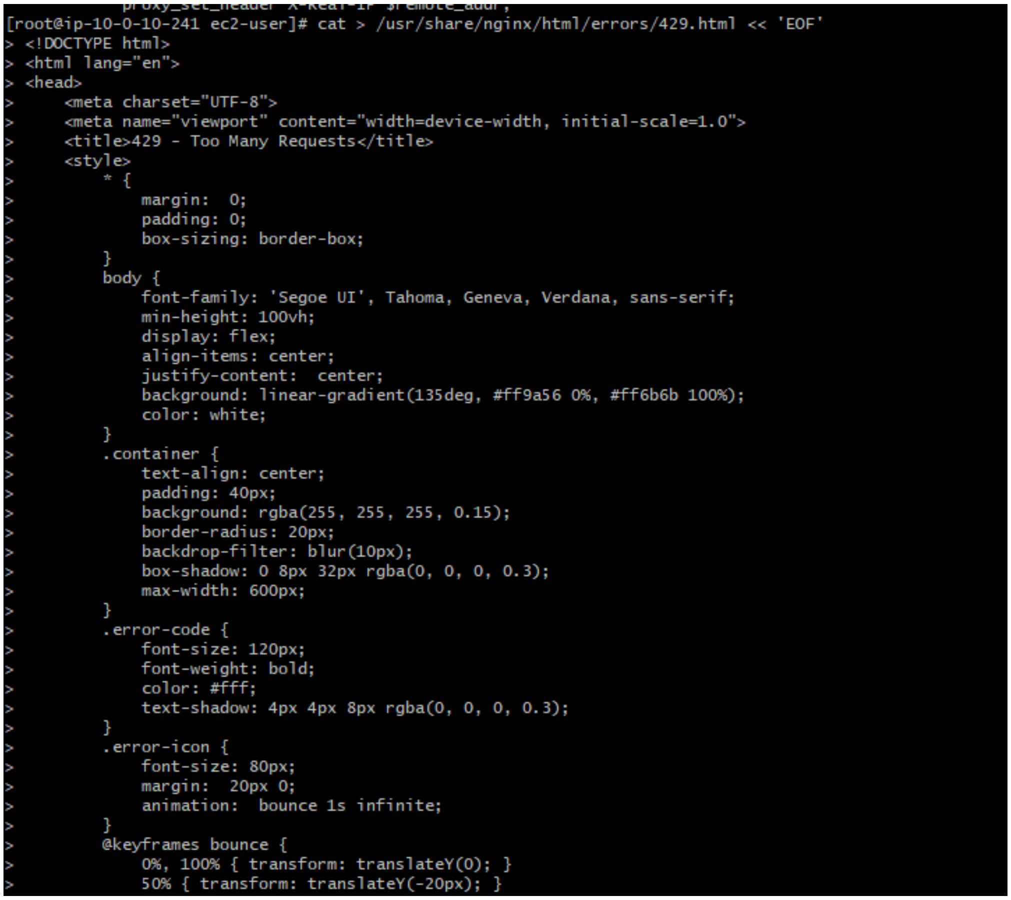
```
[root@ip-10-0-10-241 ec2-user]# cat > /etc/nginx/nginx.conf << 'EOF'
> user nginx;
> worker_processes auto;
> error_log /var/log/nginx/error.log notice;
> pid /run/nginx.pid;
>
> events {
>     worker_connections 1024;
> }
>
> http {
>     # Logging
>     log_format main '$remote_addr - $remote_user [$time_local] "$request" '
>                     '$status $body_bytes_sent "$http_referer" '
>                     '"$http_user_agent" "$http_x_forwarded_for" '
>                     'Cache: $upstream_cache_status RateLimit: $limit_req_status';
>
>     access_log /var/log/nginx/access. log main;
>
>     # Basic settings
>     sendfile on;
>     tcp_nopush on;
>     keepalive_timeout 65;
>     types_hash_max_size 4096;
>
>     include /etc/nginx/mime.types;
>     default_type application/octet-stream;
>
>     # Gzip compression
>     gzip on;
>     gzip_vary on;
>     gzip_types text/plain text/css application/json application/javascript text/xml application/
xml;
>
>     # ============================================
>     # RATE LIMITING CONFIGURATION
>     # ============================================
>     # Zone:  mylimit - 10MB shared memory zone
>     # Rate:  10 requests per second per IP
>     limit_req_zone $binary_remote_addr zone=mylimit: 10m rate=10r/s;
>
>     # Zone for stricter limiting on sensitive endpoints
>     limit_req_zone $binary_remote_addr zone=strict_limit:10m rate=2r/s;
>
>     # Custom error page for rate limiting (429)
>     limit_req_status 429;
>     # ============================================
>
>     # Cache configuration
```

Display the rate limiting configuration



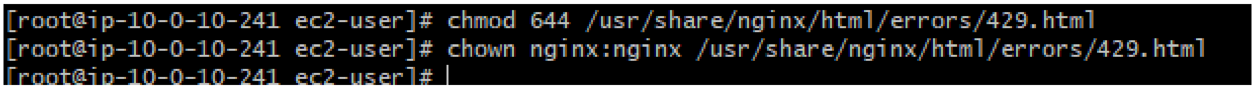
```
[root@ip-10-0-10-241 ec2-user]# grep -A 10 "RATE LIMITING" /etc/nginx/nginx.conf
    # RATE LIMITING CONFIGURATION
    # ============================================
    # Zone:  mylimit - 10MB shared memory zone
    # Rate:  10 requests per second per IP
    limit_req_zone $binary_remote_addr zone=mylimit: 10m rate=10r/s;

    # Zone for stricter limiting on sensitive endpoints
    limit_req_zone $binary_remote_addr zone=strict_limit:10m rate=2r/s;

    # Custom error page for rate limiting (429)
    limit_req_status 429;
--
        # Proxy settings with RATE LIMITING
        location / {
            # ============================================
            # RATE LIMITING APPLIED HERE
            # burst=20:  Allow burst of 20 requests
            # nodelay: Don't delay burst requests
            # ============================================
            limit_req zone=mylimit burst=20 nodelay;

            proxy_pass http://backend_servers;

            # Proxy headers
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
[root@ip-10-0-10-241 ec2-user]#
```

4. Create Custom 429 Error Page



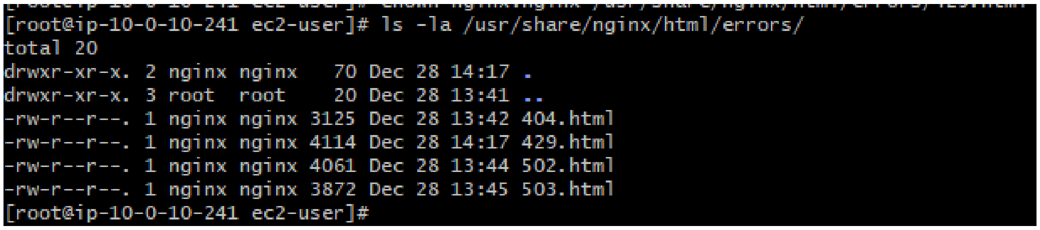
```
            proxy_set_header X-Real-IP $remote_addr;
[root@ip-10-0-10-241 ec2-user]# cat > /usr/share/nginx/html/errors/429.html << 'EOF'
> <!DOCTYPE html>
> <html lang="en">
> <head>
>     <meta charset="UTF-8">
>     <meta name="viewport" content="width=device-width, initial-scale=1.0">
>     <title>429 - Too Many Requests</title>
>     <style>
>         * {
>             margin:  0;
>             padding: 0;
>             box-sizing: border-box;
>         }
>         body {
>             font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
>             min-height: 100vh;
>             display: flex;
>             align-items: center;
>             justify-content:  center;
>             background: linear-gradient(135deg, #ff9a56 0%, #ff6b6b 100%);
>             color: white;
>         }
>         .container {
>             text-align: center;
>             padding: 40px;
>             background: rgba(255, 255, 255, 0.15);
>             border-radius: 20px;
>             backdrop-filter: blur(10px);
>             box-shadow: 0 8px 32px rgba(0, 0, 0, 0.3);
>             max-width: 600px;
>         }
>         .error-code {
>             font-size: 120px;
>             font-weight: bold;
>             color: #fff;
>             text-shadow: 4px 4px 8px rgba(0, 0, 0, 0.3);
>         }
>         .error-icon {
>             font-size: 80px;
>             margin:  20px 0;
>             animation:  bounce 1s infinite;
>         }
>         @keyframes bounce {
>             0%, 100% { transform: translateY(0); }
>             50% { transform: translateY(-20px); }
```

Set permissions



```
[root@ip-10-0-10-241 ec2-user]# chmod 644 /usr/share/nginx/html/errors/429.html
[root@ip-10-0-10-241 ec2-user]# chown nginx:nginx /usr/share/nginx/html/errors/429.html
[root@ip-10-0-10-241 ec2-user]# |
```

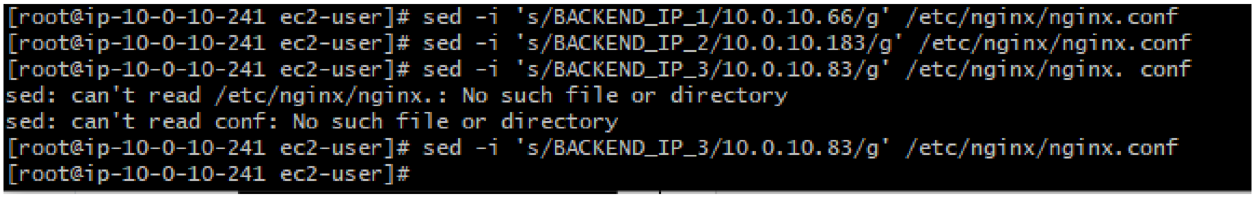
Verify file



```
[root@ip-10-0-10-241 ec2-user]# ls -la /usr/share/nginx/html/errors/
total 20
drwxr-xr-x. 2 nginx nginx   70 Dec 28 14:17 .
drwxr-xr-x. 3 root  root    20 Dec 28 13:41 ..
-rw-r--r--. 1 nginx nginx 3125 Dec 28 13:42 404.html
-rw-r--r--. 1 nginx nginx 4114 Dec 28 14:17 429.html
-rw-r--r--. 1 nginx nginx 4061 Dec 28 13:44 502.html
-rw-r--r--. 1 nginx nginx 3872 Dec 28 13:45 503.html
[root@ip-10-0-10-241 ec2-user]#
```

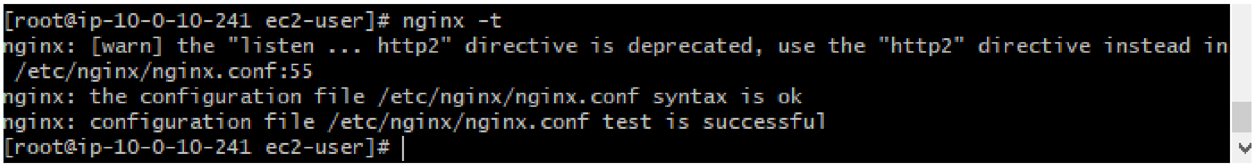
5. Update Backend IPs and Test Configuration

   Replace with your actual backend private IPs



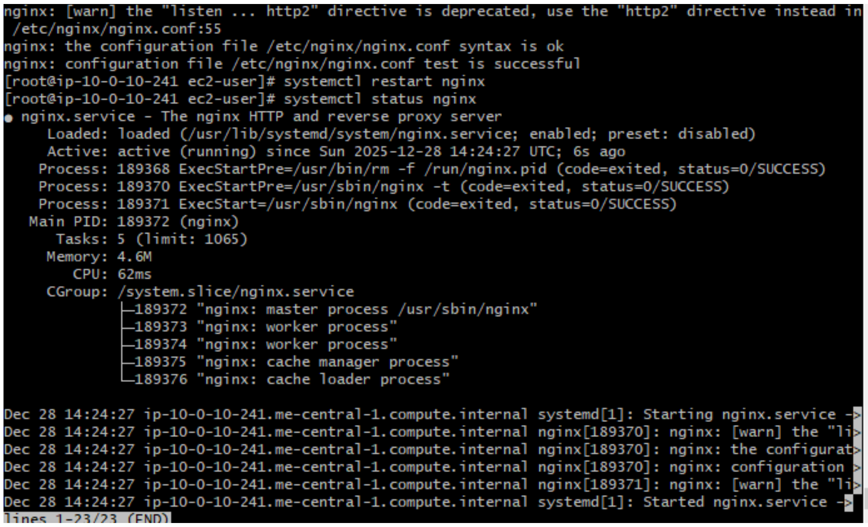
```
[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_1/10.0.10.66/g' /etc/nginx/nginx.conf
[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_2/10.0.10.183/g' /etc/nginx/nginx.conf
[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_3/10.0.10.83/g' /etc/nginx/nginx. conf
sed: can't read /etc/nginx/nginx.: No such file or directory
sed: can't read conf: No such file or directory
[root@ip-10-0-10-241 ec2-user]# sed -i 's/BACKEND_IP_3/10.0.10.83/g' /etc/nginx/nginx.conf
[root@ip-10-0-10-241 ec2-user]#
```

   Test Nginx configuration



```
[root@ip-10-0-10-241 ec2-user]# nginx -t
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in
 /etc/nginx/nginx.conf:55
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@ip-10-0-10-241 ec2-user]# |
```

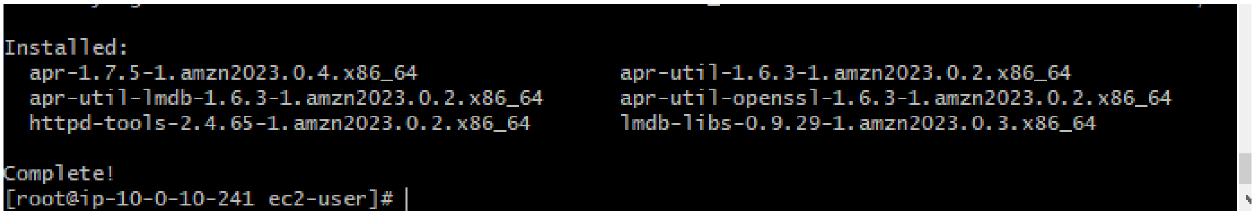
   Restart Nginx



```
nginx: [warn] the "listen ... http2" directive is deprecated, use the "http2" directive instead in
 /etc/nginx/nginx.conf:55
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@ip-10-0-10-241 ec2-user]# systemctl restart nginx
[root@ip-10-0-10-241 ec2-user]# 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 14:24:27 UTC; 6s ago
    Process: 189368 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 189370 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 189371 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 189372 (nginx)
      Tasks: 5 (limit: 1065)
     Memory: 4.6M
        CPU: 62ms
     CGroup: /system.slice/nginx.service
             ├─189372 "nginx: master process /usr/sbin/nginx"
             ├─189373 "nginx: worker process"
             ├─189374 "nginx: worker process"
             ├─189375 "nginx: cache manager process"
             └─189376 "nginx: cache loader process"

Dec 28 14:24:27 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Starting nginx.service ->
Dec 28 14:24:27 ip-10-0-10-241.me-central-1.compute.internal nginx[189370]: nginx: [warn] the "li>
Dec 28 14:24:27 ip-10-0-10-241.me-central-1.compute.internal nginx[189370]: nginx: the configurat>
Dec 28 14:24:27 ip-10-0-10-241.me-central-1.compute.internal nginx[189370]: nginx: configuration >
Dec 28 14:24:27 ip-10-0-10-241.me-central-1.compute.internal nginx[189371]: nginx: [warn] the "li>
Dec 28 14:24:27 ip-10-0-10-241.me-central-1.compute.internal systemd[1]: Started nginx.service ->
lines 1-23/23 (END)
```

6. Test Rate Limiting with Rapid Requests

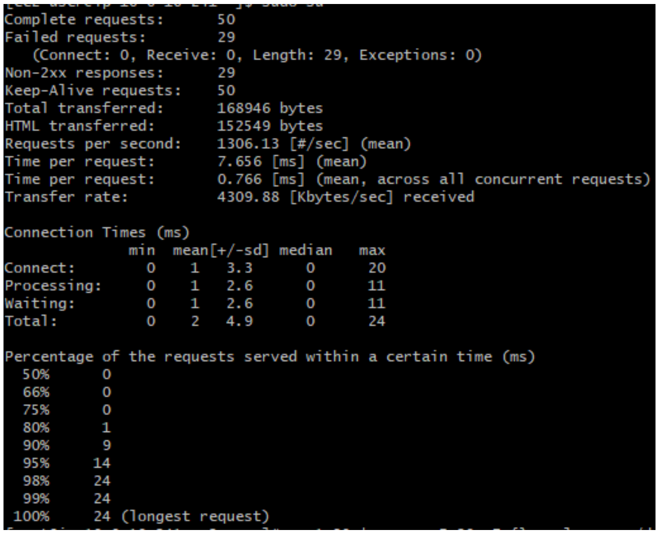
   Install Apache Benchmark tool



```
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
  httpd-tools-2.4.65-1.amzn2023.0.2.x86_64        lmdb-libs-0.9.29-1.amzn2023.0.3.x86_64

Complete!
[root@ip-10-0-10-241 ec2-user]# |
```

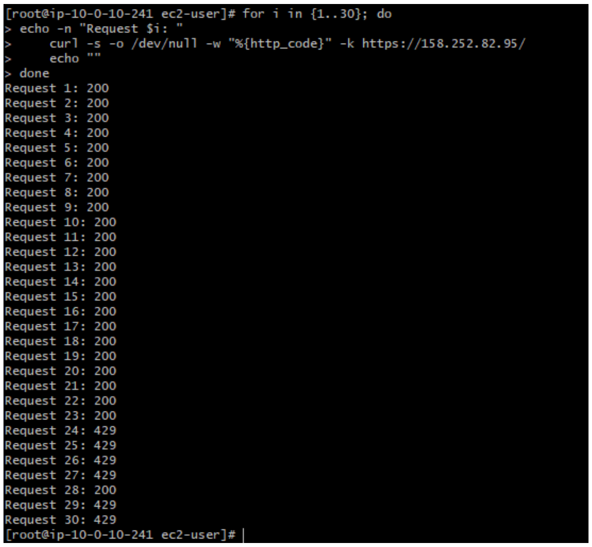
Test rate limiting - Send 50 requests rapidly

This should trigger rate limiting after the burst limit is exceeded



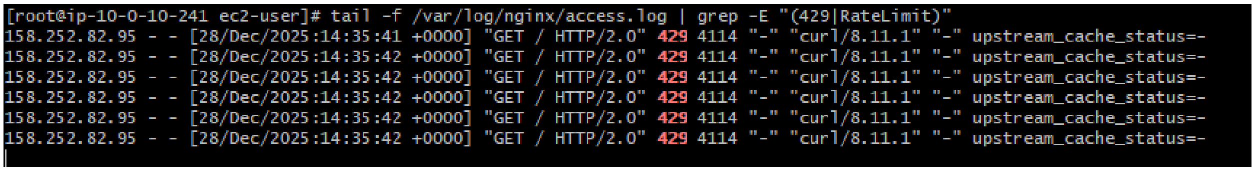
```
Complete requests:      50
Failed requests:        29
   (Connect: 0, Receive: 0, Length: 29, Exceptions: 0)
Non-2xx responses:      29
Keep-Alive requests:    50
Total transferred:      168946 bytes
HTML transferred:       152549 bytes
Requests per second:    1306.13 [#/sec] (mean)
Time per request:       7.656 [ms] (mean)
Time per request:       0.766 [ms] (mean, across all concurrent requests)
Transfer rate:          4309.88 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    1   3.3      0      20
Processing:     0    1   2.6      0      11
Waiting:        0    1   2.6      0      11
Total:          0    2   4.9      0      24

Percentage of the requests served within a certain time (ms)
  50%      0
  66%      0
  75%      0
  80%      1
  90%      9
  95%     14
  98%     24
  99%     24
 100%     24 (longest request)
```

Alternative Testing with curl in rapid succession



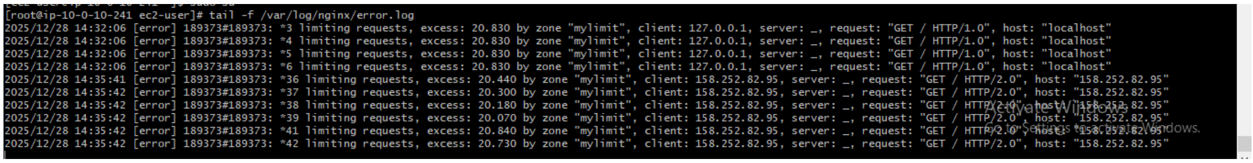
```
[root@ip-10-0-10-241 ec2-user]# for i in {1..30}; do
> echo -n "Request $i: "
>     curl -s -o /dev/null -w "%{http_code}" -k https://158.252.82.95/
>     echo ""
> done
Request 1: 200
Request 2: 200
Request 3: 200
Request 4: 200
Request 5: 200
Request 6: 200
Request 7: 200
Request 8: 200
Request 9: 200
Request 10: 200
Request 11: 200
Request 12: 200
Request 13: 200
Request 14: 200
Request 15: 200
Request 16: 200
Request 17: 200
Request 18: 200
Request 19: 200
Request 20: 200
Request 21: 200
Request 22: 200
Request 23: 200
Request 24: 429
Request 25: 429
Request 26: 429
Request 27: 429
Request 28: 200
Request 29: 429
Request 30: 429
[root@ip-10-0-10-241 ec2-user]# |
```

7. Monitor Rate Limiting in Logs



```
[root@ip-10-0-10-241 ec2-user]# tail -f /var/log/nginx/access.log | grep -E "(429|RateLimit)"
158.252.82.95 - - [28/Dec/2025:14:35:41 +0000] "GET / HTTP/2.0" 429 4114 "-" "curl/8.11.1" "-" upstream_cache_status=-
158.252.82.95 - - [28/Dec/2025:14:35:42 +0000] "GET / HTTP/2.0" 429 4114 "-" "curl/8.11.1" "-" upstream_cache_status=-
158.252.82.95 - - [28/Dec/2025:14:35:42 +0000] "GET / HTTP/2.0" 429 4114 "-" "curl/8.11.1" "-" upstream_cache_status=-
158.252.82.95 - - [28/Dec/2025:14:35:42 +0000] "GET / HTTP/2.0" 429 4114 "-" "curl/8.11.1" "-" upstream_cache_status=-
158.252.82.95 - - [28/Dec/2025:14:35:42 +0000] "GET / HTTP/2.0" 429 4114 "-" "curl/8.11.1" "-" upstream_cache_status=-
158.252.82.95 - - [28/Dec/2025:14:35:42 +0000] "GET / HTTP/2.0" 429 4114 "-" "curl/8.11.1" "-" upstream_cache_status=-
```

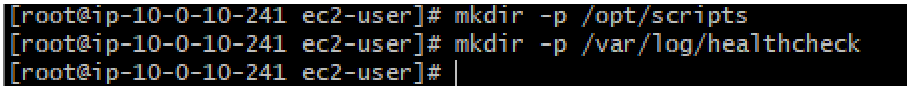
Check error logs for rate limit messages



```
[root@ip-10-0-10-241 ec2-user]# tail -f /var/log/nginx/error.log
2025/12/28 14:32:06 [error] 189373#189373: *3 limiting requests, excess: 20.830 by zone "mylimit", client: 127.0.0.1, server: _, request: "GET / HTTP/1.0", host: "localhost"
2025/12/28 14:32:06 [error] 189373#189373: *4 limiting requests, excess: 20.830 by zone "mylimit", client: 127.0.0.1, server: _, request: "GET / HTTP/1.0", host: "localhost"
2025/12/28 14:32:06 [error] 189373#189373: *5 limiting requests, excess: 20.830 by zone "mylimit", client: 127.0.0.1, server: _, request: "GET / HTTP/1.0", host: "localhost"
2025/12/28 14:32:06 [error] 189373#189373: *6 limiting requests, excess: 20.830 by zone "mylimit", client: 127.0.0.1, server: _, request: "GET / HTTP/1.0", host: "localhost"
2025/12/28 14:35:41 [error] 189373#189373: *36 limiting requests, excess: 20.440 by zone "mylimit", client: 158.252.82.95, server: _, request: "GET / HTTP/2.0", host: "158.252.82.95"
2025/12/28 14:35:42 [error] 189373#189373: *37 limiting requests, excess: 20.300 by zone "mylimit", client: 158.252.82.95, server: _, request: "GET / HTTP/2.0", host: "158.252.82.95"
2025/12/28 14:35:42 [error] 189373#189373: *38 limiting requests, excess: 20.180 by zone "mylimit", client: 158.252.82.95, server: _, request: "GET / HTTP/2.0", host: "158.252.82.95"
2025/12/28 14:35:42 [error] 189373#189373: *39 limiting requests, excess: 20.070 by zone "mylimit", client: 158.252.82.95, server: _, request: "GET / HTTP/2.0", host: "158.252.82.95"
2025/12/28 14:35:42 [error] 189373#189373: *41 limiting requests, excess: 20.840 by zone "mylimit", client: 158.252.82.95, server: _, request: "GET / HTTP/2.0", host: "158.252.82.95"
2025/12/28 14:35:42 [error] 189373#189373: *42 limiting requests, excess: 20.730 by zone "mylimit", client: 158.252.82.95, server: _, request: "GET / HTTP/2.0", host: "158.252.82.95"
```

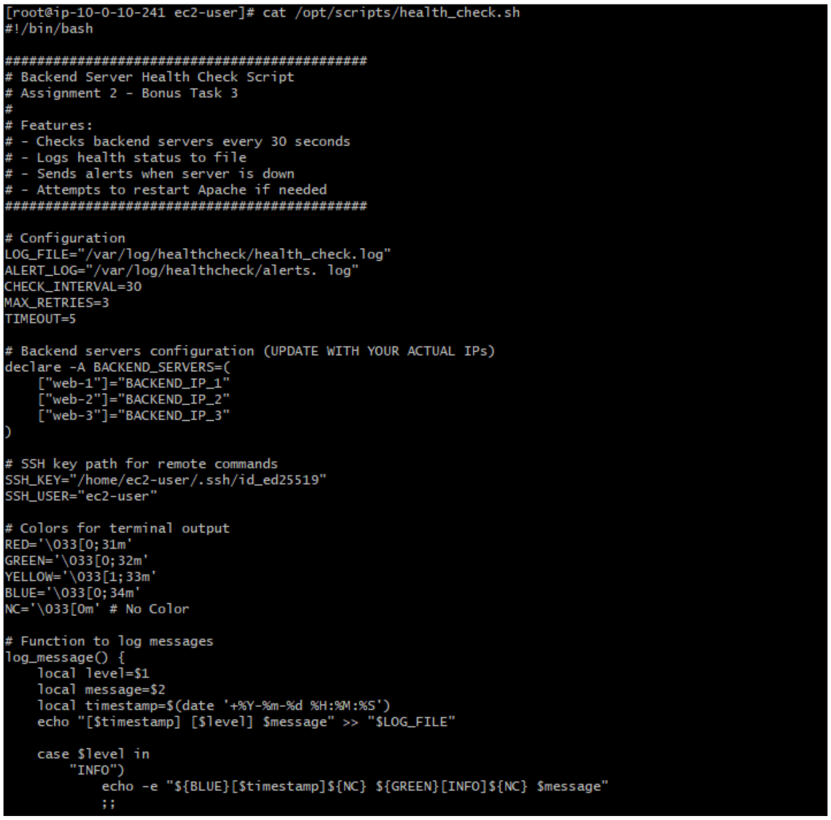
Bonus 3: Health Check Automation

1. Create Scripts Directory



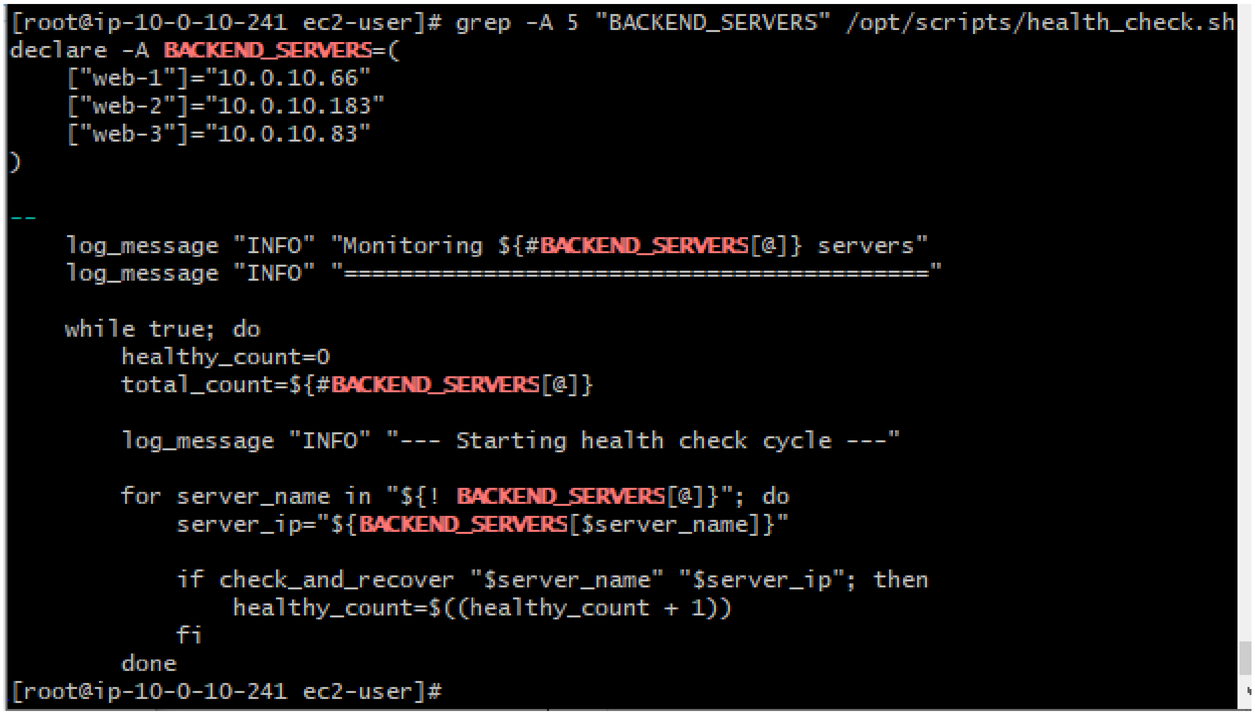
```
[root@ip-10-0-10-241 ec2-user]# mkdir -p /opt/scripts
[root@ip-10-0-10-241 ec2-user]# mkdir -p /var/log/healthcheck
[root@ip-10-0-10-241 ec2-user]# |
```

2. Create Health Check Script



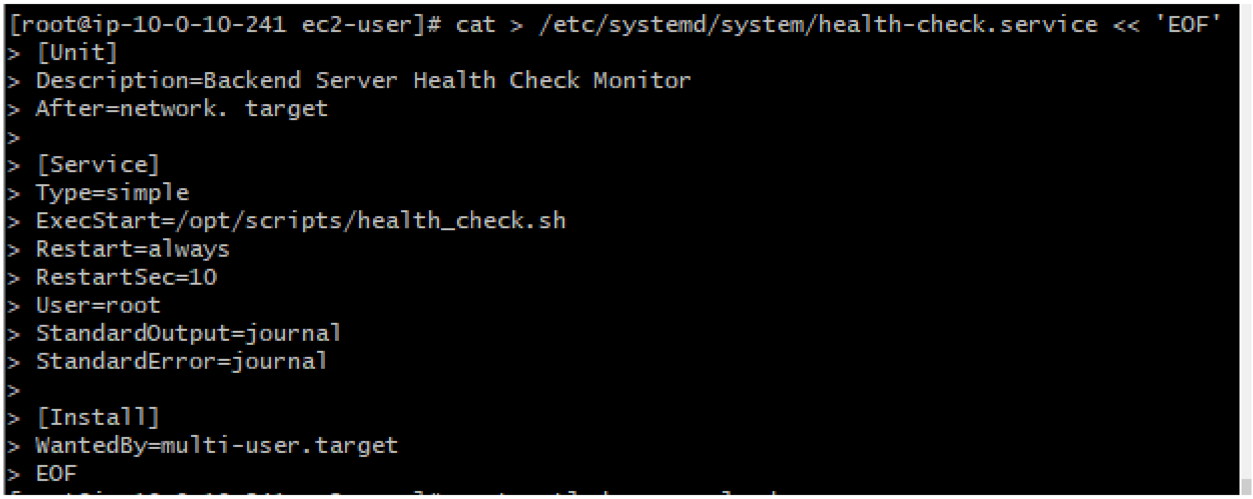
```
[root@ip-10-0-10-241 ec2-user]# cat /opt/scripts/health_check.sh
#!/bin/bash

###############################################
# Backend Server Health Check Script
# Assignment 2 - Bonus Task 3
#
# Features:
# - Checks backend servers every 30 seconds
# - Logs health status to file
# - Sends alerts when server is down
# - Attempts to restart Apache if needed
###############################################

# Configuration
LOG_FILE="/var/log/healthcheck/health_check.log"
ALERT_LOG="/var/log/healthcheck/alerts. log"
CHECK_INTERVAL=30
MAX_RETRIES=3
TIMEOUT=5

# Backend servers configuration (UPDATE WITH YOUR ACTUAL IPs)
declare -A BACKEND_SERVERS=(
    ["web-1"]="BACKEND_IP_1"
    ["web-2"]="BACKEND_IP_2"
    ["web-3"]="BACKEND_IP_3"
)

# SSH key path for remote commands
SSH_KEY="/home/ec2-user/.ssh/id_ed25519"
SSH_USER="ec2-user"

# Colors for terminal output
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
BLUE='\033[0;34m'
NC='\033[0m' # No Color

# Function to log messages
log_message() {
    local level=$1
    local message=$2
    local timestamp=$(date '+%Y-%m-%d %H:%M:%S')
    echo "[$timestamp] [$level] $message" >> "$LOG_FILE"

    case $level in
        "INFO")
            echo -e "${BLUE}[$timestamp]${NC} ${GREEN}[INFO]${NC} $message"
            ;;
```

3. Update Script with Actual Backend IPs



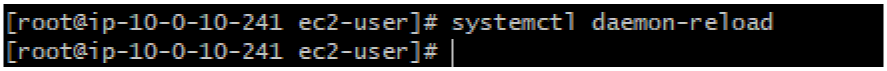
```
[root@ip-10-0-10-241 ec2-user]# grep -A 5 "BACKEND_SERVERS" /opt/scripts/health_check.sh
declare -A BACKEND_SERVERS=(
    ["web-1"]="10.0.10.66"
    ["web-2"]="10.0.10.183"
    ["web-3"]="10.0.10.83"
)

--
    log_message "INFO" "Monitoring ${#BACKEND_SERVERS[@]} servers"
    log_message "INFO" "==========================================="

    while true; do
        healthy_count=0
        total_count=${#BACKEND_SERVERS[@]}

        log_message "INFO" "--- Starting health check cycle ---"

        for server_name in "${! BACKEND_SERVERS[@]}"; do
            server_ip="${BACKEND_SERVERS[$server_name]}"

            if check_and_recover "$server_name" "$server_ip"; then
                healthy_count=$((healthy_count + 1))
            fi
        done
[root@ip-10-0-10-241 ec2-user]#
```

4. Create Systemd Service for Health Check



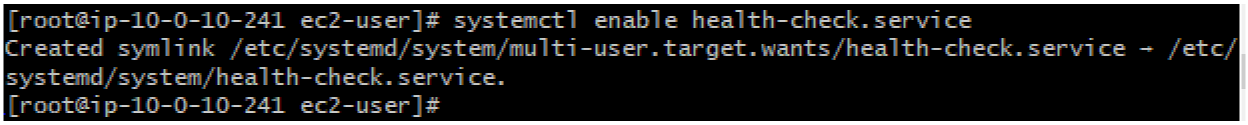
```
[root@ip-10-0-10-241 ec2-user]# cat > /etc/systemd/system/health-check.service << 'EOF'
> [Unit]
> Description=Backend Server Health Check Monitor
> After=network. target
>
> [Service]
> Type=simple
> ExecStart=/opt/scripts/health_check.sh
> Restart=always
> RestartSec=10
> User=root
> StandardOutput=journal
> StandardError=journal
>
> [Install]
> WantedBy=multi-user.target
> EOF
```

Reload systemd



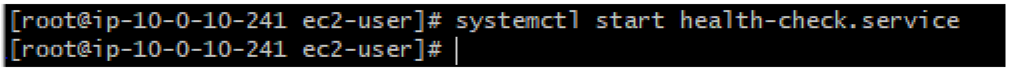
```
[root@ip-10-0-10-241 ec2-user]# systemctl daemon-reload
[root@ip-10-0-10-241 ec2-user]# 
```

Enable service to start on boot



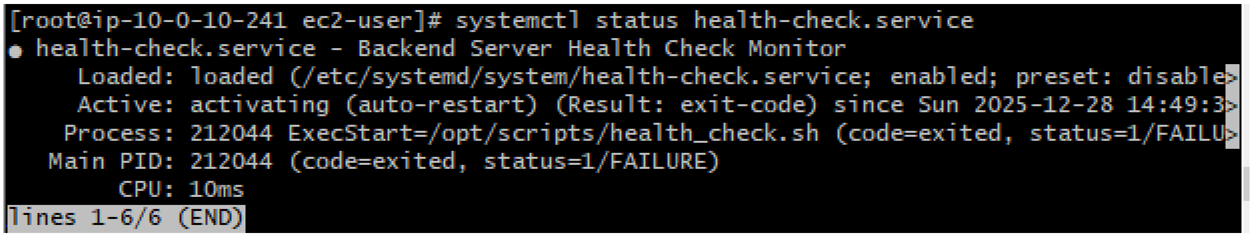
```
[root@ip-10-0-10-241 ec2-user]# systemctl enable health-check.service
Created symlink /etc/systemd/system/multi-user.target.wants/health-check.service → /etc/
systemd/system/health-check.service.
[root@ip-10-0-10-241 ec2-user]#
```

Start the service



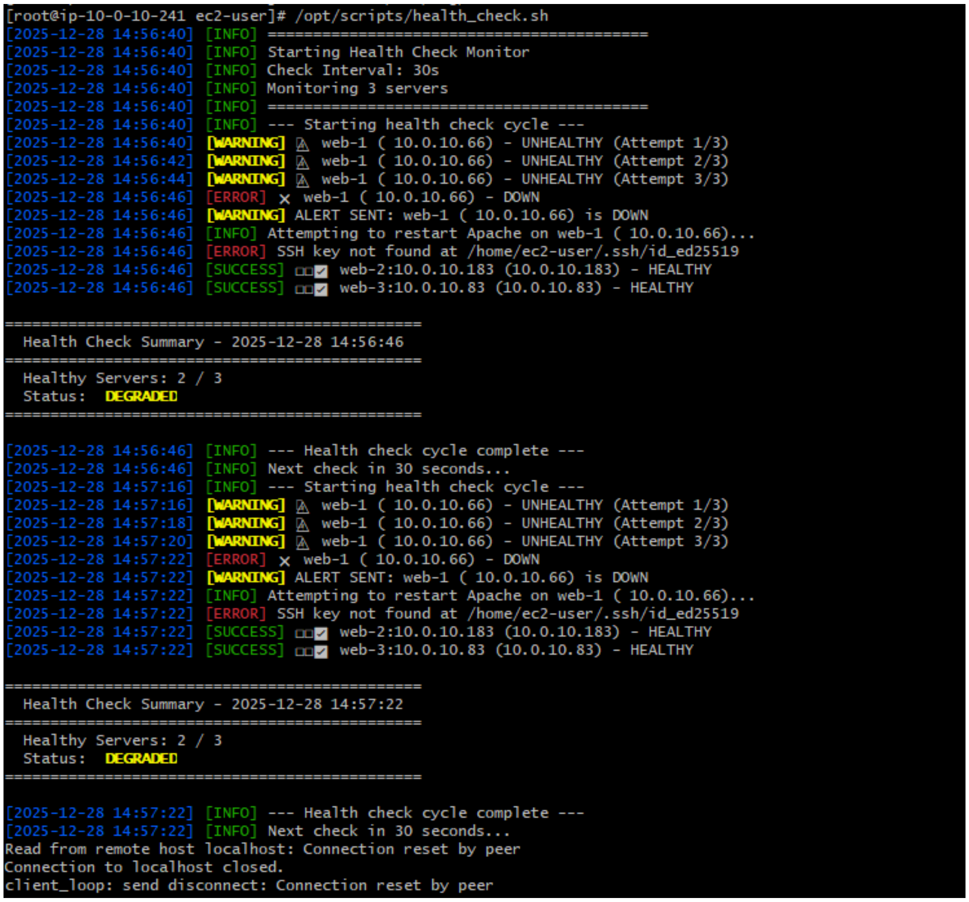
```
[root@ip-10-0-10-241 ec2-user]# systemctl start health-check.service
[root@ip-10-0-10-241 ec2-user]# |
```

Check status



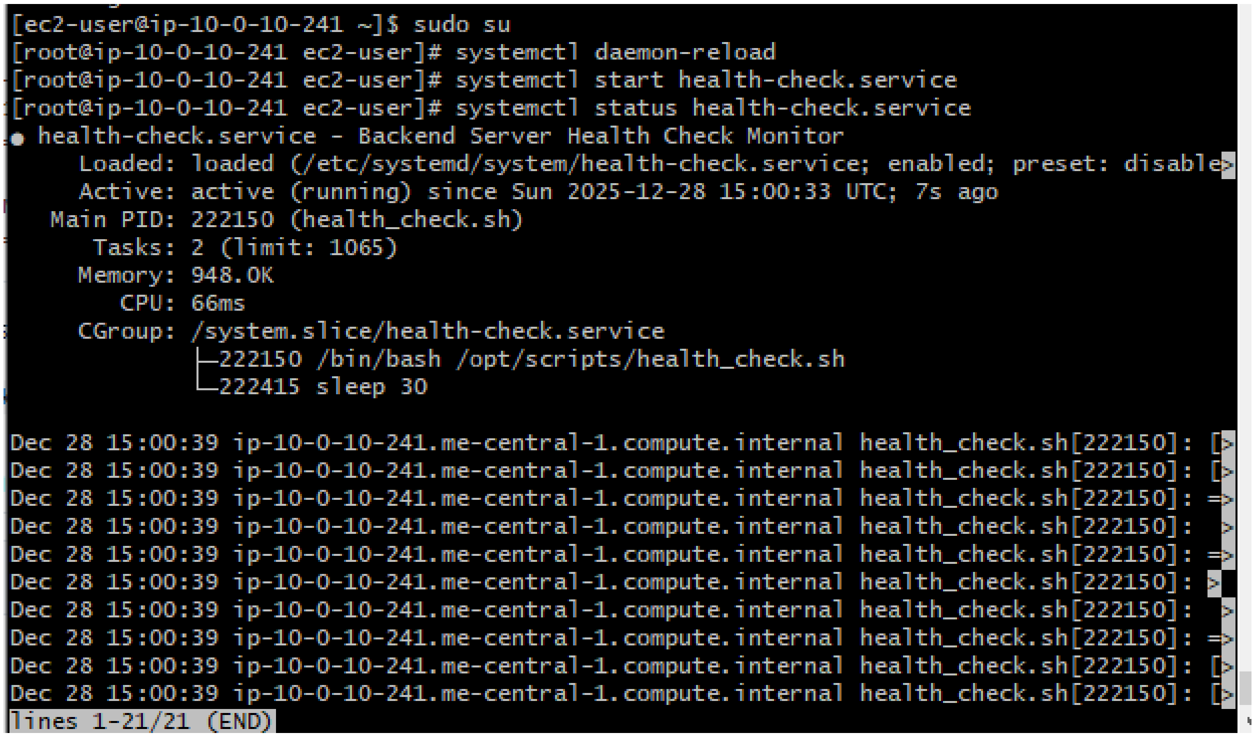
```
[root@ip-10-0-10-241 ec2-user]# systemctl status health-check.service
● health-check.service - Backend Server Health Check Monitor
     Loaded: loaded (/etc/systemd/system/health-check.service; enabled; preset: disable>
     Active: activating (auto-restart) (Result: exit-code) since Sun 2025-12-28 14:49:3>
    Process: 212044 ExecStart=/opt/scripts/health_check.sh (code=exited, status=1/FAILU>
   Main PID: 212044 (code=exited, status=1/FAILURE)
        CPU: 10ms
lines 1-6/6 (END)
```

5. Run Health Check Manually for Testing



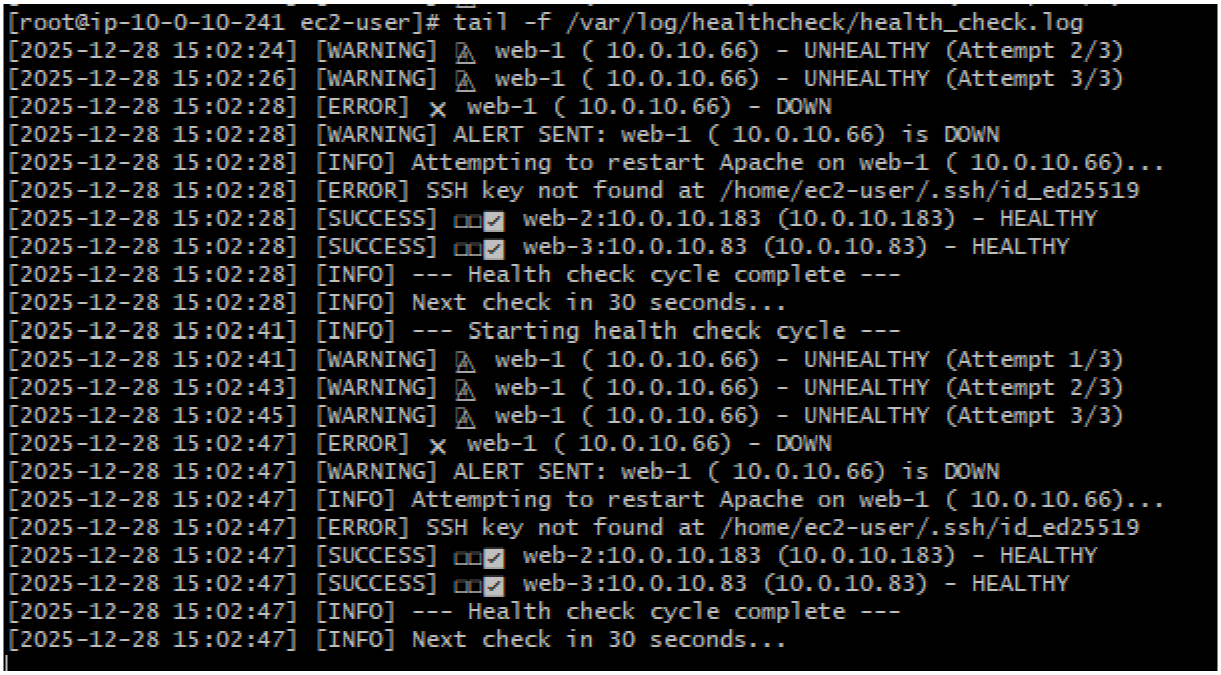
```
[root@ip-10-0-10-241 ec2-user]# /opt/scripts/health_check.sh
[2025-12-28 14:56:40] [INFO] ========================================
[2025-12-28 14:56:40] [INFO] Starting Health Check Monitor
[2025-12-28 14:56:40] [INFO] Check Interval: 30s
[2025-12-28 14:56:40] [INFO] Monitoring 3 servers
[2025-12-28 14:56:40] [INFO] ========================================
[2025-12-28 14:56:40] [INFO] --- Starting health check cycle ---
[2025-12-28 14:56:40] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 1/3)
[2025-12-28 14:56:42] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 2/3)
[2025-12-28 14:56:44] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 3/3)
[2025-12-28 14:56:46] [ERROR] ✗ web-1 ( 10.0.10.66) - DOWN
[2025-12-28 14:56:46] [WARNING] ALERT SENT: web-1 ( 10.0.10.66) is DOWN
[2025-12-28 14:56:46] [INFO] Attempting to restart Apache on web-1 ( 10.0.10.66)...
[2025-12-28 14:56:46] [ERROR] SSH key not found at /home/ec2-user/.ssh/id_ed25519
[2025-12-28 14:56:46] [SUCCESS] □□☑ web-2:10.0.10.183 (10.0.10.183) - HEALTHY
[2025-12-28 14:56:46] [SUCCESS] □□☑ web-3:10.0.10.83 (10.0.10.83) - HEALTHY

==============================================
  Health Check Summary - 2025-12-28 14:56:46
==============================================
  Healthy Servers: 2 / 3
  Status:  DEGRADED
==============================================

[2025-12-28 14:56:46] [INFO] --- Health check cycle complete ---
[2025-12-28 14:56:46] [INFO] Next check in 30 seconds...
[2025-12-28 14:57:16] [INFO] --- Starting health check cycle ---
[2025-12-28 14:57:16] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 1/3)
[2025-12-28 14:57:18] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 2/3)
[2025-12-28 14:57:20] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 3/3)
[2025-12-28 14:57:22] [ERROR] ✗ web-1 ( 10.0.10.66) - DOWN
[2025-12-28 14:57:22] [WARNING] ALERT SENT: web-1 ( 10.0.10.66) is DOWN
[2025-12-28 14:57:22] [INFO] Attempting to restart Apache on web-1 ( 10.0.10.66)...
[2025-12-28 14:57:22] [ERROR] SSH key not found at /home/ec2-user/.ssh/id_ed25519
[2025-12-28 14:57:22] [SUCCESS] □□☑ web-2:10.0.10.183 (10.0.10.183) - HEALTHY
[2025-12-28 14:57:22] [SUCCESS] □□☑ web-3:10.0.10.83 (10.0.10.83) - HEALTHY

==============================================
  Health Check Summary - 2025-12-28 14:57:22
==============================================
  Healthy Servers: 2 / 3
  Status:  DEGRADED
==============================================

[2025-12-28 14:57:22] [INFO] --- Health check cycle complete ---
[2025-12-28 14:57:22] [INFO] Next check in 30 seconds...
Read from remote host localhost: Connection reset by peer
Connection to localhost closed.
client_loop: send disconnect: Connection reset by peer
```

6. Start the Systemd Service



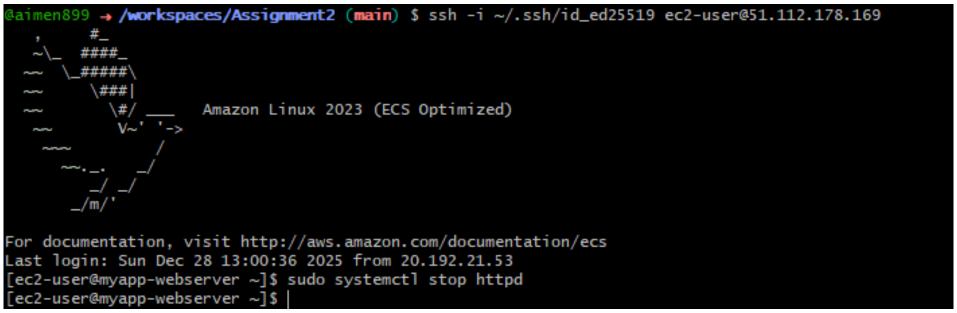
```
[ec2-user@ip-10-0-10-241 ~]$ sudo su
[root@ip-10-0-10-241 ec2-user]# systemctl daemon-reload
[root@ip-10-0-10-241 ec2-user]# systemctl start health-check.service
[root@ip-10-0-10-241 ec2-user]# systemctl status health-check.service
● health-check.service - Backend Server Health Check Monitor
     Loaded: loaded (/etc/systemd/system/health-check.service; enabled; preset: disable>
     Active: active (running) since Sun 2025-12-28 15:00:33 UTC; 7s ago
   Main PID: 222150 (health_check.sh)
      Tasks: 2 (limit: 1065)
     Memory: 948.0K
        CPU: 66ms
     CGroup: /system.slice/health-check.service
             ├─222150 /bin/bash /opt/scripts/health_check.sh
             └─222415 sleep 30

Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: [>
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: [>
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: =>
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]:  >
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: =>
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: >
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]:  >
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: =>
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: [>
Dec 28 15:00:39 ip-10-0-10-241.me-central-1.compute.internal health_check.sh[222150]: [>
lines 1-21/21 (END)
```

7. View the Logs



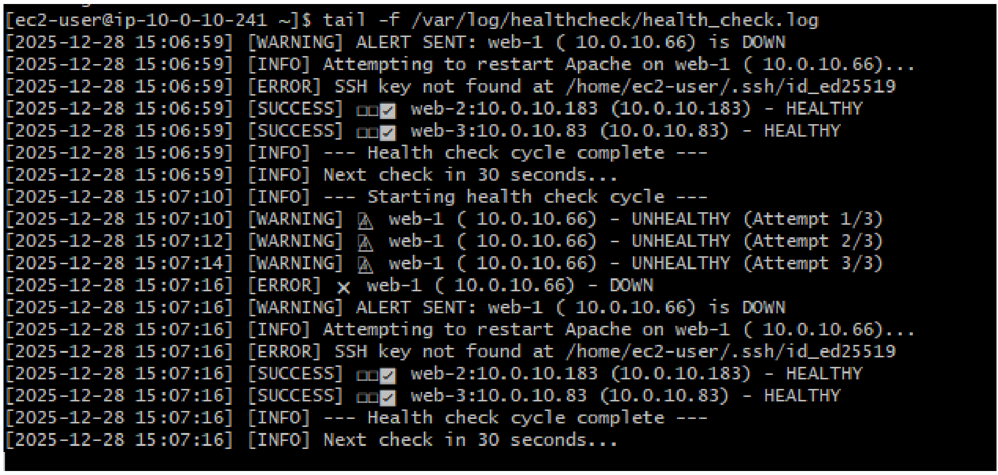
```
[root@ip-10-0-10-241 ec2-user]# tail -f /var/log/healthcheck/health_check.log
[2025-12-28 15:02:24] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 2/3)
[2025-12-28 15:02:26] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 3/3)
[2025-12-28 15:02:28] [ERROR] ✕ web-1 ( 10.0.10.66) - DOWN
[2025-12-28 15:02:28] [WARNING] ALERT SENT: web-1 ( 10.0.10.66) is DOWN
[2025-12-28 15:02:28] [INFO] Attempting to restart Apache on web-1 ( 10.0.10.66)...
[2025-12-28 15:02:28] [ERROR] SSH key not found at /home/ec2-user/.ssh/id_ed25519
[2025-12-28 15:02:28] [SUCCESS] □□✓ web-2:10.0.10.183 (10.0.10.183) - HEALTHY
[2025-12-28 15:02:28] [SUCCESS] □□✓ web-3:10.0.10.83 (10.0.10.83) - HEALTHY
[2025-12-28 15:02:28] [INFO] --- Health check cycle complete ---
[2025-12-28 15:02:28] [INFO] Next check in 30 seconds...
[2025-12-28 15:02:41] [INFO] --- Starting health check cycle ---
[2025-12-28 15:02:41] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 1/3)
[2025-12-28 15:02:43] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 2/3)
[2025-12-28 15:02:45] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 3/3)
[2025-12-28 15:02:47] [ERROR] ✕ web-1 ( 10.0.10.66) - DOWN
[2025-12-28 15:02:47] [WARNING] ALERT SENT: web-1 ( 10.0.10.66) is DOWN
[2025-12-28 15:02:47] [INFO] Attempting to restart Apache on web-1 ( 10.0.10.66)...
[2025-12-28 15:02:47] [ERROR] SSH key not found at /home/ec2-user/.ssh/id_ed25519
[2025-12-28 15:02:47] [SUCCESS] □□✓ web-2:10.0.10.183 (10.0.10.183) - HEALTHY
[2025-12-28 15:02:47] [SUCCESS] □□✓ web-3:10.0.10.83 (10.0.10.83) - HEALTHY
[2025-12-28 15:02:47] [INFO] --- Health check cycle complete ---
[2025-12-28 15:02:47] [INFO] Next check in 30 seconds...
```

8. Test Alert by Stopping a Backend Server
   Open another terminal and stop Apache on one of the backend servers:



```
@aimen899 → /workspaces/Assignment2 (main) $ ssh -i ~/.ssh/id_ed25519 ec2-user@51.112.178.169
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Sun Dec 28 13:00:36 2025 from 20.192.21.53
[ec2-user@myapp-webserver ~]$ sudo systemctl stop httpd
[ec2-user@myapp-webserver ~]$ |
```

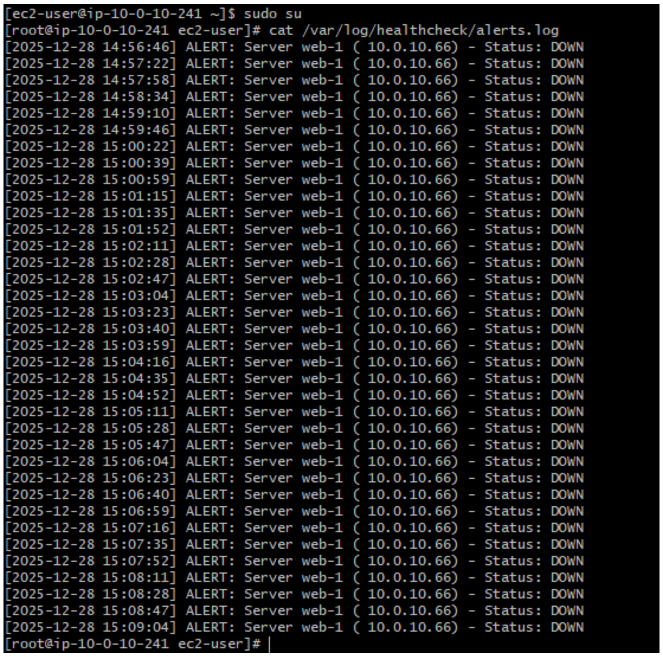
Then watch the health check logs on the Nginx server:



```
[ec2-user@ip-10-0-10-241 ~]$ tail -f /var/log/healthcheck/health_check.log
[2025-12-28 15:06:59] [WARNING] ALERT SENT: web-1 ( 10.0.10.66) is DOWN
[2025-12-28 15:06:59] [INFO] Attempting to restart Apache on web-1 ( 10.0.10.66)...
[2025-12-28 15:06:59] [ERROR] SSH key not found at /home/ec2-user/.ssh/id_ed25519
[2025-12-28 15:06:59] [SUCCESS] □□✓ web-2:10.0.10.183 (10.0.10.183) - HEALTHY
[2025-12-28 15:06:59] [SUCCESS] □□✓ web-3:10.0.10.83 (10.0.10.83) - HEALTHY
[2025-12-28 15:06:59] [INFO] --- Health check cycle complete ---
[2025-12-28 15:06:59] [INFO] Next check in 30 seconds...
[2025-12-28 15:07:10] [INFO] --- Starting health check cycle ---
[2025-12-28 15:07:10] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 1/3)
[2025-12-28 15:07:12] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 2/3)
[2025-12-28 15:07:14] [WARNING] ⚠ web-1 ( 10.0.10.66) - UNHEALTHY (Attempt 3/3)
[2025-12-28 15:07:16] [ERROR] ✗ web-1 ( 10.0.10.66) - DOWN
[2025-12-28 15:07:16] [WARNING] ALERT SENT: web-1 ( 10.0.10.66) is DOWN
[2025-12-28 15:07:16] [INFO] Attempting to restart Apache on web-1 ( 10.0.10.66)...
[2025-12-28 15:07:16] [ERROR] SSH key not found at /home/ec2-user/.ssh/id_ed25519
[2025-12-28 15:07:16] [SUCCESS] □□✓ web-2:10.0.10.183 (10.0.10.183) - HEALTHY
[2025-12-28 15:07:16] [SUCCESS] □□✓ web-3:10.0.10.83 (10.0.10.83) - HEALTHY
[2025-12-28 15:07:16] [INFO] --- Health check cycle complete ---
[2025-12-28 15:07:16] [INFO] Next check in 30 seconds...
```

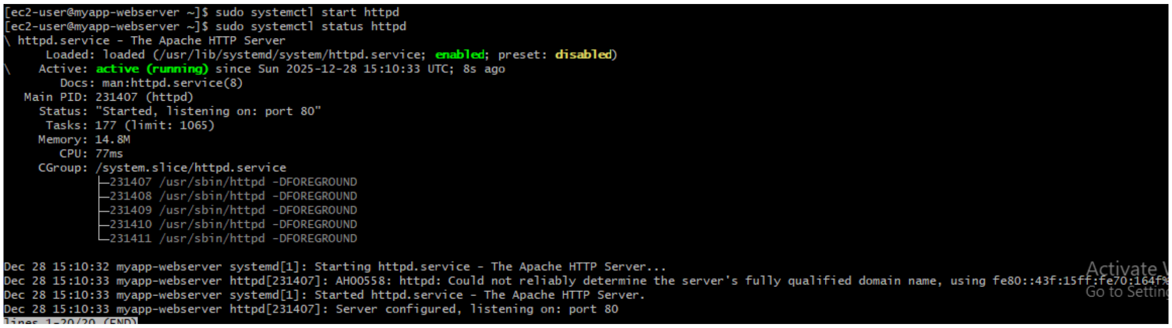
(ALERT: web-1 is down)

9. View Alert Log



```
[ec2-user@ip-10-0-10-241 ~]$ sudo su
[root@ip-10-0-10-241 ec2-user]# cat /var/log/healthcheck/alerts.log
[2025-12-28 14:56:46] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 14:57:22] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 14:57:58] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 14:58:34] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 14:59:10] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 14:59:46] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:00:22] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:00:39] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:00:59] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:01:15] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:01:35] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:01:52] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:02:11] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:02:28] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:02:47] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:03:04] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:03:23] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:03:40] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:03:59] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:04:16] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:04:35] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:04:52] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:05:11] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:05:28] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:05:47] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:06:04] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:06:23] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:06:40] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:06:59] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:07:16] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:07:35] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:07:52] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:08:11] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:08:28] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:08:47] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[2025-12-28 15:09:04] ALERT: Server web-1 ( 10.0.10.66) - Status: DOWN
[root@ip-10-0-10-241 ec2-user]# |
```

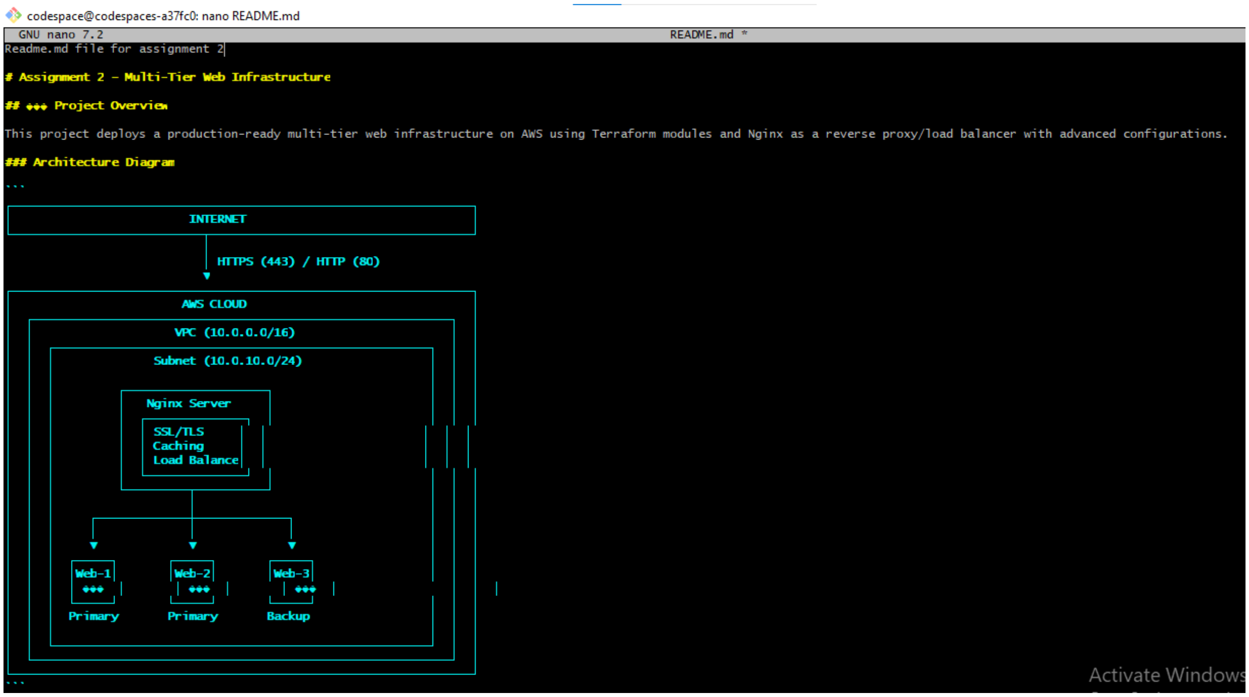
10. Restart the Backend Server



```
[ec2-user@myapp-webserver ~]$ sudo systemctl start httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
\ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
\    Active: active (running) since Sun 2025-12-28 15:10:33 UTC; 8s ago
       Docs: man:httpd.service(8)
   Main PID: 231407 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 1065)
     Memory: 14.8M
        CPU: 77ms
     CGroup: /system.slice/httpd.service
             ├─231407 /usr/sbin/httpd -DFOREGROUND
             ├─231408 /usr/sbin/httpd -DFOREGROUND
             ├─231409 /usr/sbin/httpd -DFOREGROUND
             ├─231410 /usr/sbin/httpd -DFOREGROUND
             └─231411 /usr/sbin/httpd -DFOREGROUND

Dec 28 15:10:32 myapp-webserver systemd[1]: Starting httpd.service - The Apache HTTP Server...
Dec 28 15:10:33 myapp-webserver httpd[231407]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using fe80::43f:15ff:fe70:164f%
Dec 28 15:10:33 myapp-webserver systemd[1]: Started httpd.service - The Apache HTTP Server.
Dec 28 15:10:33 myapp-webserver httpd[231407]: Server configured, listening on: port 80
lines 1-20/20 (END)
```

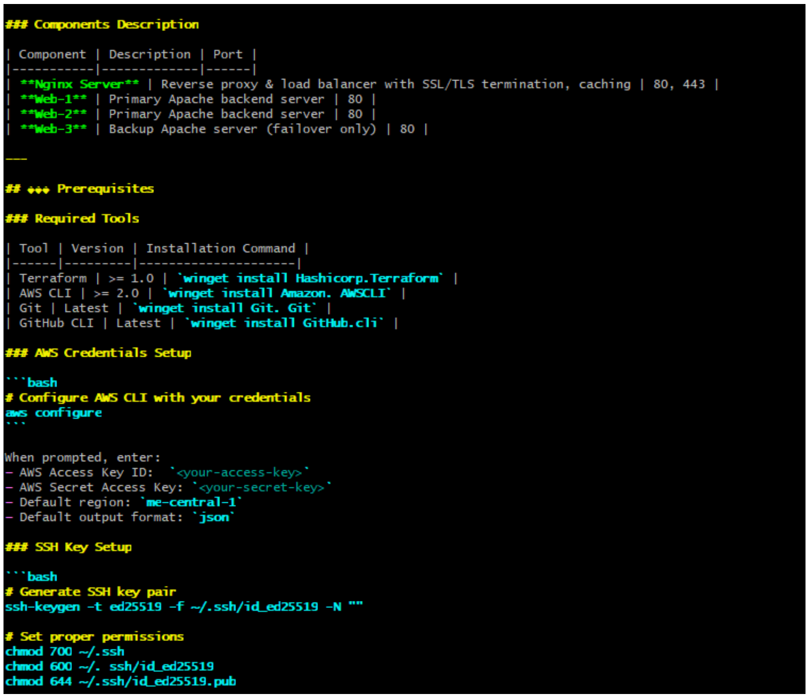
Part 6: Documentation & Cleanup

6.1 Create README. md



codespace@codespaces-a37fc0: nano README.md

```
  GNU nano 7.2                                   README.md *
Readme.md file for assignment 2

# Assignment 2 - Multi-Tier Web Infrastructure

## ◆◆◆ Project Overview

This project deploys a production-ready multi-tier web infrastructure on AWS using Terraform modules and Nginx as a reverse proxy/load balancer with advanced configurations.

### Architecture Diagram
```

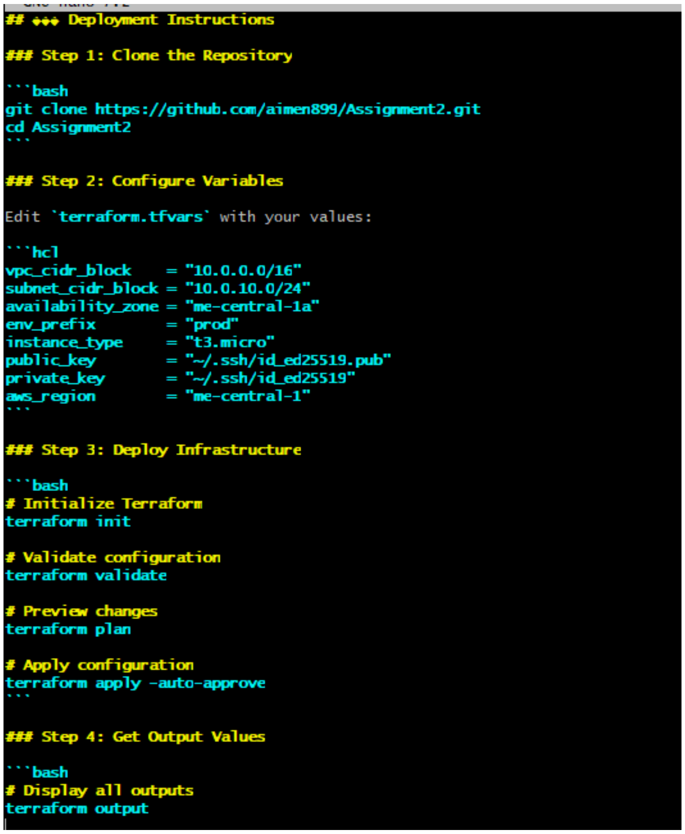


~~~
### Components Description

| Component | Description | Port |
|-----------|-------------|------|
| **Nginx Server** | Reverse proxy & load balancer with SSL/TLS termination, caching | 80, 443 |
| **Web-1** | Primary Apache backend server | 80 |
| **Web-2** | Primary Apache backend server | 80 |
| **Web-3** | Backup Apache server (failover only) | 80 |

---

## ◆◆◆ Prerequisites

### Required Tools

| Tool | Version | Installation Command |
|------|---------|--------------------|
| Terraform | >= 1.0 | `winget install Hashicorp.Terraform` |
| AWS CLI | >= 2.0 | `winget install Amazon. AWSCLI` |
| Git | Latest | `winget install Git. Git` |
| GitHub CLI | Latest | `winget install GitHub.cli` |

### AWS Credentials Setup

```bash
# Configure AWS CLI with your credentials
aws configure
~~~

When prompted, enter:
- AWS Access Key ID: `<your-access-key>`
- AWS Secret Access Key: `<your-secret-key>`
- Default region: `me-central-1`
- Default output format: `json`

SSH Key Setup

```bash
# Generate SSH key pair
ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""

# Set proper permissions
chmod 700 ~/.ssh
chmod 600 ~/. ssh/id_ed25519
chmod 644 ~/.ssh/id_ed25519.pub
```



~~~
  GNU nano 7.2
## ◆◆◆ Deployment Instructions

### Step 1: Clone the Repository

```bash
git clone https://github.com/aimen899/Assignment2.git
cd Assignment2
~~~

Step 2: Configure Variables

Edit `terraform.tfvars` with your values:

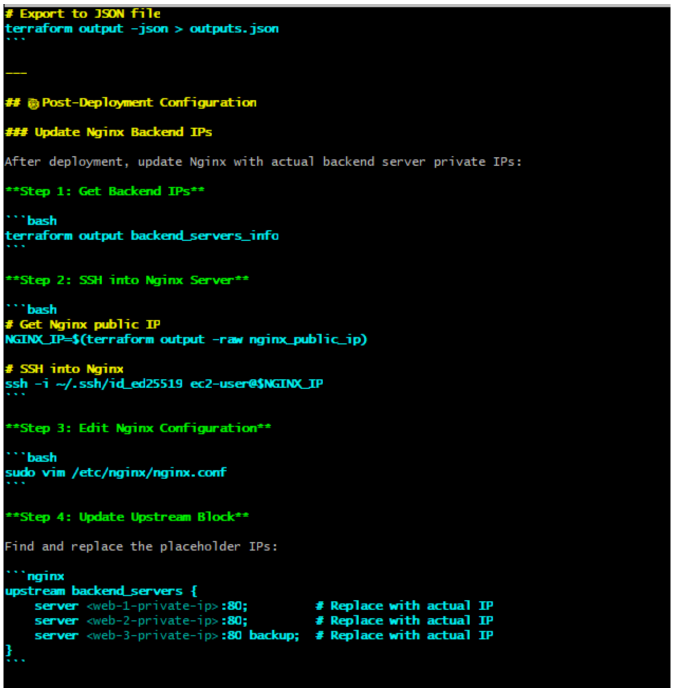
```hcl
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block  = "10.0.10.0/24"
availability_zone  = "me-central-1a"
env_prefix         = "prod"
instance_type      = "t3.micro"
public_key         = "~/.ssh/id_ed25519.pub"
private_key        = "~/.ssh/id_ed25519"
aws_region         = "me-central-1"
```

Step 3: Deploy Infrastructure

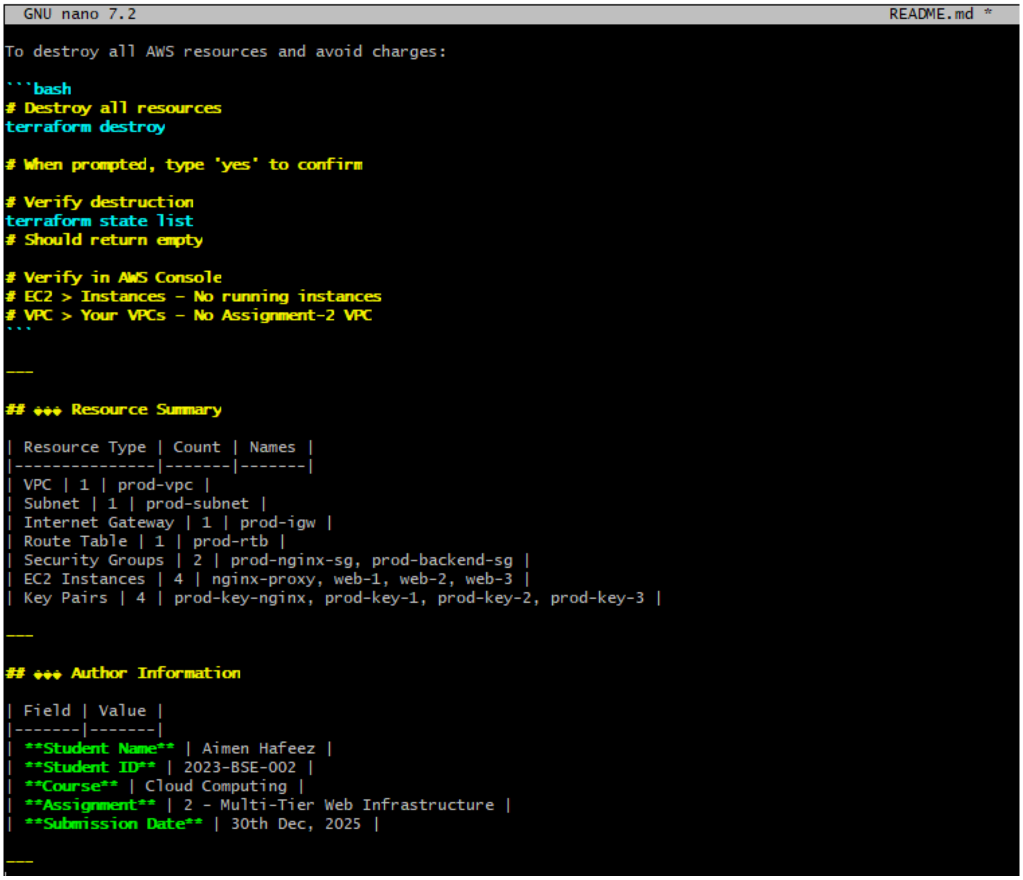
```bash
# Initialize Terraform
terraform init

# Validate configuration
terraform validate

# Preview changes
terraform plan

# Apply configuration
terraform apply -auto-approve
```

Step 4: Get Output Values

```bash
# Display all outputs
terraform output
```

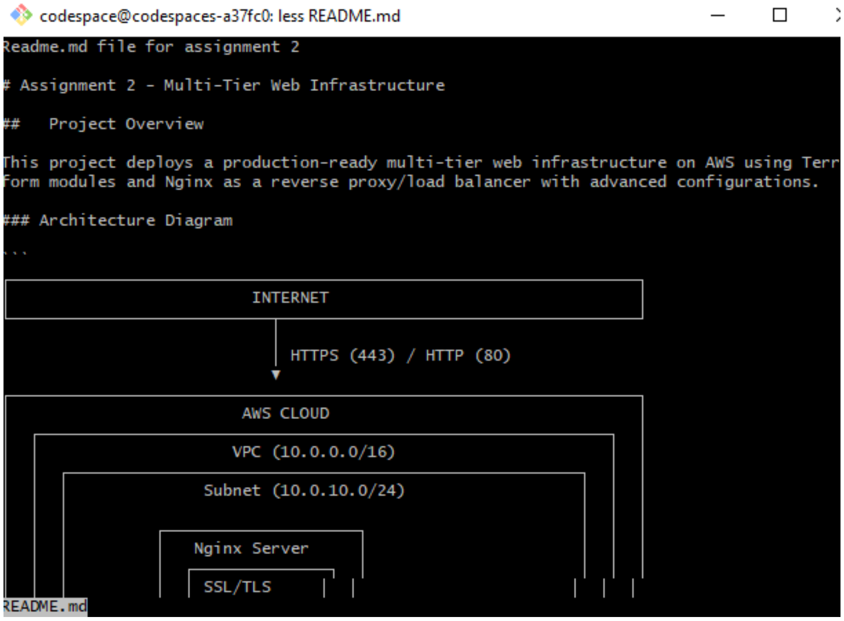


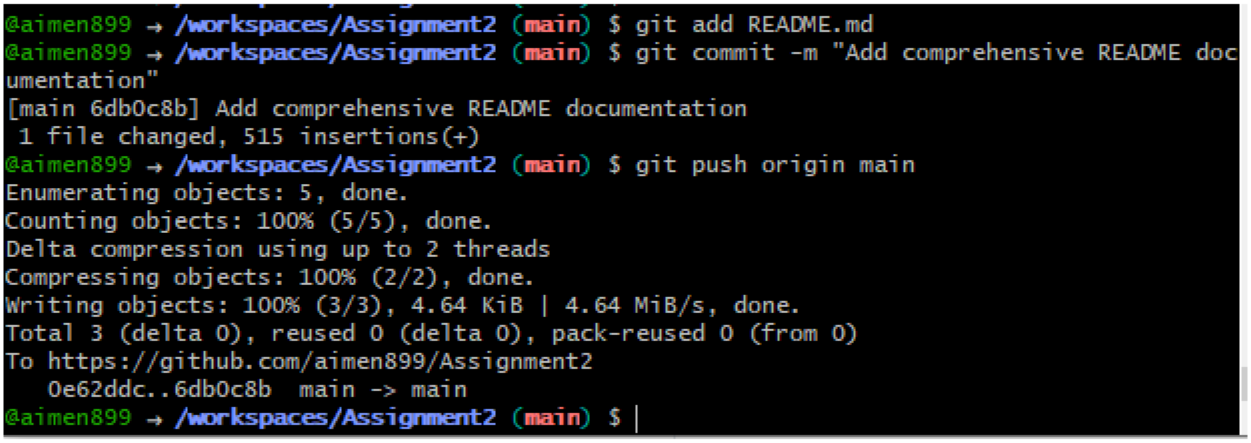
~~~
# Export to JSON file
terraform output -json > outputs.json
```

---

## ⚙ Post-Deployment Configuration

### Update Nginx Backend IPs

After deployment, update Nginx with actual backend server private IPs:

**Step 1: Get Backend IPs**

```bash
terraform output backend_servers_info
```

**Step 2: SSH into Nginx Server**

```bash
# Get Nginx public IP
NGINX_IP=$(terraform output -raw nginx_public_ip)

# SSH into Nginx
ssh -i ~/.ssh/id_ed25519 ec2-user@$NGINX_IP
```

**Step 3: Edit Nginx Configuration**

```bash
sudo vim /etc/nginx/nginx.conf
```

**Step 4: Update Upstream Block**

Find and replace the placeholder IPs:

```nginx
upstream backend_servers {
    server <web-1-private-ip>:80;         # Replace with actual IP
    server <web-2-private-ip>:80;         # Replace with actual IP
    server <web-3-private-ip>:80 backup;  # Replace with actual IP
}
```
~~~

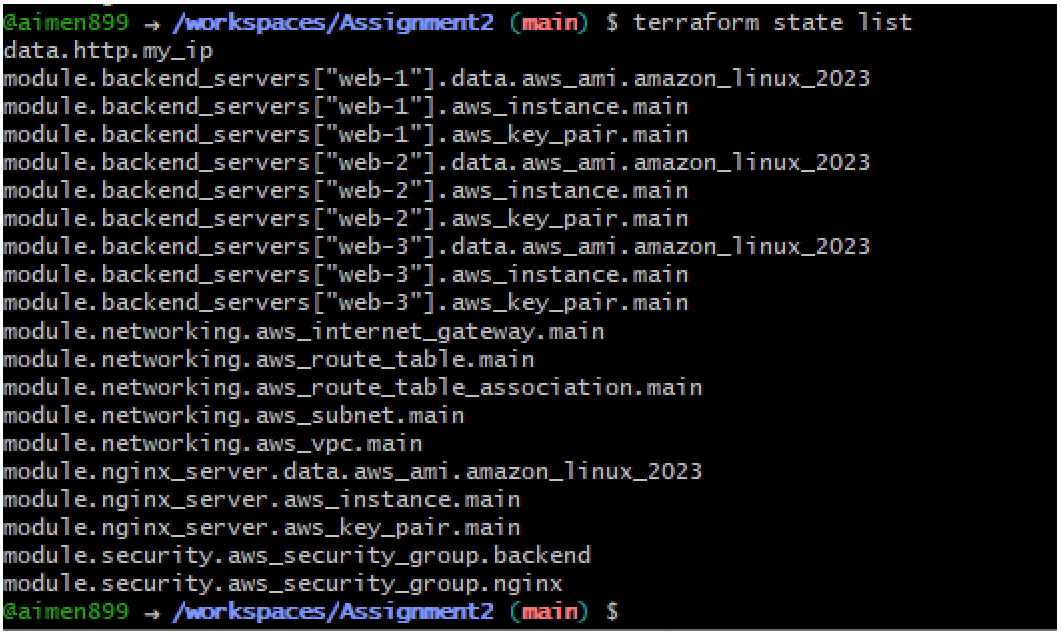


~~~
  GNU nano 7.2                                                    README.md *

To destroy all AWS resources and avoid charges:

```bash
# Destroy all resources
terraform destroy

# When prompted, type 'yes' to confirm

# Verify destruction
terraform state list
# Should return empty

# Verify in AWS Console
# EC2 > Instances - No running instances
# VPC > Your VPCs - No Assignment-2 VPC
```

---

## ��� Resource Summary

| Resource Type | Count | Names |
|---------------|-------|-------|
| VPC | 1 | prod-vpc |
| Subnet | 1 | prod-subnet |
| Internet Gateway | 1 | prod-igw |
| Route Table | 1 | prod-rtb |
| Security Groups | 2 | prod-nginx-sg, prod-backend-sg |
| EC2 Instances | 4 | nginx-proxy, web-1, web-2, web-3 |
| Key Pairs | 4 | prod-key-nginx, prod-key-1, prod-key-2, prod-key-3 |

---

## ��� Author Information

| Field | Value |
|-------|-------|
| **Student Name** | Aimen Hafeez |
| **Student ID** | 2023-BSE-002 |
| **Course** | Cloud Computing |
| **Assignment** | 2 - Multi-Tier Web Infrastructure |
| **Submission Date** | 30th Dec, 2025 |

---
~~~

Alternatively, view README in a pager

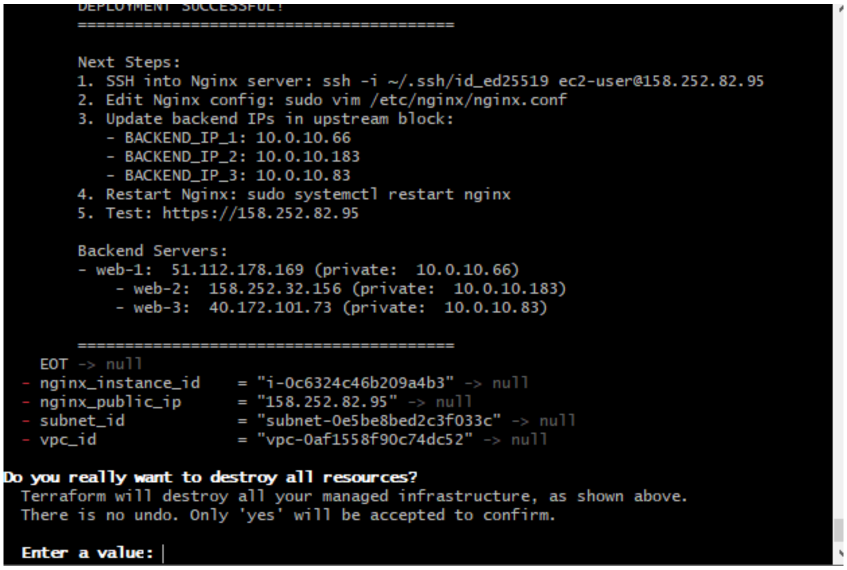


```
Readme.md file for assignment 2

# Assignment 2 - Multi-Tier Web Infrastructure

##   Project Overview

This project deploys a production-ready multi-tier web infrastructure on AWS using Terr
form modules and Nginx as a reverse proxy/load balancer with advanced configurations.

### Architecture Diagram

```
┌─────────────────────────────────────────────────────────────────┐
│                            INTERNET                             │
└─────────────────────────────────────────────────────────────────┘
                            │
                            │ HTTPS (443) / HTTP (80)
                            ▼
┌─────────────────────────────────────────────────────────────────┐
│                           AWS CLOUD                             │
│  ┌───────────────────────────────────────────────────────────┐  │
│  │                     VPC (10.0.0.0/16)                     │  │
│  │  ┌─────────────────────────────────────────────────────┐  │  │
│  │  │                Subnet (10.0.10.0/24)                │  │  │
│  │  │                                                     │  │  │
│  │  │         ┌────────────────────┐                      │  │  │
│  │  │         │   Nginx Server     │                      │  │  │
│  │  │         │  ┌──────────────┐  │                      │  │  │
│  │  │         │  │ SSL/TLS      │  │                      │  │  │
README.md
```

Verify README renders correctly on GitHub



```
@aimen899 → /workspaces/Assignment2 (main) $ git add README.md
@aimen899 → /workspaces/Assignment2 (main) $ git commit -m "Add comprehensive README doc
umentation"
[main 6db0c8b] Add comprehensive README documentation
 1 file changed, 515 insertions(+)
@aimen899 → /workspaces/Assignment2 (main) $ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 4.64 KiB | 4.64 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/aimen899/Assignment2
   0e62ddc..6db0c8b  main -> main
@aimen899 → /workspaces/Assignment2 (main) $ |
```

6.2 Infrastructure Cleanup

Prepare for Destruction

Before destroying, verify what resources exist:
```



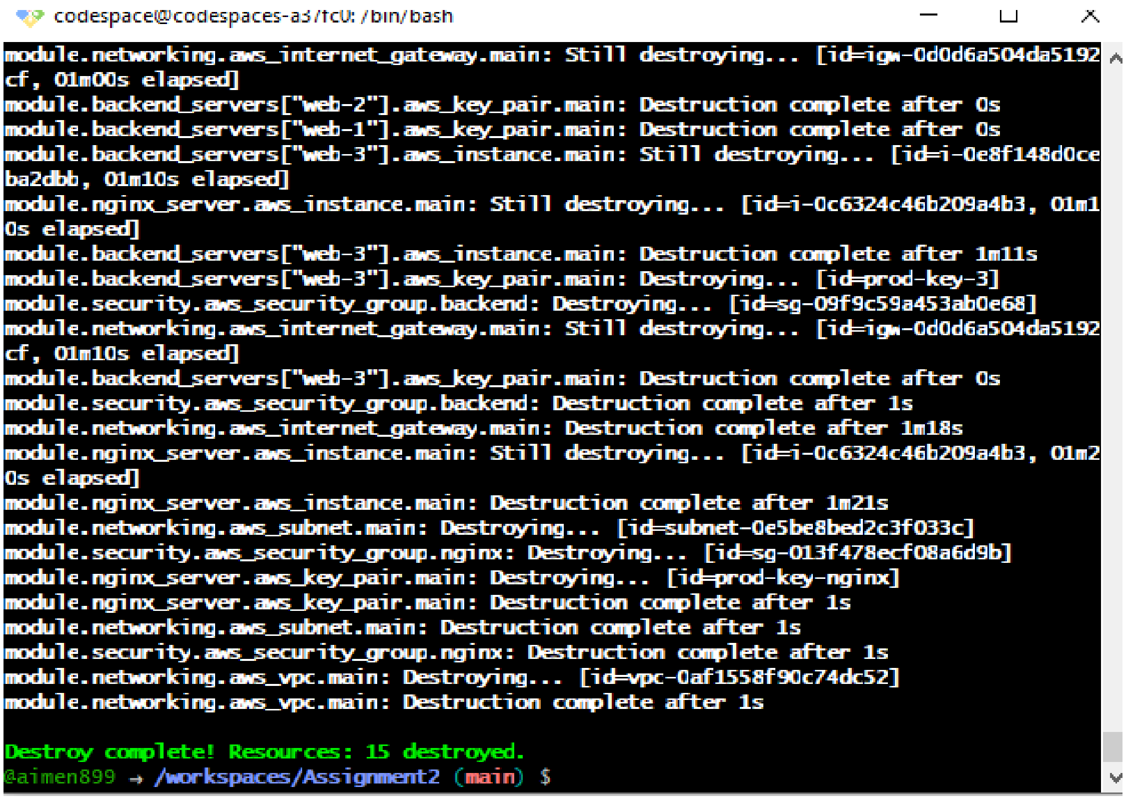
```
@aimen899 → /workspaces/Assignment2 (main) $ terraform state list
data.http.my_ip
module.backend_servers["web-1"].data.aws_ami.amazon_linux_2023
module.backend_servers["web-1"].aws_instance.main
module.backend_servers["web-1"].aws_key_pair.main
module.backend_servers["web-2"].data.aws_ami.amazon_linux_2023
module.backend_servers["web-2"].aws_instance.main
module.backend_servers["web-2"].aws_key_pair.main
module.backend_servers["web-3"].data.aws_ami.amazon_linux_2023
module.backend_servers["web-3"].aws_instance.main
module.backend_servers["web-3"].aws_key_pair.main
module.networking.aws_internet_gateway.main
module.networking.aws_route_table.main
module.networking.aws_route_table_association.main
module.networking.aws_subnet.main
module.networking.aws_vpc.main
module.nginx_server.data.aws_ami.amazon_linux_2023
module.nginx_server.aws_instance.main
module.nginx_server.aws_key_pair.main
module.security.aws_security_group.backend
module.security.aws_security_group.nginx
@aimen899 → /workspaces/Assignment2 (main) $
```

Run Terraform Destroy



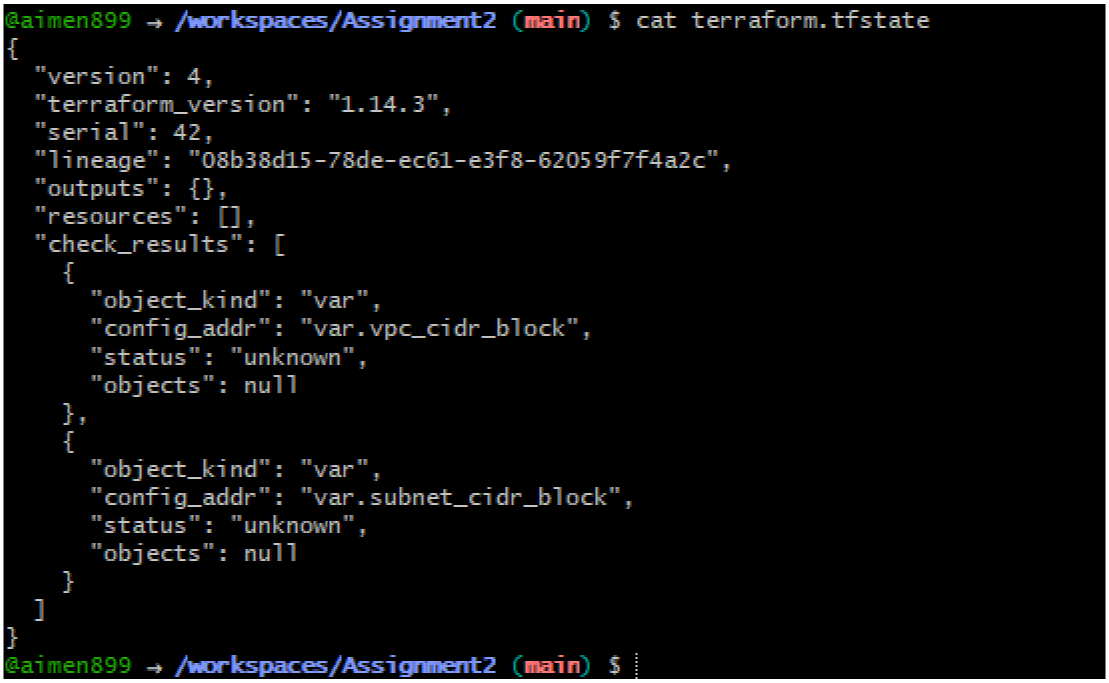
```
        DEPLOYMENT SUCCESSFUL!
        ========================================

        Next Steps:
        1. SSH into Nginx server: ssh -i ~/.ssh/id_ed25519 ec2-user@158.252.82.95
        2. Edit Nginx config: sudo vim /etc/nginx/nginx.conf
        3. Update backend IPs in upstream block:
           - BACKEND_IP_1: 10.0.10.66
           - BACKEND_IP_2: 10.0.10.183
           - BACKEND_IP_3: 10.0.10.83
        4. Restart Nginx: sudo systemctl restart nginx
        5. Test: https://158.252.82.95

        Backend Servers:
        - web-1:  51.112.178.169 (private:  10.0.10.66)
            - web-2:  158.252.32.156 (private:  10.0.10.183)
            - web-3:  40.172.101.73 (private:  10.0.10.83)

        ========================================
    EOT -> null
  - nginx_instance_id    = "i-0c6324c46b209a4b3" -> null
  - nginx_public_ip      = "158.252.82.95" -> null
  - subnet_id            = "subnet-0e5be8bed2c3f033c" -> null
  - vpc_id               = "vpc-0af1558f90c74dc52" -> 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:
```

Confirm Destruction



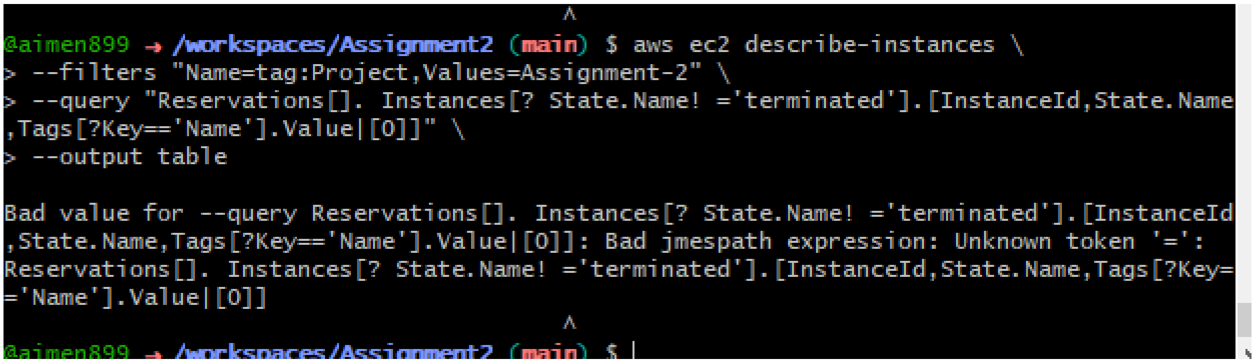
```
module.networking.aws_internet_gateway.main: Still destroying... [id=igw-0d0d6a504da5192
cf, 01m00s elapsed]
module.backend_servers["web-2"].aws_key_pair.main: Destruction complete after 0s
module.backend_servers["web-1"].aws_key_pair.main: Destruction complete after 0s
module.backend_servers["web-3"].aws_instance.main: Still destroying... [id=i-0e8f148d0ce
ba2dbb, 01m10s elapsed]
module.nginx_server.aws_instance.main: Still destroying... [id=i-0c6324c46b209a4b3, 01m1
0s elapsed]
module.backend_servers["web-3"].aws_instance.main: Destruction complete after 1m11s
module.backend_servers["web-3"].aws_key_pair.main: Destroying... [id=prod-key-3]
module.security.aws_security_group.backend: Destroying... [id=sg-09f9c59a453ab0e68]
module.networking.aws_internet_gateway.main: Still destroying... [id=igw-0d0d6a504da5192
cf, 01m10s elapsed]
module.backend_servers["web-3"].aws_key_pair.main: Destruction complete after 0s
module.security.aws_security_group.backend: Destruction complete after 1s
module.networking.aws_internet_gateway.main: Destruction complete after 1m18s
module.nginx_server.aws_instance.main: Still destroying... [id=i-0c6324c46b209a4b3, 01m2
0s elapsed]
module.nginx_server.aws_instance.main: Destruction complete after 1m21s
module.networking.aws_subnet.main: Destroying... [id=subnet-0e5be8bed2c3f033c]
module.security.aws_security_group.nginx: Destroying... [id=sg-013f478ecf08a6d9b]
module.nginx_server.aws_key_pair.main: Destroying... [id=prod-key-nginx]
module.nginx_server.aws_key_pair.main: Destruction complete after 1s
module.networking.aws_subnet.main: Destruction complete after 1s
module.security.aws_security_group.nginx: Destruction complete after 1s
module.networking.aws_vpc.main: Destroying... [id=vpc-0af1558f90c74dc52]
module.networking.aws_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 15 destroyed.
@aimen899 → /workspaces/Assignment2 (main) $
```

Verify State File is Empty



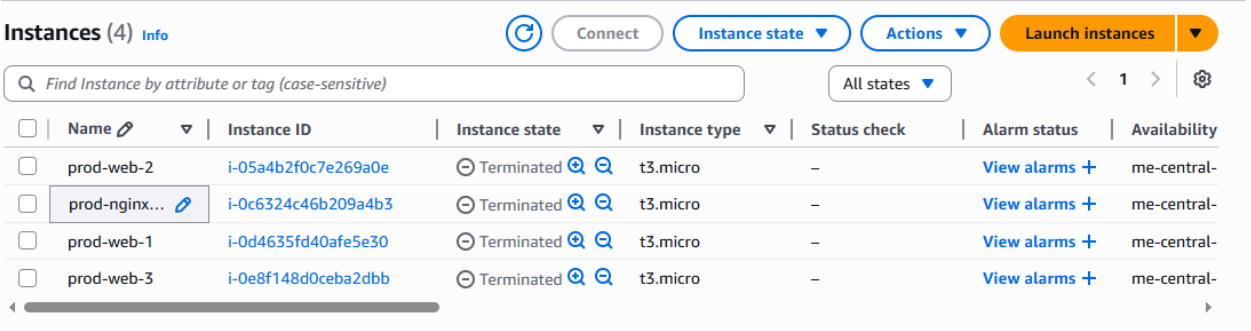
```
@aimen899 → /workspaces/Assignment2 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 42,
  "lineage": "08b38d15-78de-ec61-e3f8-62059f7f4a2c",
  "outputs": {},
  "resources": [],
  "check_results": [
    {
      "object_kind": "var",
      "config_addr": "var.vpc_cidr_block",
      "status": "unknown",
      "objects": null
    },
    {
      "object_kind": "var",
      "config_addr": "var.subnet_cidr_block",
      "status": "unknown",
      "objects": null
    }
  ]
}
@aimen899 → /workspaces/Assignment2 (main) $
```

Verify No Resources Remain Using AWS CLI



```
                                                           ^
@aimen899 → /workspaces/Assignment2 (main) $ aws ec2 describe-instances \
> --filters "Name=tag:Project,Values=Assignment-2" \
> --query "Reservations[]. Instances[? State.Name! ='terminated'].[InstanceId,State.Name
,Tags[?Key=='Name'].Value|[0]]" \
> --output table

Bad value for --query Reservations[]. Instances[? State.Name! ='terminated'].[InstanceId
,State.Name,Tags[?Key=='Name'].Value|[0]]: Bad jmespath expression: Unknown token '=':
Reservations[]. Instances[? State.Name! ='terminated'].[InstanceId,State.Name,Tags[?Key=
='Name'].Value|[0]]
                                                           ^
@aimen899 → /workspaces/Assignment2 (main) $ |
```

Verify in AWS Console



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
☐	prod-web-2	i-05a4b2f0c7e269a0e	Terminated	t3.micro	–	View alarms +	me-central-
☐	prod-nginx...	i-0c6324c46b209a4b3	Terminated	t3.micro	–	View alarms +	me-central-
☐	prod-web-1	i-0d4635fd40afe5e30	Terminated	t3.micro	–	View alarms +	me-central-
☐	prod-web-3	i-0e8f148d0ceba2dbb	Terminated	t3.micro	–	View alarms +	me-central-

Github Repsitry Link:
https://github.com/aimen899/Assignment2

Conclusion:

To sum up, this assignment showed the design and deployment of a production-ready multi-tier web architecture with Terraform and Nginx on AWS successfully. The application of Infrastructure as Code principles granted the project a scalable, secure, and highly available environment with modular Terraform configurations, automated provisioning, and efficient resource management. The integration of Nginx as a reverse proxy and load balancer allowed for effective traffic distribution along with SSL/TLS security, caching, failover handling, and performance optimization. The architecture was proven to be reliable and resilient through systematic testing of load balancing, caching, high availability, and security features, thus meeting real-world cloud deployment standards and reinforcing the practical understanding of modern cloud infrastructure design.

THE END